



A
u
b
u
r
n

U
n
i
v
e
r
s
i
t
y

IR-02-03

DESIGN LOADING FOR DEEPLY BURIED BOX CULVERTS

Prepared by

Kyungsik Kim
Chai H. Yoo

Prepared for

Highway Research Center
Auburn University, AL

Highway Research Center
Harbert Engineering Center
Auburn University, Alabama 36849-5337

MAY 2002

Design Loading for Deeply Buried Box Culverts

Kyungsik Kim

and

Chai H. Yoo

Highway Research Center
Auburn University
Auburn University, Alabama

May 2002

Acknowledgments

The investigation which led to the results presented herein was supported jointly by the Huff Eminent Scholar fund and the Highway Research Center at Auburn University. The authors gratefully acknowledge the assistance provided by D. L. Allen of Kentucky Transportation Research Center including the source code of PC version ISBILD.

Disclaimer

The opinions and conclusions expressed or implied in the report are those of the authors. They are not necessarily those of the funding agencies.

EXECUTIVE SUMMARY

An investigation was conducted to characterize and quantify the load intensity on deeply buried concrete box culverts. For deeply buried structures, the dead weight of soil itself is the main design load and the effect of live loads is not considered significant. The current AASHTO Standard Specifications for Highway Bridges and AASHTO LRFD Bridge Design Specifications stipulate the computation of the design load on the top flange of the box culvert based primarily on research work by Marston and Spangler. Depending on the installation method, trench installation or embankment installation, the effective density (or the design soil-structure interaction factor) on the concrete box culverts can be substantially different, as the settlement of the backfill is affected by the vertical shear between the adjacent soil columns. Although the AASHTO procedure may be applied conservatively for most ordinary culverts, an opportunity exists to evaluate a more rational soil-structure interaction behavior based on modern methods of finite element analyses.

The primary objectives are: 1) to evaluate factors directly influencing load intensities on concrete box culverts under high backfill, including the effect of different installation methods; 2) to examine the performance of the traditional method of predicting loads on concrete box culverts under high backfill; and 3) to develop a method that includes new design equations or figures to predict the effective densities or soil-structure interaction factors for deeply buried concrete box culverts.

To meet the objectives of this research project, three main tasks were carried out: 1) a literature survey, 2) numerical analysis and evaluation of analytical data, and 3) the development of design equations and graphical charts. In the first task, a thorough literature search was

conducted to reveal existing information regarding innovative design guidelines pertinent to calculating more rational loads on concrete box culverts. This task has uncovered a substantial number of research documents published on the subject, including the original Marston-Spangler theory, soil modeling techniques, and applications of finite element analysis programs. For the second task, a number of selected hypothetical culvert-soil systems with different installation methods and foundation conditions as well as various backfill heights, were modeled and analyzed using finite element codes such as ABAQUS, a general purpose commercial finite element code, CANDE-89 (Culvert ANalysis and DEsign Program), and ISBILD (a computer program for the analysis of static stresses and movements in embankments). The data collected were quantified and synthesized to formulate equations for predicting effective densities or soil-structure interaction factors in the final task. A selected portion of the data was then characterized using a regression analysis. Design equations were formulated and example problems were presented to illustrate the validity and reliability of the derived equations.

All three finite element programs, ABAQUS, CANDE-89, and ISBILD predicted the effective densities or soil-structure interaction factors that were reasonably close to one another. The results showed that the effective densities for embankment installations are most sensitive to the foundation characteristics, i.e., whether on a yielding or unyielding foundation was used. Concrete box culverts are most likely to be installed on a yielding foundation unless an obvious solid bedrock layer is encountered. Contrary to the current AASHTO provisions, the compactness of fill along the sides of the box section does not appear to affect the effective density significantly. The finite element analysis indicates that the effect of possible slips of the backfill materials along the exterior culvert wall is negligibly small. The results also show that imperfect trench installations reduce the effective density on the box culverts significantly.

However, the size of the soft zone that is needed in the imperfect trench installations need not be unnecessarily large. It has also been reported that the selection of the lightweight compression material to be used in the soft zone and subsequent field installations demands considerable planning and care. The current AASHTO equations, along with most other references, stipulate the effective densities for the trench installations under the assumption that the trench walls are excavated vertically. This study showed that the effect of sloping trench walls on the soil-structure interaction factors is quite high. As it is practically impossible and/or too expensive to excavate deep trench walls vertically, the validity and usefulness of the current AASHTO equations for trench installations is therefore questionable.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	ii
EXECUTIVE SUMMARY	iii
TABLE OF CONTENTS	vi
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER 1	
INTRODUCTION	1
1.1 Statement of the Problem.....	1
1.2 Objectives of Research	2
1.3 Research Approach	2
CHAPTER 2	
LITERATURE REVIEW	4
2.1 General	4
2.2 Soil Models	5
2.3 Interface Behavior	9
2.4 Imperfect Trench Installation	10
2.5 Finite Element Codes	11
CHAPTER 3	
BACKGROUND AND THEORY	14
3.1 Marston-Spangler Theory	14
3.2 Current Design Loads	19
3.3 Duncan Soil Model	26
3.4 Tangent Stiffness for Interface Shear Behavior	35
CHAPTER 4	
SOIL-STRUCTURE MODELING	42
4.1 Finite Element Model	42
4.2 Soil-Structure Interface Modeling	44
4.3 Model Verification	50
CHAPTER 5	
NUMERICAL ANALYSIS	52
5.1 General	52
5.2 Embankment Installations	53
5.3 Imperfect Trench Installations	57

5.4 Trench Installations	65
5.5 Effect of Sloping Trench Walls	69
5.6 Examples	69
CHAPTER 6	
CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE STUDY	77
6.1 Conclusions	77
6.2 Recommendations for Future Study	78
REFERENCES	80
APPENDICES	
APPENDIX A	
Typical Input for ABAQUS	84
APPENDIX B	
Typical Input for ISBILD	95
APPENDIX C	
Typical Input for CANDE-89	110
APPENDIX D	
ISBILD Source Code	111
APPENDIX E	
CANDE-89 Source Code	124

LIST OF TABLES

Table 2.1: Young's Modulus vs. Vertical Stress for the Overburden Dependent Model	8
Table 3.1: Design Values of Settlement Ratio	20
Table 3.2: Values of C_c in Terms of H/B_c	20
Table 3.3: Summary of the Hyperbolic Parameters	33
Table 3.4: Stress-Strain and Strength Parameters for Soils under Drained Conditions	36
Table 3.5: Stress-Strain and Strength Parameters for Soils under Unconsolidated-Undrained Conditions	38
Table 4.1: Soil-Structure Interface Factor Comparison	47
Table 4.2: Values of Parameters of K_{st}	48
Table 5.1: Hyperbolic Parameters Used in the Analysis	53
Table 5.2: Pressure Reduction Rates (%) Due to Imperfect Trench Installation	60
Table 5.3: Vertical Displacements at Soft Zone (ft)	64
Table 5.4: Summary of Results for Example 1	73

LISTS OF FIGURES

Figure 3.1: Pressure Transfer within a Soil-Structure System	15
Figure 3.2: Construction Conditions of Underground Conduits	16
Figure 3.3: Various Classes of Conduit Installations	17
Figure 3.4: Diagram for Coefficient C_d for Ditch Conduits	21
Figure 3.5: Diagram for Coefficient C_c for Positive Projecting Conduits	22
Figure 3.6: Diagrams for Coefficient C_n for Negative Projection Conduits and Imperfect Ditch Conditions	23
Figure 3.7: Hyperbolic Stress-Strain Curve	27
Figure 3.8: Transformed Hyperbolic Stress-Strain Curve	27
Figure 3.9: Hyperbolic Axial Strain-Radial Strain Curve	31
Figure 3.10: Transformed Hyperbolic Axial Strain-Radial Strain Curve	31
Figure 4.1: Typical Soil-Structure Finite Element Model	45
Figure 4.2: Soil-Structure Interaction Factor vs. W_t/B_c	46
Figure 4.3: Incremental Sequences for Embankment and Trench Installation	46
Figure 4.4: Spring Interface Modeling	49
Figure 4.5: Schematics of Test Setup and Data Comparison	50
Figure 5.1: Effective Density for Embankment Installations	55
Figure 5.2: Effective Density Curves from Proposed Equations for Embankment Installations	55
Figure 5.3: Deformed shape of Soil-Structure System	56
Figure 5.3: Imperfect Trench Installation	60
Figure 5.4: Load Reduction Rate vs. h/H_c (Different w/B_c)	61

Figure 5.5: Load Reduction Rate versus h/H_c (Different Young's Modulus)	61
Figure 5.6: Effective Density Distribution on the Box (Different w/B_c)	62
Figure 5.7: Effective Density Distribution on the Box (Different Young's Modulus)	62
Figure 5.8: Load Reduction Rate vs. h/H_c	63
Figure 5.9: Effective Density Distribution ($w/B_c=1$, $h/H_c=1.25$)	63
Figure 5.10: Close-up View of Deformed Shape near the Soft Zone	64
Figure 5.11: Effective Density vs. B_d/B_c	66
Figure 5.12: Effective Density vs. H/B_d on Yielding Foundation	68
Figure 5.13: Effective Density vs. H/B_d on Unyielding Foundation	68
Figure 5.14: Trench Installation with Sloping Sidewalls	70
Figure 5.15: Effect of Sloping Sidewall in Trench Installations ($H=50$ feet)	70
Figure 5.16: Effect of Sloping Sidewall in Trench Installations ($H=100$ feet)	71
Figure 5.17: Effect of Sloping Sidewall in Trench Installations ($H=150$ feet)	71

CHAPTER 1

INTRODUCTION

1.1 Statement of the Problem

Cast-in-place or precast reinforced concrete box culverts are widely used throughout the world to provide safe and relatively economical structures for the conveyance of water, vehicles, utilities or pedestrians. Although the single cell or multi-cell box culverts are rather simple structures, the loadings applied to these structures during their construction and subsequent service life can be complex. These structures are subjected to substantial vertical and lateral earth pressures, and are often subjected to significant temporary loadings during the construction of the embankment. In spite of the complex nature of the loading around these structures, simple routine design procedures would be highly desirable if they could be made possible.

When the relative settlement of the soil prism directly above the structure is greater than that of the adjacent soil prisms, as in trench installations, the layers of soil in the central prism are subjected to a reverse arch shape deformation and consequently the earth pressure on the structure is reduced by the amount of the shearing forces exerted to the central soil prism. Likewise, when the relative settlement of the soil prism directly above the structure is less than that of the adjacent soil prisms, as is usually found in embankment installations, the layers of soil on the central prism are subjected to an arch shape deformation and the earth pressure on the structure is increased. Although Marston^[1,2] and Spangler^[3,31,32] successfully quantified these phenomena by solving differential equations based on the equilibrium conditions of a simplified free body of prisms, an opportunity exists to revisit the problem using the modern-day numerical techniques made possible by the development of finite element methods.

1.2 Objectives of Research

Both AASHTO Standard Specifications for Highway Bridges^[6] and AASHTO LRFD Bridge Design Specifications^[5] stipulate procedures for calculating the design loads on box culverts based on the Marston-Spangler theory. It is desirable, however, to evaluate a more realistic soil-structure interaction behavior based on modern finite element analysis. The main objective of this research project is, therefore, to develop new design equations and/or graphical charts for the prediction of a more realistic value for the backfill load that is expected to exert on concrete box culverts, especially under high backfills. The primary objectives are as follows:

- 1) To evaluate the factors directly influencing load intensities on concrete box culverts under high backfill, including installation methods,
- 2) To examine the performance of the traditional method of predicting loads on concrete box culverts under high backfill, and
- 3) To develop a method that includes new design equations or figures to predict the effective densities for deeply buried concrete box culverts.

1.3 Research Approach

To meet the stated objectives in this research project, three main tasks were carried out as follows:

- 1) A literature survey,
- 2) Numerical analysis and evaluation, and
- 3) The development of design equations or figures.

In the first task, a thorough literature search was conducted to collect existing information regarding innovative design guidelines that can be applied to a more realistic design load evaluation for concrete box culverts. This task has resulted in a collection of a substantial

amount of material published on the subject, including the original Marston-Spangler theory, soil modeling techniques, and applications of finite element analysis programs.

In the second task, a number of selected hypothetical culvert-soil systems were modeled with different installation methods and foundation conditions, as well as a range of backfill heights. The models were analyzed using finite element codes such as ABAQUS, a general-purpose commercial finite element code, CANDE-89 (Culvert ANalysis and DEsign program), and ISBILD (a computer program for the analysis of static stresses and movements in embankments). This study presents results that are based on two-dimensional finite element analyses, linear elastic analysis by ABAQUS, CANDE-89, and nonlinear analysis by ISBILD. The construction layers were considered in these three programs. The backfill heights were varied from 50 to 200 feet for the embankment condition and 50 to 150 feet for the trench condition. In the case of the trench condition, the distance between the concrete box culvert wall and the adjacent existing trench soil column was also varied.

In the final task, the data collected in the second task were quantified and used to formulate a set of equations for predicting effective densities.

CHAPTER 2

LITERATURE REVIEW

2.1 General

A culvert, in the form of an underground structure, in a uniform soil field causes a redistribution of the free field stresses and deflections. The nature of this redistribution influences the load that reaches the structure, and the proportion of the load that reaches the structure is governed by the characteristics of the soil, and the geometry and stiffness of the structure itself. Marston pioneered an investigation of underground conduits both analytically and experimentally in the early years of the 20th century. Most codes of practice for the design and construction of underground structures have made use of the extensive experimental and analytical studies carried out originally by Marston. Marston and Anderson^[1] published an authoritative paper on the theory of external loads on pipes in ditches and tests of cement and clay drain tiles and sewer pipes in 1913. Marston issued a series of bulletins from then on; one of the most useful works was a review of theories of external loads on closed conduits in 1930.^[2] The work was carried forward by Spangler, who published a bulletin concerning field measurements of the settlement ratio of various highway culverts in 1950.^[3] Marston discussed the loads on rigid conduits in terms of the shearing forces developed in the prism of soil above the structure, making use of the following soil properties: density, the ratio between active lateral and vertical pressure, cohesion, the angle of internal friction, and the stress-strain characteristics of the soil. Marston also took into account the relative settlement of the soil above and below the structure and the soil adjacent to the structure, the vertical deflection of the top surface of the structure with respect to the base, and the compression in the soil prisms above

and adjacent to the structure. More details concerning the Marston-Spangler theory are presented in Chapter 3.

Up until the twelfth edition of the AASHTO Standard Specifications for Highway Bridges,^[4] the stipulated vertical loading was essentially 70 percent of the weight of the earth prism (120 lb/ft³) above the top slab. Article 1.2.2(A) permits the application of Marston formulas for definite conditions of bedding and backfill. The current AASHTO LRFD Bridge Design Specifications^[5] and Standard Specifications for Highway Bridges^[6] requirements on the design loadings on cast-in-place or precast concrete box culverts are based on the Marston-Spangler theory of the soil-structure interaction. These criteria were adopted by AASHTO in the thirteenth edition of the Standard Specifications^[7] in 1983. At the same time, the 0.7, vertical soil pressure reduction factor, was eliminated. The current AASHTO Standard Specifications for highway bridges^[6] provide dead and wheel loads on the culvert transferred through the filling soil in Section 6, “Culverts,” and design guidelines of culverts including soil-structure interaction factors, taking into account the type and conditions of installation, in Section 17, “Soil-Reinforced Concrete Structure Interaction Systems.” The current AASHTO LRFD Bridge Design Specifications^[5] also provide methods of calculating total unfactored loading on concrete box culverts, either cast-in-place or precast, in Section 12, “Buried Structures and Tunnel Liners.”

2.2 Soil model

Linear elastic isotropic material properties were used in earlier finite element analyses of soil behavior. Two independent elastic constants are required to characterize the soil; Poisson’s ratio and Young’s modulus, or shear modulus. It was readily recognized that successful modeling required a judicious choice of the elastic moduli of various soil types. More common

representations of soil behavior, incorporated in the finite element codes, are the nonlinear models represented by piece-wise elastic or stress-dependent moduli that are capable of handling the laboratory stress-strain conditions. There are two general categories of stiffness and/or compressibility models: (1) models based on constant confining pressure triaxial tests, and (2) models based on uniaxial strain (confined) compression tests. The soil models in the first category usually incorporate a failure condition, whereas those found in the second category do not, because the failure condition is not defined in the uniaxial tests. Examples of the first category are the Hardin model,^[8] the Duncan model,^[9,10] and the bilinear model,^[11] while the overburden-dependent model^[12] falls into the second category. Since the Duncan model is capable of obtaining its compressibility parameters from the uniaxial compression test, it is the most general of the models in the first category and one of the most frequently used models in recent studies.

In the linear elastic model, where the soil is represented as a linear elastic material, it is required that two independent elastic constants be defined, for example, Young's modulus, E , and Poisson's ratio, ν . Based on a synthesis of the results of the plate load field tests, one-dimensional compression tests, and triaxial compression laboratory tests, Sanderson and his coworkers^[13] showed the best estimate of a representative Young's modulus for all of the backfill to be in the range from 3,000 to 7,000 psi. The value of Poisson's ratio was determined from the coefficient of lateral earth pressure at rest, K_0 , which is required to define the initial stress states of all of the newly placed soil elements for the finite element analysis. This coefficient of lateral earth pressure relates the lateral stress, σ_H , to the vertical stress, σ_V , using the relationship:

$$\sigma_H = K_0 \sigma_V \quad (2.1)$$

The vertical stress, σ_v , is calculated as the product of the total unit soil weight, γ_T , and half the vertical element height, H . The value of K_0 was determined from the Mohr-Coulomb strength parameter, ϕ , and the empirical correlation as:

$$K_0 = 1 - \sin \phi \quad (2.2)$$

The Poisson's ratio, ν , for the plane strain case may be calculated from K_0 as follows:^[24]

$$\nu = \frac{K_0}{1 + K_0} \quad (2.3)$$

For example, for a ϕ of 44 degrees, typical values of K_0 and ν are calculated to be 0.35 and 0.25 from Equations 2.2 and 2.3, respectively.

The Duncan model is the most frequently used soil model based on a nonlinear constitutive relationship. Duncan and Chang^[9] expanded the basic hyperbolic stress-strain relationship of soil suggested by Kondner^[14] to include failure ratios of soil, modulus numbers, modulus exponent, and Poisson ratio parameters. In the Duncan model, triaxial and one-dimensional compression tests are employed to determine the stress-strain, strength and compressibility parameters. Details of the Duncan model are given in Chapter 3.

In a bilinear model, the stress-strain behavior of an individual triaxial test is represented as a bilinear function consisting of a constant slope from zero to the peak deviator stress, followed by a small positive slope after the peak. The constant slope represents the average slope from zero to the peak stress. The reduction in stress following the peak is not modeled; instead, a reduction of Young's modulus, i.e., the slope in deviator stress-axial strain plot to a value of 1/1,000 of its original value is incorporated to handle failure. The relationship between the constant average slope and the confining pressure was described by Janbu^[15] in terms of

confining stress, atmospheric pressure, and an exponent determining the rate of variation of constant modulus with confining stress.

One inconvenience of the overburden model is that the overburden-dependent, incrementally elastic parameters, Young's modulus, E , and Poisson's ratio, ν , cannot be determined directly from a one-dimensional compression test. Sanderson and his coworkers^[13] used a bilinear model for back-calculation in order to reproduce one-dimensional compression test results. McVay^[16] proposed some vertical stress-dependent Young's moduli for the overburden dependent model using one-dimensional compression test with a single finite element. McVay's model incorporates a constant Poisson's ratio of 0.25. Table 2.1 lists the moduli with their respective vertical stresses.

Table 2.1: Young's Modulus vs. Vertical Stress for the Overburden Dependent Model

Vertical Stress (psi)	Young's Modulus (psi)
0.0	44
0.5	623
1.0	840
3.0	1,100
5.0	1,170
10.0	1,720
15.0	2,200
20.0	2,700
25.0	3,200
30.0	3,600
40.0	4,600

2.3 Interface behavior

An interface phenomenon that is frequently discussed is slip between the soil and culvert, which is characterized by the relative movement along the surface of contact when the shear stress tangent to the interface exceeds the corresponding interface frictional resistance. The modeling of the interface behavior for finite element applications is classified into two distinct methods, i.e., the method of stiffness and the method of constraints.^[12]

Goodman, Taylor and Brekke^[17] were the first to develop the stiffness interface element to account for the relative movements between rock joints. The element consists of two lines, each with two nodal points. The two lines initially occupy the same position before deformation and each node has two degrees of freedom, horizontal and vertical displacement, which can simulate both tangential slippage across the interface and debonding of the soil and culvert. However, the element suffers from certain difficulties. In order to prevent penetration of the two contacting bodies when subject to compressive forces, a very large interface normal stiffness must be supplied. On the other hand, some penetration is required in order to recover the normal force at the interface. Leonards and Wu^[18] have indicated that due to the large normal stiffness, the significant digits of penetration become truncated, hence the resulting interface normal force would be in error. It has been shown by studies employing the method of stiffness that in most cases the effects of slip between the soil and conduits are small.^[16]

The concept of using constraint equations to represent interface behavior was first introduced by Chan and Tuba.^[19] It is believed that Katona and his coworkers^[12] were the first to develop a general theory for treating constraint equations in the formulation of interface elements, and devised an iterative procedure for characterizing the interface behavior. The interface element is defined by a set of paired nodes, one node belonging to the conduits and the

other the soil, each of which allows both horizontal and vertical displacements. A third node is assigned to the interior of the paired nodes with the distinct purpose of providing equation numbers for the normal and tangential interface forces. The resulting element stiffness matrix is of the order of 6 by 6. Four kinematic states were defined to represent the interface behavior based on permutations of either “fixed or “free” to describe the relative movements in both the normal and tangential directions. However, the free state in the normal direction and the fixed state in the tangential direction were disallowed from a physical point of view. For a given load increment, the choice of correct interface states (kinematically) is determined by a trial and error process. The trial responses are then used to determine if the assumed state is correct and, if not, what the new trial state should be.^[12] The interface behavior by constraint equation has been programmed in the CANDE program^[36] but has met with only limited success because of difficulties of convergence. The linear and overburden-dependent soil models appear to be the only two soil models free of the convergence problems with the interface elements in the CANDE program.^[18, 20]

2.4 Imperfect Trench Installation

During the late 1800s and the early 1900s, the significant effect of the relative settlements of the prisms directly above and adjacent to the buried conduits on the soil pressure was noted. Consequently, it became a common practice to place the fill to a height of about 12 ft above the structure, excavate a trench, and back fill this with an organic material such as hay, also known as imperfect trench installation. It was intuitively expected that the result of this organic back fill was to introduce a reverse arching effect with a radical alteration of the culvert pressure distribution due to the fill subsequently placed above the hay level. With the completion of the fill and the passage of time, the organic material would rot away and provide little effective

stiffness. It is believed that Brown^[21] quantified for the first time in 1967 the pressure reduction effect of the hay layers based on the finite element method of plane elasticity. An analytical method incorporating the finite element analysis of plane elasticity was developed to account for the practical features concerning how to describe the pressure distribution on a rigid culvert that is buried with the light material zone. To verify the accuracy of the theoretical treatment and the computer programs, the method developed was applied to a 200-ft-high fill embankment with organic inclusions over a 10-ft high, barrel-shaped rigid culvert. Various properties of fill and organic hay zones with rigid and flexible rock-boundaries were tried for a series of practical examples. The success of the method is largely dependent on the knowledge of the properties of fill, rock, and hay inclusions. Lately, Vaslestad and his coworkers^[22] revisited this imperfect trench installation method to examine the long-term behavior of the load reduction characteristics on rigid culverts beneath high fills. Super light expanded polystyrene blocks were used as the compressible material. The full-scale measurements showed a considerable reduction, 30%-50%, in the vertical earth pressure. The long-term observations showed that there was no increase in earth pressure and deformation of the pipes beneath the rock fill and only a slight increase in the deformation of the expanded polystyrene in the clay.

2.5 Finite Element Codes

The finite element method was originally introduced in the late 1950s as a tool for the solution of stress problems related primarily to the aircraft industry. Since that time, the finite element method has become a useful and accepted tool in many areas of civil engineering. Linear and nonlinear stress analysis applications in geotechnical engineering for static and dynamic loading was introduced in the late 1960s and early 1970s.

Kulhawy and his coworkers^[23] developed LSBUILD, a finite element program for stresses and movements in embankments during construction, in 1969. Ozawa and Duncan^[24] followed the work and developed ISBILD, which is an updated version of the older computer program LSBUILD taking into account additional characteristics such as incompatible displacement modes, a more accurate procedure for assigning initial stresses to elements, and more efficient computational techniques. The computer program ISBILD was developed originally in order to incorporate improvements in finite element analysis techniques into a computer program for analysis of stress and movements in dams. This program incorporates a nonlinear incremental finite element method employing a hyperbolic stress-strain relationship and incremental analysis procedures. Allen and Meade^[25] used the program ISBILD to predict loads on and settlements of concrete culverts that had been actually installed in the Kentucky area. Soil parameters for the original ground were obtained from triaxial tests of materials from the actual construction site. Their results indicated that the finite element method predicted the pressure on box culverts more accurately than other analytical methods available at the time.

Katona and his coworkers^[12] pioneered the application of the finite element method for the solution of buried pipe problems. Their FHWA-sponsored project produced CANDE, which is a special-purpose finite element program primarily intended for the design and analysis of buried culverts. Some useful features of the program include incremental construction to model the physical process of constructing soil layers in a progressive manner, linear and nonlinear soil models, nonlinear beam-rod elements to model yielding and cracking of structural components, and frictional interfaces between soil and structure to simulate frictional sliding during the loading. The CANDE program has been widely used for soil-structure analysis and evaluation of buried box culvert designs by a number of researchers, including Chang and his coworkers,^[20]

Katona and Vittes,^[26] and Tadros and his coworkers.^[27] CANDE has been upgraded several times, with the latest version being CANDE-89^[28] (available from McTrans, University of Florida).

ABAQUS,^[30] developed by Hibbit, Karsson & Sorensen, Inc., is one of the most broadly used general-purpose commercial finite element codes. Yang and his coworkers^[29] made use of ABAQUS to model and predict loads on a double-cell concrete box culvert under a backfill of 36 feet. A non-associative Drucker-Prager elasto-plastic model with linear yield criteria was chosen for the gravel surrounding the culvert, the backfill material, and the existing excavated slope. The interface between the structure and the surrounding fill was represented by contact pair surfaces, which allow sliding and loss of contact during the loading.

In order to investigate the soil-structure interaction effect on load intensity on concrete box culverts, the following three finite element programs will be used in this study, i.e., ABAQUS Version 5.8,^[30] ISBILD (Computer Program for Analysis of Static Stresses and Movements in Embankment),^[24] and CANDE-89 (Culvert ANALysis and DEsign Program).^[28] Analyses made in this study using these three finite element programs are based on the plane strain geometry and the small displacement theory.

CHAPTER 3

BACKGROUND AND THEORY

3.1 Marston-Spangler Theory

One of the outstanding contributions of the Marston theory of loads on buried conduits is its demonstration, by the principles of mechanics, that the load on a structure is affected by certain environmental conditions of installation in addition to the height of fill over the structure. The factors influencing the load are, for the most part, associated with the installation conditions which control the magnitude and direction of settlements of the soil prism over the structure relative to the settlements of the exterior soil prisms immediately adjacent to this central soil prism. These relative settlements generate friction forces or shearing stresses that are added to or subtracted from the dead weight of the central prism to produce the resultant load on the structure, as shown in Figure 3.1. When the relative settlement of the soil prism directly above the structure is greater than that of the adjacent soil prisms, as depicted in the trench installations in Figure 3.1, the layers of soil in the central prism are subjected to a reverse arch shape deformation and consequently the earth pressure on the structure is reduced by the amount of the shearing forces exerted to the central soil prism. Likewise, when the relative settlement of the soil prism directly above the structure is less than that of the adjacent soil prisms, as usually found in embankment installations, the layers of soil in the central prism are subjected to an arch shape deformation and the earth pressure on the structure is increased. The transfer of pressure due to these relative movements within the soil-structure system is illustrated in Figure 3.1.

Because of the influence of these installation conditions and the importance of recognizing them when determining loads, installations of buried structures are classified into

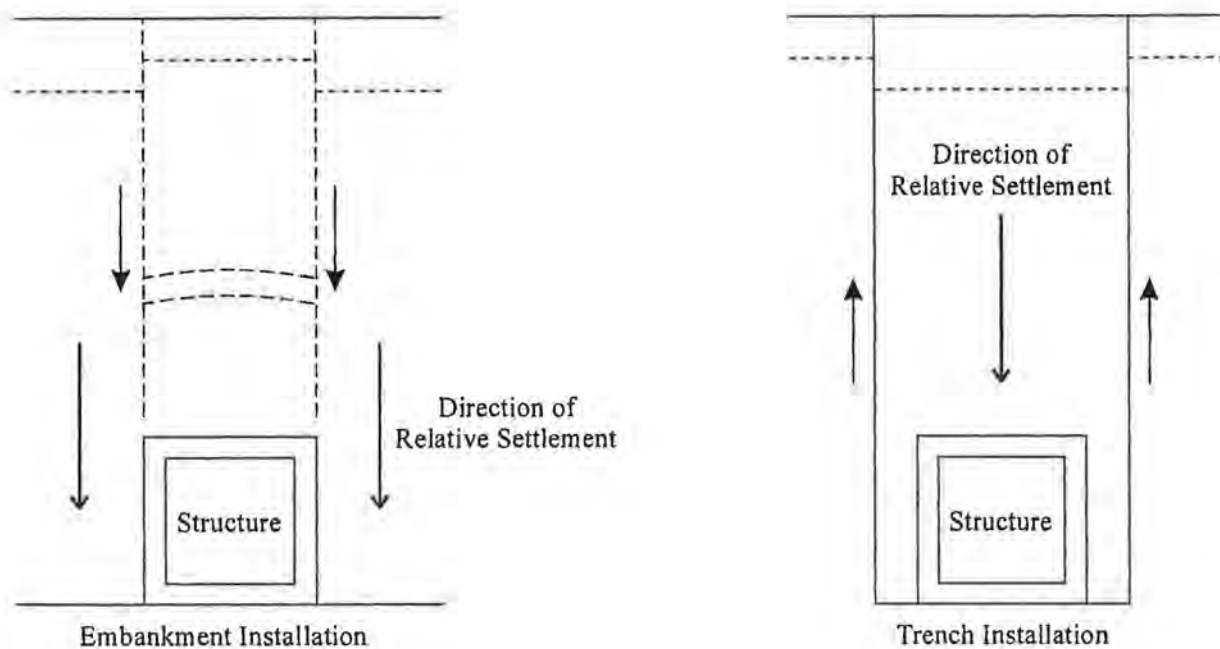


Figure 3.1: Pressure Transfer within a Soil-Structure System

two broad categories, trench installations and embankment installations. Spangler and Handy^[31] categorized buried conduits into several groups and sub-groups based on the various construction conditions that influence loads on underground conduits. Figure 3.2 shows their classification for underground conduits.

A ditch conduit is defined as one that is installed in a relatively narrow ditch dug in passive or undisturbed soil and which is then covered with earth backfill. A positive projecting conduit is one which is installed in shallow bedding with its top projecting above the surface of the natural ground and which is then covered with an embankment. Railway and highway culverts are frequently installed in this manner. A negative projecting conduit is installed in a relatively narrow and shallow ditch, with its top at an elevation below the natural ground surface,

which is then covered with an embankment. This is a very favorable method of installing a railway or highway culvert, since the load produced by a given height of fill is generally less than it would be in the case of a positive projecting conduit. This negative projecting method of construction is most effective in minimizing the load if the ditch between the top of the conduits and the natural ground surface is backfilled with loose uncompacted soil or organic material, such as baled hay. The imperfect ditch conduit, sometimes called the induced trench conduit, is

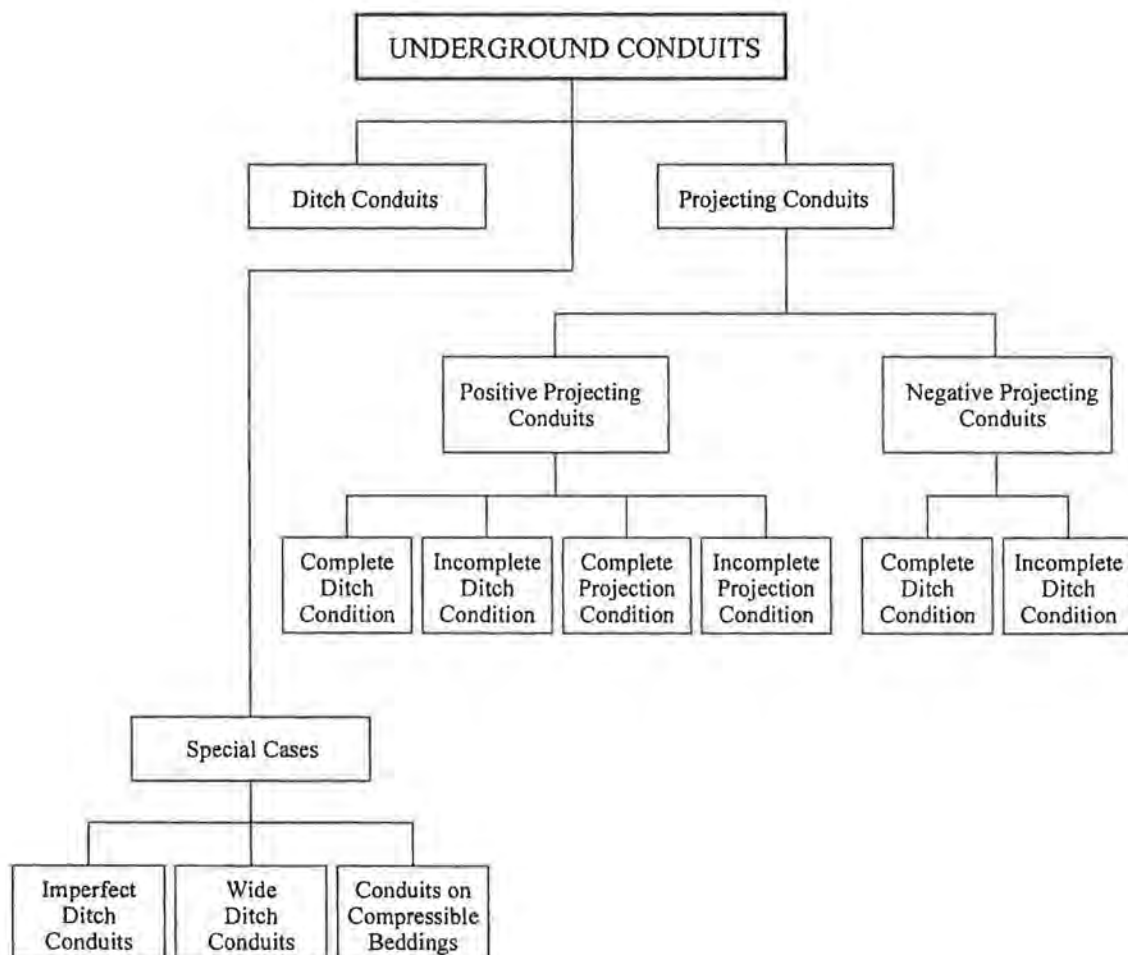


Figure 3.2: Construction Conditions of Underground Conduits^[31]

an important special case that is somewhat similar to the negative projecting conduit, but is even more favorable from the standpoint of reduction of load on the structure. Obviously, this method of construction should not be employed in embankments that serve as a water barrier, since the loosely placed backfill may encourage channeling of seepage water through the embankment. The essential elements of these four main classes of conduit are shown in Figure 3.3.

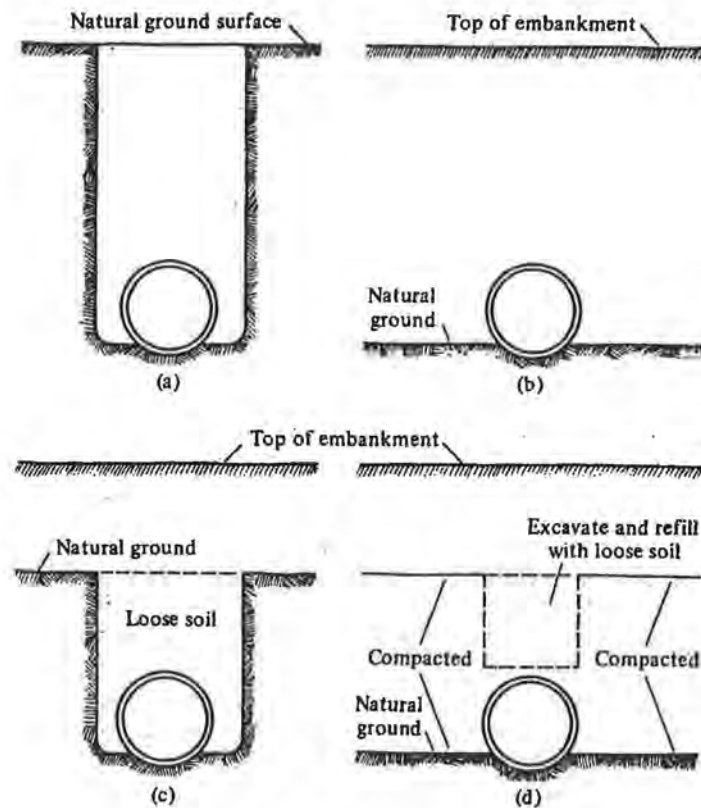


Figure 3.3: Various Classes of Conduit Installations

- | | |
|---------------------------------|---------------------------------|
| (a) Ditch Conduit | (b) Positive Projecting Conduit |
| (c) Negative Projecting Conduit | (d) Imperfect Ditch Conduit |

Marston^[2] and Spangler^[32] quantified the load on conduits installed by different construction conditions by solving differential equations based on the equilibrium condition of a simplified free body of prisms, and proposed equations for predicted loads on conduits due to earth fill as follows:

$$W = C_d \gamma B_d^2 \quad \text{for ditch conduits} \quad (3.1)$$

$$W = C_c \gamma B_c^2 \quad \text{for positive projecting conduits} \quad (3.2)$$

$$W = C_n \gamma B_c^2 \quad \text{for imperfect ditch or negative projecting conduits} \quad (3.3)$$

where C_d , C_c , and C_n = load coefficients; B_d = horizontal width of ditch; and B_c = width of ditch. Although graphical diagrams are provided for the computation of coefficients, there still exist many practical difficulties because the load coefficients proposed contain certain parameters that cannot be determined readily, such as settlement ratio and height of plane of equal settlement. Graphical diagrams for C_d , C_c , and C_n are presented in Figures 3.4, 3.5, and 3.6. For load coefficients, C_c in Figure 3.5, the rays are straight lines that can be represented by equations for use when the value of H/B_c exceeds the limit of the diagram. These equations are given in Table 3.2. Symbols used in the figures are defined as follows:

- H = height of fill above top of conduit,
- B_d = horizontal width of ditch at top of conduit,
- B_c = horizontal breadth (outside) of conduit,
- K = ratio of active lateral unit pressure to vertical and sides of ditch,
- μ = $\tan \phi$ = coefficient of internal friction of fill material,
- μ' = $\tan \phi'$ = coefficient of friction between fill material and sides of ditch,

p = projection ratio, the vertical distance from natural ground surface to the top of the structure divided by the structure height,

p' = projection ratio in negative projection or imperfect ditch installation, the depth of the ditch divided by its width,

r_{sd} = settlement ratio defined by

$$r_{sd} = \frac{(s_m + s_g) - (s_f + d_c)}{s_m}$$

where

s_m = compression strain of the side columns of soil of height pB_c ,

s_g = settlement of the natural ground surface adjacent to the conduit,

s_f = settlement of the conduit into its foundation,

d_c = shortening of the vertical height of the conduit.

In order to use Marston-Spangler's equations, it is essential to predetermine the value of the settlement ratio. Although the settlement ratio, r_{sd} , is a rational quantity in the development of the load formula, it is very difficult to predetermine the actual value that will be developed in a specific case. Spangler and Handy^[31] recommended design values of the settlement ratio based on observations of the performance of actual culverts under embankments, as shown in Table 3.1.

3.2 Current Design Specifications

The current AASHTO Standard Specifications for Highway Bridges^[6] stipulate dead loads and wheel loads on the culverts through earth fills in Section 6, "Culverts," and culvert design guidelines are provided in Section 17, "Soil-Reinforced Concrete Structure Interaction Systems." Vertical and horizontal earth pressures on culverts may be computed by recognized or

Table 3.1: Design Values of Settlement Ratio

Conditions	Settlement Ratio
Rigid culvert on foundation of rock or unyielding soil	+1.0
Rigid culvert on foundation of ordinary soil	+0.5 ~ +0.8
Rigid culvert on foundation of material that yields with respect to adjacent natural ground	0.0 ~ +0.5
Flexible culvert with poorly compacted side fills	-0.4 ~ 0.0
Flexible culvert with well-compacted side fills	-0.2 ~ +0.2

Table 3.2: Values of C_c in Terms of H/B_c

Incomplete Projection Condition $K\mu = 0.19$		Incomplete Ditch Condition $K\mu = 0.13$	
$r_{sd}p$	Equation	$r_{sd}p$	Equation
+0.1	$C_c = 1.23 H/B_c - 0.02$	-0.1	$C_c = 0.82 H/B_c + 0.05$
+0.3	$C_c = 1.39 H/B_c - 0.05$	-0.3	$C_c = 0.69 H/B_c + 0.11$
+0.5	$C_c = 1.50 H/B_c - 0.07$	-0.5	$C_c = 0.61 H/B_c + 0.20$
+0.7	$C_c = 1.59 H/B_c - 0.09$	-0.7	$C_c = 0.55 H/B_c + 0.25$
+1.0	$C_c = 1.69 H/B_c - 0.12$	-1.0	$C_c = 0.47 H/B_c + 0.40$
+2.0	$C_c = 1.93 H/B_c - 0.17$		

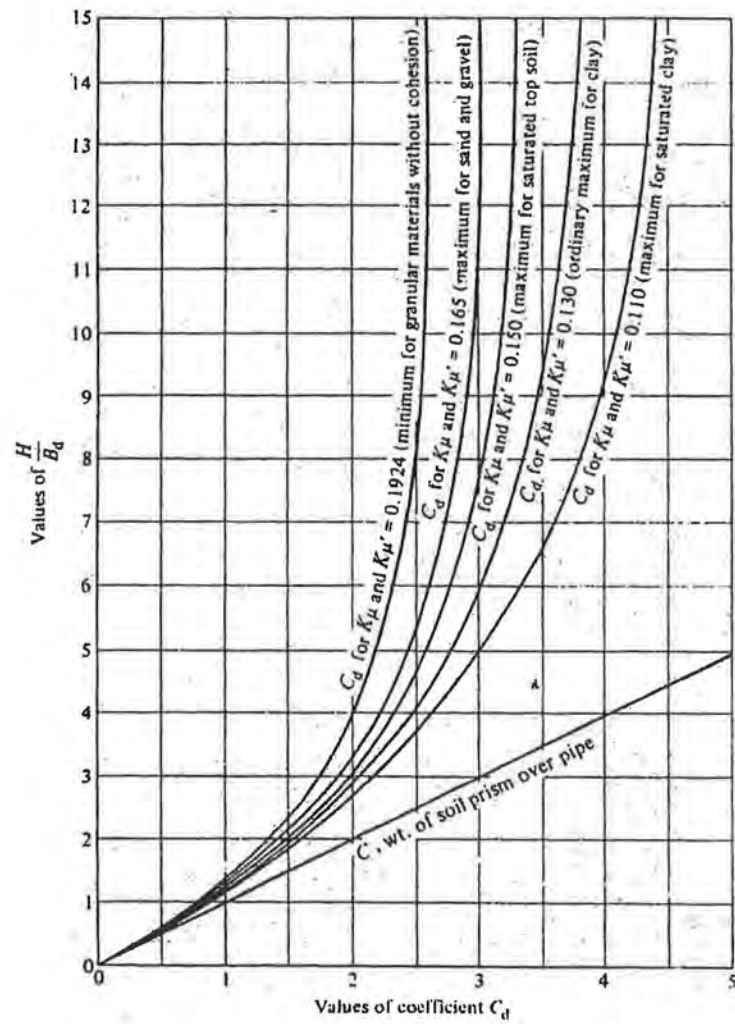


Figure 3.4: Diagram for Coefficient C_d for Ditch Conduits

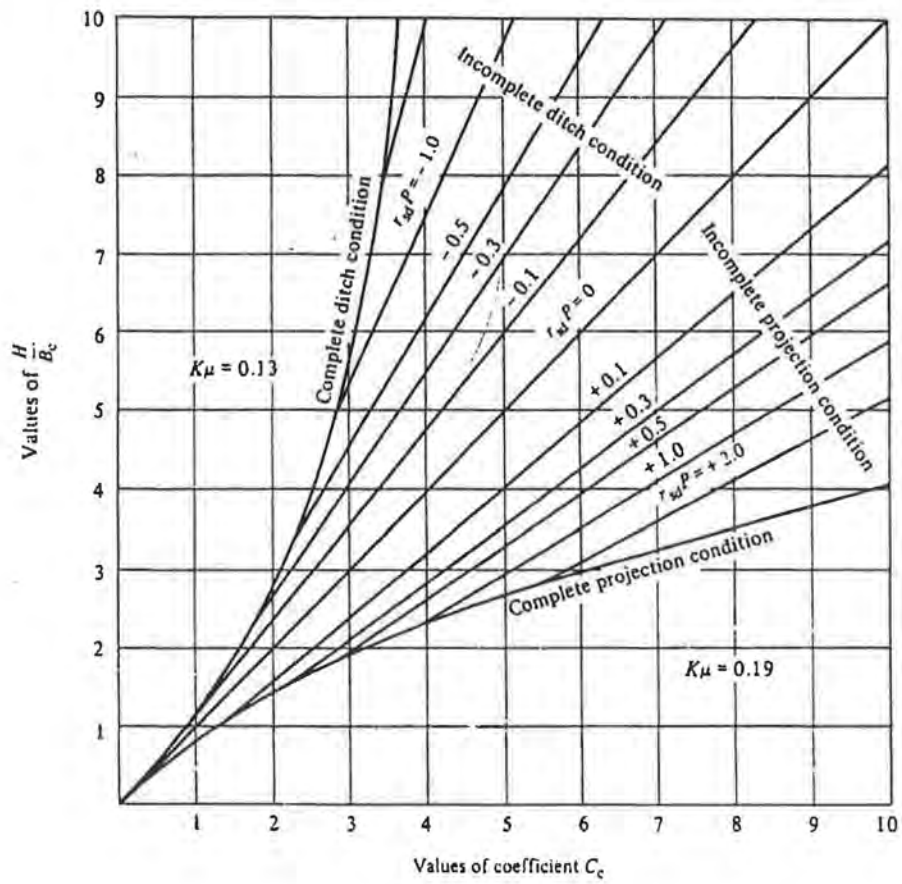


Figure 3.5: Diagram for Coefficient C_e for Positive Projecting Conduits

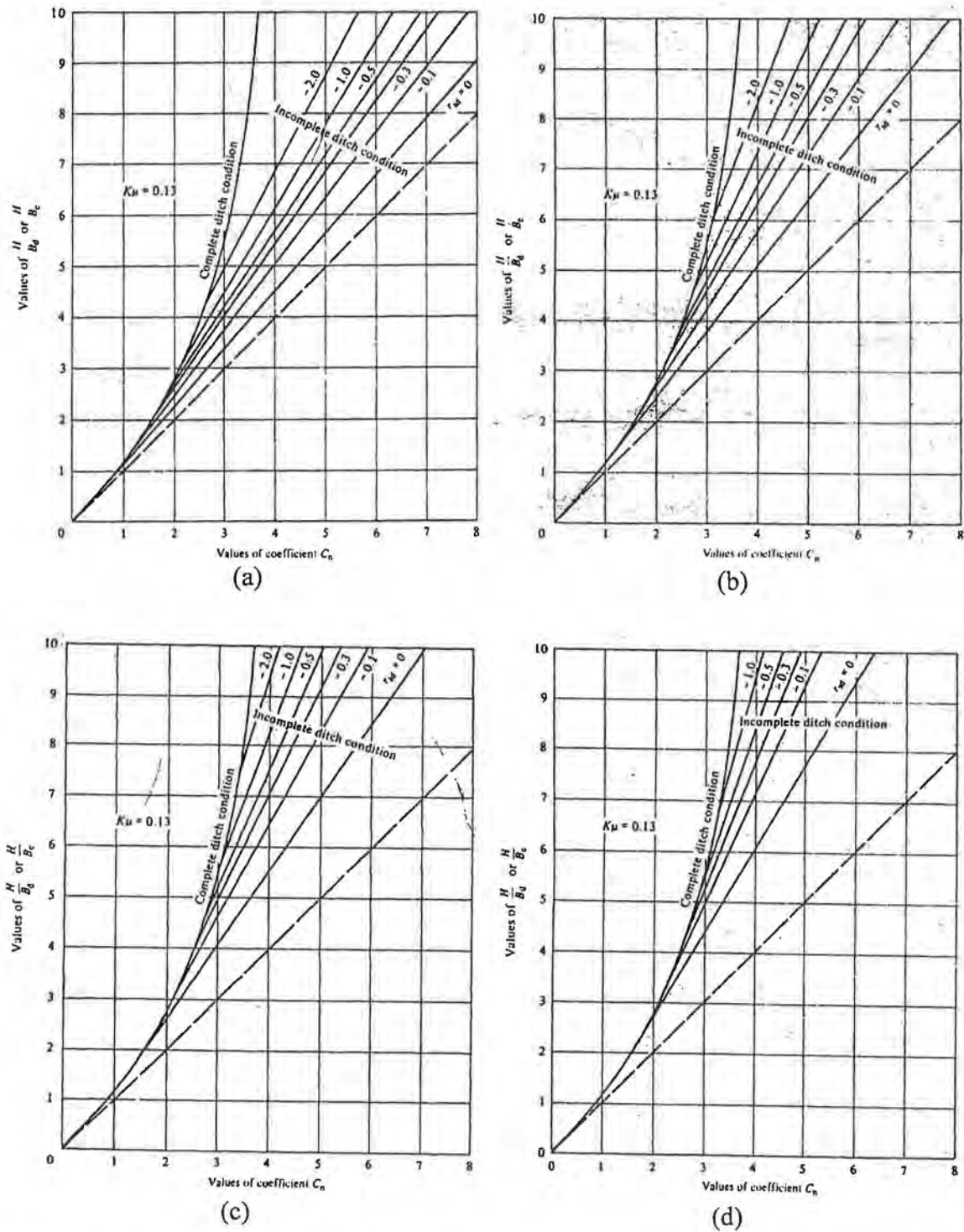


Figure 3.6: Diagrams for Coefficient C_n for Negative Projection Conduits and Imperfect Ditch Conditions (a) $p' = 0.5$ (b) $p' = 1.0$ (c) $p' = 1.5$ (d) $p' = 2.0$

approximately documented analytical techniques based on the principles of soil mechanics and soil structure interaction, or design pressure can be calculated as being the results of an equivalent fluid weight as follows:

Culvert in trench, or culvert untrenched on yielding foundation

A. Rigid culverts, except reinforced concrete boxes:

(1) For vertical earth pressure 120 pcf

For lateral earth pressure 30 pcf

(2) For vertical earth pressure 120 pcf

For lateral earth pressure 120 pcf

B. Reinforced concrete boxes:

(1) For vertical earth pressure 120 pcf

For lateral earth pressure 30 pcf

(2) For vertical earth pressure 120 pcf

For lateral earth pressure 60 pcf

C. Flexible Culverts

For vertical earth pressure 120 pcf

For lateral earth pressure 120 pcf

Culvert untrenched on unyielding foundation

A special analysis is required.

The effects of soil-structure interactions can be taken into account based on the design earth cover, sidefill compaction, and bedding characteristics. These parameters may be

determined by a soil-structure interaction analysis of the system. The loads given above may be used if they are multiplied by a soil-structure interaction factor, F_e , that accounts for the type and conditions of installation, so that the total earth load, W_e , on the reinforced concrete box section, either cast-in-place or precast, is:

$$W_e = F_e w B_c H \quad (3.4)$$

F_e may be determined by the Marston-Spangler theory of earth loads, as follows:

Embankment Installation

$$F_{e1} = 1 + 0.20 \frac{H}{B_c} \quad (3.5)$$

Trench Installation

$$F_{e2} = \frac{C_d B_d^2}{H B_c} \quad (3.6)$$

where H = backfill height;

B_c = width of the structure;

B_d = horizontal width of trench installation; and

C_d = load coefficient for trench installation.

F_{e1} need not be greater than 1.15 for installations with compacted fill at the sides of the box section, and need not be greater than 1.4 for installations with uncompacted fill at the sides of the box section. The load coefficient for trench installation, C_d , is included in Equation 3.2 as a variable. Although the current AASHTO Equation is based on the Marston-Spangler theory, the load coefficient, C_d , based on the Marston theory, remains the same. Load coefficients for trench installation, C_d , provided in the graphical diagram for several soil types in AASHTO, are identical to those of the Marston-Spangler theory as described in the previous section, Figure 3.4.

The maximum value of F_{e2} need not exceed F_{e1} .

3.3 Duncan Soil Model

Kondner^[14] has shown that the nonlinear stress-strain curves of both clay and sand may be approximated by a hyperbola with a high degree of accuracy. This hyperbola can be represented by an equation of the form:

$$(\sigma_1 - \sigma_3) = \frac{\varepsilon}{a + b\varepsilon} \quad (3.7)$$

where σ_1, σ_3 = the maximum and minimum principal stresses, respectively; ε = the axial strain; and a, b = constants whose values may be determined experimentally. Both of these constants a and b have readily discernible physical meanings. As shown in Figures 3.7 and 3.8, a is the reciprocal of the initial tangent modulus, E_t , and b is the reciprocal of the asymptotic value of the stress difference which the stress-strain curve approaches at infinite strain $(\sigma_1 - \sigma_3)_{ult}$. The values of the coefficients, a and b , may be determined readily if the stress-strain data are plotted on transformed axes, as shown Figure 3.8. When Equation 3.7 is rewritten in the following form:

$$\frac{\varepsilon}{(\sigma_1 - \sigma_3)} = a + b\varepsilon \quad (3.8)$$

it may be noted that a and b are the intercept and the slope of the resulting straight line, respectively. By plotting stress-strain data in the form shown in Figure 3.8, it is straightforward to determine the values of the parameters, a and b , corresponding to the best fit between a hyperbola and the test data. It is commonly found that the asymptotic value of $(\sigma_1 - \sigma_3)$ is larger than the compressive strength of the soil by a small amount, because the hyperbola remains below the asymptotic at all finite values of strain. The asymptotic value may be related to the compressive strength, however, by means of a factor R_f :

$$(\sigma_1 - \sigma_3)_f = R_f (\sigma_1 - \sigma_3)_{ult} \quad (3.9)$$

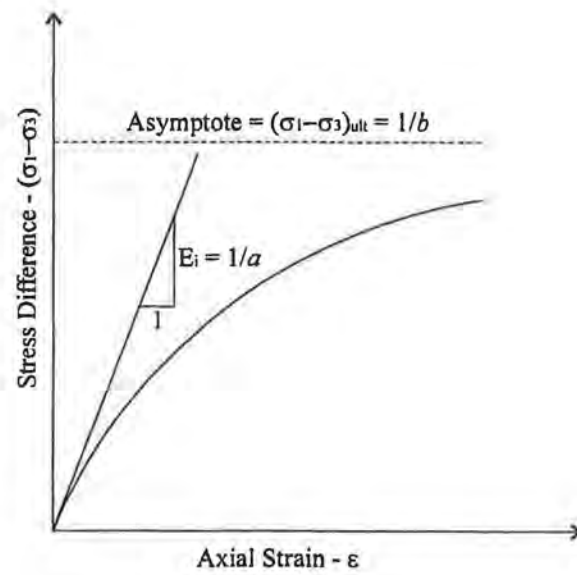


Figure 3.7: Hyperbolic Stress-Strain Curve

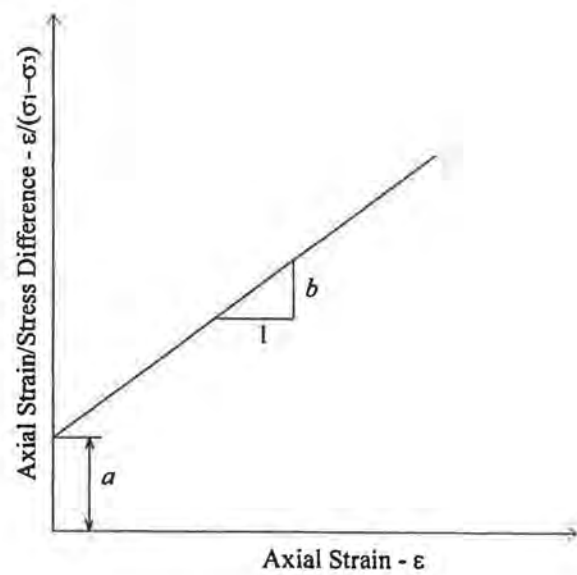


Figure 3.8: Transformed Hyperbolic Stress-Strain Curve

where $(\sigma_1 - \sigma_3)_f$ = the compressive strength, or stress difference at failure; $(\sigma_1 - \sigma_3)_{ult}$ = the asymptotic value of stress difference; and R_f = the failure ratio, which always has a value less than unity. For a number of different soils, the value of R_f has been found to be between 0.75 and 1.00, and is essentially independent of confining pressure. By expressing the parameters a and b in terms of the initial tangent modulus value and the compressive strength, Equation 3.7 may be rewritten as

$$(\sigma_1 - \sigma_3) = \frac{\varepsilon}{\left[\frac{1}{E_i} + \frac{\varepsilon R_f}{(\sigma_1 - \sigma_3)_f} \right]} \quad (3.10)$$

This hyperbolic representation of stress-strain curves has been found to be fairly useful in representing the nonlinearity of soil stress-strain behavior. Except in the case of unconsolidated-undrained tests on saturated soils, both the tangent modulus value and the compressive strength of soils have been found to vary with the confining pressure employed in the tests. Experimental studies by Janbu^[15] have shown that the relationship between tangent modulus and confining pressure may be expressed as

$$E_i = K \cdot p_a \left(\frac{\sigma_3}{p_a} \right)^n \quad (3.11)$$

where E_i = the initial tangent modulus; σ_3 = the minimum principal stress; p_a = atmospheric pressure expressed in the same pressure units as E_i and σ_3 ; K = a modulus number; and n = the exponent determining the rate of variation of E_i with σ_3 .

If it is assumed that failure will occur with no change in the value of σ_3 , the relationship between compressive strength and confining pressure may be expressed conveniently in terms of the Mohr-Coulomb failure criterion as

$$(\sigma_1 - \sigma_3)_f = \frac{2c \cos \phi + 2\sigma_3 \sin \phi}{1 - \sin \phi} \quad (3.12)$$

where c, ϕ = the Mohr-Coulomb strength parameters. Equations 3.11 and 3.12, in combination with Equation 3.4, provide a means of relating stress to strain and confining pressure using the five parameters K, n, c, ϕ , and R_f . Nonlinear, stress-dependent stress-strain behavior may be approximated in finite element analyses by assigning different modulus values to each of the elements into which the soil is subdivided for purposes of analysis. The modulus value assigned to each element is selected on the basis of the stresses or strains in each element. Because the modulus values depend on the stresses, and the stresses in turn depend on the modulus values, it is necessary to make repeated analyses to ensure that the modulus values and stress conditions correspond for each element in the system.

The stress-strain relationship expressed by Equation 3.10 may be employed in incremental stress analyses because it is possible to determine the value of the tangent modulus corresponding to any point on the stress-strain curve. If the value of the minimum principal stress is constant, the tangent modulus, E_t , may be expressed as:

$$E_t = \frac{\partial(\sigma_1 - \sigma_3)}{\partial \varepsilon} \quad (3.13)$$

If Equation 3.13 is then differentiated, and the strain, ε , derived from Equation 3.10, the initial tangent modulus, E_t , in Equation 3.10 and the compressive strength, $(\sigma_1 - \sigma_3)_f$, in Equation 3.12 are substituted into the result of the differentiation, the following expression is obtained for the tangent modulus:

$$E_t = \left[1 - \frac{R_f (1 - \sin \phi) (\sigma_1 - \sigma_3)}{2c \cos \phi + 2\sigma_3 \sin \phi} \right]^2 K p_a \left(\frac{\sigma_3}{p_a} \right)^n \quad (3.14)$$

For the hyperbolic stress-strain relationships, the same value for the unloading-reloading modulus, E_{ur} , is used for both unloading and reloading. The value of E_{ur} is related to the confining pressure by an equation of the same form as Equation 3.11:

$$E_{ur} = K_{ur} p_a \left(\frac{\sigma_3}{p_a} \right)^n \quad (3.15)$$

where K_{ur} is the unloading-reloading modulus number. The value of K_{ur} is always larger than the value of K (for primary loading). K_{ur} may be 20% greater than K for stiff soils such as dense sands. For soft soils such as loose sands, K_{ur} may be three times as large as K . The value of the exponent n is always very similar for primary loading and unloading, and in the hyperbolic relationships it is assumed to be the same.

The value of the tangent Poisson's ratio may be determined by analyzing the volume changes that occur during a triaxial test. For this purpose it is convenient to calculate the radial strains during the test using the equation

$$\varepsilon_r = \frac{1}{2}(\varepsilon_v - \varepsilon_a) \quad (3.16)$$

where ε_v and ε_a are the volumetric and axial strains. Taking compressive strains as positive, the value of ε_a is positive and the value of ε_r is negative. The value of ε_v may be either positive or negative. If the variation of ε_a versus ε_r is plotted, as shown in Figure 3.9, the resulting curve can usually be reasonably accurately represented by a hyperbolic equation of the form:

$$\varepsilon_a = \frac{-\varepsilon_r}{v_i - d\varepsilon_r} \quad (3.17)$$

As shown in Figures 3.9 and 3.10, this equation may be transformed as follows:

$$-\frac{\varepsilon_r}{\varepsilon_a} = v_i - d\varepsilon_r \quad (3.18)$$

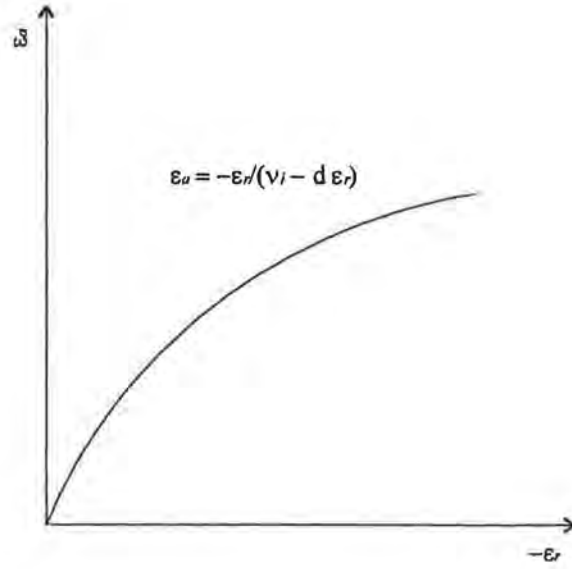


Figure 3.9: Hyperbolic Axial Strain-Radial Strain Curve

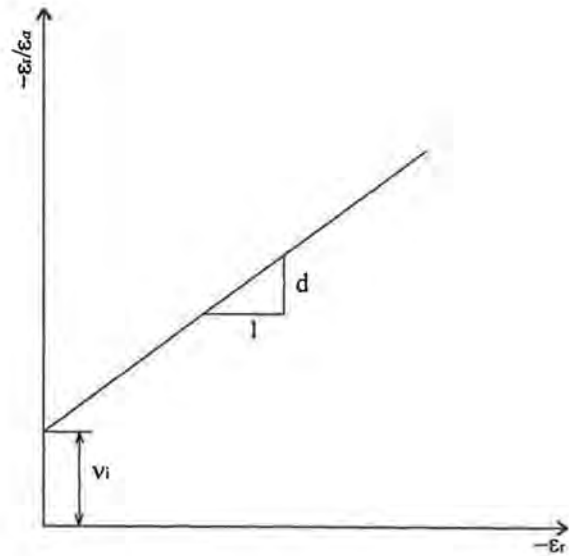


Figure 3.10: Transformed Hyperbolic Axial Strain-Radial Strain Curve

In Equation 3.18, ν_i is the initial Poisson's ratio at zero strain and d is a parameter representing the change in the value of Poisson's ratio with radial strain. For saturated soils under undrained conditions, there is no volume change and ν_i is equal to one half for any value of confining pressure. This variation of ν_i with respect to σ_3 may be expressed by the equation

$$\nu_i = G - F \log_{10} \left(\frac{\sigma_3}{p_a} \right) \quad (3.19)$$

where G is the value of ν_i at a confining pressure of one atmosphere, and F is the reduction in ν_i for a ten-fold increase in σ_3 . After differentiating Equation 3.17 with respect to ϵ_r , substituting Equation 3.19, and eliminating the strain using Equations 3.8 to 3.11, the tangent value of Poisson's ratio may be expressed in terms of the stresses as follows:

$$\nu_t = \frac{G - F \log \left(\frac{\sigma_3}{p_a} \right)}{\left\{ 1 - K p_a \left(\frac{\sigma_3}{p_a} \right)^n \left[\frac{d(\sigma_1 - \sigma_3)(1 - \sin \phi)}{2c \cos \phi + 2\sigma_3 \sin \phi} \right] \right\}^2} \quad (3.20)$$

where σ_1 , σ_3 = maximum and minimum principal stresses, respectively; K = modulus number; n = modulus exponent; c = cohesion intercept; ϕ = friction angle; G , F , d = Poisson's ratio parameters; p_a = atmospheric pressure. There are nine parameters involved in the hyperbolic stress-strain and volume change relationships, and the roles of these parameters are summarized in Table 3.3.

Nonlinear volume change can also be accounted for by employing the constant bulk modulus instead of Poisson's ratio parameters. The assumption that the bulk modulus of the soil is independent of stress level (σ_1 - σ_3) and that it varies with confining pressure provides reasonable approximation to the shape of the volume change curves. According to the theory of

Table 3.3: Summary of the Hyperbolic Parameters

Parameter	Name	Function
K, K_{ur}	Modulus number	Relate E_i and E_{ur} to σ_3
n	Modulus exponent	
c	Cohesion intercept	Relate $(\sigma_1 - \sigma_3)$ to σ_3
ϕ	Friction angle	
R_f	Failure ratio	Relate $(\sigma_1 - \sigma_3)_{ult}$ to $(\sigma_1 - \sigma_3)_f$
G	Poisson's ratio parameter	Value of ν_i at $\sigma_3 = p_a$
F	Poisson's ratio parameter	Decrease in ν_i for ten-fold increase in σ_3
d	Poisson's ratio parameter	Rate of increase of ν_i with strain

elasticity, the value of the bulk modulus is defined by

$$B = \frac{\Delta\sigma_1 + \Delta\sigma_2 + \Delta\sigma_3}{3\varepsilon_v} \quad (3.21)$$

where B is the bulk modulus, $\Delta\sigma_1$, $\Delta\sigma_2$, and $\Delta\sigma_3$ are the changes in the values of the principal stress, and $\Delta\varepsilon_v$ is the corresponding change in volumetric strain. For a conventional triaxial test, in which the deviator stress $(\sigma_1 - \sigma_3)$ increases while the confining pressure is held constant,

Equation 3.21 may be expressed as:

$$B = \frac{(\sigma_1 - \sigma_3)}{3\varepsilon_v} \quad (3.22)$$

The value of bulk modulus for a conventional triaxial compression test may be calculated using the value of $(\sigma_1 - \sigma_3)$ corresponding to any point on the stress-strain curve. When values of B are

calculated from tests on the same soil specimen at various confining pressures, the bulk modulus will usually be found to increase with increasing confining pressure. The variation of B with confining pressure can be approximated by an equation of the form:

$$B = K_b p_a \left(\frac{\sigma_3}{p_a} \right)^m \quad (3.23)$$

where K_b is the bulk modulus number and m is the bulk modulus exponent, both of which are dimensionless. p_a is atmospheric pressure. Experimental study of this soil model, sometimes called the modified Duncan model, has resulted in, for most soils, values of m varying between 0.0 and 1.0. If a bulk modulus is known, the tangent Poisson's ratio can be determined from the basic theory of elasticity by the following equation:

$$\nu_t = \frac{1}{2} - \frac{E_t}{6B} \quad (3.24)$$

Some parameters for this Duncan model are listed in Tables 3.4 and 3.5 from drained and undrained conditions, respectively. These tables have been reproduced from Duncan and his coworkers^[33] and Wong and Duncan.^[10]

Although the hyperbolic relationship outlined previously has proven to be quite useful for a wide variety of practical problems, it has some significant limitations:^[33] 1) Being based on the generalized Hooke's law, the relationships are most suitable for analysis of stresses and movements prior to failure. The relationships are capable of predicting accurately nonlinear relationships between loads and movements, and it is possible to continue the analyses up to the stage where there is local failure in some elements. However, when a stage is reached where the behavior of the soil mass is controlled to a large extent by properties assigned to elements which have already failed, the results will no longer be reliable, and they may be unrealistic in terms of the behavior of real soils at and after failure. 2) The hyperbolic relationships do not include

volume changes due to changes in shear stress, or shear dilatancy. They may, therefore, be limited in the accuracy with which they can be used to predict deformations in dilatant soils, such as dense sands under low confining pressures. The values of tangent Poisson's ratio calculated using Equation 3.13 may exceed 0.5 for some combinations of parameter values and stress values, so it needs to be specified to be less than 0.5 in the computer program. 3) The parameters are not fundamental soil properties, but only values of empirical coefficients that represent the behavior of the soil under a limited range of conditions. The values of the parameters depend on the density of the soil, its water content, the range of pressures used in testing, and the drain conditions. In order that the parameters will be representative of the behavior of the soil under field conditions, the laboratory test conditions must correspond to the field conditions with regard to these factors.

3.4 Tangent Stiffness for Interface Shear Behavior^[34]

The nonlinear stress dependent interface behavior may be represented by means of equations similar to those for Duncan's hyperbolic stress-strain relationships of soils. The nonlinear stress-displacement curves may be approximated by a hyperbola having an equation of the form:

$$\tau = \frac{\Delta_s}{a + b\Delta_s} \quad (3.25)$$

where τ = interface shear stress; Δ_s = interface displacement; and a and b = empirical coefficients whose values are determined experimentally. The values of these coefficients may be determined most readily if the stress-displacement data are transformed to another set of axes, on which the hyperbola plot as straight lines. By transposing Equation 3.25, the following relationship may be obtained:

Table 3.4: Stress-Strain and Strength Parameters for Soils under Drained Conditions

Soil	Group	Soil Description	Compaction					C (TSF)	$\phi_0(\Delta\phi)$	K	n	R_f	G	F	d	K_0	m
			Type	Max. Dry Unit Wt. (PCF)	Opt. W/C	Dry U. Wt. (PCF)	W/C										
GW	GW-1	Conglomerate Rockfill (Netzahu. Dam)				118.9		.00	50(10)	540	.43	.64				135	.34
GW	GW-2	Granitic Gneiss Rockfill (Mica Dam)				123.7		.00	44(9)	210	.51	.64	.25	.09	4.5	100	.34
GW	GW-3	Quartzite Rockfill (Furnas Dam Shell)						.00	49(6)	560	.48	.65	.33	.10	6.4	330	.33
GW	GW-4	Quartzite Rockfill (Furnas Dam Transit.)						.00	53(7)	950	.52	.59	.21	-.01	13.7	470	.52
GW	GW-5	Furnas Dam Transition						.00	50(7)	690	.57	.51	.38	.11	11.2	360	.57
GW	GW-6	Pinzandapan Gravel				132.1		.00	51(9)	690	.45	.59	.14	-.10	27.3	170	.22
GW	GW-7	Diorite Rockfill (El Infiernillo Dam)				105.7		.00	46(9)	340	.28	.71	.18	.04	7.4	52	.18
GP	GP-2	Sandy Gravel (Mica Dam Shell)						.00	41(3)	420	.50	.78	.22	.06	4.7	125	.46
GP	GP-3	Basalt Rockfill				133.8		.00	52(10)	450	.37	.61	.34	.16	4.8	255	.18
GP	GP-6	Silty Sandy Gravel (Oroville Dam)				148.0		.00	53(8)	1300	.40	.72	.33	.07	7.1	900	.22
GP	GP-7	Amphibolite Gravel (Oroville Dam Shell)				152.0		.00	51(6)	1780	.39	.67	.40	.15	15.5	1300	.16
GP	GP-11	Crushed Basaltic Rock (Round Butte Dam)		91.6		99.0	3.2	.00	51(14)	410	.21	.71	.59	.46	2.7	195	0
GP	GP-13	Sandy Gravel (Rowallan Dam)				135.0		.00	58(10)	2500	.21	.75	.35	.17	14.6	1400	0
GC	GC-1	Clayey Gravel (New Hogan Dam Core)		113.0	10.8	107.0	10.8	.28	19(-)	99	.70	.86	.29	.25	4.2	45	0
SW	SW-1	Argillite Rockfill (Pyramid Dam Shell)				111.6		.00	53(9)	1600	.08	.72	.33	.14	5.5	600	0
SW	SW-2	Crushed Olivine Basalt				125.4		.00	55(10)	1000	.22	.70	.31	.13	5.6	390	.14
SW	SW-3	Silty Sand, Some Gravel(Round Butte Dam)	16,450	120.0	13.2	108.7	13.5	.00	38(3)	260	.50	.76	.35	.11	3.0	100	.5
SW	SW-5	Vanato Sandstone (0.5 in. max. size)		118.3		117.5		.00	43(4)	330	.46	.51	.20	.07	6.9	110	.46
SP	SP-3	Glacial Cutwash Sand				112.3		.00	44(4)	190	.70	.57	.37	.16	4.8	190	.35
SP	SP-4A	Sacramento River Sand				89.5		.00	35(2)	430	.27	.84	.42	.21	2.9	230	.02
SP	SP-4B	Sacramento River Sand				94.0		.00	37(2)	410	.69	.90	.35	.17	5.9	260	.15
SP	SP-4C	Sacramento River Sand				97.8		.00	41(5)	1100	.36	.85	.46	.22	8.9	900	0
SP	SP-4D	Sacramento River Sand				103.9		.00	45(7)	1200	.48	.85	.50	.23	11.7	1500	0
SP	SP-5A	Ham River Sand						.00	31(2)	890	.26	.78					
SP	SP-5B	Ham River Sand						.00	47(9)	1100	.57	.86	.69	.29	7.1	2250	0
SP	SP-7A	Poorly Graded Sand (Port Allen Lock)		100.0	13.0	95.5		.00	39(0)	410	.65	.84					
SP	SP-7B	Poorly Graded Sand (Port Allen Lock)				100.0		.00	40(1)	400	.49	.77					
SP	SP-7C	Poorly Graded Sand (Port Allen Lock)		100.0	13.0	105.1		.00	44(3)	750	.77	.83					
SP	SP-12	Coarse to Fine Sand (Round Butte Dam)		100.0	13.0	74.8		.00	39(6)	280	.37	.71	.20	.12	7.7	95	.21
SP	SP-13	Pumicecus Sand (Round Butte Dam)		87.4		84.2	18.0	.00	48(10)	340	.45	.70	.33	.21	6.1	230	.06
SP	SP-14	Pumicecus Sand (Round Butte Dam)		80.7		76.9	25.0	.00	49(12)	650	.38	.77	.28	.07	6.9	380	.05
SP	SP-16A	Fine Silica Sand (Loose)						.00	30(0)	280	.65	.93	.35	.07	3.5	110	.65
SP	SP-16B	Fine Silica Sand (Dense)						.00	37(0)	1400	.74	.90	.32	-.05	28.2	1080	.15
SP	SP-17A	Monterey No. 0 Sand (Cylind. Specimen)						.00	35(0)	920	.79	.96	.37	.12	10.5	465	.32
SP	SP-17B	Monterey No. 0 Sand (Cubical. Specimen)						.00	39(0)	510	.51	.97	.40	.09	8.0	370	.22
SP	SP-17C	Monterey No. 0 Sand (Cylind. Specimen)						.00	45(3)	3200	.78	.92	.34	.12	75.9	1400	.45
SP	SP-17D	Monterey No. 0 Sand (Cubical. Specimen)						.00	47(5)	1500	.76	.91	.48	.30	13.5	1100	.52
SP	SP-18	Basaltic Sand (Round Butte Dam)		120.1	9.5	120.0	9.5	.00	39(13)	1600	.08	.63	.34	.16	12.3	750	0

Table 3.4 (continued)

Soil	Group	Soil Description	Compaction					C (TSF)	$\phi_0(\Delta\phi)$	K	n	R_f	G	F	d	K_s	m
			Type	Max. Dry Unit Wt. (PCF)	Opt. W/C	Dry U. Wt. (PCF)	W/C										
SM	SM-4	Silty Sand (Chatfield Dam)	Std.AASHO	123.0	9.5	116.7	9.4	.00	37(0)	100	1.07	.62					
SM	SM-5	Silty Gravelly Sand (Chatfield Dam)	Std.AASHO	132.0	8.1	124.5	7.53	.00	41(0)	530	.51	.62					
SM	SM-6	Silty Sand w/Pebbles (Round Butte Dam)	16,450	110.6	17.5	108.1	17.5	.00	46(8)	700	.35	.75	.41	.18	6.0	640	0
SM	SM-9	Silty Sand w/Pumice (Round Butte Dam)	16,450	91.7	19.5	88.4	19.0	.00	43(8)	670	.25	.72	.39	.17	6.1	500	0
SM	SM-13	Silty Sand (Round Butte Dam)		105.6	16.4	104.5	15.	.00	36(5)	530	.28	.74	.38	.11	3.9	470	0
SM	SM-16	Silty Sand & Gravel (Round Butte Dam)		109.3	12.9	109.	12.	.00	36(11)	800	.20	.67	.39	.10	4.7	600	0
SM-SC	SM-SC-1A	Silty Clayey Sand (Mica Dam Core)	Std.AASHTO	136.0	9.8	131.1	7.7	.31	33(-)	700	.37	.80	.25	.02	5.5	280	.19
SM-SC	SM-SC-1B	Silty Clayey Sand (Mica Dam Core)	Std.AASHTO	136.0	9.8	134.0	9.7	.85	34(-)	425	.58	.70	.27	.05	7.9	205	.44
SM-SC	SM-SC-1C	Silty Clayey Sand (Mica Dam Core)	Std.AASHTO	136.0	9.8	128.2	11.9	.40	45(-)	160	.81	.63	.26	.04	6.2	65	.81
ML	ML-1	Cannonsville Silt (Undisturbed)				108.0		.00	45(6)	200	1.07	.57	.33	.02	7.3	200	.89
ML	ML-4	Sandy Silty w/Pumice (Round Butte Dam)	16,450	97.0	19.0	92.8	17.7	.00	42(7)	500	.45	.82	.47	.39	6.9	400	0
ML	ML-5	Sandy Silty w/Pumice (Round Butte Dam)	16,450	102.5	16.5	99.2	17.0	.00	36(1)	530	.35	.71	.39	.03	3.2	520	.23
CL	CL-29C	Silty Clay (Canyon Dam)	Harvard	116.2	15.2	111.2	13.1	.17	30(-)	550	.05	.82					
CL	CL-29D	Silty Clay (Canyon Dam)	Harvard	116.2	15.2	116.2	13.3	.59	29(-)	690	.10	.71					
CL	CL-30E	Silty Clay (Canyon Dam)	Harvard	112.8	16.7	115.1	15.2	.51	33(-)	150	.62	.61	.58	.33	2.3	360	0
CL	CL-30F	Silty Clay (Canyon Dam)	Harvard	112.8	16.7	110.0	17.4	.39	30(-)	160	.50	.63	.41	.14	2.5	210	0
CL	CL-34E	Silty Clay (Canyon Dam)	Harvard	105.6	19.8	106.3	19.0	.26	31(-)	130	.59	.72				45	.59

Table 3.5: Stress-Strain and Strength Parameters for Soils under Unconsolidated-Undrained Conditions

Soil	Group	Soil Description	Compaction					C (TSF)	$\phi_u(\Delta\phi)$	K	n	R _r	G	F	d	K _s	m
			Type	Max. Dry Unit Wt. (PCF)	Opt. W/C	Dry U. Wt. (PCF)	W/C										
GC	GC-2A	Sandy Gravel (Oroville Dam Core)	Std.AASHO	138.6	8.1	139.0	8.1	1.50	24	540	.51	.84					
GC	GC-2B	Sandy Gravel (Oroville Dam Core)	Std.AASHO	138.6	8.1	139.0	8.1	10.01	3	190	.95	.97					
SP	SP-8D	Poorly Graded Sand (Rodman Dam)	Mod.AASHO	109.5	11.8	104.0	11.8	.00	37 (4)	590	1.10	.89					
SP	SP-8E	Poorly Graded Sand (Rodman Dam)	Mod.AASHO	109.5	11.8	98.6	11.8	.00	37 (8)	770	-.14	.87					
SP	SP-8F	Poorly Graded Sand (Rodman Dam)	Mod.AASHO	109.5	11.8	110.0	11.2	.00	43 (9)	940	.00	.82					
SP	SP-9D	Poorly Graded Silty Sand (Rodman Dam)	Mod.AASHO	101.1	13.6	101.3	13.6	.00	44 (6)	420	.67	.76					
SP	SP-9E	Poorly Graded Silty Sand (Rodman Dam)	Mod.AASHO	101.1	13.6	96.2	13.3	.00	44 (11)	850	.79	.92					
SP	SP-9F	Poorly Graded Silty Sand (Rodman Dam)	Mod.AASHO	101.1	13.6	92.0	12.4	.00	40 (8)	470	.51	.86					
SM	SM-1	Gravelly Silty Sand (Ball Mountain Dam)	Std.AASHO	122.9	10.0	124.0	9.4	.00	42 (5)	430	.38	.57					
SM	SM-3A	Silty Sand (Somerville Dam)	Std.AASHO	109.1	13.4	109.3	13.4	.00	40 (2)	350	.91	.69					
SM	SM-3B	Silty Sand (Somerville Dam)	Std.AASHO	109.1	13.4	104.1	13.2	.00	40 (6)	420	.84	.75					
SM	SM-3C	Silty Sand (Somerville Dam)	Std.AASHO	109.1	13.4	103.6	16.7	.00	39 (4)	340	.64	.72					
SM-SC	SM-SC-2	Silty Clayey Sand (Hopkinton Dam)	Std.AASHO	129.2	9.2	131.0	8.8	.98	31	320	.35	.86					
SC	SC-2	Clayey Sand (Thomaston Dam)	Std.AASHO	123.3	12.0	122.0	12.0	.92	18	39	.61	.55					
SC	SC-3	Clayey Sand (New Don Pedro Dam Core)	20,000	125.8	9.8	123.2	9.6	2.60	26	3900	-.08	.93	.71	.39	2.5	12000	-.99
SC	SC-5	Clayey Gravelly Sand (Proctor Dam)	Std.AASHO	120.1	11.2	126.0	8.3	1.80	4	510	.37	.63	.78	.03	13.3	250	.00
SC	SC-6A	Clayey Sand (Chatfield Dam)	Std.AASHO	122.0	11.7	116.2	14.7	1.30	0	52	.00	.76					
SC	SC-7B	Clayey Sand (Chatfield Dam)	Std.AASHO	115.0	15.0	110.0	17.0	1.10	0	250	.00	.97					
ML	ML-2A	Sandy Silt (Chatfield Dam)	Std.AASHO	115.0	12.8	108.7	15.6	1.80	19	200	.59	.86					
ML	ML-2B	Sandy Silt (Chatfield Dam)	Std.AASHO	115.0	12.8	109.3	12.7	.39	30	27	1.43	.72					
ML	ML-3A	Sandy Silt (Birch Dam Shell)	Std.AASHO	108.8	13.6	104.0	11.6	.42	31	240	.31	.83					
ML	ML-3B	Sandy Silt (Birch Dam Shell)	Std.AASHO	108.8	13.6	104.0	13.6	.19	31	270	.38	.82					
ML	ML-3C	Sandy Silt (Birch Dam Shell)	Std.AASHO	108.8	13.6	104.0	16.6	.54	27	100	.84	.77					
CL	CL-1A	Silty Clay (Arkabutla Dam)	Std.AASHO	110.0	18.0	108.7	16.7	.53	29	260	.60	.87					
CL	CL-1B	Silty Clay (Arkabutla Dam)	Std.AASHO	110.0	18.0	107.0	19.5	1.20	14	39	.48	.58					
CL	CL-2A	Lean Clay (Monroe Dam)	Std.AASHO	110.5	16.4	107.1	19.1	.95	0	66	.00	.75					
CL	CL-2B	Lean Clay (Monroe Dam)	Std.AASHO	110.5	16.4	104.0	21.2	.42	0	10	.03	.52					
CL	CL-3	Lean Clay (Monroe Dam)	Std.AASHO	106.8	18.0	102.0	21.7	1.00	0	36	.00	.57					
CL	CL-5A	Pittsburg Silty Clay	Mod.AASHO	118.9	13.5	105.4	11.5	.92	31	650	-.68	.90	.29	.20	2.8	190	-.81
CL	CL-5B	Pittsburg Silty Clay	Mod.AASHO	118.9	13.5	109.1	14.3	1.50	17	760	-.14	.93	.31	.09	3.1	240	-.21
CL	CL-5C	Pittsburg Silty Clay	Mod.AASHO	118.9	13.5	109.0	16.8	1.30	6	430	.10	.93	.28	.06	5.8	115	.10
CL	CL-5E	Pittsburg Silty Clay	Mod.AASHO	118.9	13.5	112.7	11.5	1.80	24	2400	-.74	.92	.30	.14	6.2	740	-.96
CL	CL-5F	Pittsburg Silty Clay	Mod.AASHO	118.9	13.5	114.7	14.5	1.90	13	2000	-.30	.97	.25	.16	10.1	460	-.64
CL	CL-5H	Pittsburg Silty Clay	Mod.AASHO	118.9	13.5	108.8	8.7	1.50	32	8900	-1.10	.97	.22	.07	19.7	1900	-1.10
CL	CL-5I	Pittsburg Silty Clay	Mod.AASHO	118.9	13.5	119.3	11.7	3.30	18	5000	-.28	.95	.28	.02	12.4	1400	-.33
CL	CL-6A	Sandy Clay (Birch Dam Core)	Std.AASHO	110.3	14.5	105.0	12.5	.64	29	320	-.21	.80					
CL	CL-6B	Sandy Clay (Birch Dam Core)	Std.AASHO	110.3	14.5	105.0	14.5	.50	29	190	.02	.81					
CL	CL-7A	Sandy Clay (Somerville Dam)	Std.AASHO	107.5	17.2	107.9	17.2	1.00	2	74	.23	.87					
CL	CL-7B	Sandy Clay (Somerville Dam)	Std.AASHO	107.5	17.2	107.2	17.0	1.00	1	68	-.05	.84					

Table 3.5 (continued)

Soil	Group	Soil Description	Compaction					C (TSF)	$\phi_0(\Delta\phi)$	K	n	R _r	G	F	d	K _s	m
			Type	Max. Dry Unit Wt. (PCF)	Opt. W/C	Dry U. Wt. (PCF)	W/C										
CL	CL-10B	Sandy Clay (Somerville Dam)	Std.AASHTO	110.7	15.0	106.5	15.0	.55	22	290	.27	.91					
CL	CL-11A	Sandy Clay (Somerville Dam)	Std.AASHTO	107.5	16.8	100.3	13.5	.78	28	680	-.36	.84					
CL	CL-11B	Sandy Clay (Somerville Dam)	Std.AASHTO	107.5	16.8	106.5	13.3	1.50	25	600	.18	.68					
CL	CL-11C	Sandy Clay (Somerville Dam)	Std.AASHTO	107.5	16.8	102.6	19.3	.74	6	23	.32	.61					
CL	CL-11D	Sandy Clay (Somerville Dam)	Std.AASHTO	107.5	16.8	106.7	16.7	.91	18	280	.60	.93					
CL	CL-11E	Sandy Clay (Somerville Dam)	Std.AASHTO	107.5	16.8	101.5	16.3	.66	20	220	.23	.90					
CL	CL-12A	Sandy Clay (Somerville Dam)	Std.AASHTO	106.1	17.2	105.0	18.6	1.30	8	140	.20	.84					
CL	CL-12C	Sandy Clay (Somerville Dam)	Std.AASHTO	106.1	17.2	101.9	17.1	1.00	13	120	.09	.83					
CL	CL-12D	Sandy Clay (Somerville Dam)	Std.AASHTO	106.1	17.2	103.0	19.7	.80	2	47	.33	.82					
CL	CL-12E	Sandy Clay (Somerville Dam)	Std.AASHTO	106.1	17.2	106.5	13.9	1.50	24	950	-.15	.90					
CL	CL-12F	Sandy Clay (Somerville Dam)	Std.AASHTO	106.1	17.2	108.3	16.9	1.50	8	470	.00	.95					
CL	CL-13A	Sandy Clay (Somerville Dam)	Std.AASHTO	104.9	17.6	98.7	20.8	.67	4	75	.44	.88					
CL	CL-13B	Sandy Clay (Somerville Dam)	Std.AASHTO	104.9	17.6	104.9	14.8	1.80	23	840	-.19	.84					
CL	CL-13C	Sandy Clay (Somerville Dam)	Std.AASHTO	104.9	17.6	101.2	17.4	1.20	12	270	.06	.87					
CL	CL-13D	Sandy Clay (Somerville Dam)	Std.AASHTO	104.9	17.6	100.5	14.2	1.40	29	1100	-.36	.83					
CL	CL-13E	Sandy Clay (Somerville Dam)	Std.AASHTO	104.9	17.6	104.4	17.5	1.40	13	410	.15	.87					
CL	CL-14	Lean Clay (Clinton Dam)	Std.AASHTO	103.0	21.2	98.0	24.0	.77	2	57	.43	.86					
CL	CL-16C	Lean Clay (Clinton Dam)	Std.AASHTO	105.0	20.2	99.7	22.9	.97	1	110	.43	.90					
CL	CL-17A	Lean Clay (Clinton Dam)	Std.AASHTO	101.0	20.1	99.1	22.7	1.10	2	100	.27	.89					
CL	CL-17B	Lean Clay (Clinton Dam)	Std.AASHTO	101.0	20.1	98.1	23.9	.99	1	160	.54	.97					
CL	CL-17C	Lean Clay (Clinton Dam)	Std.AASHTO	101.0	20.1	98.9	22.7	1.10	3	130	.46	.91					
CL	CL-19A	Lean Clay (Clinton Dam)	Std.AASHTO	102.0	19.9	96.8	22.7	.78	2	53	.41	.85					
CL	CL-24A	Sandy Clay (Chatfield Dam)	Std.AASHTO	104.0	19.3	97.6	23.4	1.20	0	240	.00	.95					
CL	CL-25A	Sandy Clay (Chatfield Dam)	Std.AASHTO	113.0	15.1	107.4	18.1	.95	0	160	.00	.93					
CL	CL-28	Sandy Clay (Proctor Dam)	Std.AASHTO	115.0	14.6	114.8	12.2	1.60	12	150	.16	.79					
CL	CL-29A	Silty Clay (Canyon Dam)	Harvard	116.2	15.2	110.9	13.0	2.00	20	440	.17	.85					
CL	CL-29B	Silty Clay (Canyon Dam)	Harvard	116.2	15.2	115.8	13.1	2.50	20	440	.34	.86					
CL	CL-30A	Silty Clay (Canyon Dam)	Harvard	112.8	16.7	111.0	16.2	1.00	16	110	.94	.91					
CL	CL-30B	Silty Clay (Canyon Dam)	Harvard	112.8	16.7	112.2	16.6	1.40	11	67	.71	.77					
CL	CL-30D	Silty Clay (Canyon Dam)	Harvard	112.8	16.7	110.3	17.3	1.00	9	37	.37	.65					
CL	CL-33B	Silty Clay (Canyon Dam)	Harvard	108.8	18.0	106.3	16.2	2.20	3	71	1.06	.98					
CH	CH-1	Fat Clay (Clinton Dam)	Std.AASHTO	94.0	26.5	90.0	28.8	.61	4	92	.21	.89					
CH	CH-3A	Fat Clay (Montee Dam)	Std.AASHTO	95.5	26.5	89.3	31.1	.37	0	21	.00	.65					
CH	CH-3B	Fat Clay (Montee Dam)	Std.AASHTO	95.5	26.5	92.6	28.6	.51	1	67	.02	.79					
CH	CH-4	Fat Clay (Montee Dam)	Std.AASHTO	100.0	22.7	96.4	26.5	.63	1	65	.14	.77					
CH	CH-5A	Fat Clay (Chatfield Dam)	Std.AASHTO	95.0	24.4	90.3	27.4	1.20	0	36	.72	.91					
CH	CH-5B	Fat Clay (Chatfield Dam)	Std.AASHTO	95.0	24.4	90.7	24.4	1.50	2	52	.66	.89					

$$\frac{\Delta_s}{\tau} = a + b\Delta_s \quad (3.26)$$

If values of Δ_s/τ are plotted against values of Δ_s , the resulting variation will be a straight line while the shear stress varies hyperbolically with respect to displacement. It may be readily seen that coefficient a is the intercept, and coefficient b is the slope of the straight line on the transformed plot. The reciprocal of coefficient a is the initial slope of the shear stress-displacement curve, and is analogous to the initial tangent modulus of a stress-strain curve. This quantity, called the initial shear stiffness, K_{si} , has units of force per cubic length, or pounds per cubic inch. The reciprocal of coefficient b is the asymptote approached by the shear stress-displacement curve at very large values of displacement. This asymptotic shear stress, τ_{ult} , is larger than the peak value of shear stress or strength of the interface, τ_f . It has been found to be convenient to express τ_{ult} in terms of τ_f by means of a failure ratio, R_f , as follows:

$$\tau_f = R_f \tau_{ult} \quad (3.27)$$

The values of R_f are found to range from 0.82 to 0.95. The initial shear stiffness, K_{si} , and the shear strength of the interface depend on the value of normal stress on the interface. The variation of initial shear stiffness with normal stress may be expressed using an equation of the general form suggested by Janbu:^[15]

$$K_{si} = K_I \gamma_w \left(\frac{\sigma_n}{p_a} \right)^n \quad (3.28)$$

where K_I = dimensionless stiffness number; n = stiffness exponent; γ_w = unit weight of water expressed in the same units as K_{si} ; and p_a = atmospheric pressure expressed in the same units as the normal stress σ_n . Clough and Duncan^[34] showed in their experimental research that the stiffness number has a value of 75,000 and the stiffness exponent has a value of 1.0. The shear

strength of the interface is proportional to normal stress, and the strength characteristics of the interface may be expressed in terms of the angle of wall friction, δ , as follows:

$$\tau_f = \sigma_n \tan \delta \quad (3.29)$$

Substituting Equations 3.28, and 29 into Equation 3.26, the variation of shear stress with respect to displacement may be expressed in terms of the parameters K_I , n , R_f , and δ as follows:

$$\tau = \frac{\Delta_s}{\frac{I}{K_I \gamma_w \left(\frac{\sigma_n}{p_a} \right)^n} + \frac{R_f \Delta_s}{\sigma_n \tan \delta}} \quad (3.30)$$

Using Equation 3.30, variations of shear stress with displacement may be calculated for any value of normal stress in the range of values employed in the tests. Thus, the nonlinear, stress-dependent behavior of the interface may be expressed in terms of the four parameters K_I , n , R_f , and δ , whose values may be readily determined from a series of interface shear tests.

A way of performing approximate nonlinear analyses of soil behavior is to perform incremental analyses, changing the soil property values as appropriate for each stage of the analysis. Tangent stiffness values, representing the slope of a tangent to the shear stress-displacement curve, may be readily employed in such analysis. Differentiating Equation 3.30 with respect to Δ_s , and eliminating Δ_s from the resulting equation, the tangent stiffness value may be expressed as:

$$K_{st} = \frac{\partial \tau}{\partial \Delta_s} = K_I \gamma_w \left(\frac{\sigma_n}{p_a} \right)^n \left(1 - \frac{R_f \tau}{\sigma_n \tan \delta} \right)^2 \quad (3.31)$$

The tangent stiffness values are calculated using Equation 3.31 and used for the spring constants in interface modeling in the next chapter.

CHAPTER 4

SOIL-STRUCTURE MODELING

4.1 Finite Element Model

ABAQUS Version 5.8,^[30] a general-purpose commercial finite element code, and ISBILD,^[24] a computer program for analysis of static stresses and movements in embankments, were used to investigate the pressure intensity for the concrete box culverts under high backfill based on the soil-structure interaction. CANDE-89^[28] (Culvert ANalysis and DEsign Program, available from McTrans, Univ. of FL) was also run for verification/comparison. In both ABAQUS and ISBILD codes, the small displacement theory is assumed, and the incremental layered construction scheme is incorporated. The ISBILD code incorporates nonlinear incremental finite element method employing the hyperbolic stress-strain relationship and incremental analysis procedures based on plane strain elements. Plane strain linear elastic elements are used in ABAQUS. For constant material properties of ABAQUS input, tangent Young's modulus and tangential Poisson's ratio are evaluated for each construction layer using Equations 3.14 and 3.20 given in Chapter 3 by assuming that soil layers are in principal states without any live loads, i.e., for constant material property values of the i^{th} incremental soil layer's element group, the following principal stresses are assumed:

$$\sigma_1^{(i)} = \gamma_i (H_i / 2) + \sum_{j=i+1}^n \gamma_j H_j \quad \text{and} \quad \sigma_3^{(i)} = K \sigma_1^{(i)} \quad (4.1)$$

where H_i , γ_i = depth and density of i^{th} soil layer (numbering from the bottom to the top of the backfill or original ground). These values are substituted for the principal stresses in Equations 3.14 and 3.20 for the evaluation of tangent Young's modulus and tangential Poisson's ratio. The coefficient of lateral pressure, K , is 0.5. Adding each element group to previously existing

element groups for incremental procedures can be performed by using the "element remove and reactivation" option of ABAQUS.

Taking advantage of the symmetry of the concrete culvert and soil, only one half of the structure and the surrounding soil were modeled. Exactly the same geometry mesh, boundary conditions, and number of construction layers were considered in both ABAQUS and ISBILD programs. It is known that the width of half the soil-structure finite element model affects the analysis results.^[16] To examine the effect of the width of soil layers in the finite element model, a number of finite element models with constant height but different widths were run using ABAQUS. Figure 4.2 shows that soil-structure interaction factors as the ratio of half soil layer width to box culvert width, W_H/B_c , varies. It also shows that the value of the soil-structure interaction factor increases until W_H/B_c reaches approximately six or seven, after which no significant increase is shown. In the one half model, the width of soil layer was taken to be seven times the culvert width because it has been shown in a series of exploratory models that any wider widths are not likely to affect the soil pressure distribution in the embankment installations. The width of in-situ soil was taken to be six times the culvert width. The depth of the yielding foundation was taken to be four times the culvert height and an unyielding foundation was modeled by placing pin-ended boundaries immediately under the structure base. A typical soil-structure finite element model is shown in Figure 4.1, and incremental sequences considered for the embankment and trench installations are illustrated in Figure 4.3. Heights of backfill varied from 50 feet to 200 feet for the embankment installations, and from 50 feet to 150 feet for the trench installations on both yielding and unyielding foundations. The constant exterior dimensions of the culvert were 8 feet by 8 feet, with 12-inch thick walls and slabs. In the case where the height of the backfill exceeds 150 feet, the thickness of walls and slabs may

have to be increased to resist the induced shearing force near the wall face. It was found from a series of numerical analyses that the thickness of the culvert did not significantly alter the pressure distribution at the top of the culvert, having an effect in the order of 1%. This phenomenon may be explained that the relative displacement at the mid-span of the top slab is not greatly affected as the thickness of the wall is varied. It should be noted that the displacement at the top of the walls remains practically the same regardless of the different values of the wall thickness used. As the pressure distribution at the top slab of a culvert is a function of the relative vertical displacement of the slab, only small differences in the soil pressure distribution would be expected. The modulus of elasticity of concrete box culverts was computed to be 3,605 ksi, assuming the 28-day strength of concrete, f'_c , is 4,000 psi and Poisson's ratio, ν , is 0.25 for soil layers that follow Duncan's model.

4.2 Soil-Structure Interface Modeling

To investigate the effect of the soil-structure interface between the exterior box culvert wall and the filling soil, two different methods were considered; one was a shear interface model and the other a spring interface model.

For the first modeling technique for the soil-culvert interface, the shear resistant capacity of the soil elements adjacent to the sidewalls of the concrete box culvert was modified to look for a possible slippage between the culvert web and soil elements. One of the extreme cases provided a very low value for the modulus. A linear elastic analysis was run on ABAQUS and CANDE-89 to examine the effects of interface behaviors. In the ABAQUS analysis, very narrow elements (1/20th of the width of the wall element) were placed between the culvert outside wall and the backfills. The value of Young's modulus of the interface element was varied from 1/1,000th to 1/10th of the filling material. The unit weight was set to be 1/1,000th of

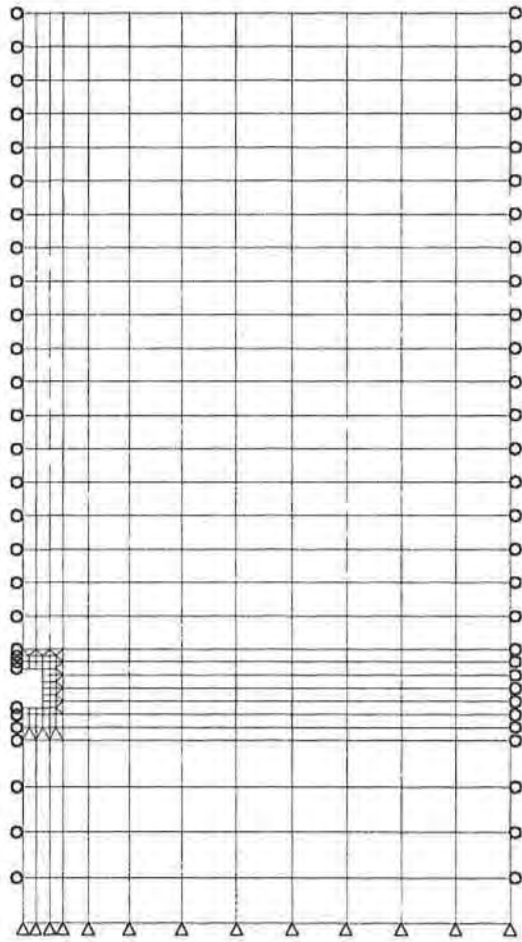


Figure 4.1: Typical Soil-Structure Finite Element Model

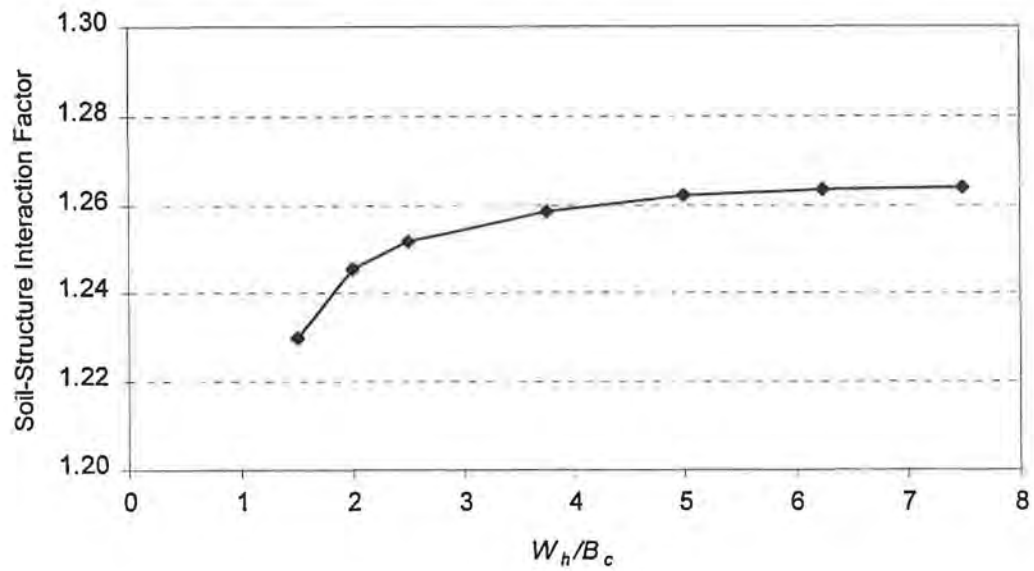


Figure 4.2: Soil-Structure Interaction Factor vs. W_h/B_c

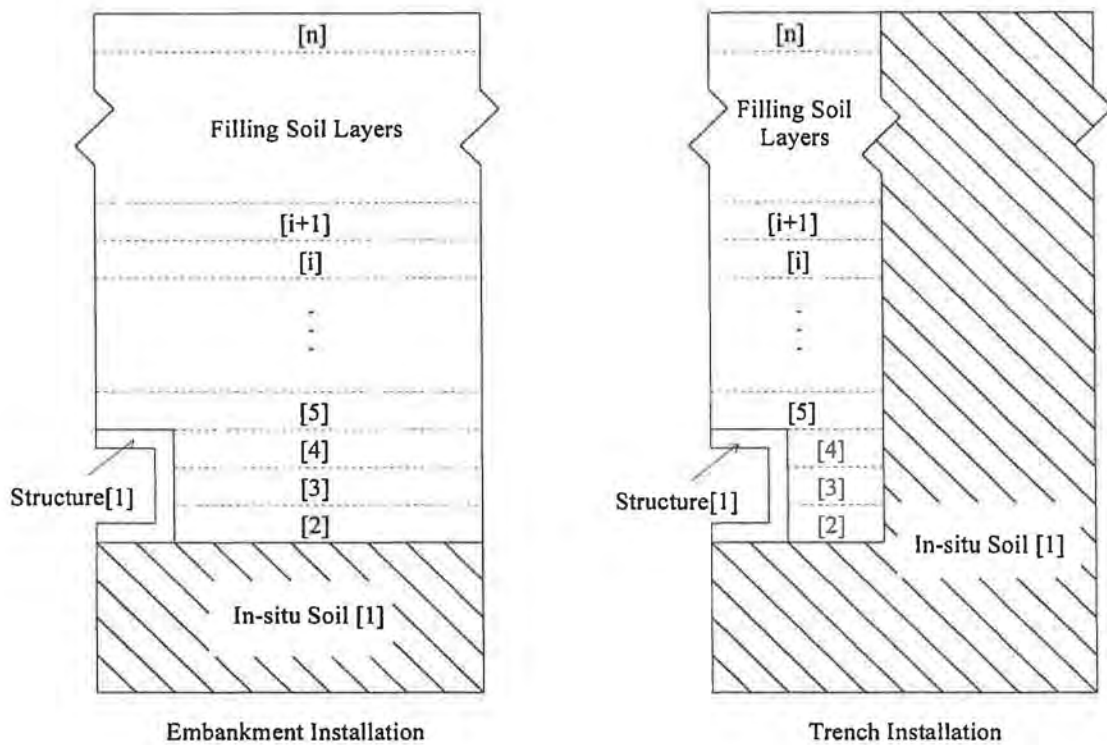


Figure 4.3: Incremental Sequences for Embankment and Trench Installation

soil. The interior dimensions of the concrete box culvert were taken to be 4 ft wide and 4 ft high. The thickness of the flange and web were 10 in. The height of backfill was 7.25 ft measured from the top of the top slab. In the CANDE-89 program, the interface behavior was taken into account by treating constraint equations in the finite formulation based on a generalized principle of virtual work. Table 1 shows the results of analyses of different shear interface models from ABAQUS and CANDE-89, and the differences are calculated on the basis of the value from CANDE-89.

For the other interface model, spring elements were placed at the nodes between the exterior wall and the adjacent soil elements, as shown in Figure 4.4. On the vertical interface line, two nodes for every geometric point were defined when constructing the finite element mesh, one for the culvert wall element and the other for the soil element. A very small spring constant was assigned for the vertical degree of freedom. To prevent extension and compression modes of interface behavior, a very large spring constant was assigned for the horizontal degree

Table 4.1: Soil-Structure Interface Factor Comparison

ABAQUS Shear Interface Model (Difference)	$E_{\text{interface element}}/E_{\text{soil}}$					
	1	1/10	1/20	1/50	1/100	1/1,000
	1.1125 (-1.00 %)	1.1288 (0.45 %)	1.1364 (1.13 %)	1.1488 (2.23 %)	1.1577 (3.03 %)	1.1740 (4.48 %)
ABAQUS Spring Interface Model (Difference)	1.1253 (0.14 %)					
CANDE-89	1.1237					

of freedom.

Applying tangent stiffness for shear interface movement (vertical), presented in the form of K_{st} in Section 3.4, each spring constant for a two elements system (at the top and bottom of the culvert element and the soil element as shown in Figure 4.4 (b)) may be written as:

$$k = \frac{d}{2} K_{st} \quad (4.1)$$

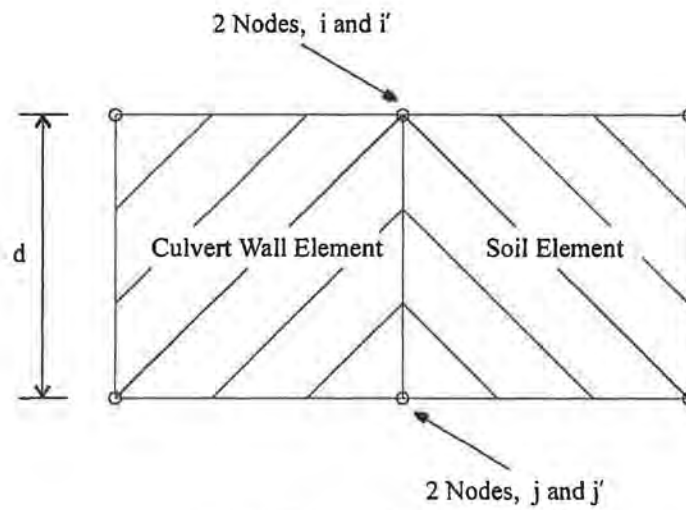
where d = depth of finite element, and rewriting this equation by substituting Equation 3.31 gives:

$$k = \frac{d}{2} K_I \gamma_w \left(\frac{\sigma_n}{p_a} \right)^n \left(1 - \frac{R_f \tau}{\sigma_n \tan \delta} \right)^2 \quad (4.2)$$

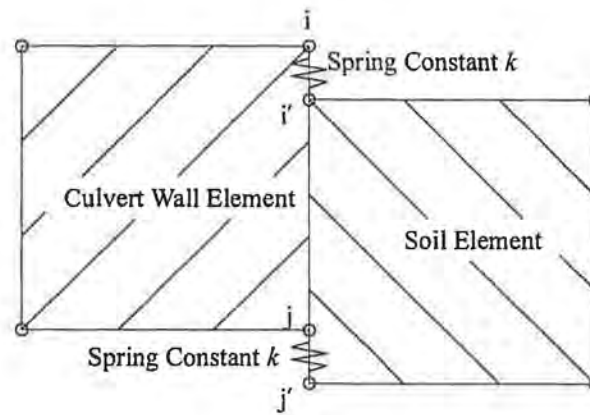
Note that K_{st} is a function of normal stress, σ_n and shear stress, τ , and these values should be replaced by normal and shear stresses of the soil element.^[34] Each model was run twice using ABAQUS. The normal and shear stresses of soil elements immediately adjacent to culvert elements were obtained from the first run, and the spring constants were calculated for the spring elements that were used. The second analysis obtained actual pressure values on the culvert. The values of other parameters of K_{st} are shown in Table 4.2. The result from this spring interface modeling is also shown in Table 4.1. As can be seen from Table 4.1, the effect of a

Table 4.2 Values of Parameters of K_{st}

Parameters	Values
δ	24°
K_I	40,000
N	1.0
R_f	0.9



(A) Constructing Elements and Nodes



(B) After Interface Shear Displacement

Figure 4.4: Spring Interface Modeling

potential slip of the backfill soil along the exterior culvert wall is insignificant. Therefore, this possibility was not considered in the subsequent analyses.

4.3 Model Verification

In order to assess the validity of the soil modeling technique adopted in this study, an example considered by Katona and Vittes^[26] for their verification study to validate the CANDE prediction versus the test data for soil pressure round a box culvert was reanalyzed using ABAQUS and ISBILD. As Katona and Vittes presented their comparison in a graphical form, the same graphical comparisons are given in Figure 4.5. As can be seen from Figure 4.5, the analytically predicted values from ABAQUS are fairly close to those from CANDE. Therefore, it was concluded that the finite element analysis results obtained using ABAQUS and the soil

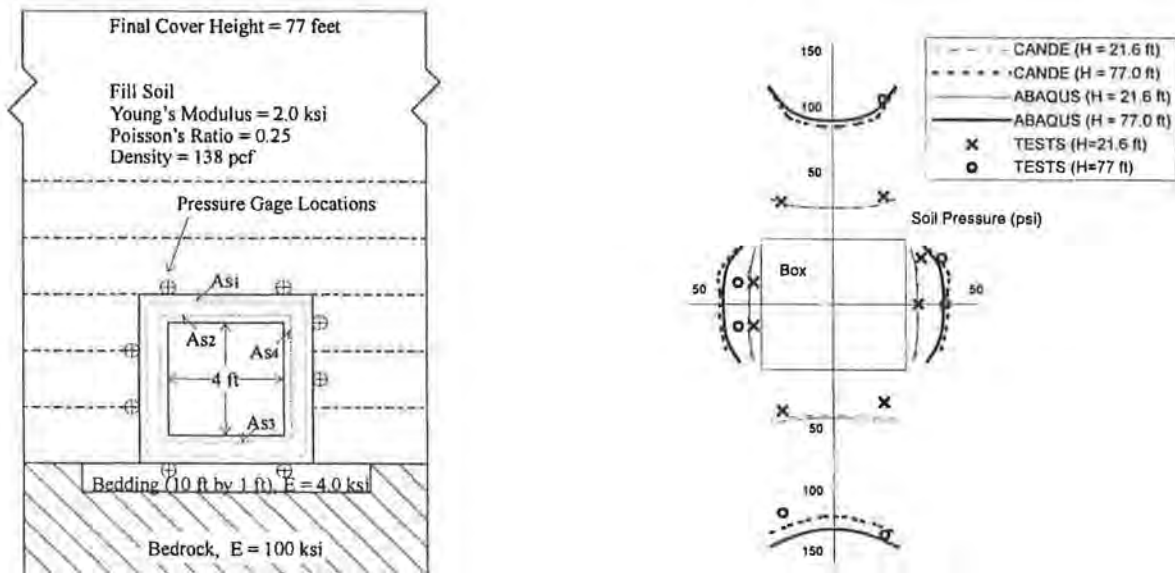


Figure 4.5: Schematics of Test Setup and Data Comparison

modeling technique adopted are valid. The soil pressure directly above the sidewall is substantially higher than the soil pressure at the center of the top slab, where the largest relative vertical deflection is expected to occur. It appears that the inclusion of reinforcing steel in the calculation of the slab stiffness has a negligible effect on the soil pressure distribution, as expected from ACI^[35] Article 8.6, "Stiffness."

CHAPTER 5

NUMERICAL ANALYSIS

5.1 General

The soil pressure or the effective density distribution at the top slab (flange) of deeply buried concrete box culverts is affected very sensitively by the installation method used, either embankment installation or trench installation. Therefore, the numerical analysis of a series of hypothetical concrete box culvert models and subsequent data reduction were carried out separately for the cases of trench installation and embankment installation. This independent treatment permits the development of an optimum finite element model for each group. A typical soil-structure model and incremental sequence of embankment and trench installation for finite element analysis are shown in Figure 4.1. The properties of soil layers were computed following the Duncan model, as discussed in Chapter 3. The following apply to the analyses of both the trench and embankment installations:

- 1) Two-dimensional plane strain elements,
- 2) Retain symmetry of the model,
- 3) Plain concrete properties,
- 4) Neglect any slip between the soil and the culvert and between the placed backfill and pre-existing ground,
- 5) Box culverts on both yielding and unyielding foundations,
- 6) Hyperbolic soil model and its parameters,
- 7) Incremental placement of the soil in layers to simulate construction sequence, and
- 8) Analysis by both ABAQUS and ISBILD (Computer Program for Analysis of

5.2 Embankment Installations

A number of hypothetical deeply buried concrete box culverts were analyzed to investigate the soil pressure intensity. The dimensions of the model box culvert were fixed at eight feet high and eight feet wide (external dimensions), with a thickness for the walls and slabs of 12 inches. The backfill height varied from 50 ft to 200 ft. Each model had a total of 20 construction-layers, regardless of backfill height.

The pressure values on the top flange of the box culverts were conservatively averaged and converted to effective densities, which are numerical values representing an equivalent hydro-static pressure. The value of the effective density is identical in nature to F_{el} given by Equation 3.5. As shown in Figure 5.1, the value of the effective density increased as backfill

Table 5.1 Hyperbolic Parameters Used in the Analysis^[25]

Parameters	Filling Soil	Original Ground
K, K_{ur}	160.00	78.00
N	0.81	1.24
c (pcf)	800.00	0.00
ϕ (degree)	34.00	26.00
R_f	0.63	0.92
G	0.26	0.5
F	0.04	0.0014
d	6.20	0.0077
w (pcf)	120.00	126.00

heights increased. The values determined from the ABAQUS analysis were higher than those obtained from ISBILD. ASSHTO^[6] provides soil-structure interaction factors of uncompacted and compacted side-fill, as given in Equation 3.5. The predicted values on yielding foundations from both ABAQUS and ISBILD analyses were between the values given for the compacted and the uncompacted side-fill by AASHTO. It appears from the analysis that the value of the effective density is most sensitive to the foundation characteristics. In the case of an unyielding foundation, the value of the effective density from ABAQUS analysis showed somewhat higher value than that obtained from ISBILD. Analysis results also indicate that the value of the effective density for an unyielding foundation is much higher than that for a yielding foundation.

Two proposed predictor equations, Equations 5.1 and 5.2, are derived based on the data from ABAQUS and ISBILD analyses by means of a linear regression method.

$$D_E = 1.009H^{0.048} \text{ on Yielding Foundation} \quad (5.1)$$

$$D_E = 1.016H^{0.068} \text{ on Unyielding Foundation} \quad (5.2)$$

where D_E and H stand for effective density value and filling height, respectively.

Figure 5.2 shows the curves of predicted values of the effective density computed from a linear regression method, along with the raw values of the predicted effective densities from ABAQUS and ISBILD. Figure 5.3 shows deformed shapes of soil-culvert systems on yielding and unyielding foundations. It can be seen from Figure 5.3 that the height of equal settlement on a yielding foundation is less than that on an unyielding foundation. This is an important characteristic that can explain why the effective density or the soil-structure interaction factor is higher for an unyielding foundation than that for a yielding foundation. There are no relative movements between the soil column on the top of culvert and adjacent soil columns above the horizontal line of equal settlement. It is evident that the foundation condition of the construction

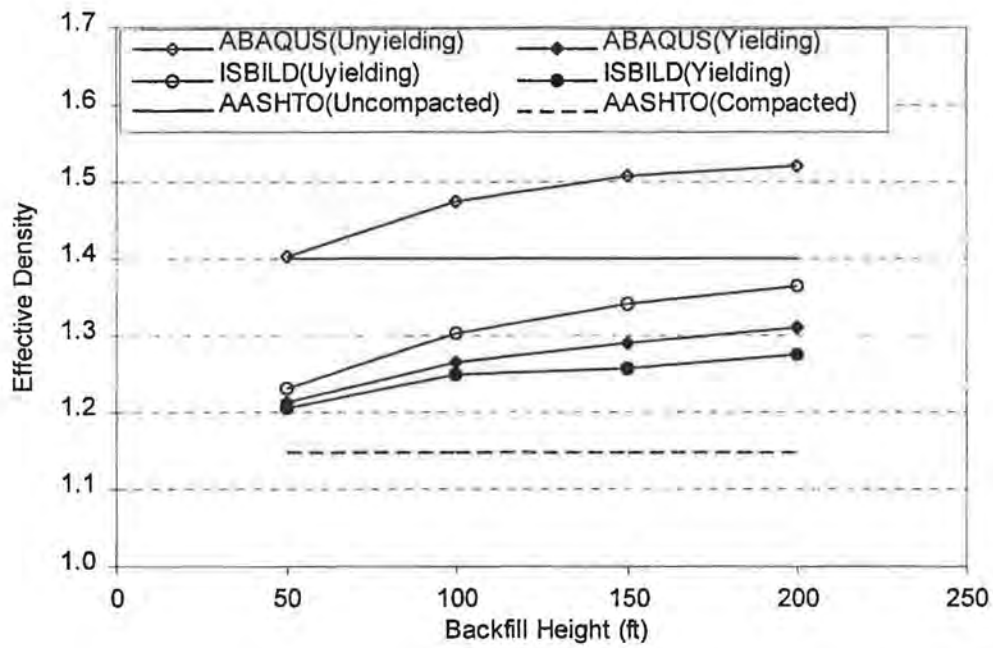


Figure 5.1: Effective Density for Embankment Installations

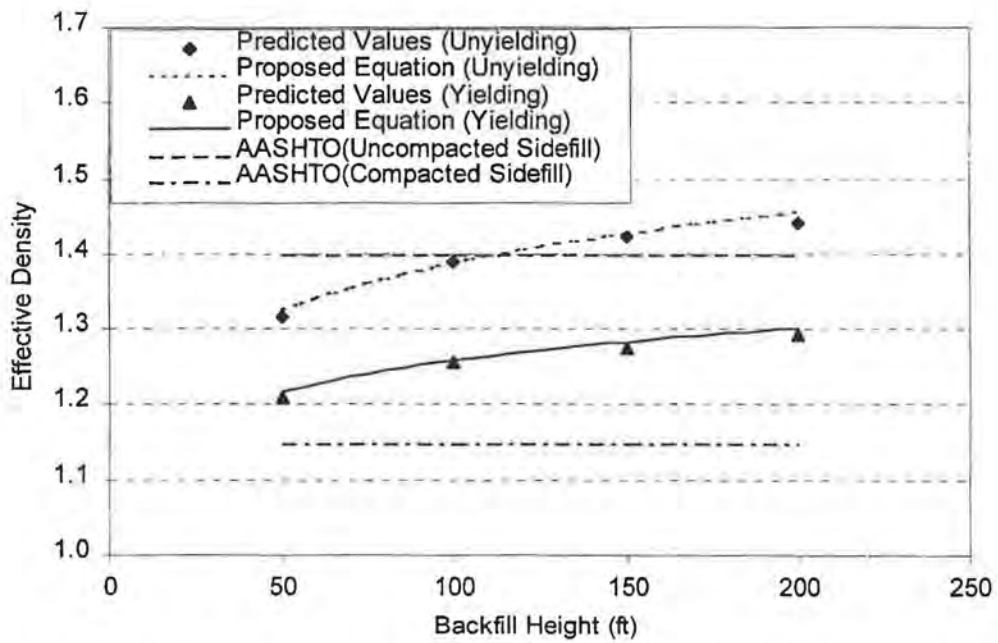
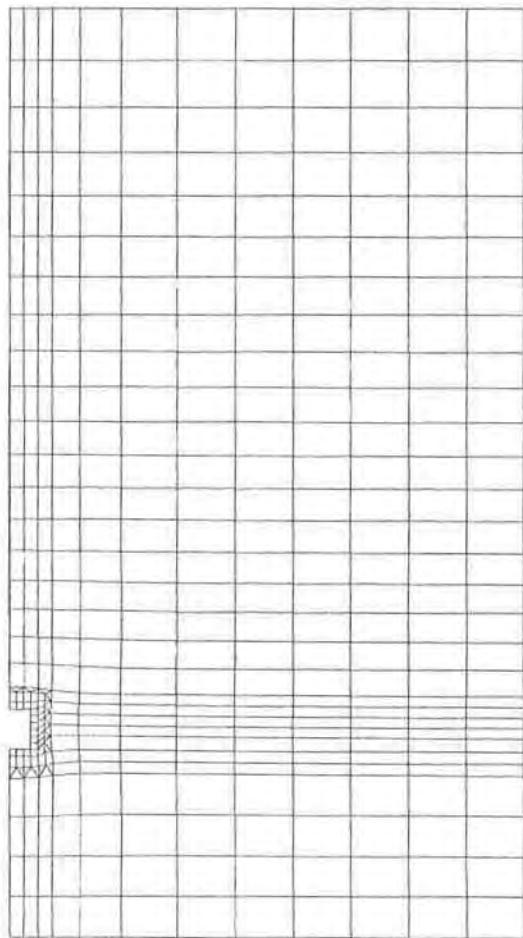
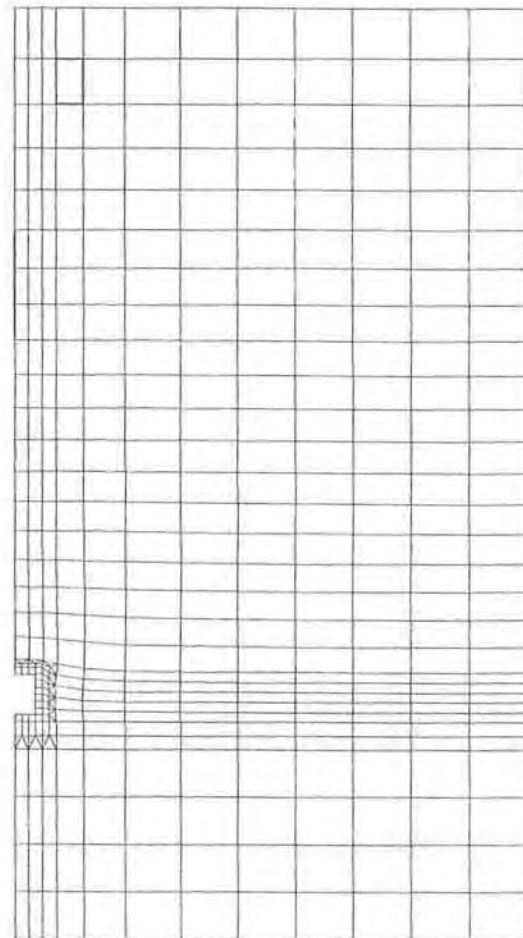


Figure 5.2: Effective Density Curves from Proposed Equations for Embankment Installations



(a) Yielding Foundation



(b) Unyielding Foundation

Figure 5.3: Deformed shape of Soil-Structure System

site is a prominent factor affecting the soil pressure on the structure, especially under high backfill conditions. However, concrete box culverts are most likely to be installed on yielding foundations unless a solid rock layer is encountered immediately under the concrete box culverts. Unlike the current AASHTO^[6] provisions, the compactness of fill along the sides of the box section does not affect the effective density significantly. It should be noted that AASHTO Standard Specifications for Highway Bridges^[6] stipulates that a special analysis is required for culverts untrenched on unyielding foundation.

5.3 Imperfect Trench Installations

The loads on concrete box culverts in embankment installations are greater than the weight of the soil directly above the structure, as stated previously. The pressure intensity on buried concrete culverts is very sensitive to the relative vertical displacement of the soil prism above the culvert. An attempt to avoid increasing the load on the structure led to the development of the imperfect trench method (sometimes called the induced trench method) of construction, where larger relative vertical displacements of the soil prism above the structure are induced by replacing some part of the fill with lightweight material. The most common type of light or compression material used in the imperfect trench method is organic, such as baled straw, leaves, hay, or compressive soil. The traditional imperfect trench method involves installing a compression layer above the culvert within the backfill, as shown in Figure 5.3. As the embankment is constructed, the soft zone compresses more than the surrounding fill and thus induces an arching action above the culvert.

In order to demonstrate the versatility of the finite element modeling, concrete box culverts with light back fill materials were analyzed for several different geometric configurations and back fill material properties. As exact values of the modulus of elasticity and

the specific weight of the baled straw were not known, the value of the modulus of elasticity were varied from 100 ksf to 10 ksf and the weight of the baled straw taken to be 10 pounds per cubic foot. A typical average value of the modulus of elasticity of soil, E , is about 30 to 40 ksf at ground level and 500 ksf at a depth of 100 feet. The width of the light back fill or loose material zone, w , was varied from 1 to 2.5 times the width of the culvert, B_c , and the height of the lightweight back fill was set at 10 feet. Examination of the literature reveals that the loose material zone is traditionally placed above a 2 feet-deep filling cover above the top flange of the box culvert. The depth of the filling cover, h' , in Figure 5.3 was thus 2 feet. The exterior dimensions of box culvert were 8 feet high and 8 feet wide, and the thickness of each wall was 12 inches. Table 5.2 shows the load reduction rate of the imperfect trench method in terms of percentage. It can be seen from Table 5.2 that the width of the lightweight backfill need not be greater than 1.5 times the width of the culvert.

The effect of the height of the soft zone and the depth of filling cover was examined. h and h' were varied from 2.5 feet to 15 feet, and from 0 to 8 feet, respectively. w/B_c was varied from 1.0 to 2.0. h was set at 10 feet when h' was varied, and h' was set at two feet when h was varied. The load reduction rates as functions of the ratio of the height of the soft zone to that of the culvert, h/H_c , are plotted with different values of w/B_c in Figure 5.4. The effect of Young's modulus is given in Figure 5.5. As Figures 5.4 and 5.5 shows that variations of load reduction have a diminishing return characteristic with regard to increasing h/H_c . Typical effective density distributions on the box culverts with $h/H_c = 1.25$ are shown in Figures 5.6 and 5.7, with different values of w/B_c and different values of Young's modulus, respectively. The variations of load reduction rates are shown in Figure 5.8 as a function of the ratio of filling cover depth to the height of the culvert, h'/H_c . Typical effective density distributions with $w/B_c=1$ and h/H_c

$=1.25$ are plotted in Figure 5.9. Figures 5.8 and 5.9 show that the highest load reduction occurs when the lightweight backfill or the soft zone is placed immediately on the box culvert, i.e., there is no intermediate filling between the culvert and the soft zone. It is also clear from Figures 5.4 and 5.5 that the height of the soft zone does not need to be greater than 1.5 times that of the culverts after that point, no significant load reduction is realized.

Although the artificially induced reverse arching action is desirable, Vaslestad and his coworkers^[22] recommended a judicious selection of the lightweight back fill material and its careful installations. A number of serious stress concentration problems have been reported due to ill-advised arrangements of compression layers and improper material selection.

It may require a considerable effort to develop a reasonable force-displacement relationship and a volume change rate for the material used in the soft zone, as there are practically no experimental data available that can be used to establish a constitutive relationship for the material in the soft zone. It is highly likely that the constitutive relationship of the soft zone material would be nonlinear, as for relationship describing backfill soil. A close up view of the deformed shape of a typical model around the soft zone is shown in Figure 5.10. The displacement values are magnified with an amplification factor of 4. Young's modulus and Poisson's ratio of the soft zone were taken to be 10 ksi and 0.1, respectively, for this case. Table 5.3 presents the vertical displacement at the top, mid-height, and bottom of the soft zone along the vertical centerline of symmetry of a soil-structure model. Three different Young's moduli and two different Poisson's ratios were used to generate Table 5.3. The vertical displacements at the top and mid-height increase for the softer material, as expected. However, it is of interest to note that the vertical displacement at the bottom of the soft material zone decreases, resulting in a volume squeeze. As presented in Table 5.3, the value of Poisson's ratio does not appear to

Table 5.2: Pressure Reduction Rates (%) Due to Imperfect Trench Installation

Modulus E (ksf)	Width of Back Fill Layer, w/B_c			
	1	1.5	2	2.5
100	32.8	41.1	39.1	36.7
50	43.6	58.6	57.0	54.3
10	56.9	83.7	85.3	84.8

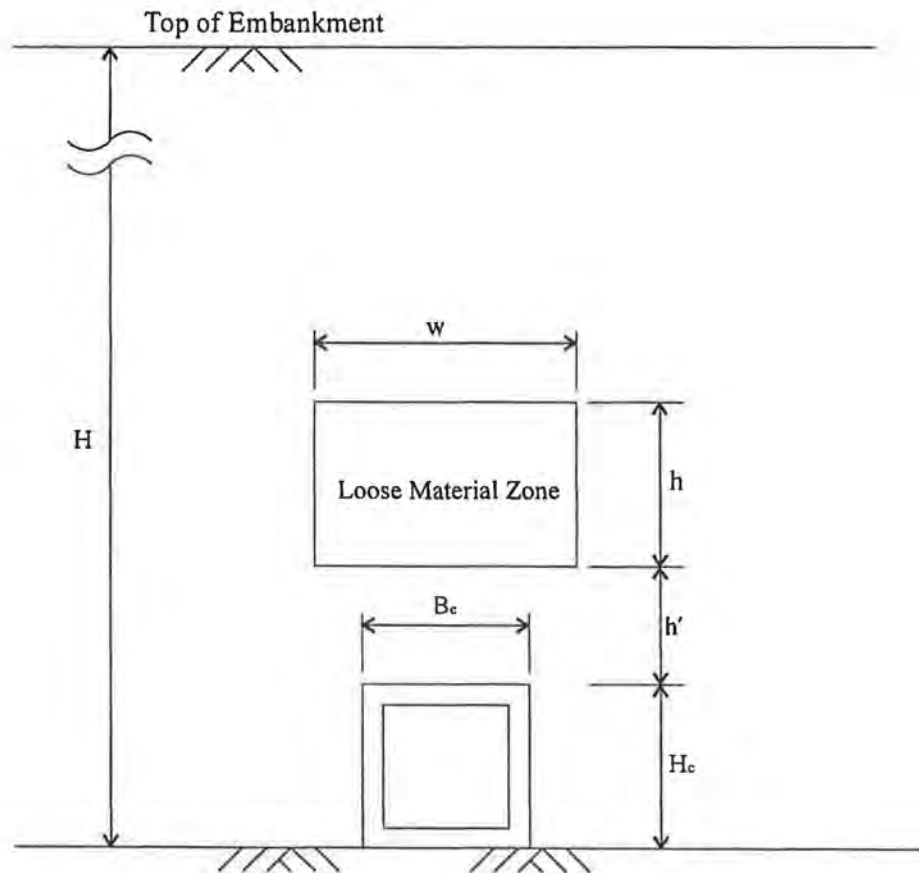


Figure 5.3: Imperfect Trench Installation

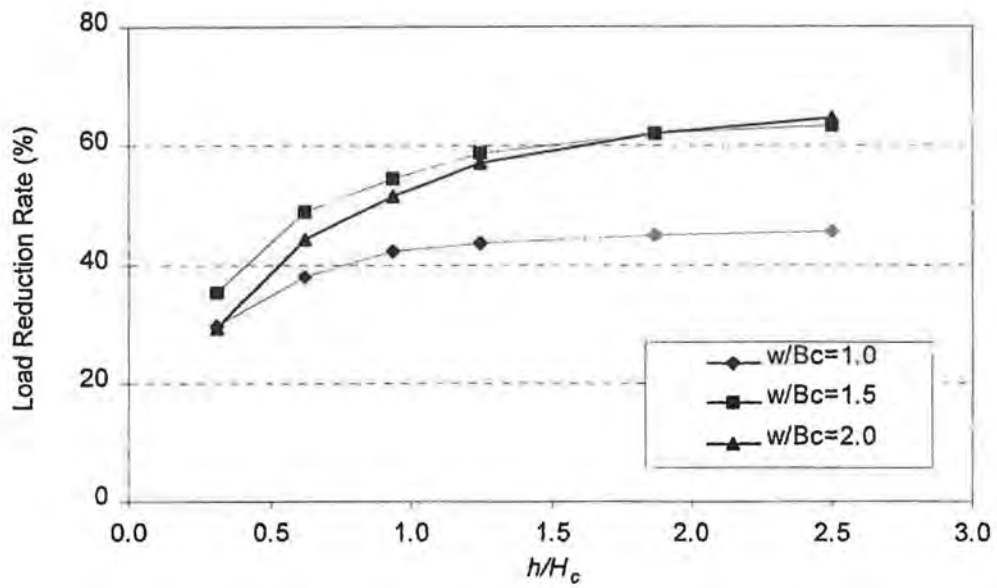


Figure 5.4: Load Reduction Rate vs. h/H_c (Different w/B_c)

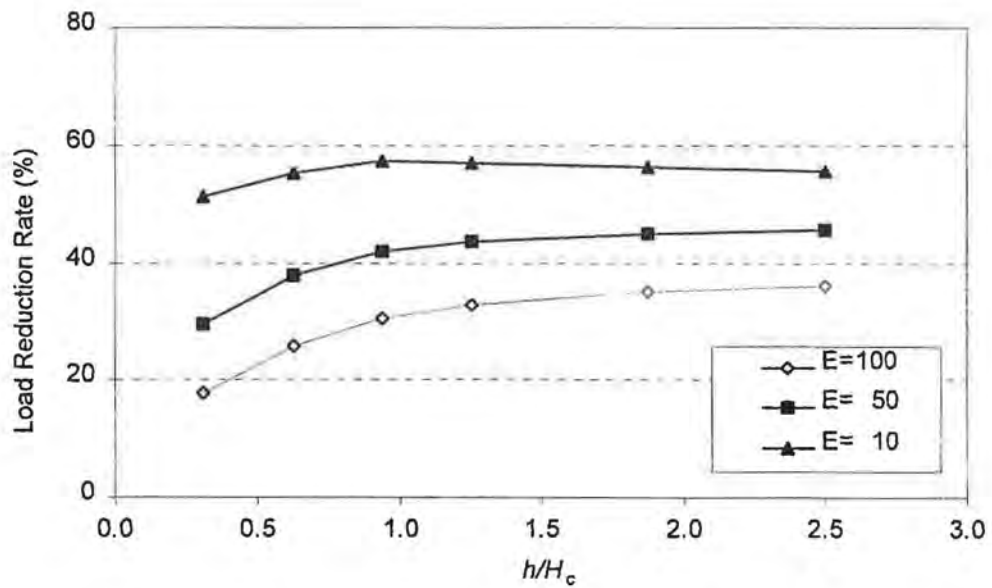


Figure 5.5: Load Reduction Rate vs. h/H_c (Different Young's Modulus)

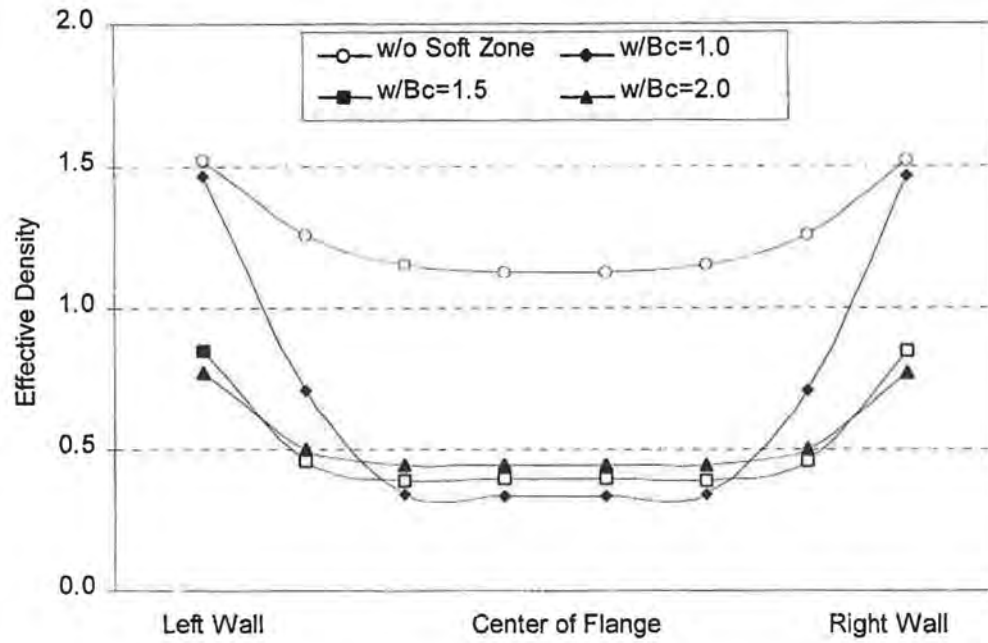


Figure 5.6: Effective Density Distribution on the Box (Different w/B_c)

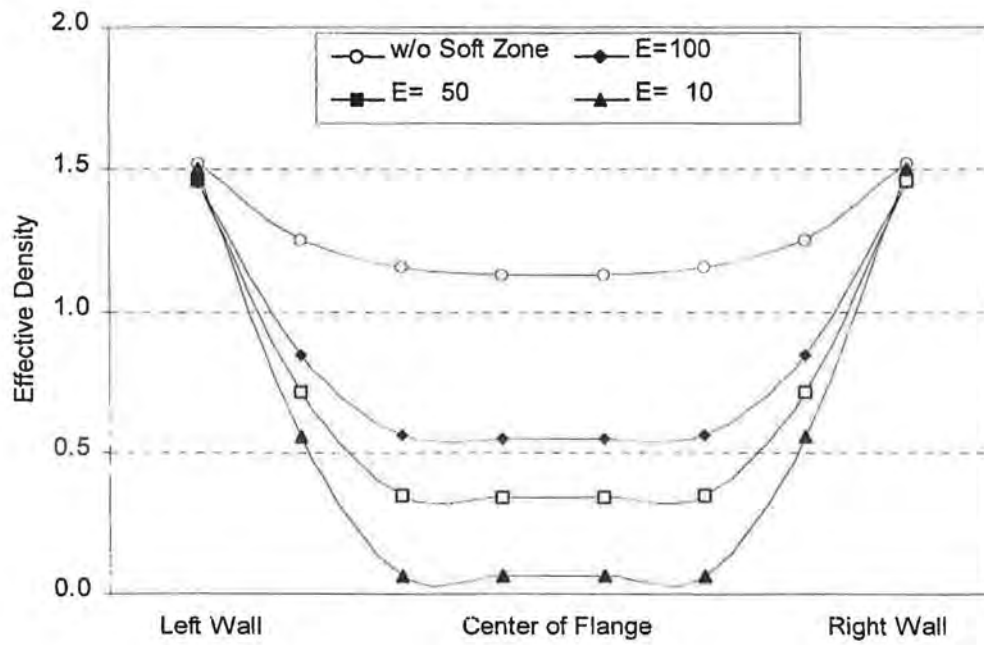


Figure 5.7: Effective Density Distribution on the Box (Different Young's Modulus)

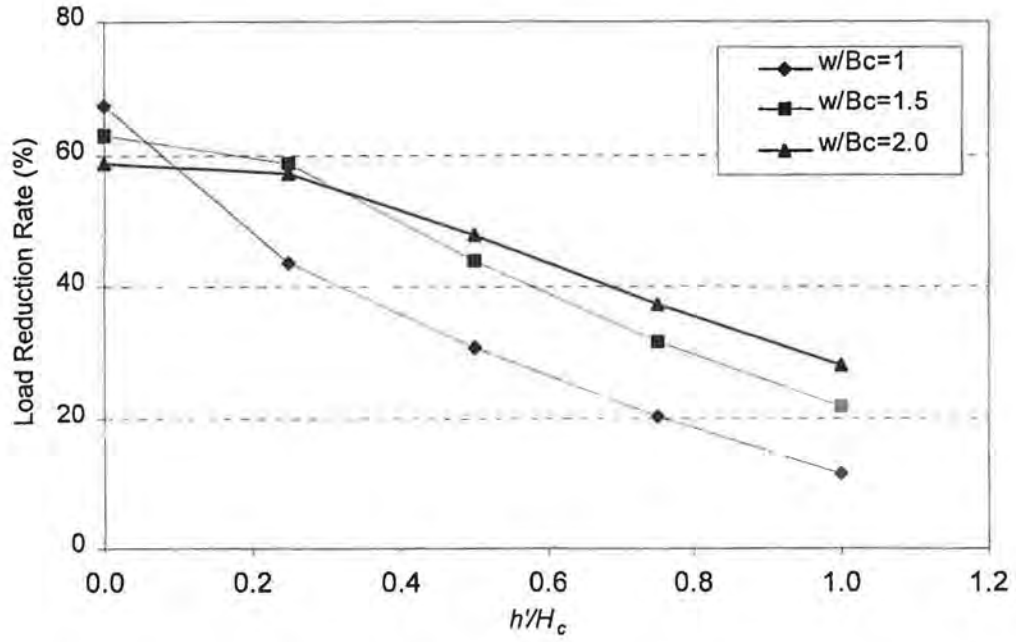


Figure 5.8: Load Reduction Rate vs. h'/H_c

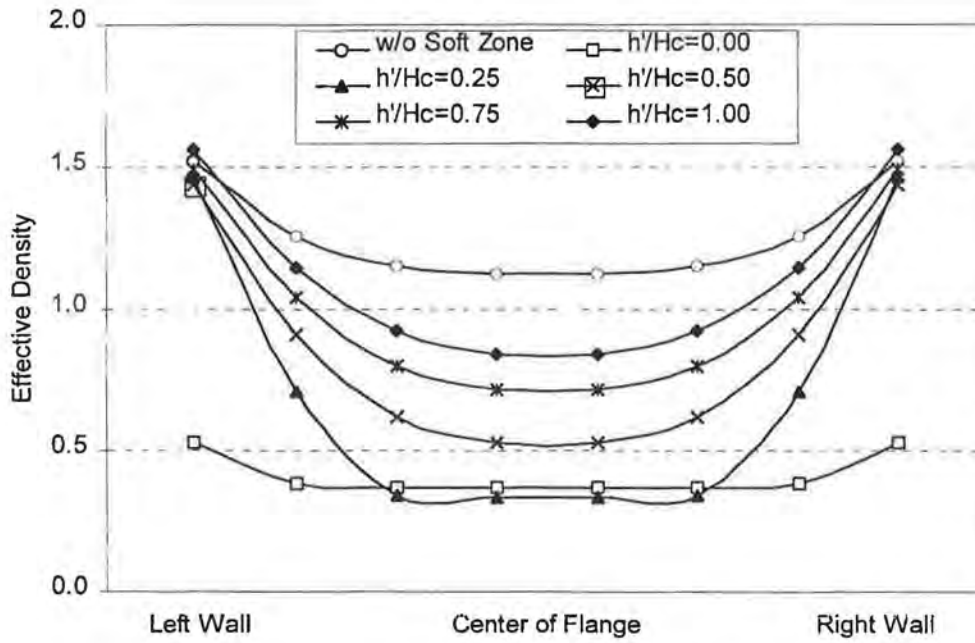


Figure 5.9: Effective Density Distribution ($w/B_c=1, h'/H_c=1.25$)

Table 5.3 Vertical Displacements at Soft Zone (ft)

	E=100 (ksf)			E=50 (ksf)			E=10 (ksf)		
	$\nu=0.4$	$\nu=0.1$	Diff. (%)	$\nu=0.4$	$\nu=0.1$	Diff. (%)	$\nu=0.4$	$\nu=0.1$	Diff. (%)
Top	0.656	0.705	7.4	0.712	0.753	5.7	0.799	0.824	3.1
Middle	0.466	0.485	4.1	0.482	0.498	3.2	0.514	0.526	2.3
Bottom	0.289	0.274	5.3	0.270	0.258	4.3	0.247	0.243	1.5

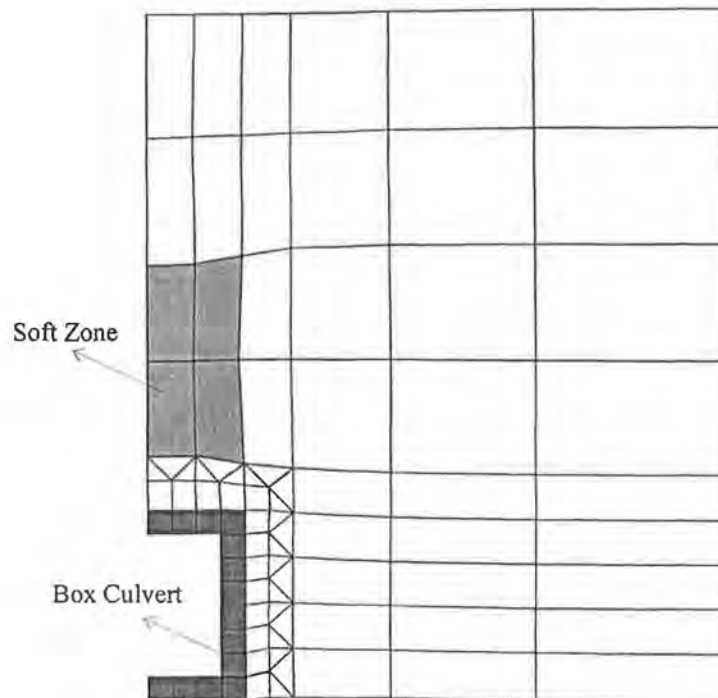


Figure 5.10: Close-up View of Deformed Shape near the Soft Zone

affect the vertical displacements significantly. It should also be noted that the soil layers between the soft zone and the concrete culvert appear to be swollen. This phenomenon is likely to be caused by the effect of Poisson's ratio of these soil layers, as they are subjected to very high lateral pressures exerted by the adjacent backfill elements.

5.4 Trench Installations

The effective density of soil pressure or the value of the soil-structure interaction factor for concrete box culverts in trench installations was also examined. The ratio of the horizontal width of the trench to the box width, B_d/B_c , was varied from two to twelve. The height of the backfill varied from 50 ft to 150 ft. Exactly the same parameters in the Duncan soil model of the original ground and filling soil were used for analyses of both embankment installations and trench installations. In this trench installation model, the sides of the trench were assumed to be vertical, and no slip was permitted between the backfill soil and the sides of the trench. The pressure values on the top flange of the box culverts were again conservatively averaged and converted to effective densities. Figure 5.11 shows the value of the effective density at the top of box culverts installed on a yielding foundation as a function of the ratio of the trench width to the box culvert width, B_d/B_c . Similar information on unyielding foundation models is not shown. The value of effective density approaches asymptotically that for embankment installations as B_d/B_c increases, as expected.

The transition width can be readily determined from Figure 5.11, where the effective density for trench installations becomes close to that for embankment installations. As in the case of embankment installations, the values of effective density determined from ABAQUS (represented by solid lines in Figure 5.11), are higher than those determined from ISBILD (represented by dotted lines in Figure 5.11).

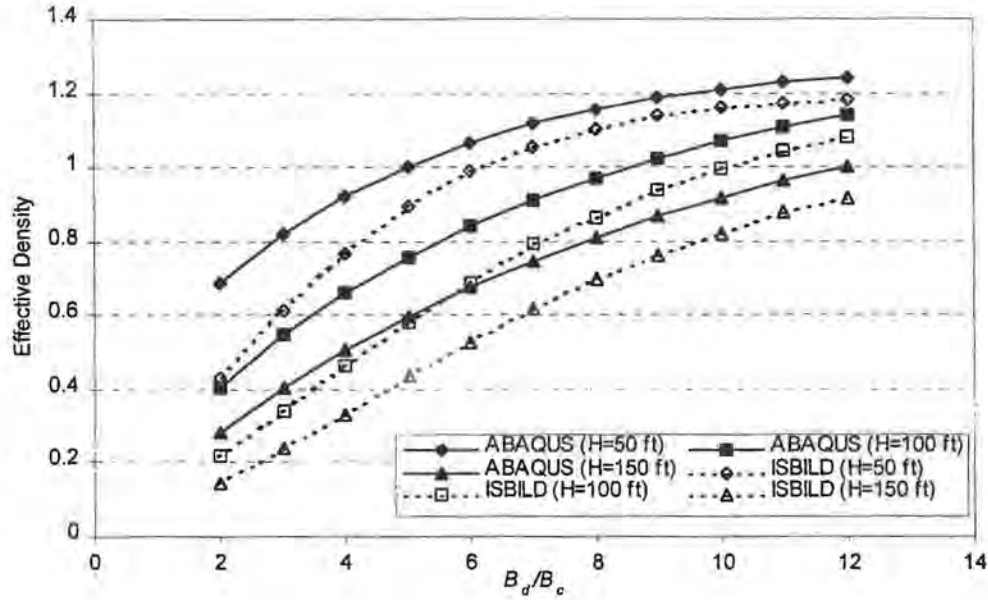


Figure 5.11: Effective Density vs. B_d/B_c .

The value of effective density is plotted as a function of the ratio of backfill height to horizontal width, H/B_d , on yielding and unyielding foundations in Figures 5.12 and 5.13, respectively. It should be noted in Figures 5.12 and 5.13 that the effective density varies only slightly for different values of the backfill height if H/B_d remains the same. The load coefficient for trench installation, C_d , is given in Equation 3.6 as a variable. Although Equation 3.6 (adopted in the current AASHTO Specifications^[5,6]) is based on Marston-Spangler theory, the load coefficient, C_d , based on the Marston theory remains unchanged. Most values of the soil-structure interaction factor from AASHTO Specifications are 1.4, the limit value, except for a few cases. It is clear from Figures 5.12 and 5.13 that the values of effective density (or structure-soil interaction factors) given by the current AASHTO Specifications appear to be conservative compared to the analytically predicted values from this study. In Figures 5.12 and 5.13, values computed from AASHTO provisions are given for the case yielding minimum values, obtained

by selecting granular materials without cohesion. The curves shown are obtained from a regression analysis on the data from ABAQUS and ISBILD as functions of H/B_d . The following regression equations are found to best fit the data relating the effective density and H/B_d .

Yielding Foundation

$$\log(D_E) = 0.012\left(\frac{H}{B_d}\right)^2 - 0.288\left(\frac{H}{B_d}\right) + 0.375; \text{ ABAQUS} \quad (5.1)$$

$$\log(D_E) = 0.017\left(\frac{H}{B_d}\right)^2 - 0.414\left(\frac{H}{B_d}\right) + 0.429; \text{ ISBILD} \quad (5.2)$$

Unyielding Foundation

$$\log(D_E) = 0.011\left(\frac{H}{B_d}\right)^2 - 0.273\left(\frac{H}{B_d}\right) + 0.465; \text{ ABAQUS} \quad (5.3)$$

$$\log(D_E) = 0.018\left(\frac{H}{B_d}\right)^2 - 0.412\left(\frac{H}{B_d}\right) + 0.476; \text{ ISBILD} \quad (5.4)$$

It appears possible that representative effective density values for yielding and unyielding foundations may be expressed by taking mean values of those two predicted values from ABAQUS and ISBILD. Equations 5.1 and 5.2 can be combined into Equation 5.5. Likewise, Equations 5.3 and 5.4 can be combined into Equation 5.6.

$$D_E = \exp\left[0.015\left(\frac{H}{B_d}\right)^2 - 0.352\left(\frac{H}{B_d}\right) + 0.402\right]; \text{ Yielding Foundation} \quad (5.5)$$

$$D_E = \exp\left[0.015\left(\frac{H}{B_d}\right)^2 - 0.342\left(\frac{H}{B_d}\right) + 0.471\right]; \text{ Unyielding Foundation} \quad (5.6)$$

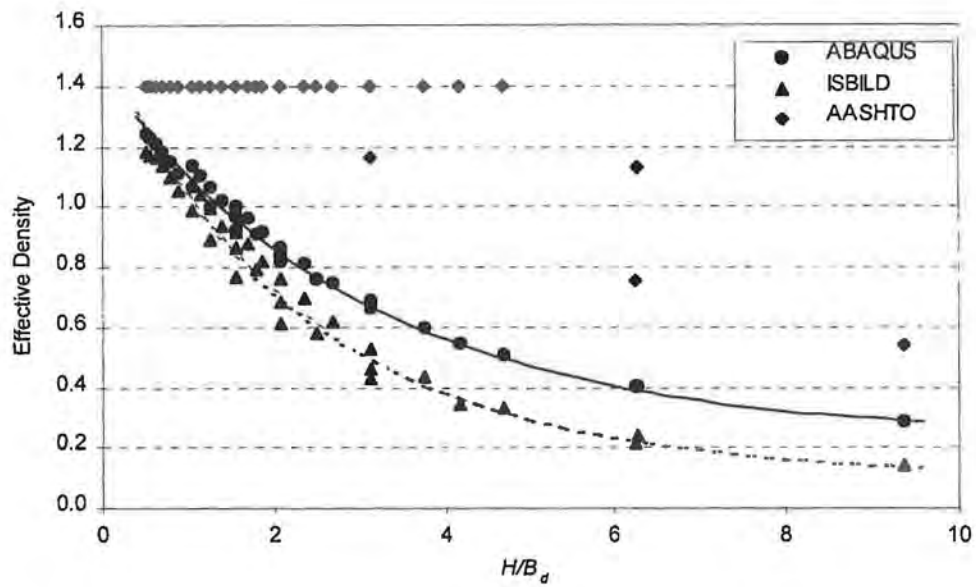


Figure 5.12: Effective Density vs. H/B_d on Yielding Foundation.

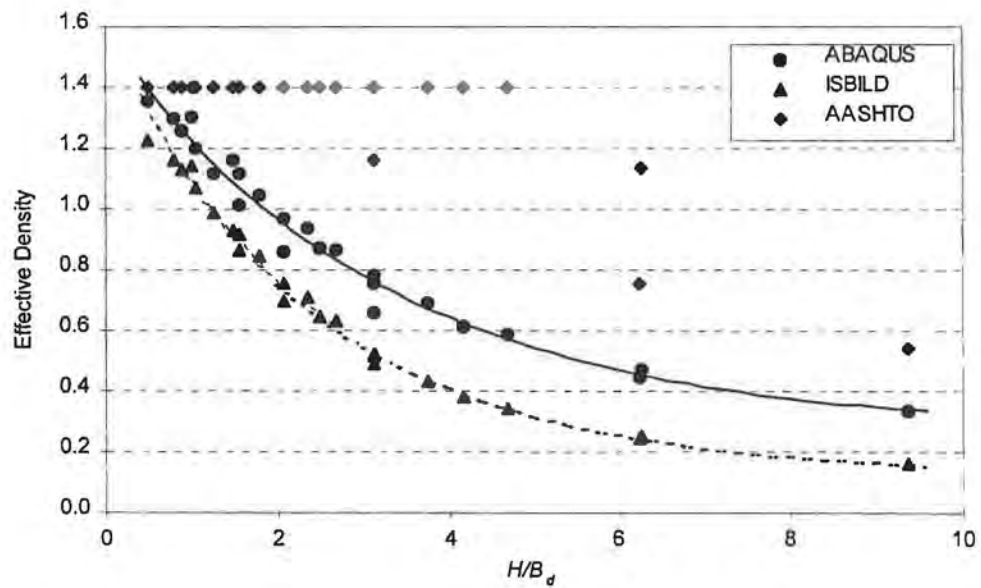


Figure 5.13: Effective Density vs. H/B_d on Unyielding Foundation

5.5 Effect of Sloping Trench Walls

In theoretical treatments, the trench walls are generally taken to be vertical. However, for deep trenches, it is almost impossible to maintain the trench walls to be vertical in practice.

Sloping trench sidewalls are expected to have a significant impact on the soil pressure on culverts. In the case where the trench is constructed with sloping sides, as shown in Figure 5.14, experiments have shown that the width of the trench at or slightly below the top of the conduit is the proper width to use for B_d in calculating the load.

To investigate how sensitive those effects are to the load intensity, boxes installed in a trench with sloping sidewalls were modeled. The resulting effective densities for $B_d/B_c = 2, 3$, and 4 with fill heights of 50, 100, and 150 ft are shown in Figures 5.15, 5.16, and 5.17, respectively. θ stands for the angle between the vertical line and the sloping sidewall of the trench as shown in Figure 5.14. Values of the effective density increase to approximately 90% of that for embankment installation when θ approaches 45 degrees. The current AASHTO Equation 3.6, along with most other references, stipulates the effective densities for the trench installations under the assumption that the trench walls are excavated vertically but the effect of the sloping trench walls on the soil-structure interface factors is quite high. Although Figures 5.12 and 5.13 show large discrepancies between the value of the effective density computed from AASHTO provisions and finite element analyses where the trench walls are assumed to be vertical, the actual differences are likely to be much smaller as the sloping trench walls are taken into account.

5.6 Examples

Examples for predicting approximate values for design loads on the box culvert are given in this section. Predicted values are compared to those obtained using the current AASHTO and

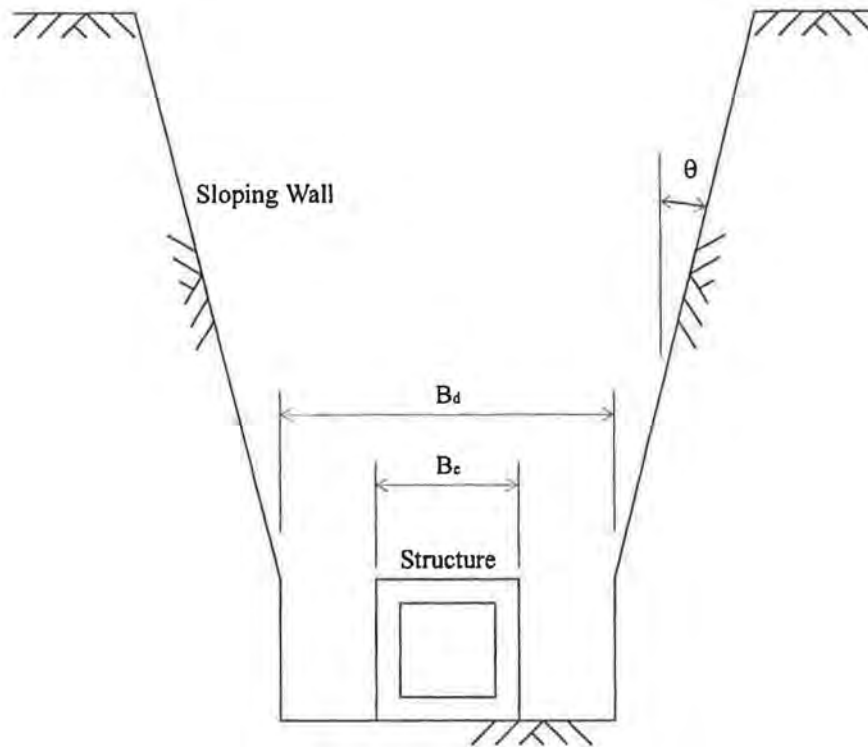


Figure 5.14: Trench Installation with Sloping Sidewalls

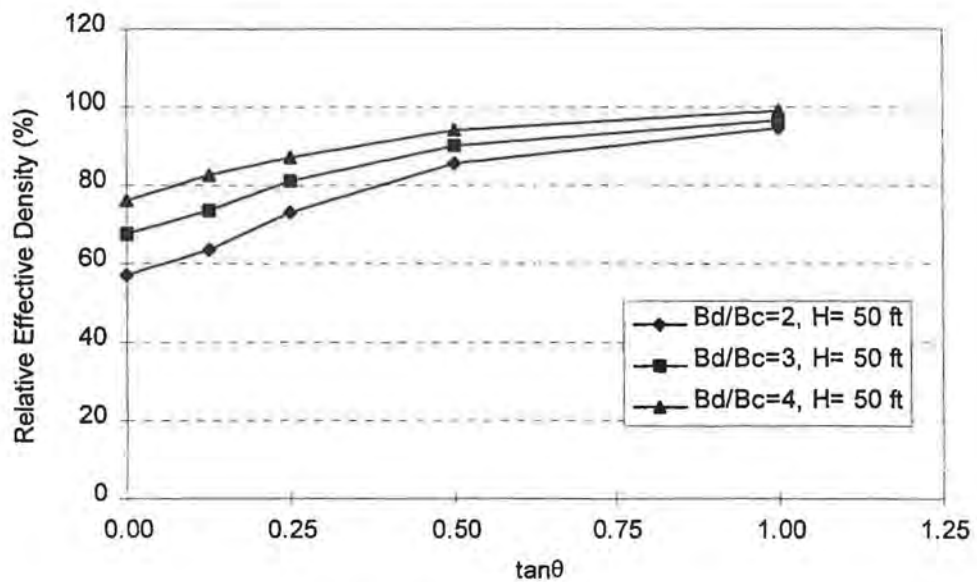


Figure 5.15: Effect of Sloping Sidewall in Trench Installations ($H=50$ feet)

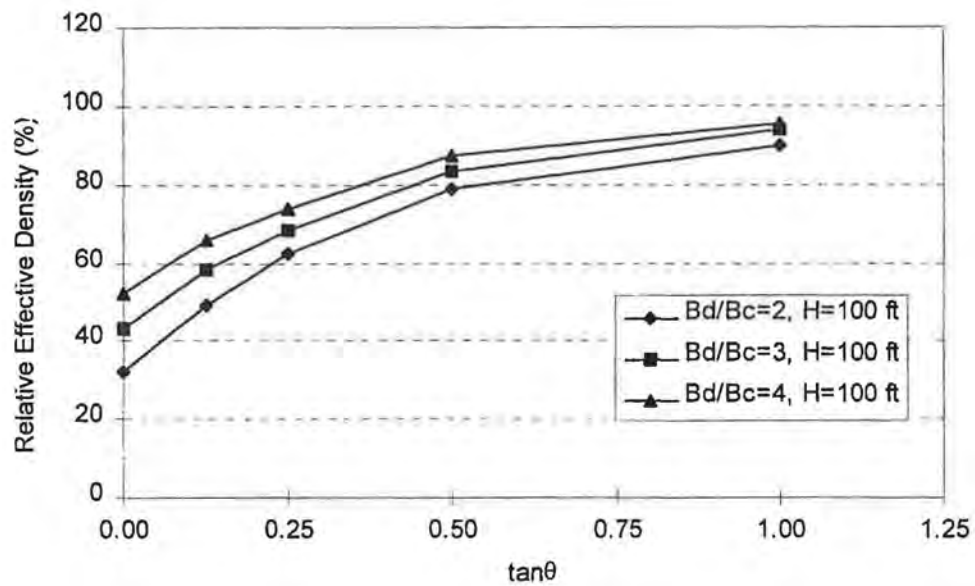


Figure 5.16: Effect of Sloping Sidewall in Trench Installations ($H=100$ feet)

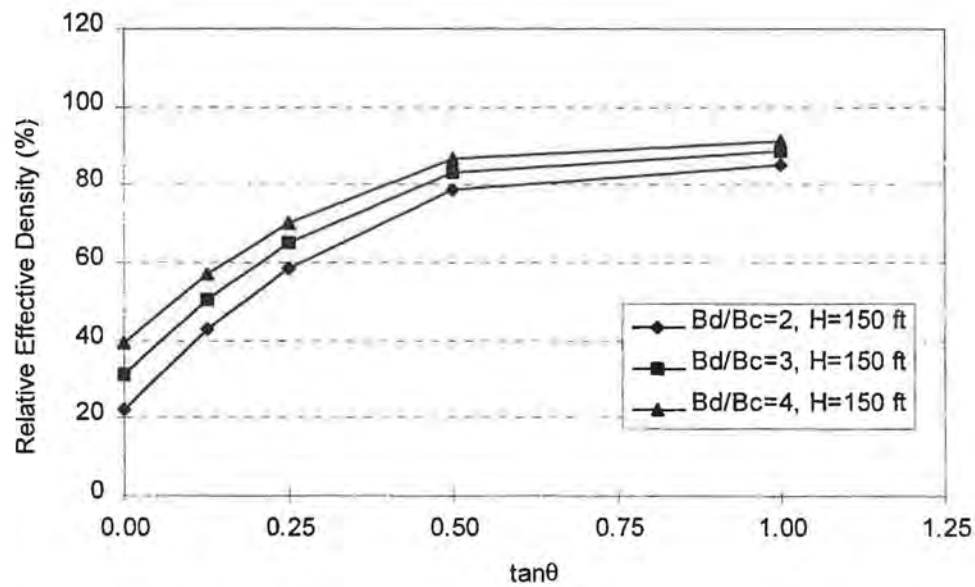


Figure 5.17: Effect of Sloping Sidewall in Trench Installations ($H=150$ feet)

Marston-Spangler recommendations. The hypothetical construction conditions for embankment and trench installations were taken to be as follows:

Example 1. (Embankment Installation)

Given:

Concrete box exterior dimension, $B_c \times H_c$:	8 ft $\times$ 8 ft
Fill height, H :	160 ft
Weight of filling soil, γ :	120 pcf
Foundation condition:	Yielding foundation

Solution:

From Equation 5.1, the effective density value, D_E can be determined as follows:

$$D_E = (1.009)(160^{0.048}) = 1.287$$

Total earth load on the culvert, W_e is calculated as

$$W_e = D_E \gamma B_c H = (1.287)(0.12)(8)(160) = 197.68 \text{ (k / ft)}$$

From AASHTO provisions, Equations 3.4 and 3.5,

$$F_{el} = 1 + 0.20 \frac{H}{B_c} = 1 + 0.20 \frac{160}{8} = 5.00 > 1.40$$

$$W_{AASHTO} = F_{el} \gamma B_c H = (1.40)(0.12)(8)(160) = 215.04 \text{ (k / ft)}$$

Equation 3.2 from the Marston-Spangler theory can be applied if the settlement ratio, r_{sd} is known. Based on Spangler's recommendation, Table 3.1 gives a settlement ratio of between +0.5 and +0.8. If the settlement ratio, r_{sd} , is assumed to be +0.7, then the load coefficient for

positive projection, C_c , can be determined from Figure 3.5, to be in the region of $C_c = 31.7$. From Equation 3.2,

$$W_{Spangler} = C_c \gamma B_c^2 = (31.7)(0.12)(8^2) = 243.45 \text{ (k / ft)}$$

The results from this study, AASHTO, and Marston-Spangler theory are compared in Table 5.4.

Table 5.4: Summary of Results for Example 1

Total Load on the Culvert (Difference)	
This study	197.68 k / ft
AASHTO	215.04 k / ft (8%)
Marston-Spangler Theory	243.45 k / ft (19%)

Example 2. (Embankment Installation)

Given:

Concrete box exterior dimension, $B_c \times H_c$:	8 ft $\times$ 8 ft
Fill height, H :	175 ft
Weight of filling soil, γ :	120 pcf
Foundation condition:	Unyielding foundation

Solution:

From Equation 5.2, the effective density D_E can be determined as follows:

$$D_E = (1.016)(175^{0.068}) = 1.444$$

The total earth load on the culvert, W_e is calculated as:

$$W_e = D_E \gamma B_c H = (1.444)(0.12)(8)(175) = 242.59 \text{ (k / ft)}$$

There are no specific equations for unyielding foundations in the current AASHTO specifications, so it is suggested that a rational method is to be used. Spangler's recommendations for settlement ratio can be applied from Table 3.1. The settlement ratio, r_{sd} can be taken to be +1.0, and the load coefficient, C_c , can be determined from Figure 3.5 to be $C_c=36.8$. Using Equation 3.2, the total load is computed according to Spangler's recommendation as:

$$W_{Spangler} = C_c \gamma B_c^2 = (36.8)(0.12)(8^2) = 282.62 \text{ (k / ft)}$$

This value is greater than the predicted value from this study by 16%. However, it should be noted that concrete box culverts are most likely to be installed on yielding foundation unless an obvious solid rock layer is encountered.

Example 3. (Trench Installation)

Given:

Concrete box exterior dimension, $B_c \times H_c$:	8 ft $\times$ 8 ft
Fill height, H :	120 ft
Weight of filling soil, γ :	120 pcf
Width of trench, B_d :	24 ft
Slope of trench wall, $\tan \theta$:	1/2
Foundation condition:	Yielding foundation

Solution:

From the given data,

$$B_d / B_c = 16 / 8 = 2.0$$

$$H / B_d = 120 / 16 = 7.5$$

From Equation 5.1, the effective density, D_E , of embankment installation can be determined as

$$D_E = (1.009)(120^{0.048}) = 1.270$$

From Figures 5.16 and 5.17, one can determine the relative effective density, say α ,

corresponding to $H = 120$ ft and $B_d/B_c = 2$. Using a linear interpolation, the relative effective density, α , is computed to be 78%. Therefore, the total earth load is calculated as follows:

$$W_e = \alpha D_E \gamma B_c H = (0.78)(1.270)(0.12)(8)(120) = 114.12 \text{ (k / ft)}$$

The total earth load can also be determined from current AASHTO specification using Equations 3.4, 3.6 and Figure 3.4. From Figure 3.4, $C_d = 2.75$ is taken for $K\mu' = 0.165$ (maximum for sand and gravel), and from Equations 3.4 and 3.6, the soil-structure interaction factor and total earth load for trench installation are computed as follows:

$$F_{e2} = \frac{C_d B_d^2}{H B_c} = \frac{(2.75)(16^2)}{(120)(8)} = 0.733$$

$$W_{AASHTO} = F_{e2} \gamma B_c H = (0.733)(0.12)(8)(120) = 84.44 \text{ (k / ft)}$$

It appears that there is some difference between the values of total load from this study and AASHTO. However, it should be also noted that the load values obtained by this study include the effect of sloping walls by incorporating $\tan\theta = 1/2$, while the value from AASHTO is determined under the assumption that the trench walls are excavated to be completely vertical. In the case where the trench wall is assumed to be vertical, Equations 5.5 and 5.6 can be used to obtain the effective density for yielding foundations and unyielding foundations, respectively.

For a vertical trench wall, i.e., $\tan\theta = 0$, on the yielding foundation, the total earth load is calculated using Equation 5.5 as follows:

$$\begin{aligned} D_E &= \exp \left[0.015 \left(\frac{H}{B_d} \right)^2 - 0.352 \left(\frac{H}{B_d} \right) + 0.402 \right] \\ &= \exp \left[0.015 (7.5)^2 - 0.352 (7.5) + 0.402 \right] \\ &= 0.248 \end{aligned}$$

$$W_e = D_E \gamma B_c H = (0.248)(0.12)(8)(120) = 28.57 \text{ (k / ft)}$$

However, it should be noted that the assumption of a deep vertical trench wall is not very realistic one.

CHAPTER 6

CONCLUSIONS AND RECOMMENDED FUTURE STUDY

6.1 Conclusions

The findings and concluding remarks are summarized below for the finite element numerical investigation on the effective density or soil-structure interaction factor of deeply buried concrete box culverts. All three finite element programs, ABAQUS, CANDE-89, and ISBILD predict effective densities or soil-structure interaction factors that are reasonably close to each other.

- 1) The values of the effective density for embankment installations are more sensitively affected by the foundation characteristics, i.e., whether a yielding or unyielding foundation is used. The values of effective density on unyielding foundations are much higher than those on yielding foundations. AASHTO stipulates that a special analysis is required for embankment installations on unyielding foundations.
- 2) Finite element analysis results show that the greatest effect of load reduction can be obtained in imperfect trench installations when the soft zone is placed immediately above the box culvert, i.e., there is no soil cover between the top of the box culvert and the bottom of the soft zone.
- 3) The variation of the load reduction rate due to imperfect trench installations shows a diminishing return characteristic. The ratio of the height of the soft zone to that of the box culvert, h/H_c , does not need not to be greater than 1.5; after that point no significant load reduction is realized.

- 4) Although imperfect trench installations reduce the effective density significantly, it has been reported that the selection of the lightweight compression materials and field installations demands considerable planning and care.
- 5) To simulate the soil-structure interface between the exterior box culvert wall and the filling soil, two different methods of interface modeling have been applied. The effect of possible slips of the backfill materials along the exterior culvert wall appears to be negligibly small.
- 6) Finite element analysis indicates that the effective density is only marginally affected by the properties of the sidefill materials, i.e., compacted or uncompacted fill at the sides of the box section.
- 7) The current AASHTO provision reproduced in Equation 3.4 and most other references stipulate the effective densities for the trench installations under the assumption that the trench walls are excavated vertically. However, it has been shown in this study that the effect of the sloping trench walls on the soil-structure interface factors is actually quite high.
- 8) Although Figures 5.5 and 5.6 show large discrepancies between the values of the effective density computed from AASHTO provisions and finite element analyses, the actual differences are likely to be much smaller if the effect of the sloping trench wall is reflected in the finite element analyses.

6.2 Recommendations for Future Study

It is hoped that the knowledge gained and the data presented herein will lead to a better understanding of the complex soil-structure interaction behavior of concrete box culverts.

In view of the remarkable decrease in computational expenses in recent years, the finite element

method approach can be applied to everyday design practice in deeply or shallow buried concrete box culvert designs. However, a considerable effort is still required to properly determine the subsoil conditions and assign the appropriate parameters for the constitutive model. The following recommendations are offered for future research.

- 1) Experimental field studies would be desirable for a verification of the present study.
- 2) It is expected that the constitutive relationship, as well as the volume change rate of the compression or soft zone material, is likely to be nonlinear as for those of other soil materials considered in the present study. Material nonlinearity of the soft zone should be examined and considered in future more detailed models.
- 3) Laboratory performed soil property tests for constitutive models may be necessary to augment the results obtained from finite element analyses.

REFERENCES

- [1] Marston, A., and Anderson, A. O., "The Theory of Loads on Pipes in Ditches and Tests of Cement and Clay Drain Tile and Sewer Pipes." Bulletin 31, Iowa Engineering Experiment Station, Ames, Iowa, 1913.
- [2] Marston, A., "The Theory of External Loads on Closed Conduits in the Light of the Latest Experiments," Bulletin 96, Iowa Engineering Experiment Station, February 19, 1930.
- [3] Spangler, M. G., "Field Measurements of the Settlement Ratios of Various Highway Culverts," Bulletin 171, Iowa Engineering Experiment Station, Ames, Iowa, 1950.
- [4] AASHTO Standard Specifications for Highway Bridges, 12th ed., American Association of State Highway and Transportation Officials, Inc., Washington, D.C., 1977.
- [5] AASHTO LRFD Bridge Design Specifications, 2nd Ed., American Association of State Highway and Transportation Officials, Inc., Washington, D.C., 1998.
- [6] AASHTO Standard Specifications for Highway Bridges, 16th ed., American Association of State Highway and Transportation Officials, Inc., Washington, D.C., 1996.
- [7] AASHTO Standard Specifications for Highway Bridges, 13th ed., American Association of State Highway and Transportation Officials, Inc., Washington, D.C., 1983.
- [8] Hardin, B. O., "Characterization and Use of Shear Stress-Strain Relations for Airfield Subgrade and Base Course Material," Technical Report No. AFN-TR-71-60, Air Force Weapons Laboratory, Kirkland AFB, New Mexico, July 1971.
- [9] Duncan, J.M., and Chang, C.Y., "Nonlinear Analysis of Stress and Strain on Soils," Journal of Soil Mechanics and Foundation Div., ASCE, Vol. 96, No. SM5, September 1970, pp. 1629-1653.
- [10] Wong, K.S., and Duncan, J.M., "Hyperbolic Stress-Strain Parameters for Nonlinear Finite Element Analysis of Stresses and Movements in Soil Masses," Report No. TE-74-3, University of California, Berkeley, California, 1974.
- [11] McVay, M.C., and Selig, E.T., "Soil Model and Finite Element Boundary Studies," Report No. ACP81-283I, Dept. of Civil Engineering, University of Massachusetts, Amherst, Mass., July 1981.
- [12] Katona, M.G., Smith, J.M., Odello, R.S., and Allgood, J.R., "CANDE - A Modern Approach for the Structural Design and Analysis of Buried Culverts," Report No. FHWA-RD-77-5, FHWA, October 1976.

- [13] Sanderson, M. C., McVay, M. C., Dorwart, B. C., and Selig, E. T., "Soil Properties for Bucks County Culvert Project," Report No. RSC80-242R, Prepared for Republic Steel Corporation, Dept. of Civil Engineering, University of Massachusetts, Amherst, Massachusetts, June, 1981.
- [14] Konder, R.L., "Hyperbolic Stress-Strain Response: Cohesive Soils," Journal of the Soil Mechanics and Foundations Div., ASCE, Vol. 98, No. SM1, 1963, pp. 115-143.
- [15] Janbu, N., "Soil Compressibility as Determined by Oedometer and Triaxial Tests," European Conference on Soil Mechanics & Foundations Engineering, Wiesbaden, Germany Vol. 1, 1963, pp. 19-25.
- [16] McVay, M. C., "Evaluation of Numerical Modeling of Buried Conduits," Ph.D. Dissertation, Dept of Civil Engineering, University of Massachusetts, Amherst, Massachusetts, Feb. 1982.
- [17] Goodman, R. E., Taylor, R. L., and Brekke, T., "A Model for the Mechanics of Jointed Rock," Journal of the Soil Mechanics and Foundations Div., ASCE, Vol. 94, No. SM3, Proc. Paper 5937, May, 1968, pp. 473-496
- [18] Leonards, G. A., and Wu, T. H., "Predicting performance of Buried Conduits", Report No. 6-36-62F, Final Report to Indiana State Highway Commission and FHWA, Engineering Experiment Station, Purdue University, W. Lafayette, Indiana, March, 1981.
- [19] Chan, S. H., and Tuba, I. S., "A Finite Element Method for Contact Problems of Solid Bodies," International Journal of Mechanical Science, Vol. 13, No. 7, July, 1971, pp 615-639.
- [20] Chang, C. S., Espinoza, J. M., and Selig, E. T., "Computer Analysis of Newtown Creek Culvert," J. of the Geotechnical Engineering Div., ASCE, May 1980, pp531-556
- [21] Brown, C.B., "Forces on Rigid Culverts under High Fills," J. Structural Div., ASCE, 93(ST5), October 1967, pp. 195-215.
- [22] Vaslestad, J., Johansen, T.H., and Holm, W., "Load Reduction on Rigid Culverts Beneath High Fills: Long-term Behavior," Transportation Research Record 1415, Washington, D.C., 1993.
- [23] Kulhawy, F. H., and Duncan, J. M., and Seed, H. B., "Finite Element Analysis of Stresses and Movements in Embankments During Construction," Geotechnical Engineering Research Report No. TE-69-4, Department of Civil Engineering, University of California, Berkeley, November, 1969.
- [24] Ozawa, Y., and Duncan, J.M., "ISBILD: A Computer Program for Analysis of Static Stresses and Movements in Embankment," Report No. TE-73-4, University of California, Berkeley, California, 1973.

- [25] Allen, D.L., and Meade, B.W., "Analysis of Loads and Settlements for Reinforced Concrete Culverts," Research Report UKTRP-84-22, Kentucky Transportation Research Program, University of Kentucky, Kentucky, 1984.
- [26] Katona, M.G., and Vittes, P.D., "Soil-Structure Analysis and Evaluation of Buried Box-Culvert Designs," Transportation Research Record 878, Washington, D.C., 1982.
- [27] Tadros, M. K., Benak, J. V., and Gilliland, M. K., "Soil Pressure on Box Culverts", ACI Structural Jour., Vol. 86, No. 4, July-August 1989. pp439-450.
- [28] Musser, S.C., "CANDE-89 User Manual," Report No. FHWA-RD-89-169, FHWA, June 1989.
- [29] Yang, M. Z., Drumm, E. C., Bennett, R. M. and Mauldon, M. "Influence of Compactive Effort on Earth Pressure around a Box Culvert" Proceedings, 9th International Conference of the Association for Computer Methods and Advances in Geomechanics (IACMAG 97), Wuhan, China, 1997, pp2021-2026.9
- [30] Hibbit, Karlsson & Sorensen, Inc. ABAQUS/Standard User's Manual, Ver. 5.8, 1998.
- [31] Spangler, M.G., and Handy, R.L., Soil Engineering, 4th Ed., Harper and Row, New York, 1982.
- [32] Spangler, M.G., "A Theory on Loads on Negative Projecting Conduits," Proceedings, Highway Research Board, 30, 1950, pp. 153-161.
- [33] Duncan, J. M., Byrne, P., Wong, K. S., and Mabry, P., "Strength, Stress-Strain and Bulk Modulus Parameters for Finite Element Analysis of Stresses and Movements in Soil Masses," Report No. UCB/Gt/80-01, University of California, Berkeley California, August 1980.
- [34] Clough, G. W., and Duncan, J. M., "Finite Element Analysis of Retaining Wall Behavior," J. of Soil Mechanics and Foundations Div., ASCE, December 1971, pp.1657-1673.
- [35] ACI Building Code Requirements for Structural Concrete (318-99) and Commentary (318R-99), American Concrete Institute, Farmington Hills, Mich., 1999.
- [36] Katona, M. G., Meinhert, D. F., Orillac, R. and Lee, C. H., "Structural Evaluation of New Concepts for Long-Span Culverts and Culvert Installation," Report No. FHWA-RD-79-115, Intrim Report to Federal Highway Administration, December 1979.
- [37] Watkins, R. K., and Anderson, L. R., "Structural Mechanics of Buried Pipes," CRC Press LLC, Boca Raton, Florida, 2000.

APPENDICES

APPENDIX A

Typical Input for ABAQUS

APPENDIX B

Typical Input for ISBILD

APPENDIX C

Typical Input for CANDE-89

APPENDIX D

ISBILD Source Code

APPENDIX E

CANDE-89 Source Code

APPENDIX A

TYPICAL INPUT for ABAQUS

```
*HEADING
LOAD ON THE CONCRETE BOX CULVERT
*RESTART,WRITE,FREQUENCY=5
*NODE
1, 0.,-40.
4, 6.,-40.
13,74.,-40.
*NODE,NSET=THREENO
5, 10.,-40.
6, 16.,-40.
7, 24.,-40.
*NGEN
1,4,1
7,13,1
*NSET,NSET=BOT,GENERATE
1,13,1
*NCOPY,CHANGE NUMBER=52,SHIFT,OLD SET=BOT,NEW SET=HLINE1
0.,28.,0.
0.,0.,0.,0.,0.,0.,0.
*NFILL
BOT,HLINE1,4,13
*NCOPY,CHANGE NUMBER=68,SHIFT,OLD SET=THREENO
0.,30.,0.
0.,0.,0.,0.,0.,0.,0.
*NCOPY,CHANGE NUMBER=84,SHIFT,OLD SET=THREENO
0.,32.,0.
0.,0.,0.,0.,0.,0.,0.
*NCOPY,CHANGE NUMBER=111,SHIFT,OLD SET=THREENO
0.,34.,0.
0.,0.,0.,0.,0.,0.,0.
*NCOPY,CHANGE NUMBER=176,SHIFT,OLD SET=THREENO
0.,42.,0.
0.,0.,0.,0.,0.,0.,0.
*NODE
66, 0.,-10.
72, 6.,-10.
81,74.,-10.
82, 0.,-8.
88, 6.,-8.
97,74.,-8.
98, 0.,-7.
101,3.,-7.
107,3.,-1.
110,0.,-1.
167,0.,0.
170,3.,0.
171,0.,1.
176,5.,1.
177,0.,2.
180,6.,2.
189,74.,2.
111, 4.,-7.
112, 5.,-7.
113, 4.,-6.
114, 5.,-6.
115, 6.,-6.
124,74.,-6.
```



```

*NGEN
66,72
75,81
82,88
91,97
98,101
101,107
107,110
167,170
171,176
177,180
183,189
118,124
*NSET,NSET=LANO,GENERATE
111,124
*NCOPY,CHANGE NUMBER=14,SHIFT,OLD SET=LANO
0.,2.,0.
0.,0.,0.,0.,0.,0.,0.
*NCOPY,CHANGE NUMBER=28,SHIFT,OLD SET=LANO
0.,4.,0.
0.,0.,0.,0.,0.,0.,0.
*NCOPY,CHANGE NUMBER=42,SHIFT,OLD SET=LANO
0.,6.,0.
0.,0.,0.,0.,0.,0.,0.
*NSET,NSET=FILLBOT,GENERATE
177,189
*NCOPY,CHANGE NUMBER=247,SHIFT,OLD SET=FILLBOT, NEW SET=FILLTOP
0.,98.,0.
0.,0.,0.,0.,0.,0.,0.
*NFILL
FILLBOT,FILLTOP,19,13
*NSET,NSET=SI
1,14,27,40,53,66,82,98,110,167,171,
26,39,52,65,81,97,124,138,152,166
*NSET,NSET=DES,GENERATE
177,424,13
189,436,13
*NSET,NSET=SIDESBC
SI,DES
*NSET,NSET=BOTTOMBC
BOT
*NSET,NSET=CHECKNO,GENERATE
98,110,1
*ELEMENT,TYPE=CPE4,ELSET=CULV
82,82,83,99,98
83,83,84,100,99
84,84,85,101,100
85,85,86,111,101
86,101,111,113,102
87,102,113,125,103
88,103,125,127,104
89,104,127,139,105
90,105,139,141,106
91,106,141,153,107
92,107,153,155,170
93,108,107,170,169
94,109,108,169,168
95,110,109,168,167
*ELEMENT,TYPE=CPE4
1,1,2,15,14
58,56,57,73,72
67,66,67,83,82
101,88,89,116,115

```

```

115,115,116,130,129
167,157,158,181,180
96,86,87,112,111
97,111,112,114,113
110,113,114,126,125
111,125,126,128,127
124,127,128,140,139
125,139,140,142,141
138,141,142,154,153
139,153,154,156,155
152,167,168,172,171
153,168,169,173,172
154,169,170,174,173
155,170,155,175,174
156,155,156,176,175
*ELGEN
1,12,1,1,4,13,12
58,9,1,1
67,15,1,1
101,9,1,1
115,9,1,1,3,14,14
167,9,1,1
*ELEMENT,TYPE=CPE3
98,112,115,114
99,112,88,115
100,87,88,112
112,126,129,128
113,126,115,129
114,114,115,126
157,171,172,177
158,172,178,177
159,172,173,178
160,173,174,178
161,174,179,178
162,174,175,179
163,175,176,179
164,176,180,179
165,176,157,180
166,156,157,176
49,53,67,66
50,53,54,67
51,54,68,67
52,54,69,68
53,54,55,69
54,69,55,70
55,55,71,70
56,55,56,71
57,71,56,72
*ELGEN
112,3,14,14
113,3,14,14
114,3,14,14
*ELEMENT,TYPE=CPE4
176,177,178,191,190
*ELGEN
176,12,1,1,19,13,12
*ELSET,ELSET=F1,GENERATE
1,12
*ELSET,ELSET=F2,GENERATE
13,24
*ELSET,ELSET=F3,GENERATE
25,36
*ELSET,ELSET=F4,GENERATE

```

37, 48
 *ELSET, ELSET=F5, GENERATE
 49, 66
 *ELSET, ELSET=F6, GENERATE
 67, 81
 *ELSET, ELSET=LACOM, GENERATE
 96, 102
 *ELSET, ELSET=LBCOM, GENERATE
 110, 116
 *ELSET, ELSET=LCCOM, GENERATE
 124, 130
 *ELSET, ELSET=LDCOM, GENERATE
 138, 144
 *ELSET, ELSET=LA, GENERATE
 103, 109
 *ELSET, ELSET=LB, GENERATE
 117, 123
 *ELSET, ELSET=LC, GENERATE
 131, 137
 *ELSET, ELSET=LD, GENERATE
 145, 151
 *ELSET, ELSET=L1, GENERATE
 152, 175
 *ELSET, ELSET=L2, GENERATE
 176, 187
 *ELSET, ELSET=L3, GENERATE
 188, 199
 *ELSET, ELSET=L4, GENERATE
 200, 211
 *ELSET, ELSET=L5, GENERATE
 212, 223
 *ELSET, ELSET=L6, GENERATE
 224, 235
 *ELSET, ELSET=L7, GENERATE
 236, 247
 *ELSET, ELSET=L8, GENERATE
 248, 259
 *ELSET, ELSET=L9, GENERATE
 260, 271
 *ELSET, ELSET=L10, GENERATE
 272, 283
 *ELSET, ELSET=L11, GENERATE
 284, 295
 *ELSET, ELSET=L12, GENERATE
 296, 307
 *ELSET, ELSET=L13, GENERATE
 308, 319
 *ELSET, ELSET=L14, GENERATE
 320, 331
 *ELSET, ELSET=L15, GENERATE
 332, 343
 *ELSET, ELSET=L16, GENERATE
 344, 355
 *ELSET, ELSET=L17, GENERATE
 356, 367
 *ELSET, ELSET=L18, GENERATE
 368, 379
 *ELSET, ELSET=L19, GENERATE
 380, 391
 *ELSET, ELSET=L20, GENERATE
 392, 403
 *ELSET, ELSET=CHECKEL
 152, 153, 154, 155, 156,

92,93,94,95,
 96,97,110,111,124,125,138,139
 *SOLID SECTION,ELSET=L1,MATERIAL=L1
 *SOLID SECTION,ELSET=L2,MATERIAL=L2
 *SOLID SECTION,ELSET=L3,MATERIAL=L3
 *SOLID SECTION,ELSET=L4,MATERIAL=L4
 *SOLID SECTION,ELSET=L5,MATERIAL=L5
 *SOLID SECTION,ELSET=L6,MATERIAL=L6
 *SOLID SECTION,ELSET=L7,MATERIAL=L7
 *SOLID SECTION,ELSET=L8,MATERIAL=L8
 *SOLID SECTION,ELSET=L9,MATERIAL=L9
 *SOLID SECTION,ELSET=L10,MATERIAL=L10
 *SOLID SECTION,ELSET=L11,MATERIAL=L11
 *SOLID SECTION,ELSET=L12,MATERIAL=L12
 *SOLID SECTION,ELSET=L13,MATERIAL=L13
 *SOLID SECTION,ELSET=L14,MATERIAL=L14
 *SOLID SECTION,ELSET=L15,MATERIAL=L15
 *SOLID SECTION,ELSET=L16,MATERIAL=L16
 *SOLID SECTION,ELSET=L17,MATERIAL=L17
 *SOLID SECTION,ELSET=L18,MATERIAL=L18
 *SOLID SECTION,ELSET=L19,MATERIAL=L19
 *SOLID SECTION,ELSET=L20,MATERIAL=L20
 *SOLID SECTION,ELSET=LA,MATERIAL=LA
 *SOLID SECTION,ELSET=LB,MATERIAL=LB
 *SOLID SECTION,ELSET=LC,MATERIAL=LC
 *SOLID SECTION,ELSET=LD,MATERIAL=LD
 *SOLID SECTION,ELSET=LACOM,MATERIAL=LACOM
 *SOLID SECTION,ELSET=LBCOM,MATERIAL=LBCOM
 *SOLID SECTION,ELSET=LCCOM,MATERIAL=LCCOM
 *SOLID SECTION,ELSET=LDCOM,MATERIAL=LDCOM
 *SOLID SECTION,ELSET=F1,MATERIAL=F1
 *SOLID SECTION,ELSET=F2,MATERIAL=F2
 *SOLID SECTION,ELSET=F3,MATERIAL=F3
 *SOLID SECTION,ELSET=F4,MATERIAL=F4
 *SOLID SECTION,ELSET=F5,MATERIAL=F5
 *SOLID SECTION,ELSET=F6,MATERIAL=F6
 *MATERIAL,NAME=L20
 *ELASTIC
 238.,0.425
 *DENSITY
 0.125
 *MATERIAL,NAME=L19
 *ELASTIC
 313.,0.401
 *DENSITY
 0.125
 *MATERIAL,NAME=L18
 *ELASTIC
 356.,0.388
 *DENSITY
 0.125
 *MATERIAL,NAME=L17
 *ELASTIC
 387.,0.378
 *DENSITY
 0.125
 *MATERIAL,NAME=L16
 *ELASTIC
 412.,0.370
 *DENSITY
 0.125
 *MATERIAL,NAME=L15
 *ELASTIC

433., 0.364
 *DENSITY
 0.125
 *MATERIAL, NAME=L14
 *ELASTIC
 452., 0.358
 *DENSITY
 0.125
 *MATERIAL, NAME=L13
 *ELASTIC
 468., 0.353
 *DENSITY
 0.125
 *MATERIAL, NAME=L12
 *ELASTIC
 483., 0.348
 *DENSITY
 0.125
 *MATERIAL, NAME=L11
 *ELASTIC
 497., 0.344
 *DENSITY
 0.125
 *MATERIAL, NAME=L10
 *ELASTIC
 509., 0.340
 *DENSITY
 0.125
 *MATERIAL, NAME=L9
 *ELASTIC
 521., 0.336
 *DENSITY
 0.125
 *MATERIAL, NAME=L8
 *ELASTIC
 532., 0.332
 *DENSITY
 0.125
 *MATERIAL, NAME=L7
 *ELASTIC
 542., 0.329
 *DENSITY
 0.125
 *MATERIAL, NAME=L6
 *ELASTIC
 552., 0.326
 *DENSITY
 0.125
 *MATERIAL, NAME=L5
 *ELASTIC
 561., 0.323
 *DENSITY
 0.125
 *MATERIAL, NAME=L4
 *ELASTIC
 570., 0.320
 *DENSITY
 0.125
 *MATERIAL, NAME=L3
 *ELASTIC
 578., 0.318
 *DENSITY
 0.125

```

*MATERIAL,NAME=L2
*ELASTIC
587.,0.315
*DENSITY
0.125
*MATERIAL,NAME=L1
*ELASTIC
592.,0.313
*DENSITY
0.125
*MATERIAL,NAME=LD
*ELASTIC
595.,0.313
*DENSITY
0.125
*MATERIAL,NAME=LC
*ELASTIC
598.,0.312
*DENSITY
0.125
*MATERIAL,NAME=LB
*ELASTIC
601.,0.311
*DENSITY
0.125
*MATERIAL,NAME=LA
*ELASTIC
604.,0.310
*DENSITY
0.125
*MATERIAL,NAME=LDCOM
*ELASTIC
595.,0.313
*DENSITY
0.125
*MATERIAL,NAME=LCCOM
*ELASTIC
598.,0.312
*DENSITY
0.125
*MATERIAL,NAME=LBCOM
*ELASTIC
601.,0.311
*DENSITY
0.125
*MATERIAL,NAME=LACOM
*ELASTIC
604.,0.310
*DENSITY
0.125
*MATERIAL,NAME=F6
*ELASTIC
353.,0.450
*DENSITY
0.126
*MATERIAL,NAME=F5
*ELASTIC
362.,0.450
*DENSITY
0.126
*MATERIAL,NAME=F4
*ELASTIC
376.,0.450

```

```

*DENSITY
0.126
*MATERIAL,NAME=F3
*ELASTIC
400.,0.450
*DENSITY
0.126
*MATERIAL,NAME=F2
*ELASTIC
431.,0.450
*DENSITY
0.126
*MATERIAL,NAME=F1
*ELASTIC
465.,0.450
*DENSITY
0.126
*SOLID SECTION,ELSET=CULV,MATERIAL=CULV
*MATERIAL,NAME=CULV
*ELASTIC
519120,0.250
*DENSITY
0.15
*STEP
  1ST ANALYSIS
*STATIC
*BOUNDARY
  BOTTOMBC,1,2
  SIDESBC,1
*DLOAD
  F1,  GRAV,1.0,0.,-1.,0.
  F2,  GRAV,1.0,0.,-1.,0.
  F3,  GRAV,1.0,0.,-1.,0.
  F4,  GRAV,1.0,0.,-1.,0.
  F5,  GRAV,1.0,0.,-1.,0.
  F6,  GRAV,1.0,0.,-1.,0.
  CULV, GRAV,1.0,0.,-1.,0.
*MODEL CHANGE, REMOVE
  LA, LB, LC, LD, LACOM, LBCOM, LCCOM, LDCOM,
  L1, L2, L3, L4, L5, L6, L7, L8, L9, L10,
  L11, L12, L13, L14, L15, L16, L17, L18, L19, L20
*NODE PRINT, NSET=CHECKNO
  U, RF
*EL PRINT, ELSET=CHECKEL, POSITION=CENTROIDAL
  S, E
*NODE FILE, NSET=CHECKNO
  U
*EL FILE, ELSET=CHECKEL
  S, E
*END STEP
*STEP
  add construction layer LB
*STATIC
*MODEL CHANGE, ADD=STRAINFREE
  LA, LACOM,
*DLOAD, OP=MOD
  LA, GRAV, 1.0, 0., -1., 0.
  LACOM, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
  add construction layer LB
*STATIC
*MODEL CHANGE, ADD=STRAINFREE

```



```

LB,LBCOM,
*DLOAD,OP=MOD
LB, GRAV, 1.0, 0., -1., 0.
LBCOM, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
add construction layer LC
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
LC,LCCOM,
*DLOAD,OP=MOD
LC, GRAV, 1.0, 0., -1., 0.
LCCOM, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
add construction layer LD
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
LD,LDCOM
*DLOAD,OP=MOD
LD, GRAV, 1.0, 0., -1., 0.
LDCOM, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
add 1st construction layer
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
L1,
*DLOAD,OP=MOD
L1, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
add 2nd construction layer
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
L2,
*DLOAD,OP=MOD
L2, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
add 3rd construction layer
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
L3,
*DLOAD,OP=MOD
L3, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
add 4th construction layer
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
L4,
*DLOAD,OP=MOD
L4, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
add 5th construction layer
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
L5,
*DLOAD,OP=MOD
L5, GRAV, 1.0, 0., -1., 0.
*END STEP

```

```

*STEP
  add 6th construction layer
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
  L6,
*DLOAD,OP=MOD
  L6, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
  add 7th construction layer
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
  L7,
*DLOAD,OP=MOD
  L7, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
  add 8th construction layer
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
  L8,
*DLOAD,OP=MOD
  L8, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
  add 9th construction layer
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
  L9,
*DLOAD,OP=MOD
  L9, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
  add 11th construction layer
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
  L10,
*DLOAD,OP=MOD
  L10, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
  add construction layer L11
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
  L11,
*DLOAD,OP=MOD
  L11, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
  add construction layer L12
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
  L12,
*DLOAD,OP=MOD
  L12, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
  add construction layer L13
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
  L13,
*DLOAD,OP=MOD
  L13, GRAV, 1.0, 0., -1., 0.

```

```

*END STEP
*STEP
  add construction layer L14
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
  L14,
*DLOAD,OP=MOD
  L14, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
  add construction layer L15
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
  L15,
*DLOAD,OP=MOD
  L15, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
  add construction layer L16
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
  L16,
*DLOAD,OP=MOD
  L16, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
  add construction layer L17
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
  L17,
*DLOAD,OP=MOD
  L17, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
  add construction layer L18
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
  L18,
*DLOAD,OP=MOD
  L18, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
  add construction layer L19
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
  L19,
*DLOAD,OP=MOD
  L19, GRAV, 1.0, 0., -1., 0.
*END STEP
*STEP
  add construction layer L20
*STATIC
*MODEL CHANGE,ADD=STRAINFREE
  L20,
*DLOAD,OP=MOD
  L20, GRAV, 1.0, 0., -1., 0.
*END STEP

```

APPENDIX B

TYPICAL INPUT for ISBILD

CULVERT ANALYSIS

```

403,436,81,97,14,30,4,24,0,0,1,872,
2.116,
1,0.126,78.,78.,1.24,0.0077,0.50,0.0014,0.0,26.,0.92,0.5,
2,0.125,300.,300.,0.25,6.2000,0.26,0.0400,0.0,32.,0.70,0.0,
3,0.125,300.,300.,0.25,6.2000,0.26,0.0400,0.0,32.,0.70,0.0,
4,0.150,245330.,245330.,0.01,6.2,0.26,0.04,0.8,89.,0.63,0.,
1,0.,-40.000,1,1,
2,2.0000,-40.000,1,1,
3,4.0000,-40.000,1,1,
4,6.0000,-40.000,1,1,
5,10.000,-40.000,1,1,
6,16.000,-40.000,1,1,
7,24.000,-40.000,1,1,
8,32.333,-40.000,1,1,
9,40.667,-40.000,1,1,
10,49.000,-40.000,1,1,
11,57.333,-40.000,1,1,
12,65.667,-40.000,1,1,
13,74.000,-40.000,1,1,
14,0.,-33.000,1,,
15,2.0000,-33.000,,
16,4.0000,-33.000,,
17,6.0000,-33.000,,
18,10.000,-33.000,,
19,16.000,-33.000,,
20,24.000,-33.000,,
21,32.333,-33.000,,
22,40.667,-33.000,,
23,49.000,-33.000,,
24,57.333,-33.000,,
25,65.667,-33.000,,
26,74.000,-33.000,1,
27,0.,-26.000,1,
28,2.0000,-26.000,,
29,4.0000,-26.000,,
30,6.0000,-26.000,,
31,10.000,-26.000,,
32,16.000,-26.000,,
33,24.000,-26.000,,
34,32.333,-26.000,,
35,40.667,-26.000,,
36,49.000,-26.000,,
37,57.333,-26.000,,
38,65.667,-26.000,,
39,74.000,-26.000,1,
40,0.,-19.000,1,
41,2.0000,-19.000,,
42,4.0000,-19.000,,
43,6.0000,-19.000,,
44,10.000,-19.000,,
45,16.000,-19.000,,
46,24.000,-19.000,,
47,32.333,-19.000,,
48,40.667,-19.000,,
49,49.000,-19.000,,
50,57.333,-19.000,,

```

51,	65.667,	-19.000,	,	,
52,	74.000,	-19.000,	1,	,
53,	0.,	-12.000,	1,	,
54,	2.0000,	-12.000,	,	,
55,	4.0000,	-12.000,	,	,
56,	6.0000,	-12.000,	,	,
57,	10.000,	-12.000,	,	,
58,	16.000,	-12.000,	,	,
59,	24.000,	-12.000,	,	,
60,	32.333,	-12.000,	,	,
61,	40.667,	-12.000,	,	,
62,	49.000,	-12.000,	,	,
63,	57.333,	-12.000,	,	,
64,	65.667,	-12.000,	,	,
65,	74.000,	-12.000,	1,	,
66,	0.,	-10.000,	1,	,
67,	1.0000,	-10.000,	,	,
68,	2.0000,	-10.000,	,	,
69,	3.0000,	-10.000,	,	,
70,	4.0000,	-10.000,	,	,
71,	5.0000,	-10.000,	,	,
72,	6.0000,	-10.000,	,	,
73,	10.000,	-10.000,	,	,
74,	16.000,	-10.000,	,	,
75,	24.000,	-10.000,	,	,
76,	32.333,	-10.000,	,	,
77,	40.667,	-10.000,	,	,
78,	49.000,	-10.000,	,	,
79,	57.333,	-10.000,	,	,
80,	65.667,	-10.000,	,	,
81,	74.000,	-10.000,	1,	,
82,	0.,	-8.0000,	1,	,
83,	1.0000,	-8.0000,	,	,
84,	2.0000,	-8.0000,	,	,
85,	3.0000,	-8.0000,	,	,
86,	4.0000,	-8.0000,	,	,
87,	5.0000,	-8.0000,	,	,
88,	6.0000,	-8.0000,	,	,
89,	10.000,	-8.0000,	,	,
90,	16.000,	-8.0000,	,	,
91,	24.000,	-8.0000,	,	,
92,	32.333,	-8.0000,	,	,
93,	40.667,	-8.0000,	,	,
94,	49.000,	-8.0000,	,	,
95,	57.333,	-8.0000,	,	,
96,	65.667,	-8.0000,	,	,
97,	74.000,	-8.0000,	1,	,
98,	0.,	-7.0000,	1,	,
99,	1.0000,	-7.0000,	,	,
100,	2.0000,	-7.0000,	,	,
101,	3.0000,	-7.0000,	,	,
102,	3.0000,	-6.0000,	,	,
103,	3.0000,	-5.0000,	,	,
104,	3.0000,	-4.0000,	,	,
105,	3.0000,	-3.0000,	,	,
106,	3.0000,	-2.0000,	,	,
107,	3.0000,	-1.0000,	,	,
108,	2.0000,	-1.0000,	,	,
109,	1.0000,	-1.0000,	,	,
110,	0.,	-1.0000,	1,	,
111,	4.0000,	-7.0000,	,	,
112,	5.0000,	-7.0000,	,	,
113,	4.0000,	-6.0000,	,	,

114,	5.0000,	-6.0000,	,	,
115,	6.0000,	-6.0000,	,	,
116,	10.000,	-6.0000,	,	,
117,	16.000,	-6.0000,	,	,
118,	24.000,	-6.0000,	,	,
119,	32.333,	-6.0000,	,	,
120,	40.667,	-6.0000,	,	,
121,	49.000,	-6.0000,	,	,
122,	57.333,	-6.0000,	,	,
123,	65.667,	-6.0000,	,	,
124,	74.000,	-6.0000,	1,	,
125,	4.0000,	-5.0000,	,	,
126,	5.0000,	-5.0000,	,	,
127,	4.0000,	-4.0000,	,	,
128,	5.0000,	-4.0000,	,	,
129,	6.0000,	-4.0000,	,	,
130,	10.000,	-4.0000,	,	,
131,	16.000,	-4.0000,	,	,
132,	24.000,	-4.0000,	,	,
133,	32.333,	-4.0000,	,	,
134,	40.667,	-4.0000,	,	,
135,	49.000,	-4.0000,	,	,
136,	57.333,	-4.0000,	,	,
137,	65.667,	-4.0000,	,	,
138,	74.000,	-4.0000,	1,	,
139,	4.0000,	-3.0000,	,	,
140,	5.0000,	-3.0000,	,	,
141,	4.0000,	-2.0000,	,	,
142,	5.0000,	-2.0000,	,	,
143,	6.0000,	-2.0000,	,	,
144,	10.000,	-2.0000,	,	,
145,	16.000,	-2.0000,	,	,
146,	24.000,	-2.0000,	,	,
147,	32.333,	-2.0000,	,	,
148,	40.667,	-2.0000,	,	,
149,	49.000,	-2.0000,	,	,
150,	57.333,	-2.0000,	,	,
151,	65.667,	-2.0000,	,	,
152,	74.000,	-2.0000,	1,	,
153,	4.0000,	-1.0000,	,	,
154,	5.0000,	-1.0000,	,	,
155,	4.0000,	0.,	,	,
156,	5.0000,	0.,	,	,
157,	6.0000,	0.,	,	,
158,	10.000,	0.,	,	,
159,	16.000,	0.,	,	,
160,	24.000,	0.,	,	,
161,	32.333,	0.,	,	,
162,	40.667,	0.,	,	,
163,	49.000,	0.,	,	,
164,	57.333,	0.,	,	,
165,	65.667,	0.,	,	,
166,	74.000,	0.,	1,	,
167,	0.,	0.,	1,	,
168,	1.0000,	0.,	,	,
169,	2.0000,	0.,	,	,
170,	3.0000,	0.,	,	,
171,	0.,	1.0000,	1,	,
172,	1.0000,	1.0000,	,	,
173,	2.0000,	1.0000,	,	,
174,	3.0000,	1.0000,	,	,
175,	4.0000,	1.0000,	,	,
176,	5.0000,	1.0000,	,	,

177,	0.,	2.0000,	1, ,
178,	2.0000,	2.0000,	, ,
179,	4.0000,	2.0000,	, ,
180,	6.0000,	2.0000,	, ,
181,	10.000,	2.0000,	, ,
182,	16.000,	2.0000,	, ,
183,	24.000,	2.0000,	, ,
184,	32.333,	2.0000,	, ,
185,	40.667,	2.0000,	, ,
186,	49.000,	2.0000,	, ,
187,	57.333,	2.0000,	, ,
188,	65.667,	2.0000,	, ,
189,	74.000,	2.0000,	1, ,
190,	0.,	7.1579,	1, ,
191,	2.0000,	7.1579,	, ,
192,	4.0000,	7.1579,	, ,
193,	6.0000,	7.1579,	, ,
194,	10.000,	7.1579,	, ,
195,	16.000,	7.1579,	, ,
196,	24.000,	7.1579,	, ,
197,	32.333,	7.1579,	, ,
198,	40.667,	7.1579,	, ,
199,	49.000,	7.1579,	, ,
200,	57.333,	7.1579,	, ,
201,	65.667,	7.1579,	, ,
202,	74.000,	7.1579,	1, ,
203,	0.,	12.316,	1, ,
204,	2.0000,	12.316,	, ,
205,	4.0000,	12.316,	, ,
206,	6.0000,	12.316,	, ,
207,	10.000,	12.316,	, ,
208,	16.000,	12.316,	, ,
209,	24.000,	12.316,	, ,
210,	32.333,	12.316,	, ,
211,	40.667,	12.316,	, ,
212,	49.000,	12.316,	, ,
213,	57.333,	12.316,	, ,
214,	65.667,	12.316,	, ,
215,	74.000,	12.316,	1, ,
216,	0.,	17.474,	1, ,
217,	2.0000,	17.474,	, ,
218,	4.0000,	17.474,	, ,
219,	6.0000,	17.474,	, ,
220,	10.000,	17.474,	, ,
221,	16.000,	17.474,	, ,
222,	24.000,	17.474,	, ,
223,	32.333,	17.474,	, ,
224,	40.667,	17.474,	, ,
225,	49.000,	17.474,	, ,
226,	57.333,	17.474,	, ,
227,	65.667,	17.474,	, ,
228,	74.000,	17.474,	1, ,
229,	0.,	22.632,	1, ,
230,	2.0000,	22.632,	, ,
231,	4.0000,	22.632,	, ,
232,	6.0000,	22.632,	, ,
233,	10.000,	22.632,	, ,
234,	16.000,	22.632,	, ,
235,	24.000,	22.632,	, ,
236,	32.333,	22.632,	, ,
237,	40.667,	22.632,	, ,
238,	49.000,	22.632,	, ,
239,	57.333,	22.632,	, ,

240,	65.667,	22.632,	,	,
241,	74.000,	22.632,	1,	,
242,	0.,	27.789,	1,	,
243,	2.0000,	27.789,	,	,
244,	4.0000,	27.789,	,	,
245,	6.0000,	27.789,	,	,
246,	10.000,	27.789,	,	,
247,	16.000,	27.789,	,	,
248,	24.000,	27.789,	,	,
249,	32.333,	27.789,	,	,
250,	40.667,	27.789,	,	,
251,	49.000,	27.789,	,	,
252,	57.333,	27.789,	,	,
253,	65.667,	27.789,	,	,
254,	74.000,	27.789,	1,	,
255,	0.,	32.947,	1,	,
256,	2.0000,	32.947,	,	,
257,	4.0000,	32.947,	,	,
258,	6.0000,	32.947,	,	,
259,	10.000,	32.947,	,	,
260,	16.000,	32.947,	,	,
261,	24.000,	32.947,	,	,
262,	32.333,	32.947,	,	,
263,	40.667,	32.947,	,	,
264,	49.000,	32.947,	,	,
265,	57.333,	32.947,	,	,
266,	65.667,	32.947,	,	,
267,	74.000,	32.947,	1,	,
268,	0.,	38.105,	1,	,
269,	2.0000,	38.105,	,	,
270,	4.0000,	38.105,	,	,
271,	6.0000,	38.105,	,	,
272,	10.000,	38.105,	,	,
273,	16.000,	38.105,	,	,
274,	24.000,	38.105,	,	,
275,	32.333,	38.105,	,	,
276,	40.667,	38.105,	,	,
277,	49.000,	38.105,	,	,
278,	57.333,	38.105,	,	,
279,	65.667,	38.105,	,	,
280,	74.000,	38.105,	1,	,
281,	0.,	43.263,	1,	,
282,	2.0000,	43.263,	,	,
283,	4.0000,	43.263,	,	,
284,	6.0000,	43.263,	,	,
285,	10.000,	43.263,	,	,
286,	16.000,	43.263,	,	,
287,	24.000,	43.263,	,	,
288,	32.333,	43.263,	,	,
289,	40.667,	43.263,	,	,
290,	49.000,	43.263,	,	,
291,	57.333,	43.263,	,	,
292,	65.667,	43.263,	,	,
293,	74.000,	43.263,	1,	,
294,	0.,	48.421,	1,	,
295,	2.0000,	48.421,	,	,
296,	4.0000,	48.421,	,	,
297,	6.0000,	48.421,	,	,
298,	10.000,	48.421,	,	,
299,	16.000,	48.421,	,	,
300,	24.000,	48.421,	,	,
301,	32.333,	48.421,	,	,
302,	40.667,	48.421,	,	,

303,	49.000,	48.421,	,	,
304,	57.333,	48.421,	,	,
305,	65.667,	48.421,	,	,
306,	74.000,	48.421,	1,	,
307,	0.,	53.579,	1,	,
308,	2.0000,	53.579,	,	,
309,	4.0000,	53.579,	,	,
310,	6.0000,	53.579,	,	,
311,	10.000,	53.579,	,	,
312,	16.000,	53.579,	,	,
313,	24.000,	53.579,	,	,
314,	32.333,	53.579,	,	,
315,	40.667,	53.579,	,	,
316,	49.000,	53.579,	,	,
317,	57.333,	53.579,	,	,
318,	65.667,	53.579,	,	,
319,	74.000,	53.579,	1,	,
320,	0.,	58.737,	1,	,
321,	2.0000,	58.737,	,	,
322,	4.0000,	58.737,	,	,
323,	6.0000,	58.737,	,	,
324,	10.000,	58.737,	,	,
325,	16.000,	58.737,	,	,
326,	24.000,	58.737,	,	,
327,	32.333,	58.737,	,	,
328,	40.667,	58.737,	,	,
329,	49.000,	58.737,	,	,
330,	57.333,	58.737,	,	,
331,	65.667,	58.737,	,	,
332,	74.000,	58.737,	1,	,
333,	0.,	63.895,	1,	,
334,	2.0000,	63.895,	,	,
335,	4.0000,	63.895,	,	,
336,	6.0000,	63.895,	,	,
337,	10.000,	63.895,	,	,
338,	16.000,	63.895,	,	,
339,	24.000,	63.895,	,	,
340,	32.333,	63.895,	,	,
341,	40.667,	63.895,	,	,
342,	49.000,	63.895,	,	,
343,	57.333,	63.895,	,	,
344,	65.667,	63.895,	,	,
345,	74.000,	63.895,	1,	,
346,	0.,	69.053,	1,	,
347,	2.0000,	69.053,	,	,
348,	4.0000,	69.053,	,	,
349,	6.0000,	69.053,	,	,
350,	10.000,	69.053,	,	,
351,	16.000,	69.053,	,	,
352,	24.000,	69.053,	,	,
353,	32.333,	69.053,	,	,
354,	40.667,	69.053,	,	,
355,	49.000,	69.053,	,	,
356,	57.333,	69.053,	,	,
357,	65.667,	69.053,	,	,
358,	74.000,	69.053,	1,	,
359,	0.,	74.211,	1,	,
360,	2.0000,	74.211,	,	,
361,	4.0000,	74.211,	,	,
362,	6.0000,	74.211,	,	,
363,	10.000,	74.211,	,	,
364,	16.000,	74.211,	,	,
365,	24.000,	74.211,	,	,

366,	32.333,	74.211,	,	,
367,	40.667,	74.211,	,	,
368,	49.000,	74.211,	,	,
369,	57.333,	74.211,	,	,
370,	65.667,	74.211,	,	,
371,	74.000,	74.211,	1,	,
372,	0.,	79.368,	1,	,
373,	2.0000,	79.368,	,	,
374,	4.0000,	79.368,	,	,
375,	6.0000,	79.368,	,	,
376,	10.000,	79.368,	,	,
377,	16.000,	79.368,	,	,
378,	24.000,	79.368,	,	,
379,	32.333,	79.368,	,	,
380,	40.667,	79.368,	,	,
381,	49.000,	79.368,	,	,
382,	57.333,	79.368,	,	,
383,	65.667,	79.368,	,	,
384,	74.000,	79.368,	1,	,
385,	0.,	84.526,	1,	,
386,	2.0000,	84.526,	,	,
387,	4.0000,	84.526,	,	,
388,	6.0000,	84.526,	,	,
389,	10.000,	84.526,	,	,
390,	16.000,	84.526,	,	,
391,	24.000,	84.526,	,	,
392,	32.333,	84.526,	,	,
393,	40.667,	84.526,	,	,
394,	49.000,	84.526,	,	,
395,	57.333,	84.526,	,	,
396,	65.667,	84.526,	,	,
397,	74.000,	84.526,	1,	,
398,	0.,	89.684,	1,	,
399,	2.0000,	89.684,	,	,
400,	4.0000,	89.684,	,	,
401,	6.0000,	89.684,	,	,
402,	10.000,	89.684,	,	,
403,	16.000,	89.684,	,	,
404,	24.000,	89.684,	,	,
405,	32.333,	89.684,	,	,
406,	40.667,	89.684,	,	,
407,	49.000,	89.684,	,	,
408,	57.333,	89.684,	,	,
409,	65.667,	89.684,	,	,
410,	74.000,	89.684,	1,	,
411,	0.,	94.842,	1,	,
412,	2.0000,	94.842,	,	,
413,	4.0000,	94.842,	,	,
414,	6.0000,	94.842,	,	,
415,	10.000,	94.842,	,	,
416,	16.000,	94.842,	,	,
417,	24.000,	94.842,	,	,
418,	32.333,	94.842,	,	,
419,	40.667,	94.842,	,	,
420,	49.000,	94.842,	,	,
421,	57.333,	94.842,	,	,
422,	65.667,	94.842,	,	,
423,	74.000,	94.842,	1,	,
424,	0.,	100.00,	1,	,
425,	2.0000,	100.00,	,	,
426,	4.0000,	100.00,	,	,
427,	6.0000,	100.00,	,	,
428,	10.000,	100.00,	,	,

429,	16.000,	100.00,	,	,	
430,	24.000,	100.00,	,	,	
431,	32.333,	100.00,	,	,	
432,	40.667,	100.00,	,	,	
433,	49.000,	100.00,	,	,	
434,	57.333,	100.00,	,	,	
435,	65.667,	100.00,	,	,	
436,	74.000,	100.00,	1,	,	
1,	1,	2,	15,	14,	1,
2,	2,	3,	16,	15,	1,
3,	3,	4,	17,	16,	1,
4,	4,	5,	18,	17,	1,
5,	5,	6,	19,	18,	1,
6,	6,	7,	20,	19,	1,
7,	7,	8,	21,	20,	1,
8,	8,	9,	22,	21,	1,
9,	9,	10,	23,	22,	1,
10,	10,	11,	24,	23,	1,
11,	11,	12,	25,	24,	1,
12,	12,	13,	26,	25,	1,
13,	14,	15,	28,	27,	1,
14,	15,	16,	29,	28,	1,
15,	16,	17,	30,	29,	1,
16,	17,	18,	31,	30,	1,
17,	18,	19,	32,	31,	1,
18,	19,	20,	33,	32,	1,
19,	20,	21,	34,	33,	1,
20,	21,	22,	35,	34,	1,
21,	22,	23,	36,	35,	1,
22,	23,	24,	37,	36,	1,
23,	24,	25,	38,	37,	1,
24,	25,	26,	39,	38,	1,
25,	27,	28,	41,	40,	1,
26,	28,	29,	42,	41,	1,
27,	29,	30,	43,	42,	1,
28,	30,	31,	44,	43,	1,
29,	31,	32,	45,	44,	1,
30,	32,	33,	46,	45,	1,
31,	33,	34,	47,	46,	1,
32,	34,	35,	48,	47,	1,
33,	35,	36,	49,	48,	1,
34,	36,	37,	50,	49,	1,
35,	37,	38,	51,	50,	1,
36,	38,	39,	52,	51,	1,
37,	40,	41,	54,	53,	1,
38,	41,	42,	55,	54,	1,
39,	42,	43,	56,	55,	1,
40,	43,	44,	57,	56,	1,
41,	44,	45,	58,	57,	1,
42,	45,	46,	59,	58,	1,
43,	46,	47,	60,	59,	1,
44,	47,	48,	61,	60,	1,
45,	48,	49,	62,	61,	1,
46,	49,	50,	63,	62,	1,
47,	50,	51,	64,	63,	1,
48,	51,	52,	65,	64,	1,
49,	53,	67,	66,	66,	1,
50,	53,	54,	67,	67,	1,
51,	54,	68,	67,	67,	1,
52,	54,	69,	68,	68,	1,
53,	54,	55,	69,	69,	1,
54,	69,	55,	70,	70,	1,
55,	55,	71,	70,	70,	1,

56,	55,	56,	71,	71,	1,
57,	71,	56,	72,	72,	1,
58,	56,	57,	73,	72,	1,
59,	57,	58,	74,	73,	1,
60,	58,	59,	75,	74,	1,
61,	59,	60,	76,	75,	1,
62,	60,	61,	77,	76,	1,
63,	61,	62,	78,	77,	1,
64,	62,	63,	79,	78,	1,
65,	63,	64,	80,	79,	1,
66,	64,	65,	81,	80,	1,
67,	66,	67,	83,	82,	1,
68,	67,	68,	84,	83,	1,
69,	68,	69,	85,	84,	1,
70,	69,	70,	86,	85,	1,
71,	70,	71,	87,	86,	1,
72,	71,	72,	88,	87,	1,
73,	72,	73,	89,	88,	1,
74,	73,	74,	90,	89,	1,
75,	74,	75,	91,	90,	1,
76,	75,	76,	92,	91,	1,
77,	76,	77,	93,	92,	1,
78,	77,	78,	94,	93,	1,
79,	78,	79,	95,	94,	1,
80,	79,	80,	96,	95,	1,
81,	80,	81,	97,	96,	1,
82,	82,	83,	99,	98,	4,
83,	83,	84,	100,	99,	4,
84,	84,	85,	101,	100,	4,
85,	85,	86,	111,	101,	4,
86,	101,	111,	113,	102,	4,
87,	102,	113,	125,	103,	4,
88,	103,	125,	127,	104,	4,
89,	104,	127,	139,	105,	4,
90,	105,	139,	141,	106,	4,
91,	106,	141,	153,	107,	4,
92,	107,	153,	155,	170,	4,
93,	108,	107,	170,	169,	4,
94,	109,	108,	169,	168,	4,
95,	110,	109,	168,	167,	4,
96,	86,	87,	112,	111,	3,
97,	111,	112,	114,	113,	3,
98,	112,	115,	114,	114,	3,
99,	112,	88,	115,	115,	3,
100,	87,	88,	112,	112,	3,
101,	88,	89,	116,	115,	3,
102,	89,	90,	117,	116,	3,
103,	90,	91,	118,	117,	2,
104,	91,	92,	119,	118,	2,
105,	92,	93,	120,	119,	2,
106,	93,	94,	121,	120,	2,
107,	94,	95,	122,	121,	2,
108,	95,	96,	123,	122,	2,
109,	96,	97,	124,	123,	2,
110,	113,	114,	126,	125,	3,
111,	125,	126,	128,	127,	3,
112,	126,	129,	128,	128,	3,
113,	126,	115,	129,	129,	3,
114,	114,	115,	126,	126,	3,
115,	115,	116,	130,	129,	3,
116,	116,	117,	131,	130,	3,
117,	117,	118,	132,	131,	2,
118,	118,	119,	133,	132,	2,

119,	119,	120,	134,	133,	2,
120,	120,	121,	135,	134,	2,
121,	121,	122,	136,	135,	2,
122,	122,	123,	137,	136,	2,
123,	123,	124,	138,	137,	2,
124,	127,	128,	140,	139,	3,
125,	139,	140,	142,	141,	3,
126,	140,	143,	142,	142,	3,
127,	140,	129,	143,	143,	3,
128,	128,	129,	140,	140,	3,
129,	129,	130,	144,	143,	3,
130,	130,	131,	145,	144,	3,
131,	131,	132,	146,	145,	2,
132,	132,	133,	147,	146,	2,
133,	133,	134,	148,	147,	2,
134,	134,	135,	149,	148,	2,
135,	135,	136,	150,	149,	2,
136,	136,	137,	151,	150,	2,
137,	137,	138,	152,	151,	2,
138,	141,	142,	154,	153,	3,
139,	153,	154,	156,	155,	3,
140,	154,	157,	156,	156,	3,
141,	154,	143,	157,	157,	3,
142,	142,	143,	154,	154,	3,
143,	143,	144,	158,	157,	3,
144,	144,	145,	159,	158,	3,
145,	145,	146,	160,	159,	2,
146,	146,	147,	161,	160,	2,
147,	147,	148,	162,	161,	2,
148,	148,	149,	163,	162,	2,
149,	149,	150,	164,	163,	2,
150,	150,	151,	165,	164,	2,
151,	151,	152,	166,	165,	2,
152,	167,	168,	172,	171,	2,
153,	168,	169,	173,	172,	2,
154,	169,	170,	174,	173,	2,
155,	170,	155,	175,	174,	2,
156,	155,	156,	176,	175,	2,
157,	171,	172,	177,	177,	2,
158,	172,	178,	177,	177,	2,
159,	172,	173,	178,	178,	2,
160,	173,	174,	178,	178,	2,
161,	174,	179,	178,	178,	2,
162,	174,	175,	179,	179,	2,
163,	175,	176,	179,	179,	2,
164,	176,	180,	179,	179,	2,
165,	176,	157,	180,	180,	2,
166,	156,	157,	176,	176,	2,
167,	157,	158,	181,	180,	2,
168,	158,	159,	182,	181,	2,
169,	159,	160,	183,	182,	2,
170,	160,	161,	184,	183,	2,
171,	161,	162,	185,	184,	2,
172,	162,	163,	186,	185,	2,
173,	163,	164,	187,	186,	2,
174,	164,	165,	188,	187,	2,
175,	165,	166,	189,	188,	2,
176,	177,	178,	191,	190,	2,
177,	178,	179,	192,	191,	2,
178,	179,	180,	193,	192,	2,
179,	180,	181,	194,	193,	2,
180,	181,	182,	195,	194,	2,
181,	182,	183,	196,	195,	2,

182,	183,	184,	197,	196,	2,
183,	184,	185,	198,	197,	2,
184,	185,	186,	199,	198,	2,
185,	186,	187,	200,	199,	2,
186,	187,	188,	201,	200,	2,
187,	188,	189,	202,	201,	2,
188,	190,	191,	204,	203,	2,
189,	191,	192,	205,	204,	2,
190,	192,	193,	206,	205,	2,
191,	193,	194,	207,	206,	2,
192,	194,	195,	208,	207,	2,
193,	195,	196,	209,	208,	2,
194,	196,	197,	210,	209,	2,
195,	197,	198,	211,	210,	2,
196,	198,	199,	212,	211,	2,
197,	199,	200,	213,	212,	2,
198,	200,	201,	214,	213,	2,
199,	201,	202,	215,	214,	2,
200,	203,	204,	217,	216,	2,
201,	204,	205,	218,	217,	2,
202,	205,	206,	219,	218,	2,
203,	206,	207,	220,	219,	2,
204,	207,	208,	221,	220,	2,
205,	208,	209,	222,	221,	2,
206,	209,	210,	223,	222,	2,
207,	210,	211,	224,	223,	2,
208,	211,	212,	225,	224,	2,
209,	212,	213,	226,	225,	2,
210,	213,	214,	227,	226,	2,
211,	214,	215,	228,	227,	2,
212,	216,	217,	230,	229,	2,
213,	217,	218,	231,	230,	2,
214,	218,	219,	232,	231,	2,
215,	219,	220,	233,	232,	2,
216,	220,	221,	234,	233,	2,
217,	221,	222,	235,	234,	2,
218,	222,	223,	236,	235,	2,
219,	223,	224,	237,	236,	2,
220,	224,	225,	238,	237,	2,
221,	225,	226,	239,	238,	2,
222,	226,	227,	240,	239,	2,
223,	227,	228,	241,	240,	2,
224,	229,	230,	243,	242,	2,
225,	230,	231,	244,	243,	2,
226,	231,	232,	245,	244,	2,
227,	232,	233,	246,	245,	2,
228,	233,	234,	247,	246,	2,
229,	234,	235,	248,	247,	2,
230,	235,	236,	249,	248,	2,
231,	236,	237,	250,	249,	2,
232,	237,	238,	251,	250,	2,
233,	238,	239,	252,	251,	2,
234,	239,	240,	253,	252,	2,
235,	240,	241,	254,	253,	2,
236,	242,	243,	256,	255,	2,
237,	243,	244,	257,	256,	2,
238,	244,	245,	258,	257,	2,
239,	245,	246,	259,	258,	2,
240,	246,	247,	260,	259,	2,
241,	247,	248,	261,	260,	2,
242,	248,	249,	262,	261,	2,
243,	249,	250,	263,	262,	2,
244,	250,	251,	264,	263,	2,

245,	251,	252,	265,	264,	2,
246,	252,	253,	266,	265,	2,
247,	253,	254,	267,	266,	2,
248,	255,	256,	269,	268,	2,
249,	256,	257,	270,	269,	2,
250,	257,	258,	271,	270,	2,
251,	258,	259,	272,	271,	2,
252,	259,	260,	273,	272,	2,
253,	260,	261,	274,	273,	2,
254,	261,	262,	275,	274,	2,
255,	262,	263,	276,	275,	2,
256,	263,	264,	277,	276,	2,
257,	264,	265,	278,	277,	2,
258,	265,	266,	279,	278,	2,
259,	266,	267,	280,	279,	2,
260,	268,	269,	282,	281,	2,
261,	269,	270,	283,	282,	2,
262,	270,	271,	284,	283,	2,
263,	271,	272,	285,	284,	2,
264,	272,	273,	286,	285,	2,
265,	273,	274,	287,	286,	2,
266,	274,	275,	288,	287,	2,
267,	275,	276,	289,	288,	2,
268,	276,	277,	290,	289,	2,
269,	277,	278,	291,	290,	2,
270,	278,	279,	292,	291,	2,
271,	279,	280,	293,	292,	2,
272,	281,	282,	295,	294,	2,
273,	282,	283,	296,	295,	2,
274,	283,	284,	297,	296,	2,
275,	284,	285,	298,	297,	2,
276,	285,	286,	299,	298,	2,
277,	286,	287,	300,	299,	2,
278,	287,	288,	301,	300,	2,
279,	288,	289,	302,	301,	2,
280,	289,	290,	303,	302,	2,
281,	290,	291,	304,	303,	2,
282,	291,	292,	305,	304,	2,
283,	292,	293,	306,	305,	2,
284,	294,	295,	308,	307,	2,
285,	295,	296,	309,	308,	2,
286,	296,	297,	310,	309,	2,
287,	297,	298,	311,	310,	2,
288,	298,	299,	312,	311,	2,
289,	299,	300,	313,	312,	2,
290,	300,	301,	314,	313,	2,
291,	301,	302,	315,	314,	2,
292,	302,	303,	316,	315,	2,
293,	303,	304,	317,	316,	2,
294,	304,	305,	318,	317,	2,
295,	305,	306,	319,	318,	2,
296,	307,	308,	321,	320,	2,
297,	308,	309,	322,	321,	2,
298,	309,	310,	323,	322,	2,
299,	310,	311,	324,	323,	2,
300,	311,	312,	325,	324,	2,
301,	312,	313,	326,	325,	2,
302,	313,	314,	327,	326,	2,
303,	314,	315,	328,	327,	2,
304,	315,	316,	329,	328,	2,
305,	316,	317,	330,	329,	2,
306,	317,	318,	331,	330,	2,
307,	318,	319,	332,	331,	2,

308,	320,	321,	334,	333,	2,
309,	321,	322,	335,	334,	2,
310,	322,	323,	336,	335,	2,
311,	323,	324,	337,	336,	2,
312,	324,	325,	338,	337,	2,
313,	325,	326,	339,	338,	2,
314,	326,	327,	340,	339,	2,
315,	327,	328,	341,	340,	2,
316,	328,	329,	342,	341,	2,
317,	329,	330,	343,	342,	2,
318,	330,	331,	344,	343,	2,
319,	331,	332,	345,	344,	2,
320,	333,	334,	347,	346,	2,
321,	334,	335,	348,	347,	2,
322,	335,	336,	349,	348,	2,
323,	336,	337,	350,	349,	2,
324,	337,	338,	351,	350,	2,
325,	338,	339,	352,	351,	2,
326,	339,	340,	353,	352,	2,
327,	340,	341,	354,	353,	2,
328,	341,	342,	355,	354,	2,
329,	342,	343,	356,	355,	2,
330,	343,	344,	357,	356,	2,
331,	344,	345,	358,	357,	2,
332,	346,	347,	360,	359,	2,
333,	347,	348,	361,	360,	2,
334,	348,	349,	362,	361,	2,
335,	349,	350,	363,	362,	2,
336,	350,	351,	364,	363,	2,
337,	351,	352,	365,	364,	2,
338,	352,	353,	366,	365,	2,
339,	353,	354,	367,	366,	2,
340,	354,	355,	368,	367,	2,
341,	355,	356,	369,	368,	2,
342,	356,	357,	370,	369,	2,
343,	357,	358,	371,	370,	2,
344,	359,	360,	373,	372,	2,
345,	360,	361,	374,	373,	2,
346,	361,	362,	375,	374,	2,
347,	362,	363,	376,	375,	2,
348,	363,	364,	377,	376,	2,
349,	364,	365,	378,	377,	2,
350,	365,	366,	379,	378,	2,
351,	366,	367,	380,	379,	2,
352,	367,	368,	381,	380,	2,
353,	368,	369,	382,	381,	2,
354,	369,	370,	383,	382,	2,
355,	370,	371,	384,	383,	2,
356,	372,	373,	386,	385,	2,
357,	373,	374,	387,	386,	2,
358,	374,	375,	388,	387,	2,
359,	375,	376,	389,	388,	2,
360,	376,	377,	390,	389,	2,
361,	377,	378,	391,	390,	2,
362,	378,	379,	392,	391,	2,
363,	379,	380,	393,	392,	2,
364,	380,	381,	394,	393,	2,
365,	381,	382,	395,	394,	2,
366,	382,	383,	396,	395,	2,
367,	383,	384,	397,	396,	2,
368,	385,	386,	399,	398,	2,
369,	386,	387,	400,	399,	2,
370,	387,	388,	401,	400,	2,

371,	388,	389,	402,	401,	2,
372,	389,	390,	403,	402,	2,
373,	390,	391,	404,	403,	2,
374,	391,	392,	405,	404,	2,
375,	392,	393,	406,	405,	2,
376,	393,	394,	407,	406,	2,
377,	394,	395,	408,	407,	2,
378,	395,	396,	409,	408,	2,
379,	396,	397,	410,	409,	2,
380,	398,	399,	412,	411,	2,
381,	399,	400,	413,	412,	2,
382,	400,	401,	414,	413,	2,
383,	401,	402,	415,	414,	2,
384,	402,	403,	416,	415,	2,
385,	403,	404,	417,	416,	2,
386,	404,	405,	418,	417,	2,
387,	405,	406,	419,	418,	2,
388,	406,	407,	420,	419,	2,
389,	407,	408,	421,	420,	2,
390,	408,	409,	422,	421,	2,
391,	409,	410,	423,	422,	2,
392,	411,	412,	425,	424,	2,
393,	412,	413,	426,	425,	2,
394,	413,	414,	427,	426,	2,
395,	414,	415,	428,	427,	2,
396,	415,	416,	429,	428,	2,
397,	416,	417,	430,	429,	2,
398,	417,	418,	431,	430,	2,
399,	418,	419,	432,	431,	2,
400,	419,	420,	433,	432,	2,
401,	420,	421,	434,	433,	2,
402,	421,	422,	435,	434,	2,
403,	422,	423,	436,	435,	2,
1,	96,109,111,124,	86,113,124,	97		
2,	110,123,125,138,	113,127,138,124			
3,	124,137,139,152,	127,141,152,138			
4,	138,151,153,166,	141,155,166,152			
5,	152,175,167,189,	167,177,189,166			
6,	176,187,190,202,	177,190,202,189			
7,	188,199,203,215,	190,203,215,202			
8,	200,211,216,228,	203,216,228,215			
9,	212,223,229,241,	216,229,241,228			
10,	224,235,242,254,	229,242,254,241			
11,	236,247,255,267,	242,255,267,254			
12,	248,259,268,280,	255,268,280,267			
13,	260,271,281,293,	268,281,293,280			
14,	272,283,294,306,	281,294,306,293			
15,	284,295,307,319,	294,307,319,306			
16,	296,307,320,332,	307,320,332,319			
17,	308,319,333,345,	320,333,345,332			
18,	320,331,346,358,	333,346,358,345			
19,	332,343,359,371,	346,359,371,358			
20,	344,355,372,384,	359,372,384,371			
21,	356,367,385,397,	372,385,397,384			
22,	368,379,398,410,	385,398,410,397			
23,	380,391,411,423,	398,411,423,410			
24,	392,403,424,436,	411,424,436,423			
82,83,84,85,86,87,88,89,90,91,92,93,94,95					
82,83,84,85,86,88,99,100,101,102,103,104,105,106,107,					
108,109,110,111,113,125,127,139,141,153,155,167,168,169,170					
1,					
82, 0.0, 0.0, 0.0					
83, 0.0, 0.0, 0.0					

84, 0.0, 0.0, 0.0
85, 0.0, 0.0, 0.0
86, 0.0, 0.0, 0.0
87, 0.0, 0.0, 0.0
88, 0.0, 0.0, 0.0
89, 0.0, 0.0, 0.0
90, 0.0, 0.0, 0.0
91, 0.0, 0.0, 0.0
92, 0.0, 0.0, 0.0
93, 0.0, 0.0, 0.0
94, 0.0, 0.0, 0.0
95, 0.0, 0.0, 0.0
6,0.0
1,1, 1,12,8.0
2,1,13,24,8.0
3,1,25,36,7.0
4,1,37,48,5.0
5,1,49,66,2.0
6,1,67,81,2.0

APPENDIX C

TYPICAL INPUT for CANDE-89

ANALYS	2	CONCRE	BOX	CULVERT	8*6-8 (MEDIUM REINFORCED - H=77 FT)	2
		-1.0	10.0	STD	3 0.0001	
		4500.0			150.0 60000.0	
		10.0	10.0	10.0	10.0 0.0	
		.037	.050	.050	.017 1.0	
EMBA		EMBANKMENT - STIFF SOIL				
	0	1	3	20	29.00 29.00 77.00 138.00	12.00
	1	1		0.0	IN-SITU	
	100000.0			0.25		
	2	1		0.0	BEDDING-SOIL	
	4001.0			0.25		
L	3	1	138.0		FILL SOIL	
		2000.0		0.25		
STOP						

APPENDIX D

ISBILD SOURCE CODE

```

C THIS IS THE ORIGINAL ISBILD FINITE ELEMENT PROGRAM. SOME PRINTING CAN
C BE AVOIDED BY INPUTTING A 1(ONE) IN COLUMN 55 OF THE SECOND DATA CARD.
C THIS VARIABLE (MPRINT) WAS ADDED BY DAN EATON
C DATA FOR CONTOUR MAPPING CAN BE OUTPUT TO UNIT 7(SEARCH FOR **CNTUR**)
C
C PUNCH OUTPUT ARE STORED IN UNIT 4
C
C
C NUMELT=TOTAL NUMBER OF ELEMENTS IN THE COMPLETED STRUCTURE
C NUMNPT=TOTAL NUMBER OF NODAL POINTS IN THE COMPLETE STRUCTURE
C NFEL =NUMBER OF ELEMENTS IN THE FOUNDATION PART
C NFNP =NUMBER OF NODAL POINTS IN THE FOUNDATION PART
C NUMCEL=NUMBER OF ELEMENTS IN THE PREEXISTING PART
C NUMCNP=NUMBER OF NODAL POINTS IN THE PREEXISTING PART
C NUMMAT=NUMBER OF DIFFERENT MATERIAL TYPES
C NLAY =NUMBER OF CONSTRUCTION LAYER INCREMENTS
C NFORCE=NUMBER OF LOAD INCREMENTS AFTER CONSTRUCTION
C NPUNCH=NUMBER OF CONSTRUCTION LAYER INCREMENTS AND LOAD
C INCREMENTS FOR WHICH OUTPUT IS TO BE STORED IN A FILE. FOR
C EXAMPLE, IF THE OUTPUT FOR THE LAST CONSTRUCTION LAYER
C INCREMENT AND THE LAST LOAD INCREMENT AFTER CONSTRUCTION
C IS TO BE STORED, NPUNCH=2. ORDINARILY NPUNCH=0
C THE OUTPUT CONSISTS OF THE STRESS AND
C STRAIN FOR EACH ELEMENT AND THE DISPLACEMENT FOR EACH NODAL
C POINT. THIS OUTPUT MAY BE USED AS THE INPUT FOR
C THE PREEXISTING PART IN ANALYSES OF EMBANKMENTS ON OR
C CONTAINING PREEXISTING PARTS.
C
C INEQ=2*NUMNPT-SUM OF RESTRAINT (SUM OF ID ARRAY)
C
C IMPLICIT REAL*8 (A-H,O-Z)
C REAL*4 HEAD,T,TIME,AA
C DIMENSION HEAD(20),T(10),L(1),A(1)
C COMMON /ISOP/ EL,E2,RR(4),ZZ(4),LM(8),P(8),S(12,12),STR(3,12),
1 STS(3,8),VJAC,MPRINT
C COMMON AA(8000000)
C EQUIVALENCE (AA(1),L(1)),(AA(1),A(1))
C
C MTOTAL=RAM OF YOUR COMPUTER DIVIDED BY 4
C
C PROGRAM CAPACITY CONTROLLED BY THE FOLLOWING TWO STATEMENTS
C
C MTOTAL=8000000
C MTOTD2=MTOTAL/2
C IN=5
C IQ=6
C OPEN ( 4,FILE='F04',FORM='UNFORMATTED')
C OPEN ( 5,FILE=' ')
C OPEN ( 6,FILE=' ')
C OPEN ( 7,FILE='F07',FORM='UNFORMATTED')
C OPEN ( 8,FILE='F08',FORM='UNFORMATTED')
C OPEN ( 9,FILE='F09',FORM='UNFORMATTED')
C OPEN (10,FILE='F10',FORM='UNFORMATTED')
C OPEN (11,FILE='F11',FORM='UNFORMATTED')

```

```

OPEN (12,FILE='F12',FORM='UNFORMATTED')
OPEN (13,FILE='F13',FORM='UNFORMATTED')
OPEN (14,FILE='F14',FORM='UNFORMATTED')
OPEN (15,FILE='F15',FORM='UNFORMATTED')
OPEN (16,FILE='F16',FORM='UNFORMATTED')
C
100 READ (IN,1000,END=999) HEAD
READ (IN,*) NUMELT,NUMNPT,NFEL,NFNP,NUMCEL,NUMCNP,
1 NUMMAT,NLAY,NFORCE,NPUNCH,MPRINT,INEQ
CALL SECOND (T(1))
WRITE (IO,2000) HEAD
WRITE (IO,2010) NUMELT,NUMNPT,NFEL,NFNP,NUMCEL,NUMCNP,NUMMAT,NLAY,
1 NFORCE
NUMLD=NLAY+NFORCE
IF (NPUNCH.EQ.0) GO TO 110
WRITE (IO,2020)
GO TO 120
110 WRITE (IO,2030)
120 CONTINUE
C
C BLOCK OUT VARIABLES IN A-VECTOR
C
N1=1
N2=N1+INEQ
N3=N2+2*NUMNPT
N4=N3+5*NUMELT
N5=N4+NUMCEL+1
N6=N5+NUMCNP+1
N7=N6+2*NUMLD
N8=N7+2*NUMLD
N9=N8+4*NUMLD
NDEX=N9+NPUNCH+1
IF (NDEX/2*2.EQ.NDEX) N10=NDEX/2+1
IF (NDEX/2*2.NE.NDEX) N10=NDEX/2+2
N11=N10+13*NUMMAT
N12=N11+NUMNPT
N13=N12+NUMNPT
N14=N13+NUMELT
N15=N14+NUMELT
N16=N15+NUMELT
N17=N16+NUMELT
N18=N17+NUMNPT
N19=N18+NUMNPT
N20=N19+3*NUMELT
N21=N20+3*NUMELT
N22=N21+3*NUMELT
NN1=N21+NUMELT
MTMN17=MTOTD2-N17
NN2=N21+2*NUMNPT
IF (NN2.GT.N22) N22=NN2
IF (N22.LT.MTOTD2) GO TO 130
WRITE (IO,5000) N22
GO TO 999
C
C READ AND PRINT INPUT DATA AND SET UP INITIAL CONDITIONS
C
130 CALL LAYOUT (L(N1),L(N2),L(N3),L(N4),L(N5),L(N6),L(N7),L(N8),
1 L(N9),A(N10),A(N11),A(N12),A(N13),A(N14),A(N15),A(N16),
2 A(N17),A(N19),A(N20),A(N21),A(NN1),NUMELT,NUMNPT,NUMCEL,
3 NUMCNP,NFEL,NUMMAT,NUMLD,NLAY,NEQ,NEQB,MBAND,PATM,MTMN17,
4 NMKEQB,NPUNCH,IN,IO,INEQ)
CALL SECOND (T(2))
N23=N22+NEQ

```

```

N24=N23+NEQB
IF(N24.LT.MTODT2) GO TO 140
WRITE (IO,5000) N24
GO TO 999

C
C FORM STRAIN-DISPLACEMENT MATRIX FOR ALL ELEMENTS. STORE ON TAPE 7
C
140 CALL FORMST (A(N11),A(N12),L(N3),NUMELT)
CALL SECOND (T(3))
T(1)=T(2)-T(1)
T(2)=T(3)-T(2)
TIME=T(1)+T(2)
DO 400 LN=1,NUMLD
T(10)=0.0
WRITE (IO,2000) HEAD

C
C DETERMINE CONTROL DATA FOR EACH LAYER
C
CALL CALBLK (L(N2),L(N4),L(N5),L(N6),L(N7),L(N8),NUMELT,
1 NUMNPT,NUMCEL,NUMCNP,NUMLD,NLAY,LN,MBAND,NUMEL,NUMNP,
2 NELCAL,NNPCAL,NELRED,NNPRED,NEQ,NEQB,NBLOCK,NMXEQB,IO)
CALL SECOND (T(4))
NN1=N19+NEQ
NE2B=2*NEQB
NSB=(MBAND+1)*NEQB
NNN1=N17+NSB
NNN2=NNN1+NSB

C
C SETUP LOAD VECTOR
C
CALL FVECT (A(N10),L(N2),A(N11),A(N12),L(N3),L(N4),L(N6),A(N17),
1 A(N18),A(N19),A(NN1),NUMELT,NUMNPT,NUMCEL,NUMMAT,NUMLD,NLAY,LN,
2 NEQ,NEQB,NUMNP,IN,IO)
CALL SECOND (T(5))
T(3)=T(4)-T(3)
T(4)=T(5)-T(4)
NN1=N17+NE2B*MBAND
NN2=NN1+NE2B
DO 300 IT=1,2
CALL SECOND (T(5))

C
C CALCULATE ELEMENT STIFFNESS MATRIX FOR ALL ELEMENTS, STORE
C ON TAPE 13
C CALCULATE STRESS-DISPLACEMENT MATRIX FOR ALL ELEMENTS, STORE
C ON TAPE 11
C
CALL ISQUAD (L(N3),A(N13),A(N14),L(N4),NUMELT,NUMCEL,NUMEL,
1 NELCAL,NELRED)
CALL SECOND (T(6))

C
C FORM TOTAL STIFFNESS MATRIX, STORE ON TAPE 15
C
CALL ADDSTF (A(N17),A(NN1),NUMEL,NEQB,NE2B,NBLOCK,MBAND)
CALL SECOND (T(7))

C
C SOLVE FOR DISPLACEMENT UNKNOWNNS
C
CALL SYMBAN (A(N17),A(NN1),L(N1),NEQB,MBAND,NBLOCK,NSB,
1 15,14,12,13,13,IO)
CALL SECOND (T(8))

C
C EVALUATE RESULTS
C
CALL ISRSLT (A(N10),L(N2),L(N3),A(N13),A(N14),A(N15),A(N16),L(N4),
1 L(N5),L(N9),A(N17),A(N19),A(N20),A(N21),A(N22),A(N23),
2 PATM,NUMELT,NUMNPT,NUMCEL,NUMCNP,NUMMAT,NUMLD,NLAY,LN,IT,NPUNCH,
3 NUMEL,NUMNP,NELCAL,NNPCAL,NNPRED,NEQ,NEQB,NBLOCK,
4 A(N11),A(N12),IO)
CALL SECOND (T(9))
WRITE (IO,2000)
DO 250 I=5,8
T(I)=T(I+1)-T(I)
250 T(9)=T(5)+T(6)+T(7)+T(8)
WRITE (IO,2110) T(5),T(6),T(7),T(8),T(8)
IF(1T.LT.2) GO TO 280
T(10)=T(10)+T(3)+T(4)+T(9)
WRITE (IO,2120) T(3),T(4),T(10)
TIME=TIME+T(10)
GO TO 300
280 T(10)=T(10)+T(9)
300 CONTINUE
400 CONTINUE
WRITE (IO,2130) T(1),T(2),TIME
GO TO 100
999 CONTINUE
CLOSE ( 4, STATUS = 'DELETE')
CLOSE ( 5)
CLOSE ( 6)
CLOSE ( 7, STATUS = 'DELETE')
CLOSE ( 8, STATUS = 'DELETE')
CLOSE ( 9, STATUS = 'DELETE')
CLOSE (10, STATUS = 'DELETE')
CLOSE (11, STATUS = 'DELETE')
CLOSE (12, STATUS = 'DELETE')
CLOSE (13, STATUS = 'DELETE')
CLOSE (14, STATUS = 'DELETE')
CLOSE (15, STATUS = 'DELETE')
CLOSE (16, STATUS = 'DELETE')
1000 FORMAT (20A4)
2000 FORMAT (1H1,20A4)
2010 FORMAT (/,
1 35H0TOTAL NUMBER OF ELEMENTS*****,I3,/,
2 35H0TOTAL NUMBER OF NODES*****,I3,/,
3 35H0NUMBER OF ELEMENTS IN FOUNDATION**,I3,/,
4 35H0NUMBER OF NODES IN FOUNDATION*****,I3,/,
5 35H0NUMBER OF PREEXISTING ELEMENTS*****,I3,/,
6 35H0NUMBER OF PREEXISTING NODES*****,I3,/,
7 35H0NUMBER OF DIFF. MATERIALS*****,I3,/,
8 35H0NUMBER OF CONSTRUCTION LAYERS*****,I3,/,
9 35H0NUMBER OF LOAD CASES*****,I3)
2020 FORMAT (49H0RESULTS ARE PUNCHED OUT FOR FOLLOWING LOAD CASES,/)
2030 FORMAT (34H0FINAL RESULTS ARE NOT PUNCHED OUT,/)
2100 FORMAT ( ' ', 'SOLUTION TIME',/)
2110 FORMAT (/,
1 35H0FORM ELEMENT STIFFNESSES*****,F8.2,/,
2 35H0FORM TOTAL STIFFNESS*****,F8.2,/,
3 35H0EQUATION SOLVING*****,F8.2,/,
4 35H0CALCULATE STRESSES AND STRAINS*****,F8.2,/,
5 35H0SOLUTION TIME FOR THIS ITERATION**,F8.2)
2120 FORMAT (/,
1 35H0DETERMINE CONTROL DATA*****,F8.2,/,
2 35H0FORM LOAD VECTOR*****,F8.2,/,
3 35H0TOTAL TIME FOR THIS LOAD CASE*****,F8.2)
2130 FORMAT ( ' ', 'OVERALL LOG',/,
1 35H0DATA INPUT*****,F8.2,/,
2 35H0FORM STRAIN-DISPLACEMENT MATRIX***,F8.2,/,

```



```

3 35H0TOTAL SOLUTION TIME+*****F8.2)
5000 FORMAT (//,17H STORAGE EXCEEDED,I9)
END
SUBROUTINE ADDSTF (A,B,NUMEL,NEQB,NE2B,NBLOCK,MBAND)
IMPLICIT REAL*8 (A-H,O-Z)

C
C   FORMS GLOBAL EQUILIBRIUM EQUATIONS IN BLOCKS
C
COMMON /ISOP/ E1,E2,RR(4),Z2(4),LM(8),P(8),S(12,12),STR(3,12),
1 STA(3,8),VJAC,MPRINT
DIMENSION A(NE2B,MBAND),B(NE2B)
DATA NUM3/0/
K=NEQB+1
TEMP=NBLOCK
MB=DSQRT(TEMP)
MB=MB/2+1
NEBB=MB*NE2B
MM=1
NSHIFT=0

C
C   REWIND 10
C   REWIND 15
C
C   FORM EQUATIONS IN BLOCKS      ( 2 BLOCKS AT A TIME)
C
DO 500 M=1,NBLOCK,2
DO 100 J=1,MBAND
DO 100 I=1,NE2B
100 A(I,J)=0.0
READ (10) (B(I),I=1,NEQB)
IF (M.NE. NBLOCK) READ (10) (B(I),I=K,NE2B)

C
C   REWIND 13
C   REWIND 14
C   NA=14
C   NUME=NUM3
C   IF (MM.NE. 1) GO TO 150
C   NA=13
C   NUME=NUMEL
C   NUM3=0
150 DO 300 N=1,NUME
READ (NA) LM,((S(I,J),I=1,8),J=1,8)
DO 220 I=1,8
LMN=1-LM(I)
II=LM(I)-NSHIFT
IF (II.LE. 0 .OR. II.GT. NE2B) GO TO 220
DO 200 J=1,8
JJ=LM(J)+LMN
IF (JJ.GT. 0) A(II,JJ)=A(II,JJ)+S(I,J)
200 CONTINUE
220 CONTINUE

C
C   DETERMINE IF STIFFNESS IS TO BE PLACED ON TAPE 14
C
IF (MM.GT. 1) GO TO 300
DO 250 I=1,8
II=LM(I)-NSHIFT
IF (II.GT. NE2B .AND. II.LE. NEBB) GO TO 260
250 CONTINUE
GO TO 300
260 WRITE(14) LM,((S(I,J),I=1,8),J=1,8)
NUM3=NUM3+1
300 CONTINUE

```

```

WRITE(15) ((A(I,J),I=1,NEQB),J=1,MBAND), (B(I),I=1,NEQB)
IF (M.EQ. NBLOCK) GO TO 500
WRITE(15) ((A(I,J),I=K,NE2B),J=1,MBAND), (B(I),I=K,NE2B)
IF (MM.EQ.MM) MM=0
MM=MM+1
NSHIFT=NSHIFT+NE2B
500 CONTINUE
RETURN
END
SUBROUTINE CALBLK (ID,NCEL,NCNF,NOMEL,NOMNP,NPHUMP,NUMELT,NUMNPT,
1 NUMCEL,NUMCNP,NUMLD,NLAY,LN,MBAND,NUMEL,NUMNP,NELCAL,
2 NNPCAL,NELRED,NNPRED,NEQ,NEQB,NBLOCK,NMXEQB,IO)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION ID(NUMNPT,2),NCEL(1),NCNP(1)
DIMENSION NOMEL(NUMLD,2),NOMNP(NUMLD,2),NPHUMP(NUMLD,4)

C
C   DETERMINE CONTROL DATA
C
IF (LN.GT. NLAY) GO TO 10
WRITE (IO,2000) LN
WRITE (IO,2010) (NOMEL(LN,N),N=1,2), (NOMNP(LN,M),M=1,2),
1 (NPHUMP(LN,L),L=1,4)
GO TO 20
10 LNMLAY=LN-NLAY
WRITE (IO,2020) LNMLAY
20 CONTINUE
IF (LN.GT. NLAY) GO TO 80
IF (NUMCEL.EQ. 0) GO TO 50
NUMEL=MAX0(NOMEL(LN,2),NCEL(NUMCEL))
NUMNP=MAX0(NOMNP(LN,2),NCNP(NUMCNP))
GO TO 60
50 NUMEL=NOMEL(LN,2)
NUMNP=NOMNP(LN,2)
60 NELCAL=NOMEL(LN,2)
NNPCAL=NOMNP(LN,2)
NELRED=NOMEL(LN,1)
NNPRED=NOMNP(LN,1)
GO TO 100
80 NUMEL=NUMELT
NUMNP=NUMNPT
NELCAL=NUMELT
NNPCAL=NUMNPT
NELRED=NUMELT+1
NNPRED=NUMNPT+1
100 CONTINUE

C
C   DETERMINE NUMBER OF EQUATIONS AND BLOCKS
C
NEQB=NMXEQB
MAX=0
DO 120 I=1,2
DO 120 N=1,NUMNP
IF (ID(N,I).GT. MAX) MAX=ID(N,I)
120 CONTINUE
NEQ=MAX
IF (NEQB.GT. NEQ) NEQB=NEQ
NBLOCK=(NEQ-1)/NEQB+1
WRITE (IO,2100) MBAND,NEQ,NEQB,NBLOCK
RETURN
2000 FORMAT (//,
1 57H *****
2 57H *****
3 57H *

```

```

4 45H * LAYER NUMBER***** ,I4,8H
5 57H *
6 57H *****
7 57H *****
2010 FORMAT (/
1 35H0ADDED ELEMENTS***** ,I5,5H THRU,I5, /
2 35H0ADDED NODAL POINTS***** ,I5,5H THRU,I5, /
3 35H0NODAL POINTS OF HUMPED SURFACE*****,4I5, /)
2020 FORMAT (/
1 57H *****
2 57H *****
3 57H *
4 45H * LOAD CASE AFTER CONSTRUCTION***** ,I4,8H
5 57H *
6 57H *****
7 57H *****
2100 FORMAT (/
1 35H0BAND WIDTH***** ,I4, /
2 35H0TOTAL NUMBER OF EQUATIONS***** ,I4, /
3 35H0NUMBER OF EQUATIONS IN BLOCK***** ,I4, /
4 35H0NUMBER OF BLOCKS***** ,I4)
END
SUBROUTINE DERIVE
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /ISOP/ SX,TY,RR(4),ZZ(4),IM(8),P(8),S(12,12),STR(3,12),
1 STS(3,8),VJAC,MPRINT
DIMENSION H(6),HS(6),HT(6),HR(6),HZ(6)
DIMENSION II(6),JJ(6)
DATA II/1,3,5,7,9,11/,JJ/2,4,6,8,10,12/

MATRIX OF DERIVATIVES

SM=1.0D0-SX
SP=1.0D0+SX
TM=1.0D0-TY
TP=1.0D0+TY

H(1)=SM*TM/4.0D0
H(2)=SP*TM/4.0D0
H(3)=SP*TP/4.0D0
H(4)=SM*TP/4.0D0
H(5)=(1.0D0-SX*SX)
H(6)=(1.0D0-TY*TY)

HS(1)=-TM/4.0D0
HS(2)=-HS(1)
HS(3)=TP/4.0D0
HS(4)=-HS(3)
HS(5)=-SX-SX
HS(6)=0.0

HT(1)=-SM/4.0D0
HT(2)=-SP/4.0D0
HT(3)=-HT(2)
HT(4)=-HT(1)
HT(5)=0.0
HT(6)=-TY-TY

JACOBIAN

PZT=HT(1)*ZZ(1)+HT(2)*ZZ(2)+HT(3)*ZZ(3)+HT(4)*ZZ(4)
PZS=HS(1)*ZZ(1)+HS(2)*ZZ(2)+HS(3)*ZZ(3)+HS(4)*ZZ(4)
PRS=HS(1)*RR(1)+HS(2)*RR(2)+HS(3)*RR(3)+HS(4)*RR(4)

```

```

PRT=HT(1)*RR(1)+HT(2)*RR(2)+HT(3)*RR(3)+HT(4)*RR(4)
VJAC=PRS*PZT-PRT*PZS

C
PSR=PZT/VJAC
PTR=-PZS/VJAC
PSZ=-PRT/VJAC
PTZ=PRS/VJAC

C
DO 100 I=1,6
HR(I)=PSR*HS(I)+PTR*HT(I)
100 HZ(I)=PSZ*HS(I)+PTZ*HT(I)

C
C
C
FORM STRAIN DISPLACEMENT MATRIX

C
DO 200 K=1,6
I=I(K)
J=J(K)
STR(1,I)=HR(K)
STR(2,J)=HZ(K)
STR(3,I)=HZ(K)
200 STR(3,J)=HR(K)

C
RETURN
END
SUBROUTINE ELAW (EMPR,BULK,SHEAR,POIS,SLMAX,PRS,PATM,
1 NUMMAT,N,MTYPE,STRLEV,KK)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION EMPR(NUMMAT,13),BULK(1),SHEAR(1),POIS(1),SLMAX(1),PRS(2)

C
C
C
CALCULATE SHEAR MODULUS, BULK MODULUS AND POISSON'S RATIO VALUES

DEVSTR=PRS(1)-PRS(2)
WRITE(6,77) PRS(2)
C 77 FORMAT(2X,'PRS(2)',2X,F10.5)
DEVFH=EMPR(MTYPE,12)+EMPR(MTYPE,13)*PRS(2)
IF (DEVFH .GT. 0.0) GO TO 100
STRLEV=0.0
DEVLEV=0.0
GO TO 110
100 DEVLEV=DEVSTR/DEVFH
STRLEV=DEVLEV/EMPR(MTYPE,10)
110 CONTINUE
C
C
C
WRITE(6,88) KK
C 88 FORMAT(2X,'KK',2X,I2)
IF (KK .EQ. 1) GO TO 140
IF (PRS(2) .GT. 0.0) GO TO 120
POIS(N)=EMPR(MTYPE,6)
IF (POIS(N) .GT. 0.49D0) POIS(N) = 0.49D0
GO TO 130
120 IF (STRLEV .LT. 1.0D0 .AND. SHEAR(N) .GT. 0.0001D0) GO TO 140
C 130 PHIR=68.2-30.2*DLOG10(CF)
C
C
C
PI=355./113.
SHEAR(N)=DABS((PRS(1)*DTAN(PHIR*PI/180.)))/3.)
130 IF (SHEAR(N).LT.14400.) SHEAR(N)=360.
GO TO 200
140 CONTINUE
IF (PRS(2) .LT. 0.01D0) PRS(2)=0.01D0
PPRS2=PRS(2)/PATM
IF (KK .EQ. 3 .AND. STRLEV .LT. SLMAX(N)) GO TO 150
EINIT=PATM*EMPR(MTYPE,2)*PPRS2*EMPR(MTYPE,4)
TEMP=1.0D0-DEVLEV
EMOD=EINIT*(TEMP*TEMP)
IF (EMOD.LT.100.0) EMOD=100.0

```

```

      IF (EMOD .LE. 0.0) EMOD=1.00
      IF (EMOD.GT.9999999.0.AND.MTYPE.NE.3) EMOD=9999999.0
      GO TO 160
150  EMOD=PATM*EMPR(MTYPE,3)*PPRS2**EMPR(MTYPE,4)
      IF (EMOD.LT.100.0) EMOD=100.0
      IF (EMOD .LE. 0.0) EMOD=100.0
      IF (EMOD.GT.5000000.0.AND.MTYPE.NE.3) EMOD=5000000.0
160  CONTINUE
      POISI=EMPR(MTYPE,6)-EMPR(MTYPE,7)*DLOG10(PPRS2)
      EPSAX=DEVSTR/(EINIT*TEMP)
      TEMP=1.0D0-EMPR(MTYPE,5)*EPSAX
      POIS(N)=POISI/(TEMP*TEMP)
      IF (POIS(N) .GT. 0.49D0) POIS(N)=0.49D0
      IF (POIS(N) .LE. 0.0) POIS(N)=.10
      IF (POIS(N) .LT. 0.1) POIS(N)=0.1
      TEMP=1.0D0+POIS(N)
      SHEAR(N)=EMOD/(TEMP*TEMP)
      BULK(N)=SHEAR(N)/(1.0D0-POIS(N)-POIS(N))
      IF (KK .NE. 1) GO TO 200
      IF (STRLEV .GE. 1.0D0 .OR. PRS(2) .LE. 0.0) SHEAR(N)=0.0001D0
200  CONTINUE
      RETURN
      END
      SUBROUTINE FORMST(X,Y,INP,NUMELT)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ISOFF/ EL,E2,RR(4),ZZ(4),LM(8),P(8),S(12,12),STR(3,12),
1  STS(3,8),VJAC,MPRINT
      DIMENSION X(1),Y(1),INP(NUMELT,5)
      DIMENSION SSS(2),TTT(2)
      DATA SSS /-0.57735026918963D0,0.57735026918963D0/
      DATA TTT /-0.57735026918963D0,0.57735026918963D0/

      FORM STRAIN-DISPLACEMENT MATRIX

      REWIND 15
      REWIND 16
      DO 300 N=1,NUMELT
      DO 50 J=1,12
      DO 50 I=1,3
50  STR(I,J)=0.0
      READ (15) LM
      WRITE(16) LM
      DO 100 I=1,4
      II=INP(N,I)
      RR(I)=X(II)
100  ZZ(I)=Y(II)
      DO 200 II=1,2
      E1=SSS(II)
      DO 200 JJ=1,2
      E2=TTT(JJ)
      CALL DERIVE
      WRITE(16) VJAC,STR
200  CONTINUE
      E1=0.0
      E2=0.0
      CALL DERIVE
      WRITE(16) ((STR(I,J),I=1,3),J=1,8)
300  CONTINUE
      RETURN
      END
      SUBROUTINE FVECT (EMPR,ID,X,Y,INP,NCEL,NOMEL,FX,FY,B,R,
1  NUMELT,NUMNPT,NUMCEL,NUMMAT,NUMLD,NLAY,LN,NEQ,NEQB,NUMNE,IN,IO)
      IMPLICIT REAL*8 (A-H,O-Z)

```

```

      DIMENSION EMPR(NUMMAT,13),ID(NUMNPT,2),X(1),Y(1),INP(NUMELT,5)
      DIMENSION NCEL(1),NOMEL(NUMLD,2),FX(1),FY(1),B(1),R(NEQB)
C
C  CALCULATE NODAL POINT FORCES DUE TO WEIGHTS OF ADDED ELEMENTS
C
      DO 50 I=1,NUMNPT
      FX(I)=0.0
50  FY(I)=0.0
      IF (LN .GT. NLAY) GO TO 400
      NELS=NOMEL(LN,1)
      NELL=NOMEL(LN,2)
      DO 300 N=NELS,NELL
      IF (NUMCEL .EQ. 0) GO TO 100
      DO 80 M=1,NUMCEL
      IF (N .EQ. NCEL(M)) GO TO 300
80  CONTINUE
100  CONTINUE
      FG=0.0
      MTYPE=INP(N,5)
      KK=1
      I=INP(N,1)
      J=INP(N,2)
      K=INP(N,4)
120  AREA=X(J)*(Y(K)-Y(I))+X(I)*(Y(J)-Y(K))+X(K)*(Y(I)-Y(J))
      FG=FG-EMPR(MTYPE,1)*AREA/2.0D0
      IF (KK .NE. 1 .OR. INP(N,3) .EQ. INP(N,4)) GO TO 140
      KK=2
      I=INP(N,3)
      J=INP(N,4)
      K=INP(N,2)
      GO TO 120
140  CONTINUE
      IF (INP(N,3) .EQ. INP(N,4)) GO TO 200
      FG=FG/4.0D0
      KL=4
      GO TO 220
200  FG=FG/3.0D0
      KL=3
220  CONTINUE
      DO 250 I=1,KL
      II=INP(N,I)
250  FY(II)=FY(II)+FG
300  CONTINUE
      GO TO 650
400  CONTINUE
C
C  READ NODAL POINT FORCE DATA AND DISTRIBUTED LOAD DATA
C  FOR LOAD CASES AFTER CONSTRUCTION
C
      READ (IN,*) NUMFC,NUMPC
      WRITE (10,2000) NUMFC,NUMPC
      IF (NUMFC .EQ. 0) GO TO 550
      DO 520 I=1,NUMFC
520  READ (IN,*) MM,FX(MM),FY(MM)
550  CONTINUE
      IF (NUMPC .EQ. 0) GO TO 650
      WRITE (10,2010)
      DO 600 I=1,NUMPC
      READ (IN,*) IBC,JBC,PRES1,PRES2
      WRITE (10,2020) IBC,JBC,PRES1,PRES2
      DY=(Y(JBC)-Y(IBC))/2.0D0
      DX=(X(IBC)-X(JBC))/2.0D0
      TEMP1=PRES1+PRES1

```

```

      TEMP2=PRES2+PRES2
      FX(IBC)=FX(IBC)+(TEMP1+PRES2)*DY/3.000
      FY(JBC)=FY(JBC)+(PRES1+TEMP2)*DY/3.000
      FY(IBC)=FY(IBC)+(TEMP1+PRES2)*DX/3.000
      FY(JBC)=FY(JBC)+(PRES1+TEMP2)*DX/3.000
600  CONTINUE
650  CONTINUE
C
C      SET UP FORCE VECTOR
C
      DO 700 I=1,NEQ
700  B(I)=0.0
      DO 720 N=1,NUMNP
      DO 720 I=1,2
      II=ID(N,I)
      IF (II .LT. 1) GO TO 720
      IF (I .EQ. 2) GO TO 710
      B(II)=B(II)+FX(N)
      GO TO 720
710  B(II)=B(II)+FY(N)
720  CONTINUE
      REWIND 10
      KSHIFT=0
      DO 730 I=1,NEQB
730  R(I)=0.0
      DO 750 N=1,NEQ
      II=N-KSHIFT
      R(II)=B(N)
      IF (II .NE. NEQB) GO TO 750
      WRITE(10) R
      KSHIFT=KSHIFT+NEQB
      DO 740 I=1,NEQB
740  R(I)=0.0
750  CONTINUE
      WRITE(10) R
C
C      PRINT NODAL POINT FORCES
C
      IF (LN .LE. NLAY) GO TO 760
      WRITE (10,2030)
      GO TO 770
760  WRITE (10,2035)
770  CONTINUE
      DO 800 N=1,NUMNP
800  WRITE (10,2040) N,FX(N),FY(N)
C1000 FORMAT (2I5)
C1010 FORMAT (I5,2F10.0)
C1020 FORMAT (2I5,2F10.0)
2000 FORMAT (/,
1 35HNUMBER OF N.P. FORCE CARDS*****I3,/,
2 35HNUMBER OF PRESSURE CARDS*****I3)/,
2010 FORMAT (29H0PRESSURE BOUNDARY CONDITIONS,/,
1 34H I J PRESSURE I PRESSURE J,/)
2020 FORMAT (2I5,2F12.3)
2030 FORMAT (19H1NODAL POINT FORCES ,/,
1 25H NP X-FORCE Y-FORCE,/)
2035 FORMAT (47H1NODAL POINT FORCES (WEIGHTS OF ADDED ELEMENTS),/,
1 25H NP X-FORCE Y-FORCE,/)
2040 FORMAT (I5,2F10.3)
      RETURN
      END
      SUBROUTINE ISQUAD (INP,BULK,SHEAR,NCEL,NUMELT,NUMCEL,NUMEL,
1 NELCAL,NELRED)

```

```

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ISOE/ E1,E2,RR(4),ZZ(4),LM(6),P(8),S(12,12),STR(3,12),
1 STS(3,8),VJAC,MPRINT
      DIMENSION INP(NUMELT,5),BULK(1),SHEAR(1),NCEL(1)
      REWIND 13
      REWIND 11
      REWIND 16
C
C      INITIALIZATION
C
      DO 300 N=1,NUMEL
      KEY=0
      DO 20 J=1,12
      DO 20 I=1,12
20  S(I,J)=0.0
C
C      FORM STIFFNESS MATRIX AND WRITE ON TAPE 2 (CHANGED TO 13)
C
      IF (N .LE. NELCAL .OR. NUMCEL .EQ. 0) GO TO 40
      DO 30 M=1,NUMCEL
      IF (N .EQ. NCEL(M)) GO TO 40
30  CONTINUE
      GO TO 50
40  KEY=1
50  CONTINUE
      IF (KEY .EQ. 0) GO TO 80
      IF (N .GE. NELRED .AND. N .LE. NELCAL) GO TO 60
55  LL=1
      FMAG=1.0D0
      GO TO 80
60  IF (NUMCEL .EQ. 0) GO TO 75
      DO 70 M=1,NUMCEL
      IF (N .EQ. NCEL(M)) GO TO 55
70  CONTINUE
75  LL=0
      FMAG=0.00001D0
80  CONTINUE
      READ (16) LM
      C1=FMAG*(BULK(N)+SHEAR(N))
      C2=FMAG*(BULK(N)-SHEAR(N))
      C3=FMAG*SHEAR(N)
      DO 100 LX=1,2
      DO 100 LY=1,2
      READ (16) VJAC,STR
      IF (KEY .EQ. 0) GO TO 100
      DO 90 J=1,12
      D1=VJAC*(C1*STR(1,J)+C2*STR(2,J))
      D2=VJAC*(C2*STR(1,J)+C1*STR(2,J))
      D3=VJAC*C3*STR(3,J)
      DO 90 I=J,12
      S(I,J)=S(I,J)+STR(1,I)*D1+STR(2,I)*D2+STR(3,I)*D3
90  S(J,I)=S(I,J)
100 CONTINUE
      IF (KEY .EQ. 0) GO TO 160
      IF (INP(N,3) .EQ. INP(N,4)) GO TO 160
C
C      ELIMINATE EXTRA DEGREES OF FREEDOM
C
      DO 150 NN=1,4
      L=12-NN
      K=L+1
      DO 150 I=1,L
      C=S(I,K)/S(K,K)

```

```

      IF (C .EQ. 0.0) GO TO 151
      TEMP = DLOG10(DABS(C))
      GO TO 152
151  TEMP = -3.0D+77
152  DO 150 J=1,L
      IF (S(K,J) .EQ. 0.0) GO TO 153
      TEMP1 = TEMP + DLOG10(DABS(S(K,J)))
      GO TO 154
153  TEMP1 = TEMP-3.0D+77
154  IF (TEMP1 .GT. -78.0D0) S(I,J)=S(I,J)-C*S(K,J)
150  CONTINUE
160  CONTINUE
      WRITE(13) LM, ((S(I,J), I=1,8), J=1,8)
C
C   FORM STRAIN-DISPLACEMENT MATRIX AND WRITE ON TAPE 11
C
      READ (16) ((STR(I,J), I=1,3), J=1,8)
      IF (KEY .EQ. 0) GO TO 300
      WRITE(11) LM, ((STR(I,J), I=1,3), J=1,8), LL
300  CONTINUE
      RETURN
      END
      SUBROUTINE ISRSLT (EMPR, ID, INP, BULK, SHEAR, POIS, SLMAX, NCEL, NCNP,
1  NLDP, DISP, STRESS, STRAIN, SNEW, DELD, B, R, PATM, NUMELT, NUMNPT, NUMCEL,
2  NUMCNP, NUMMAT, NUMLD, NLAY, LN, IT, NPUNCH, NUMEL, NUMNP, NELCAL, NNPCAL,
3  NNPRED, NEQ, NEQB, NBLOCK, X, Y, IO)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /ISOP/ EL, E2, RR(4), ZZ(4), LM(8), P(8), S(12,12), STR(3,12),
1  STS(3,8), VJAC, MPRINT
      DIMENSION X(NUMNPT), Y(NUMNPT)
      DIMENSION EMPR(NUMMAT,13), ID(NUMNPT,2), BULK(1), SHEAR(1), POIS(1)
      DIMENSION INP(NUMELT,5), SLMAX(1), NCEL(1), NCNP(1), NLDP(1)
      DIMENSION DISP(NUMNPT,2), STRESS(NUMELT,3), STRAIN(NUMELT,3)
      DIMENSION SNEW(NUMELT,3), DELD(NUMNPT,2), B(1), R(1)
      DIMENSION SIG(3), EPS(3), PRS(5)
      REWIND 13
      REWIND 8
      REWIND 11
C
C   MOVE DISPLACEMENTS INTO CORE
C
      NQ=NEQB*NBLOCK
      DO 10 NN=1,NBLOCK
      READ (13) (R(I), I=1,NEQB)
      N=NEQB
      IF (NN .EQ. 1) N=NEQ-NQ+NEQB
      NQ=NQ-NEQB
      DO 10 J=1,N
      I=NQ+J
10  B(I)=R(J)
      IF (LN .GT. NLAY) GO TO 15
      WRITE (10,2000) LN, IT
      GO TO 16
15  LNMLAY=LN-NLAY
      WRITE (10,2005) LNMLAY, IT
16  CONTINUE
      IF (IT .LT. 2) GO TO 110
C
C   ADD INCREMENTAL DISPLACEMENTS AND PRINT INCREMENTAL AND TOTAL
C   DISPLACEMENTS
C
      WRITE (10,2010)
      REWIND 9

```

```

      READ (9) DISP
      READ (9) STRAIN
      DO 20 I=1,2
      DO 20 N=1,NUMNP
20  DELD(N,I)=0.0
      DO 70 N=1,NUMNP
      IF (N .LE. NNPCAL .OR. NUMCNP .EQ. 0) GO TO 40
      DO 30 M=1,NUMCNP
      IF (N .EQ. NCNP(M)) GO TO 40
30  CONTINUE
      GO TO 70
40  CONTINUE
      IF (N .LT. NNPRED .OR. N .GT. NNPCAL) GO TO 45
      IF (NUMCNP .EQ. 0) GO TO 70
      DO 42 M=1,NUMCNP
      IF (N .EQ. NCNP(M)) GO TO 45
42  CONTINUE
      GO TO 70
45  CONTINUE
      DO 50 I=1,2
      II=ID(N,I)
      IF (II .LT. 1) GO TO 50
      DELD(N,I)=B(II)
50  CONTINUE
      DO 60 J=1,2
60  DISP(N,J)=DISP(N,J)+DELD(N,J)
70  CONTINUE
      DO 100 N=1,NUMNP
      IF (N .LE. NNPCAL .OR. NUMCNP .EQ. 0) GO TO 90
      DO 80 M=1,NUMCNP
      IF (N .EQ. NCNP(M)) GO TO 90
80  CONTINUE
      GO TO 100
90  CONTINUE
      TEMP1=DISP(N,1)
      TEMP2=DISP(N,2)
      TD=DSQRT(TEMP1*TEMP1+TEMP2*TEMP2)
      IF (MPRINT.EQ.1.AND.LN.LT.NUMLD) GO TO 777
      WRITE (10,2050) N, (DELD(N,I), I=1,2), (DISP(N,M), M=1,2), TD, N
C*****CNTUR*****
      WRITE(7) N,X(N),Y(N), (DISP(N,M), M=1,2), TD
C2051 FORMAT(15,2F10.2,3F15.4)
C*****END CNTUR*****
777  CONTINUE
100  CONTINUE
110  CONTINUE
C
C   CALCULATE INCREMENTAL STRESSES AND STRAINS, ADD INCREMENTAL
C   STRESSES AND STRAINS AND PRINT STRAINS AND MODULUS VALUES
C
      READ (8) STRESS
      DO 120 I=1,3
      DO 120 N=1,NUMEL
120  SNEW(N,I)=STRESS(N,I)
      DO 300 N=1,NUMEL
      IF (N .LE. NELCAL .OR. NUMCEL .EQ. 0) GO TO 160
      DO 140 M=1,NUMCEL
      IF (N .EQ. NCEL(M)) GO TO 160
140  CONTINUE
      GO TO 300
160  CONTINUE
      READ (11) LM, ((STR(I,J), I=1,3), J=1,8), LL
      IF (LL .EQ. 0) GO TO 222

```

```

C
C FORM STRESS-DISPLACEMENT MATRIX
C
C1=BULK(N)+SHEAR(N)
C2=BULK(N)-SHEAR(N)
C3=SHEAR(N)
DO 200 K=1,8
STS(1,K)=C1*STR(1,K)+C2*STR(2,K)
STS(2,K)=C2*STR(1,K)+C1*STR(2,K)
STS(3,K)=C3*STR(3,K)
KK=LM(K)
IF (KK.EQ. 0) GO TO 180
P(K)=B(KK)
GO TO 200
180 P(K)=0.0
200 CONTINUE
DO 220 I=1,3
SIG(I)=0.0
DO 220 K=1,8
220 SIG(I)=SIG(I)+STS(I,K)*P(K)
GO TO 228
222 CONTINUE
IF (IT.EQ. 1) GO TO 225
DO 224 I=1,3
224 SIG(I)=0.0
GO TO 228
225 CONTINUE
DO 226 I=1,3
226 SIG(I)=STRESS(N,I)
228 CONTINUE
IF (IT.EQ. 2) GO TO 240
DO 230 I=1,3
230 SNEW(N,I)=STRESS(N,I)-0.5D0*SIG(I)
GO TO 300
240 CONTINUE
DO 250 I=1,3
SNEW(N,I)=STRESS(N,I)-SIG(I)
250 STRESS(N,I)=SNEW(N,I)
DO 270 I=1,3
EPS(I)=0.0
IF (LL.EQ. 0) GO TO 270
DO 260 K=1,8
260 EPS(I)=EPS(I)+STR(I,K)*P(K)
270 STRAIN(N,I)=STRAIN(N,I)-EPS(I)*100.0D0
300 CONTINUE
IF (IT.LT. 2) GO TO 400
WRITE (IO,2100)
DO 360 N=1,NUMEL
IF (N.LE. NELCAL .OR. NUMCEL.EQ. 0) GO TO 340
DO 320 M=1,NUMCEL
IF (N.EQ. NCEL(M)) GO TO 340
320 CONTINUE
GO TO 360
340 CONTINUE

C
C CALCULATE PRINCIPAL STRAINS
C
EP=(STRAIN(N,2)+STRAIN(N,1))/2.0D0
EM=(STRAIN(N,2)-STRAIN(N,1))/2.0D0
TEMP=STRAIN(N,3)/2.0D0
ER=DSQRT(TEMP*TEMP+EM*EM)
PRS(1)=EP+ER
PRS(2)=EP-ER

PRS(3)=2.0D0*ER
EMOD=2.0D0*BULK(N)*(1.0D0+POIS(N))*(1.0D0-POIS(N)-POIS(N))
IF (MPRINT.EQ.1.AND.LN.LT.NUMLD) GO TO 778
WRITE (IO,2150) N,EMOD,BULK(N),SHEAR(N),POIS(N),STRAIN(N,M),
1 M=1,3),(PRS(I),I=1,3),N
778 CONTINUE
360 CONTINUE
WRITE (IO,2200)
400 CONTINUE

C
C CALCULATE PRINCIPAL STRESSES AND NEW MODULUS VALUES
C
DO 500 N=1,NUMEL
IF (N.LE. NELCAL .OR. NUMCEL.EQ. 0) GO TO 440
DO 420 M=1,NUMCEL
IF (N.EQ. NCEL(M)) GO TO 440
420 CONTINUE
GO TO 500
440 CONTINUE
CP=(SNEW(N,2)+SNEW(N,1))/2.0D0
CM=(SNEW(N,2)-SNEW(N,1))/2.0D0
TEMP=SNEW(N,3)
CR=DSQRT(TEMP*TEMP+CM*CM)
PRS(1)=CP+CR
PRS(2)=CP-CR
MTYPE=INP(N,5)
KL=2
IF (IT.EQ. 1) KL=3
CALL ELAW (EMER,BULK,SHEAR,POIS,SLMAX,PRS,PATM,NUMMAT,N,
1 MTYPE,STRLEV,KL)
IF (IT.LT. 2) GO TO 500
IF (STRLEV.GT. SLMAX(N)) SLMAX(N)=STRLEV
PRS(3)=CR
PRS(4)=0.0
IF (SNEW(N,3).EQ. 0.0.AND. CM.EQ. 0.0) GO TO 450
PRS(4)=57.29577951D0*DATA2(SNEW(N,3),CM)/2.0D0
450 CONTINUE
IF (PRS(2).NE. 0.0) GO TO 460
PRS(5)=999.999D0
GO TO 470
460 PRS(5)=PRS(1)/PRS(2)
470 CONTINUE

C
C PRINT STRESSES
C
IF (MPRINT.EQ.1.AND.LN.LT.NUMLD) GO TO 779
WRITE (IO,2250) N,(STRESS(N,M),M=1,3),(PRS(I),I=1,5),
1 SLMAX(N),STRLEV,N
779 CONTINUE
500 CONTINUE
IF (IT.LT. 2) GO TO 530
REWIND 9
WRITE(9) DISP
WRITE(9) STRAIN
REWIND 8
WRITE(8) STRESS
IF (NPUNCH.EQ. 0) GO TO 530
DO 510 I=1,NPUNCH
IF (LN.EQ. NLDP(I)) GO TO 520
510 CONTINUE
GO TO 530
520 CONTINUE
WRITE(4) (N,(STRESS(N,M),M=1,3),N=1,NUMEL)

```

```

      WRITE(4) (N, (STRAIN(N,M), M=1, 3), N=1, NUMEL)
      WRITE(4) (N, (DISP(N,M), M=1, 2), N=1, NUMNP)
530  CONTINUE
      RETURN
2000 FORMAT (15H LAYER NUMBER =, I3, 15H ITERATION =, I3, /,
1 59H ...../)
2005 FORMAT (12H LOAD CASE =, I3, 15H ITERATION =, I3, /,
1 59H ...../)
2010 FORMAT ('0 NP DELTA-X DELTA-Y X-DISP',
1 ' Y-DISP TOTAL NE', /)
2050 FORMAT (15, 5F14.4, I4)
2100 FORMAT ('1', ' MODULUS AND POISSON'S RATIO BASED ON AVERAGE',
1 ' STRESSES DURING THE INCREMENT', /,
2 ' ', ' STRAINS FOR FINAL CONDITION AT END OF INCREMENT', /,
3 ' ', ' ELE ELAS MOD BULK MOD SHEAR MOD POIS RATIO', /,
4 ' EPS-X EPS-Y GAM-XY EPS-1 EPS-3', /,
5 ' GAMMAX ELE', /)
2150 FORMAT (15, 10F11.3, I5)
2200 FORMAT ('1', ' STRESSES AND STRESS LEVELS FOR FINAL CONDITION AT',
1 ' END OF INCREMENT', /,
2 ' ', ' ELE SIG-X SIG-Y TAU-XY SIG-1', /,
3 ' ', ' SIG-3 TAU-MAX THETA SIG1/SIG3 SIGMAX', /,
4 ' SLPRES ELE', /)
2250 FORMAT (15, 10F11.3, I5)
CXXXXXXXXXXXXX
C2500 FORMAT (15, 3F10.4)
C2550 FORMAT (15, 2F10.4)
      END
      SUBROUTINE LAYOUT (MAXB, ID, INP, NCEL, NCNP, NOMEL, NOMNP, NPHUMP, NLDP,
1 EMPR, X, Y, BULK, SHEAR, POIS, SIGMAX, DISP, STRESS, STRAIN, XCP, YCP, NUMELT,
2 NUMNPT, NUMCEL, NUMCNP, NFEL, NUMMAT, NUMLD, NLAY, NEQ, NEQB, MBAND, PATM,
3 MTMNI7, NMREQB, NPUNCH, IN, IO, INEQ)
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION EMPR(NUMMAT, 13), ID(NUMNPT, 2), X(1), Y(1), INP(NUMELT, 5)
      DIMENSION BULK(1), SHEAR(1), POIS(1), SIGMAX(1), XCP(1), YCP(1)
      DIMENSION DISP(NUMNPT, 2), STRESS(NUMELT, 3), STRAIN(NUMELT, 3)
      DIMENSION NCEL(1), NCNP(1), NOMEL(NUMLD, 2), NOMNP(NUMLD, 2)
      DIMENSION NPHUMP(NUMLD, 4), NLDP(1)
      DIMENSION MATNO(10), NLEL(10), NREL(10), HL(10), SINITX(10), SINITY(10)
      DIMENSION LM(8), PRS(5), HH(10), SIGAVE(3), MAXB(INEQ)
C
C THIS PORTION IS EXECUTED JUST TO GET RID OF WARNING FROM THE COMPILER
C
      DO 997 I=1, INEQ
997 MAXB(I)=0
C
      REWIND 15
      IF (NPUNCH .EQ. 0) GO TO 20
C
C READ AND PRINT DATA FOR LOAD CASE TO BE PUNCHED OUT
C
      READ (IN, *) (NLDP(I), I=1, NPUNCH)
      WRITE (IO, 1050) (NLDP(I), I=1, NPUNCH)
20 CONTINUE
C
C READ AND PRINT MATERIAL PROPERTY DATA
C
      READ (IN, *) PATM
      WRITE (IO, 2000) PATM
      WRITE (IO, 2010)
50 READ (IN, *) M, (EMPR(M, I), I=1, 11)
      WRITE (IO, 2020) M, (EMPR(M, I), I=1, 11)
      PHI=EMPR(M, 9)/57.29577951D0

```

```

      CONST=2.0D0/(EMPR(M, 10)*(1.0D0-DSIN(PHI)))
      EMPR(M, 12)=CONST*EMPR(M, 8)*DCOS(PHI)
      EMPR(M, 13)=CONST*DSIN(PHI)
      IF (M .LT. NUMMAT) GO TO 50
      LL=0
C
C AND PRINT NODAL POINT DATA AND BOUNDARY CONDITIONS
C
100 READ (IN, *) MM, X(MM), Y(MM), (ID(MM, I), I=1, 2)
      IF (LL .LE. 0) GO TO 110
      DIFNP=MM-LL
      DX=(X(MM)-X(LL))/DIFNP
      DY=(Y(MM)-Y(LL))/DIFNP
110 LL=LL+1
      IF (MM-LL) 150, 140, 120
120 X(LL)=X(LL-1)+DX
      Y(LL)=Y(LL-1)+DY
      DO 130 I=1, 2
130 ID(LL, I)=ID(LL-1, I)
      GO TO 110
140 IF (NUMNPT-MM) 150, 160, 100
150 WRITE (IO, 5000) MM
      GO TO 999
160 WRITE (IO, 2030)
      N=0
170 N=N+1
      WRITE (IO, 2040) N, X(N), Y(N), (ID(N, I), I=1, 2)
      IF (N .LT. NUMNPT) GO TO 170
      NN=0
C
C READ AND PRINT ELEMENT DATA
C
200 READ (IN, *) N, (INP(N, I), I=1, 5)
210 NN=NN+1
      IF (N .LE. NN) GO TO 230
      DO 220 K=1, 4
220 INP(NN, K)=INP(NN-1, K)+1
      INP(NN, 5)=INP(NN-1, 5)
230 IF (N .GT. NN) GO TO 210
      IF (NUMELT .GT. NN) GO TO 200
      WRITE (IO, 2050)
      N=0
250 N=N+1
      XCP(N)=0.0
      YCP(N)=0.0
      DO 240 I=1, 4
      II=INP(N, I)
      XCP(N)=XCP(N)+X(II)/4.0D0
240 YCP(N)=YCP(N)+Y(II)/4.0D0
      WRITE (IO, 2060) N, (INP(N, M), M=1, 5), XCP(N), YCP(N)
      IF (N .LT. NUMELT) GO TO 250
      IF (NPUNCH .EQ. 0) GO TO 260
      WRITE(4) (N, XCP(N), YCP(N), N=1, NUMELT)
260 CONTINUE
C
C SET UP EQUATION NUMBERS
C
      NEQ=0
      DO 330 N=1, NUMNPT
      DO 330 I=1, 2
      IF (ID(N, I) .LE. 1000) GO TO 310
      NN=ID(N, I)-1000
      ID(N, I)=ID(NN, I)

```



```

      GO TO 330
310 IF (ID(N,I) .EQ. 1) GO TO 320
      NEQ=NEQ+1
      ID(N,I)=NEQ
      GO TO 330
320 ID(N,I)=0
330 CONTINUE
      WRITE (IO,2070)
      WRITE (IO,2080) (N, (ID(N,I), I=1,2), N=1, NUMNPT)
C
C   DETERMINE BAND WIDTH
C
      MBAND=0
      DO 430 N=1, NUMELT
      MIN=100000
      MAX=0
      IJ=0
      DO 410 I=1,4
      II=INP(N,I)
      DO 410 J=1,2
      IJ=IJ+1
410 LM(IJ)=ID(II,J)
      WRITE(15) LM
      DO 420 L=1,8
      IF (LM(L) .EQ. 0) GO TO 420
      IF (LM(L) .GT. MAX) MAX=LM(L)
      IF (LM(L) .LT. MIN) MIN=LM(L)
420 CONTINUE
      NDIF=MAX-MIN+1
      IF (NDIF .GT. MBAND) MBAND=NDIF
430 CONTINUE
      NMKEQB=MTMN17/(MBAND+2)/2
      NEQB=NMKEQB
      IF (NEQB .GT. NEQ) NEQB=NEQ
      WRITE (IO,2085) MBAND,NEQ,NEQB
      IF (NLAY .EQ. 0) GO TO 440
C
C   READ AND PRINT CONSTRUCTION SEQUENCE INFORMATION
C
      WRITE (IO,2090)
      READ (IN,*) (LN, (NOMEL(LN,I), I=1,2), (NOMNP(LN,J), J=1,2),
      1 (NPHUMP(LN,K), K=1,4), N=1, NLAY)
      WRITE (IO,2100) (LN, (NOMEL(LN,I), I=1,2), (NOMNP(LN,J), J=1,2),
      1 (NPHUMP(LN,K), K=1,4), LN=1, NLAY)
440 CONTINUE
      IF (NUMCEL .EQ. 0) GO TO 450
C
C   READ AND PRINT DATA FOR PREEXISTING ELEMENTS AND NODAL POINTS
C
      WRITE (IO,2110)
      READ (IN,*) (NCEL(N), N=1, NUMCEL)
      WRITE (IO,1050) (NCEL(N), N=1, NUMCEL)
      WRITE (IO,2120)
      READ (IN,*) (NCNP(N), N=1, NUMCNP)
      WRITE (IO,1050) (NCNP(N), N=1, NUMCNP)
450 CONTINUE
C
C   INITIALIZATION OF STRESSES, STRAINS, AND STRESS LEVELS
C   IN ALL ELEMENTS AND DISPLACEMENTS OF ALL NODAL POINTS
C
      DO 500 I=1, NUMELT
      SLMAX(I)=0.0
      DO 500 J=1,3

```

```

      STRESS(I,J)=0.0
500 STRAIN(I,J)=0.0
      DO 510 J=1,2
      DO 510 I=1, NUMNPT
510 DISP(I,J)=0.0
      IF (NUMCEL .EQ. 0) GO TO 550
C
C   READ STRESSES, STRAINS AND DISPLACEMENTS AND CALCULATE MODULUS
C   VALUES FOR PREEXISTING PART
C
      READ (IN,*) NMODL
      READ (IN,*) (N, (STRESS(N,M), M=1,3), J=1, NUMCEL)
      IF (NMODL .NE. 0) GO TO 520
      READ (IN,*) (N, (STRAIN(N,M), M=1,3), J=1, NUMCEL)
      READ (IN,*) (N, (DISP(N,M), M=1,2), J=1, NUMCNP)
520 CONTINUE
      DO 530 I=1, NUMCEL
      N=NCEL(I)
      CP=(STRESS(N,2)+STRESS(N,1))/2.0D0
      CM=(STRESS(N,2)-STRESS(N,1))/2.0D0
      TEMP=STRESS(N,3)
      CR=DSQRT(TEMP*TEMP+CM*CM)
      PRS(1)=CP+CR
      PRS(2)=CP-CR
      MTYPE=INP(N,5)
      CALL ELAW (EMPR, BULK, SHEAR, POIS, SLMAX, PRS, PATM, NUMMAT, N,
      1 MTYPE, STRLEV, 1)
      SLMAX(N)=STRLEV
530 CONTINUE
550 CONTINUE
      IF (NFEL .EQ. 0) GO TO 700
C
C   READ AND PRINT DATA FOR FOUNDATION LAYERS USED IN CALCULATING
C   INITIAL STRESSES
C
      WRITE (IO,2200)
      READ (IN,*) NFLAY, HFLEV
      WRITE (IO,2210) NFLAY, HFLEV
      WRITE (IO,2220)
      I=0
600 I=I+1
      READ (IN,*) I, MATNO(I), NLEL(I), NREL(I), HL(I)
      WRITE (IO,2230) I, MATNO(I), NLEL(I), NREL(I), HL(I)
      IF (I .GT. 1) GO TO 610
      HH(I)=HL(I)-HFLEV
      GO TO 620
610 HH(I)=HL(I)-HL(I-1)
620 CONTINUE
      IF (I .LT. NFLAY) GO TO 600
C
C   CALCULATE INITIAL STRESSES AND MODULI FOR FOUNDATION ELEMENTS
C
      DO 670 I=1, NFLAY
      SINITY(I)=0.0
      IF (I .EQ. NFLAY) GO TO 640
      NN=I+1
      DO 630 J=NN, NFLAY
      MTYPE=MATNO(J)
630 SINITY(I)=SINITY(I)+EMPR(MTYPE,1)*HH(J)
640 MTYPE=MATNO(I)
      SINITY(I)=SINITY(I)+EMPR(MTYPE,1)*HH(I)/2.0D0
      SINITY(I)=EMPR(MTYPE,11)*SINITY(I)
     >NNL=NLEL(I)

```

```

NNR=NREL(I)
DO 650 N=NNL,NNR
STRESS(N,1)=SINITX(I)
650 STRESS(N,2)=SINITY(I)
NNL1=NNL+1
N=NNL
MTYPE=INP(NNL,5)
PRS(1)=STRESS(NNL,2)
PRS(2)=STRESS(NNL,1)
CALL ELAW (EMPR,BULK,SHEAR,POIS,SLMAX,PRS,PATM,NUMMAT,N,
1 MTYPE,STRLEV,1)
SLMAX(NNL)=STRLEV
DO 660 N=NNL1,NNR
BULK(N)=BULK(NNL)
SHEAR(N)=SHEAR(NNL)
POIS(N)=POIS(NNL)
660 SLMAX(N)=SLMAX(NNL)
670 CONTINUE
700 CONTINUE
C
C CALCULATE INITIAL STRESSES AND MODULI FOR LAYERS TO BE ADDED
C
WRITE (IO,2310)
IF (NLAY .EQ. 0) GO TO 790
DO 780 LN=1,NLAY
NNL=NOMEL(LN,1)
NNR=NOMEL(LN,2)
IJ1=NPHUMP(LN,1)
IJ2=NPHUMP(LN,2)
IJ3=NPHUMP(LN,3)
IJ4=NPHUMP(LN,4)
IF ((X(IJ2)-X(IJ1)) .NE. 0.0) TEMP1=(Y(IJ2)-Y(IJ1))/(X(IJ2)-X(IJ1))
IF ((X(IJ3)-X(IJ2)) .NE. 0.0) TEMP2=(Y(IJ3)-Y(IJ2))/(X(IJ3)-X(IJ2))
IF ((X(IJ4)-X(IJ3)) .NE. 0.0) TEMP3=(Y(IJ4)-Y(IJ3))/(X(IJ4)-X(IJ3))
DO 770 M=NNL,NNR
IF (NUMCEL .EQ. 0) GO TO 720
DO 710 M=1,NUMCEL
IF (N .EQ. NCEL(M)) GO TO 770
710 CONTINUE
720 MTYPE=INP(N,5)
IF (XCP(N) .LE. X(IJ1)) GO TO 731
IF (XCP(N) .LE. X(IJ2)) GO TO 732
IF (XCP(N) .LE. X(IJ3)) GO TO 733
IF (XCP(N) .LE. X(IJ4)) GO TO 734
SLOPE=0.0
HT=Y(IJ4)-YCP(N)
C WRITE(6,44) Y(IJ4), YCP(N)
C 44 FORMAT(3X,'Y(IJ4)',2X,F10.2,3X,'YCP(N)',2X,F10.2)
GO TO 740
731 SLOPE=0.0
HT=Y(IJ1)-YCP(N)
GO TO 740
732 SLOPE=TEMP1
HT=Y(IJ1)+(XCP(N)-X(IJ1))*SLOPE-YCP(N)
GO TO 740
733 SLOPE=TEMP2
HT=Y(IJ2)+(XCP(N)-X(IJ2))*SLOPE-YCP(N)
GO TO 740
734 SLOPE=TEMP3
HT=Y(IJ3)+(XCP(N)-X(IJ3))*SLOPE-YCP(N)
740 BETA=DTAN(SLOPE)
STRESS(N,2)=HT*EMPR(MTYPE,1)
C WRITE(6,33) HT,EMPR(MTYPE,1)

```

```

C 33 FORMAT(2F10.3)
STRESS(N,3)=0.50D0*STRESS(N,2)*DSIN(BETA)
SIGAVE(2)=STRESS(N,2)/2.0D0
SIGAVE(3)=STRESS(N,3)/2.0D0
POIS1=EMPR(MTYPE,6)
750 IF (POIS1 .GT. 0.49D0) POIS1=0.49D0
STRESS(N,1)=STRESS(N,2)*POIS1/(1.0D0-POIS1)
SIGAVE(1)=STRESS(N,1)/2.0D0
CP=(SIGAVE(2)+SIGAVE(1))/2.0D0
CM=(SIGAVE(2)-SIGAVE(1))/2.0D0
CR=DSQRT(SIGAVE(3)*SIGAVE(3)+CM*CM)
PRS(1)=CP+CR
PRS(2)=CP-CR
CALL ELAW (EMPR,BULK,SHEAR,POIS,SLMAX,PRS,PATM,NUMMAT,N,
1 MTYPE,STRLEV,1)
POIST=POIS(N)
IF (DABS(POIS1-POIST) .LT. 0.0001D0) GO TO 760
POIS1=POIS1+(POIST-POIS1)/10.0D0
GO TO 750
760 CONTINUE
SLMAX(N)=STRLEV
770 CONTINUE
780 CONTINUE
790 CONTINUE
C
C PRINT INITIAL MODULI AND STRESSES FOR ALL ELEMENTS
C
DO 800 N=1,NUMELT
EMOD=2.0D0*SHEAR(N)*(1.0D0+POIS(N))
WRITE (IO,2320) N,XCP(N),YCP(N),EMOD,BULK(N),SHEAR(N),POIS(N),
1 (STRESS(N,M),M=1,3)
800 CONTINUE
REWIND 8
REWIND 9
WRITE(8) STRESS
WRITE(9) DISP
WRITE(9) STRAIN
999 CONTINUE
RETURN
C1000 FORMAT (F10.0)
C1010 FORMAT (I5,7F10.0/5F10.0)
C1020 FORMAT (I5,2F10.0,2I5)
C1030 FORMAT (6I5)
C1040 FORMAT (9I5)
1050 FORMAT (15I5)
C1100 FORMAT (I5,3F10.0)
C1110 FORMAT (I5,2F10.0)
C1200 FORMAT (I5,F10.0)
C1210 FORMAT (4I5,F10.0)
2000 FORMAT (/,23H MATERIAL PROPERTY DATA,/,
1 22H ATMOSPHERIC PRESSURE=F10.4,/)
2010 FORMAT (' ',24X,'MODULUS',34X,'POISSON RATIO',/,
1 ' ',MAT UNIT WT K KUR N D',
2 ' G F C PHI FAIL RATIO',
3 ' KO',/)
2020 FORMAT (I4,11F11.3)
2030 FORMAT (23H1NODAL POINT INPUT DATA,/,5H NODE,3X,
1 23H1NODAL POINT COORDINATES,7X,9HB.C. CODE/,7H NUMBER,
2 40H X-ORD Y-ORD XX YY,/)
2040 FORMAT (I7,2F10.3,10X,2I5)
2050 FORMAT (30H1FOUR NODES SOLID ELEMENT DATA,/,
1 5H ELET,5X,15HCONNECTED NODES,5H MATL,
2 28H ELEMENT CENTER COORDINATES,/,

```

```

      3 30H NO.      I      J      K      L NO.,24H      X-ORD      Y-ORD,/)
2060 FORMAT (6I5,2F12.3)
C2065 FORMAT (I5,2F10.3)
2070 FORMAT (17H1EQUATION NUMBERS//,15H      N      X      Y,/)
2080 FORMAT (3I5)
2085 FORMAT (/
      1 35HOBAND WIDTH*****I4,/,
      2 35HONUMBER OF EQUATIONS*****I4,/,
      3 35HONUMBER OF EQUATIONS IN BLOCK*****I4)
2090 FORMAT (31H1CONSTRUCTION LAYER INFORMATION ,//, 6H LAYER,
      1 31H ADDED ELEMENTS ADDED NODES,
      2 34H      NODES OF HUMPED SURFACE ,/)
2100 FORMAT (I5,I9,I6,I10,I6,I0X,4I6)
2110 FORMAT (29H0ELEMENTS OF PREEXISTING PART ,//)
2120 FORMAT (26H0NODES OF PREEXISTING PART ,//)
5000 FORMAT (17H N.P. ERROR , N = ,I4)
2310 FORMAT ('1 INITIAL VALUES IN ELEMENTS',//,
      1 ' ELE      X-ORD      Y-ORD      ELAS MOD      BULK MOD',
      2 ' SHEAR MOD POIS RATIO      SIG-X      SIG-Y      TAU-XY',/)
2320 FORMAT (I5,9F11.3)
2200 FORMAT (28H1FOUNDATION PART INFORMATION)
2210 FORMAT (35HONUMBER OF LAYER IN FOUNDATION*****I4,/,
      1 35H0ELEVATION OF RIGID BOUNDARY*****F8.3,/)
2220 FORMAT (47H0LAYER MAT. NO. INCLUSIVE ELEMENTS ELEVATION,/)
2230 FORMAT (I5,3F10,F12.3)
      END
SUBROUTINE SYMBAN (A,B,MAXB,NEQB,MB,NBLOCK,NSB,NORG,NBKS,NT1,
      1 NT2,NRST,IO)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(NSB),B(NSB),MAXB(NEQB)
C
      NC=MB+1
      NBR=(MB-1)/NEQB+1
      INC=NEQB-1
      NMB=NEQB*MB
      N2=NT2
      N1=NT1
      REWIND NORG
      REWIND NBKS
C
C      REDUCE EQUATIONS BLOCK-BY-BLOCK
C
      DO 900 N=1,NBLOCK
      IF (N.GT.1.AND. NBR.EQ. 1) GO TO 110
      IF (NBR.EQ. 1) GO TO 105
      REWIND N1
      REWIND N2
105  NI=N1
      IF(N.EQ. 1) NI=NORG
      READ (NI) A
110  DO 300 I=1,NEQB
      MAXB(I)=0
      D=A(I)
      IF(D) 115,300,120
115  M=NEQB*(N-1)+I
      WRITE (IO,116) M,D
116  FORMAT (33H0SET OF EQUATIONS MAY BE SINGULAR ,//,
      26H DIAGONAL TERM OF EQUATION, I8, 8H EQUALS, 1PE12.4)
C
120  II=I
      DO 125 J=2,NC
      II=II+NEQB
125  A(II)=A(II)/D

```

```

C
      DO 130 J=I,NMB,NEQB
      IF (A(J).NE.0.0) MAXB(I)=J
130  CONTINUE
C
      JL=I+1
      IF (JL.GT.NEQB) GO TO 300
      II=I
      DO 200 J=JL,NEQB
      II=II+NEQB
      IF(II.GT.NMB) GO TO 200
      C=A(II)
      IF (C.EQ.0.0) GO TO 200
      C=C*A(I)
C
      KK=J-II
      MAX=MAXB(I)
      DO 150 JJ=II,MAX,NEQB
150  A(JJ+KK)=A(JJ+KK)-C*A(JJ)
C
      KK=J+NMB
      JJ=I+NMB
      A(KK)=A(KK)-C*A(JJ)
200  CONTINUE
300  CONTINUE
      WRITE(NBKS) MAXB,A
C
C      SUBSTITUTE INTO REMAINING EQUATIONS
C
      DO 800 NN=1,NBR
      IF(N+NN.GT.NBLOCK) GO TO 800
      NI=N1
      IF(N.EQ.1) NI=NORG
      IF(NN.EQ.NBR) NI=NORG
      READ (NI) B
      IL=1+NN*NEQB*NEQB
      DO 700 I=1,NEQB
      II=IL
      DO 690 K=1,NEQB
      IF (II.GT.NMB) GO TO 690
      C=A(II)
      IF (C.EQ.0.0) GO TO 690
      C=C*A(K)
      MAX=MAXB(K)
C
      KK=I-II
      DO 640 JJ=II,MAX,NEQB
640  B(JJ+KK)=B(JJ+KK)-C*A(JJ)
C
      KK=I+NMB
      JJ=K+NMB
      B(KK)=B(KK)-C*A(JJ)
C
      690  II=II+INC
      700  IL=IL+NEQB
C
      IF(NBR.NE.1) GO TO 750
      DO 740 I=1,NSB
740  A(I)=B(I)
      GO TO 800
750  WRITE(N2) B
800  CONTINUE
C

```

```

      M=N1
      N1=N2
900  N2=M
C
C   BACKSUBSTITUTION - RESULTS ON TAPE NRST
C
      NEB=NEQB*(NBR+1)
      NUM=NBR*NEQB
      DO 905 I=1,NEB
905  B(I)=0.0
      REWIND NRST
      REWIND NBKS
C
      DO 1000 N=1,NBLOCK
      READ (NBKS) MAXB,A
      K=NEB
      DO 910 J=1,NUM
      I=K-NEQB
      B(K)=B(I)
910  K=K-1
C
      I=NMB
      K=0
      DO 920 J=1,NEQB
      I=I+1
      K=K+1
920  B(K)=A(I)
C
      DO 950 I=1,NEQB
      J=NEQB+1-I
      MAX=MAXB(J)
      IF (A(J).EQ. 0.0) GO TO 950
      KK=J
      JJ=KK+1
      IL=J+NEQB
      C=B(KK)
      DO 940 II=IL,MAX,NEQB
      C=C-A(II)*B(JJ)
940  JJ=JJ+1
      B(KK)=C
950  CONTINUE
C
      I=0
      K=0
      DO 960 J=1,NEQB
      K=K+1
      I=I+1
960  A(I)=B(K)
C
      WRITE(NRST) (A(I),I=1,NEQB)
1000 CONTINUE
C
      RETURN
      END
      SUBROUTINE SECOND (TIM)
      USE PORTLIB
C
C   SUBROUTINE TO OBTAIN TIME
C   THIS SUBROUTINE IS COMPILER DEPENDENT, MICROSOFT FORTRAN POWER
C   STATION PORTLIB
C
      TIM=TIMEF ()
C

```

```

      RETURN
      END

```

APPENDIX E

CANDE-89 SOURCE CODE

```

* * * * *
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
* VERSION 1.0, MARCH 1990
* PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
* THE FEDERAL HIGHWAY ADMINISTRATION
* * * * *
C PROGRAM CANDE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE30,
C 1 TAPE10,TAPE11,TAPE12,TAPE13,TAPE14,TAPE16,TAPE17)
C
C CANDE... (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM..
C
C ENGLISH UNIT VERSION
C
* Uncomment this statement in all program modules for double
* precision version. SCM 3-10-90
*
* IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
* Revise dimensions of PIPMAT and RESULT arrays from 30 to 99.
* SCM 4-30-89
*
* DIMENSION HED(15),PLIST(10),PIPMAT(5,99),RESULT(20,99),SOLVE(6)
*
* NEW ADDITION COMMON STATEMENT IN 1983 FOR JOINT SLIPPING BEHAVIOR
*
* Expand array dimension statements from 30 to 99.
* SCM 5-22-89
*
* COMMON/XJOINT/DSTRN(99),XLONG(99),JOINT
*
* Add common statement for output control. SCM 3-18-89
*
* COMMON /OUTPUT/ IWRT,IPL0T,MAXIT,JCON
*
* DATA NLIST,PLIST /5,4HSTEE,4HALUM,4HCONC,4HPLAS,4HBASI,4HL
* 1 4HINUM,4HRETE,4HRTIC,4HC /
* DATA DESIGN, ANALYS, STOP/4HDESI,4HANAL,4HSTOP/
* DATA SOLVE/4HBURN,4HF.E.,4HF.E.,4HS ,4HAUTO,4HUSER/
* DATA XPLOT /4HPLTD/
*
* Add file control commands. SCM 2-24-88
*
* OPEN(5,FILE=' ')
* OPEN(6,FILE=' ')
* OPEN(10,FILE='PLOT1.PL1')
* OPEN(30,FILE='PLOT2.PL2')
* OPEN(11,FILE='DATA.SAV')
* OPEN(16,FILE='DUNCAN.DUN')
* OPEN(17,FILE='XFACES.XFC')
* OPEN(12,FORM='UNFORMATTED')
* OPEN(13,FORM='UNFORMATTED')
* OPEN(14,FORM='UNFORMATTED')

```

```

ACAN0001
ACAN0002
ACAN0003
ACAN0004
ACAN0005
ACAN0006
ACAN0007
ACAN0008
ACAN0009
ACAN0010
ACAN0011
ACAN0012
ACAN0013
ACAN0014
ACAN0015
ACAN0016
ACAN0017
ACAN0018
ACAN0019
ACAN0020
ACAN0021
ACAN0022
ACAN0023
ACAN0024
ACAN0025
ACAN0026
ACAN0027
ACAN0028
ACAN0029
ACAN0030
ACAN0031
ACAN0032
ACAN0033
ACAN0034
ACAN0035
ACAN0036
ACAN0037
ACAN0038
ACAN0039
ACAN0040
ACAN0041
ACAN0042
ACAN0043
ACAN0044
ACAN0045
ACAN0046
ACAN0047
ACAN0048
ACAN0049
ACAN0050
ACAN0051
ACAN0052
ACAN0053
ACAN0054

Delete Unit 30 output from ACANDE and add to RESOUT.
SCM 3-03-89

REWIND 30

NPROB = 0
WRITE(6,2000)
WRITE(6,2010)
100 CONTINUE
DO 105 I = 1, 99
DO 105 N = 1,20
105 RESULT(N,I) = 0.0
READ(5,1000) XMODE,LEVEL,PTYPE,HED,NPMAT,NPPT
IF (XMODE.EQ. STOP) GO TO 900
IF (LEVEL.LT.1 .OR. LEVEL.GT.3) GO TO 500
IF (XMODE.NE.DESIGN .AND. XMODE.NE.ANALYS) GO TO 500

Revise control statements for canned mesh additions.
SCM 3-18-89

IF (LEVEL.NE.3) NPMAT = 10
IF (LEVEL.NE.3) NPPT = 11

IF (LEVEL.EQ.2) THEN
LCODE = -(LEVEL*100 + NPMAT)
ELSE
LCODE = LEVEL
ENDIF
IF (LEVEL.EQ.2 .AND. NPMAT.EQ.2) THEN
NPMAT = 14
NPPT = 15
ELSE IF (LEVEL.EQ.2 .AND. NPMAT.EQ.3) THEN
NPMAT = 19
NPPT = 20
ELSE IF (LEVEL.NE.3) THEN
NPMAT = 10
NPPT = 11
ENDIF

NPIPE = 0
DO 110 N=1,NLIST
110 IF (PTYPE.EQ.PLIST(N)) NPIPE = N
IF (NPIPE.EQ.0) GO TO 500

NPROB = NPROB + 1
WRITE(6,2100) NPROB,HED,XMODE,SOLVE(LEVEL),SOLVE(LEVEL+3),
1 PLIST(NPIPE),PLIST(NPIPE+5)

READ OR ESTABLISH PIPE PROPERTIES FROM PIPE LIBRARY

ICOME = 1
GO TO 300
CONTINUE

READ SOIL AND SYSTEM PROPERTIES FROM SOLUTION LIBRARY

GO TO 400
CONTINUE

130 SOLVE SYSTEM FOR PIPE RESPONSES, ITERATE AS REQUIRED.
C
C IEXIT = 1
140 ICOME = 2

```

```

ACAN0055
ACAN0056
ACAN0057
ACAN0058
ACAN0059
ACAN0060
ACAN0061
ACAN0062
ACAN0063
ACAN0064
ACAN0065
ACAN0066
ACAN0067
ACAN0068
ACAN0069
ACAN0070
ACAN0071
ACAN0072
ACAN0073
ACAN0074
ACAN0075
ACAN0076
ACAN0077
ACAN0078
ACAN0079
ACAN0080
ACAN0081
ACAN0082
ACAN0083
ACAN0084
ACAN0085
ACAN0086
ACAN0087
ACAN0088
ACAN0089
ACAN0090
ACAN0091
ACAN0092
ACAN0093
ACAN0094
ACAN0095
ACAN0096
ACAN0097
ACAN0098
ACAN0099
ACAN0100
ACAN0101
ACAN0102
ACAN0103
ACAN0104
ACAN0105
ACAN0106
ACAN0107
ACAN0108
ACAN0109
ACAN0110
ACAN0111
ACAN0112
ACAN0113
ACAN0114
ACAN0115
ACAN0116
ACAN0117

```

ACAN0118
ACAN0119
ACAN0120
ACAN0121
ACAN0122
ACAN0123
ACAN0124
ACAN0125
ACAN0126
ACAN0127
ACAN0128
ACAN0129
ACAN0130
ACAN0131
ACAN0132
ACAN0133
ACAN0134
ACAN0135
ACAN0136
ACAN0137
ACAN0138
ACAN0139
ACAN0140
ACAN0141
ACAN0142
ACAN0143
ACAN0144
ACAN0145
ACAN0146
ACAN0147
ACAN0148
ACAN0149
ACAN0150
ACAN0151
ACAN0152
ACAN0153
ACAN0154
ACAN0155
ACAN0156
ACAN0157
ACAN0158
ACAN0159
ACAN0160
ACAN0161
ACAN0162
ACAN0163
ACAN0164
ACAN0165
ACAN0166
ACAN0167
ACAN0168
ACAN0169
ACAN0170
ACAN0171
ACAN0172
ACAN0173
ACAN0174
ACAN0175
ACAN0176
ACAN0177
ACAN0178
ACAN0179
ACAN0180

```
C    900 WRITE(6,6000)
      STOP
C
C    FORMATS
C
C    1000 FORMAT (A4,3X,I1,1X,A4,3X,15A4,2I2)
C
*    Revise title page of Unit 6 output file for 1989 version.
*    SCM 3-10-90
*
2000 FORMAT(1H1,//
X,/12X,57H          CCCC   AA       N     N DDDD EEEEE           8888 999
X,/12X,57H        CC     AAA      NN    N DD  D EE             88 8 99
X,/12X,56H        CC     AA A     NNN   N DD  D EEEE DDD 88888 9999
X,/12X,55H        CC     AAAAA  NN  N N DD  D EE              88 8 9
X,/12X,54H        CCCC   AA    A NN   NN DDDDD EEEEEE         8888 9
X,/
X,/12X,52H|||||IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
X,/12X,51H               IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
X,/12X,50H                                           III
X,/12X,49H                    (C)ULVERT (AN)ALYSIS AND (DE)SIGN II
X,/12X,48H                                                              I
X,/
X,/12X,35H                                VERSION 1.0
X,/12X,34H                               MARCH 1990
X,/
X,/12X,47H                PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
X,/12X,46H                 THE FEDERAL HIGHWAY ADMINISTRATION
X/)
2010 FORMAT(1H ,//
X,/12X,27H          PIPE TYPE LIBRARIES:
X,/12X,33H            ALUMINUM (CORRUGATED)
X,/12X,33H            BASIC (ANALYSIS ONLY)
X,/12X,46H                        (UNCONNECTED ELEMENT OPTION)
X,/12X,33H            CONCRETE (REINFORCED)
X,/12X,32H            PLASTIC (SOLID WALL)
X,/12X,30H            STEEL (CORRUGATED)
X,/12X,40H                      (SLOTTED JOINT OPTION)
X,/
X,/12X,23H          SOLUTION LEVELS:
X,/12X,43H            LEVEL 1: ELASTIC THEORY - BURNS
X,/12X,42H            LEVEL 2: FINITE ELEMENT - AUTO
X,/12X,51H                  CIRCULAR AND ELLIPTICAL (CAN1)
X,/12X,51H                  CONCRETE BOX CULVERT (CANBOX)
X,/12X,51H                  2 OR 3 SEGMENT ARCHES (CANARI)
X,/12X,42H            LEVEL 3: FINITE ELEMENT - USER
X/)
*
2100 FORMAT(1H1,/// 10X,19H *** PROBLEM NUMBER,I3,4H ****//10X,15A4//
X 10X, 21H EXECUTION MODE ..... ,A4 //
X 10X, 21H SOLUTION LEVEL ..... ,2A4 //
X 10X, 21H CULVERT TYPE ..... ,2A4)
3000 FORMAT(////28H *** FATAL ERROR IN DATA *** // 9H MODE IS ,A4,4X,
X10H LEVEL IS ,I2,4X,14H PIPE-TYPE IS ,A4)
6000 FORMAT (///// 10X,40H * * * * NORMAL EXIT FROM CANDE * * * *
END
*
* *****
* CANDE-B9 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
*
*          VERSION 1.0, MARCH 1990
*
* PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
*
* THE FEDERAL HIGHWAY ADMINISTRATION
*****
```

ACAN0181
ACAN0182
ACAN0183
ACAN0184
ACAN0185
ACAN0186
ACAN0187
ACAN0188
ACAN0189
ACAN0190
ACAN0191
ACAN0192
99ACAN0193
9ACAN0194
ACAN0195
ACAN0196
ACAN0197
ACAN0198
ACAN0199
ACAN0200
ACAN0201
ACAN0202
ACAN0203
ACAN0204
ACAN0205
ACAN0206
ACAN0207
ACAN0208
ACAN0209
ACAN0210
ACAN0211
ACAN0212
ACAN0213
ACAN0214
ACAN0215
ACAN0216
ACAN0217
ACAN0218
ACAN0219
ACAN0220
ACAN0221
ACAN0222
ACAN0223
ACAN0224
ACAN0225
ACAN0226
ACAN0227
ACAN0228
ACAN0229
ACAN0230
ACAN0231
ACAN0232
ACAN0233
ACAN0234
ACAN0235
ACAN0236
ACAN0237
AFUN0001
AFUN0002
AFUN0003
AFUN0004
AFUN0005
AFUN0006

```

      FUNCTION AF(I,J)
AFUN0007
*
*   Uncomment this statement in all program modules for double
*   precision version. SCM 3-10-90
*
*   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
COMMON /AFCALC/ XX(4), YY(4)
AF = (XX(I) - XX(J)) * (YY(I) + YY(J)) / 2.0
RETURN
END
* * * * *
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
*   VERSION 1.0, MARCH 1990
*   PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
*   THE FEDERAL HIGHWAY ADMINISTRATION
* * * * *
SUBROUTINE ALUMIN (IA, ICOME, IEXIT, LEVEL, NINC, NPMAT, NPPT, PDIA,
1 PIPMAT, RESULT, SK, SM, XMODE)
C
C   SUBROUTINE FOR CORRUGATED ALUMINUM CULVERTS.
C
C   THE FOLLOWING ADDITIONS TO THE * RESULT * ARRAY ARE CALCULATED
C   IN THIS SUBROUTINE.
C
      RESULT(10,N) = BENDING STRESSES IN OUTER FIBERS.
      RESULT(11,N) = THRUST STRESS (AVERAGE OVER SECTION)
      RESULT(12,N) = SHEAR STRESS (AVERAGE OVER SECTION)
C
*   Uncomment this statement in all program modules for double
*   precision version. SCM 3-10-90
*
*   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
*   Revise dimensions of PIPMAT and RESULT arrays from 30 to 99.
*   SCM 4-30-89
*
DIMENSION CPLIST(10), FS(4), PALIST(35), PFS(4), PHINGE(5), FILIST(35),
1 PIPMAT(5,99), PSLIST(35), RESULT(20,99)
*
DATA E, XNU, YIELD, ELONG, E2/10200000., 0.33, 24000., 0.05, 600000./
DATA NFS, FS/4, 3., 4., 2., 2.0/
DATA FF, GAGE20/0.09, 0.038/
DATA ANALYS, DESIGN /4HANAL, 4HDESI/
DATA CPLIST, BXB, NUMCP, NUMGA/ 4H 8/3, 4H 3, 4H 6, 4H 6, 4H 9,
1 4H1/2, 2*4H1, 4H2, 4H5/2, 3H X, 5, 7/
DATA PALIST/
1 0.03875, 0.05158, 0.06458, 0.08066, 0.11300, 0.14533, 0.17775,
2 0.04450, 0.05925, 0.07416, 0.09275, 0.13000, 0.16733, 0.20483,
3 0.00000, 0.00000, 0.06460, 0.08080, 0.11300, 0.14540, 0.17750,
4 0.00000, 0.00000, 0.20408, 0.22825, 0.26658, 0.30483, 0.34325,
5 0.11700, 0.14600, 0.17500, 0.20400, 0.23400, 0.26300, 0.29200/
DATA FILIST/
1 0.00112, 0.00150, 0.00189, 0.00239, 0.00342, 0.00453, 0.0057,
2 0.00515, 0.00689, 0.00865, 0.01088, 0.01545, 0.02017, 0.02508,
3 0.00000, 0.00000, 0.00850, 0.01060, 0.01490, 0.01910, 0.02340,
4 0.00000, 0.00000, 0.09616, 0.10800, 0.12691, 0.14616, 0.16583,
5 0.09100, 0.11400, 0.13600, 0.15900, 0.18200, 0.20600, 0.22700/
DATA PSLIST/
1 0.00419, 0.00549, 0.00676, 0.00831, 0.01133, 0.01428, 0.01724,
2 0.00995, 0.01315, 0.01634, 0.02025, 0.02798, 0.03557, 0.04308,
3 0.00000, 0.00000, 0.01603, 0.01974, 0.02697, 0.03366, 0.04021,
4 0.00000, 0.00000, 0.08883, 0.09891, 0.11466, 0.13016, 0.14575,
5 0.07000, 0.08685, 0.10264, 0.11888, 0.13481, 0.15119, 0.16509/
ALUM0008 C
ALUM0009 GO TO (100,200), ICODE
ALUM0010 C
ALUM0011 C
ALUM0012 100 FIRST ENTRY ESTABLISHES PIPE PROPERTIES, AND SAFETY FACTORS
ALUM0013 READ(5,1000) NONLIN, PDIA, PE, PNU, PYIELD, PELONG, PDEN, PE2
ALUM0014 IF(PE.LE.0.0) PE = E
ALUM0015 IF(PNU.LE.0.0) PNU = XNU
ALUM0016 IF(PYIELD.LE.0.0) PYIELD = YIELD
ALUM0017 IF(PELONG.LT.PYIELD*(1.-PNU**2)/PE) PELONG = ELONG
ALUM0001 IF(NONLIN.NE.1 .AND. NONLIN.NE.2) NONLIN = 0
ALUM0002 IF(NONLIN.EQ.2 .AND. PE2.LE.0.0) PE2 = E2
ALUM0003 C
ALUM0004 WRITE(6,2000) PDIA, PE, PNU, PYIELD, PELONG, PDEN, NONLIN
ALUM0005 IF(NONLIN.EQ.2) WRITE(6,2001) PE2
ALUM0006 PIMIN = PDIA**2/(PE*FF)
ALUM0007 PHINGE(1) = 0.0
ALUM0008 IF(XMODE.EQ.DESIGN) GO TO 110
ALUM0009 READ(5,1010) PA, PI, PS
ALUM0010 WRITE(6,2010) PA, PI, PS
ALUM0011 GO TO 130
ALUM0012 C
ALUM0013 110 READ(5,1010) (PFS(I), I=1, NFS)
ALUM0014 DO 120 I=1, NFS
ALUM0015 IF(PFS(I).GT.0.0) GO TO 120
ALUM0016 PFS(I) = FS(I)
ALUM0017 120 CONTINUE
ALUM0018 WRITE(6,2100) (PFS(I), I=1, NFS)
ALUM0019 C
ALUM0020 PAMIN = GAGE20
ALUM0021 PA = PAMIN
ALUM0022 PI = PIMIN
ALUM0023 C
ALUM0024 NOTE...SECTION MODULUS FOR ALUMINUM CORRUGATIONS HAS THE RELATION,
ALUM0025 PS = PI/(1.417*SQRT(PI/PA) + 0.450*PA)
ALUM0026 ITER = 0
ALUM0027 C
ALUM0028 INITIALIZE PIPE PROPERTIES ALL AROUND PIPE TO BE THE SAME
ALUM0029 C
ALUM0030 130 DO 140 N=1, NPMAT
ALUM0031 PIPMAT(1,N) = PE
ALUM0032 PIPMAT(2,N) = PNU
ALUM0033 PIPMAT(3,N) = PA
ALUM0034 PIPMAT(4,N) = PI
ALUM0035 140 PIPMAT(5,N) = PDEN*PA
ALUM0036 C
ALUM0037 DO 145 N=1, NPPT
ALUM0038 RESULT(14,N) = PA
ALUM0039 RESULT(15,N) = PI
ALUM0040 145 RESULT(16,N) = PI/PS
ALUM0041 ICON = 0
ALUM0042 PESUM = PE
ALUM0043 RETURN
ALUM0044 C
ALUM0045 SECOND ENTRY DETERMINES MAXIMUM STRUCTURAL RESPONSES FOR
ALUM0046 EVALUATING THE PIPE IN TERMS OF FAILURE CRITERIA. IN THE DESIGN
ALUM0047 MODE, THE PROGRAM ITERATES TO FIND ADEQUATE SECTION PROPERTIES.
ALUM0048 200 IEXIT = 0
ALUM0049 C
ALUM0050 IF(NONLIN.EQ.1 .AND. IA.EQ.NINC) CALL HINGE(PIPMAT, NPMAT, PS, PDIA,
ALUM0051 X PYIELD, SM, SK, RESULT, NPPT, PHINGE)
ALUM0052
ALUM0053
ALUM0054
ALUM0055
ALUM0056
ALUM0057
ALUM0058
ALUM0059
ALUM0060
ALUM0061
ALUM0062
ALUM0063
ALUM0064
ALUM0065
ALUM0066
ALUM0067
ALUM0068
ALUM0069
ALUM0070
ALUM0071
ALUM0072
ALUM0073
ALUM0074
ALUM0075
ALUM0076
ALUM0077
ALUM0078
ALUM0079
ALUM0080
ALUM0081
ALUM0082
ALUM0083
ALUM0084
ALUM0085
ALUM0086
ALUM0087
ALUM0088
ALUM0089
ALUM0090
ALUM0091
ALUM0092
ALUM0093
ALUM0094
ALUM0095
ALUM0096
ALUM0097
ALUM0098
ALUM0099
ALUM0100
ALUM0101
ALUM0102
ALUM0103
ALUM0104
ALUM0105
ALUM0106
ALUM0107
ALUM0108
ALUM0109
ALUM0110
ALUM0111
ALUM0112
ALUM0113
ALUM0114

```



```

IF(NONLIN.NE.2) GO TO 203
ICON = ICON + 1
CALL EMOD(IA, ICON, NPMAT, NPPT, PA, PE, PEZ, PESUM, PI, PIPMAT,
1 PNU, PS, PYIELD, RESULT, ICOME)
IF(ICON.EQ.-1) GO TO 203
IEXIT = -1
RETURN

C
C CHECK IF NONLINEAR SOIL MODEL HAS CONVERGED ...
C
203 IF(NINC.GT.0) GO TO 204
IEXIT = -1
RETURN

C
C EVALUATE CURRENT MAXIMUM STRESSES, DISPLACEMENTS, AND PRESSURES.
C
204 TMAX = 0.0
BMAX = 0.0
PAVG = 0.0
EPSMAX = 0.0
UYMAX = RESULT(4,1)
UYMIN = RESULT(4,1)

C
DO 210 N=1,NPPT
RESULT(10,N) = RESULT(5,N)/PS
RESULT(11,N) = RESULT(6,N)/PA
RESULT(12,N) = RESULT(7,N)/PA
IF(NONLIN.EQ.0) GO TO 205
SIGN = 1.0
IF(RESULT(10,N).LT.0.0) SIGN = -1.0
BSTRES = PYIELD
IF(NONLIN.EQ.2) BSTRES = PYIELD + RESULT(17,N)
IF(SIGN*RESULT(10,N).GT.PYIELD) RESULT(10,N) = SIGN*BSTRES
205 IF(ABS(RESULT(10,N)).GT.BMAX) BMAX = ABS(RESULT(10,N))
IF(ABS(RESULT(11,N)).GT.TMAX) TMAX = ABS(RESULT(11,N))
IF(RESULT(4,N).GT.UYMAX) UYMAX = RESULT(4,N)
IF(RESULT(4,N).LT.UYMIN) UYMIN = RESULT(4,N)
IF(NONLIN.EQ.2.AND.RESULT(18,N).GT.EPSMAX) EPSMAX = RESULT(18,N)
PAVG = PAVG + RESULT(8,N)
210 CONTINUE
PAVG = ABS(PAVG/NPPT)
DISP = UYMAX-UYMIN
IF (SM .EQ. 0.0) SM = 1.0
PBUCK = PDIA**3/(36.*PESUM*SM*(1.-SK))*(1.-PNU**2)
PCRT = SQRT(PI/PBUCK)
IF(NONLIN.EQ.0) EPSMAX = BMAX*(1.-PNU**2)/PE
IF(NONLIN.EQ.1) EPSMAX = PHINGE(5)*(1.-PNU**2)*PYIELD/PE
IF(NONLIN.EQ.2) EPSMAX = EPSMAX*(1.-PNU**2)*PYIELD/PE
IF(NONLIN.EQ.1 .AND. PHINGE(1).LT.0.0) GO TO 300
IF(XMODE.EQ.ANALYS) GO TO 300

C
C DESIGN... REVISE SECTION PROPERTIES AS REQUIRED
C
ICON = 0
IF(IA.LT.NINC) RETURN
ITER = ITER + 1
ICHKA = 0
ICHKI = 0

C
C UPDATE THRUST AREA AS REQUIRED BY MAX THRUST.
C
PANEW = PFS(1)*TMAX*PA/PYIELD
IF(PANEW.LT.PAMIN) PANEW = PAMIN
ALUM0115 IF(ABS((PANEW-PA)/PANEW).LT.0.05) ICHKA = 1
ALUM0116 PA = PANEW
ALUM0117
C
ALUM0118 UPDATE MOMENT OF INERTIA AS REQUIRED BY ... (1) DEFLECTION LIMIT
ALUM0119 AS A PERCENT OF DIAMETER, OR (2) ELASTIC BUCKLING AS GIVEN BY
ALUM0120 CHELAPATI, HIGHWAY RESEARCH RECORD, NO. 413, 1972., OR (3) STRAIN
ALUM0121 RUPTURE IN THE OUTER FIBERS.
ALUM0122
C
ALUM0123 PINEW = PFS(2)*DISP*PI/(0.2*PDIA)
ALUM0124 PINEWB = (PFS(3)*PAVG)**2*PBUCK
ALUM0125 PINEWB = PFS(4)*EPSMAX*PI/PELONG
ALUM0126
*
ALUM0127 * Revise intrinsic function to generic form.
ALUM0128 * SCM 3-10-90
ALUM0129
*
ALUM0130 PINEW = MAX(PINEW, PINEWB, PINEWB)
ALUM0131
*
ALUM0132 IF(PINEW.LT.PIMIN) PINEW = PIMIN
ALUM0133 IF(ABS((PINEW-PI)/PINEW).LT.0.05) ICHKI = 1
ALUM0134 PI = PINEW
ALUM0135 PS = PI/(1.417*SQRT(PI/PA) + 0.450*PA)
ALUM0136
C
ALUM0137 UPDATE SECTION PROPERTIES AND CHECK FOR CONVERGENCE.
ALUM0138
C
ALUM0139 IF(ICHKA*ICHKI.EQ.1) GO TO 250
ALUM0140 IF(ITER.GT.5) WRITE(6,2200) ITER, PA, PI, PS
ALUM0141 IF(ITER.GT.5) GO TO 300
ALUM0142 IEXIT = -2
ALUM0143 GO TO 130
ALUM0144
C
ALUM0145 SEARCH STANDARD CORRUGATION TABLES FOR SUITABLE SECTIONS WHICH
ALUM0146 SATISFY THE DESIGN SOLUTION, LIST THEM.
ALUM0147
C
250 WRITE(6,2300) ITER, PA, PI, PS
ALUM0148 SUMMAX = 0.0
ALUM0149 IJBEST = 0
ALUM0150 DO 280 I=1,NUMCP
ALUM0151 NGAGE = 0
ALUM0152 DO 260 J=1,NUMGA
ALUM0153 IJ = J+NUMGA*(I-1)
ALUM0154 CPA = PALIST(IJ)
ALUM0155 CPI = PILIST(IJ)
ALUM0156 IF(CPA.LT.PA .OR. CPI.LT.PI) GO TO 260
ALUM0157 NGAGE = 22-2*J
ALUM0158 IF(I.EQ.4) NGAGE = 14-2*J+J/4
ALUM0159 IF(I.EQ.5) NGAGE = 0
ALUM0160 GO TO 270
ALUM0161
260 CONTINUE
ALUM0162 WRITE(6,2400) CPLIST(I), BXB, CPLIST(I+5)
ALUM0163 GO TO 280
ALUM0164
270 WRITE(6,2500) CPLIST(I), BXB, CPLIST(I+5), NGAGE, CPA, CPI
ALUM0165 SUMS = PA/CPA + PI/CPI
ALUM0166 IF(SUMS.LT.SUMMAX) GO TO 280
ALUM0167 SUMMAX = SUMS
ALUM0168 IJBEST = IJ
ALUM0169 NGBEST = NGAGE
ALUM0170
280 CONTINUE
ALUM0171
C
ALUM0172
C
ALUM0173 CHOOSE MOST EFFICIENT SECTION FROM LIST AND ANALYSE.
ALUM0174
C
ALUM0175 IF(IJBEST.EQ.0) WRITE(6,2600)
ALUM0176 IF(IJBEST.EQ.0) GO TO 300
ALUM0177 PA = PALIST(IJBEST)
ALUM0178
ALUM0179
ALUM0180
ALUM0181
ALUM0182
ALUM0183
ALUM0184
ALUM0185
ALUM0186
ALUM0187
ALUM0188
ALUM0189
ALUM0190
ALUM0191
ALUM0192
ALUM0193
ALUM0194
ALUM0195
ALUM0196
ALUM0197
ALUM0198
ALUM0199
ALUM0200
ALUM0201
ALUM0202
ALUM0203
ALUM0204
ALUM0205
ALUM0206
ALUM0207
ALUM0208
ALUM0209
ALUM0210
ALUM0211
ALUM0212
ALUM0213
ALUM0214
ALUM0215
ALUM0216
ALUM0217
ALUM0218
ALUM0219
ALUM0220
ALUM0221
ALUM0222
ALUM0223
ALUM0224
ALUM0225
ALUM0226
ALUM0227
ALUM0228
ALUM0229
ALUM0230
ALUM0231
ALUM0232
ALUM0233
ALUM0234
ALUM0235
ALUM0236
ALUM0237
ALUM0238
ALUM0239
ALUM0240

```

```

PI = P1LIST(IJBEST)
PS = PSLIST(IJBEST)
IBEST = (IJBEST-1)/NUMGA+1
WRITE(6,2700) CPLIST(IBEST),BXH,CPLIST(IBEST+5),NGREST
XMODE = ANALYS
IEXIT = -2
GO TO 130

C
300 CONTINUE
C
C PRINT FINITE ELEMENT FIELD RESPONSES.
C
IF (LEVEL.EQ. 1) GO TO 330
ICOME = 3
CALL PRHERO(IA, ICOME, IEXIT, LEVEL, NINC, NPMAT, NPPT, PDIA, PIPMAT,
1 RESULT, SK, SM)
ICOME = 2
330 CONTINUE
ICOME=3
IF(NONLIN.EQ.2) CALL EMOD(IA, ICON, NPMAT, NPPT, PA, PE, PE2, PESUM, PI,
1 PIPMAT, PNU, PS, PYIELD, RESULT, ICOME)
ICOME=2

C
C PRINTOUT STRUCTURAL RESULTS AND EVALUATION.
C
WRITE(6,2800) IA, (RESULT(1,J), RESULT(3,J), RESULT(8,J), RESULT(5,J),
X RESULT(6,J), RESULT(7,J), RESULT(2,J), RESULT(4,J), RESULT(9,J),
X RESULT(10,J), RESULT(11,J), RESULT(12,J), J=1, NPPT)

C
IF(NONLIN.EQ.2) WRITE(6,2990) IA, (J, RESULT(19,J), RESULT(20,J),
X RESULT(18,J), RESULT(13,J), J=1, NPPT)

C
IF (TMAX*DISP*PAVG*BMAX .EQ. 0.0) GO TO 350
CIRCR = PYIELD/TMAX
DISPR = (0.2*PDIA)/DISP
BUCKR = PCRT/PAVG
STRNR = FELONG/EFSMAX
BENDR = FYIELD/BMAX
HANDR = PI/PIMIN
WRITE(6,2900) IA, CIRCR, DISPR, BUCKR, STRNR, BENDR, HANDR
350 CONTINUE
IF(NONLIN.EQ.1) GO TO 400
IF(PHINGE(1).LT.0.0 .OR. PHINGE(1).GT.1.0) GO TO 400
WRITE(6,2950) (PHINGE(I), I=1,5)
400 IEXIT = 1
IF(IA.LT.NINC) IEXIT= 0
ICON = 0
RETURN

C
C FORMATS
C
1000 FORMAT(I5,7F10.0)
1010 FORMAT(4F10.0)
2000 FORMAT(/// 10X,35H PIPE PROPERTIES ARE AS FOLLOWS ... //
X 15X,36H AVERAGE PIPE DIAMETER (IN.) ..... , E15.5 //
X 15X,36H YOUNGS MODULUS OF PIPE (PSI)..... , E15.5 //
X 15X,36H POISSONS RATIO OF PIPE (-) ..... , E15.5 //
X 15X,36H YIELD STRESS OF PIPE (PSI)..... , E15.5 //
X 15X,36H RUPTURE STRAIN OF PIPE (IN/IN) .... , E15.5 //
X 15X,36H DENSITY OF PIPE (PCI)..... , E15.5 //
X 15X,36H MATERIAL CHARACTER, NONLIN ..... , I5 /
X 17X,25H NONLIN=0, IMPLIES LINEAR /
X 17X,30H NONLIN=1, IMPLIES YIELD-HINGE /

```

```

ALUM0241 X 17X,34H NONLIN=2, IMPLIES BILINEAR CURVE // ALUM0304
ALUM0242 2001 FORMAT( /15X,36H MODULUS OF UPPER BI-CURVE ..... ,E15.5 // ) ALUM0305
ALUM0243 2010 FORMAT(/// 10X, 31H SECTION PROPERTIES OF PIPE ... // ALUM0306
ALUM0244 X 15X, 36H THRUST AREA (IN**2/IN) ..... , F10.5 // ALUM0307
ALUM0245 X 15X, 36H MOM. OF INERTIA (IN**4/IN) ..... , F10.5 // ALUM0308
ALUM0246 X 15X, 36H SECTION MODULUS (IN**3/IN) ..... , F10.5 // ALUM0309
ALUM0247 2100 FORMAT(/// 10X,43H DESIRED SAFETY FACTORS FOR PIPE DESIGN ... /// ALUM0310
ALUM0248 X 15X,36H THRUST STRESS YIELDING ..... , F10.5 // ALUM0311
ALUM0249 X 15X,36H DEFLECTION COLLAPSE, (0.2*D) ..... , F10.5 // ALUM0312
ALUM0250 X 15X,36H ELASTIC BUCKLING ..... , F10.5 // ALUM0313
ALUM0251 X 15X,36H OUTER FIBER RUPTURE ..... , F10.5 // ALUM0314
ALUM0252 2200 FORMAT(1H1,/// 10X,50H DESIGN ITERATIVE PROCEDURE DID NOT CONVERGEALUM0315
ALUM0253 X AFTER ,I3/ 10X,55H ITERATIONS. THE LAST SOLUTION IS PRINTED AND EALUM0316
ALUM0254 XVALUATED, / 10X,37H WHERE THE SECTION PROPERTIES ARE ... // ALUM0317
ALUM0255 X 15X,36H THRUST AREA PER UNIT LENGTH ..... , E15.5 // ALUM0318
ALUM0256 X 15X,36H MOM. OF INERTIA PER UNIT LENGTH ... , E15.5 // ALUM0319
ALUM0257 X 15X,36H SECTION MODULUS PER UNIT LENGTH ... , E15.5 // ALUM0320
ALUM0258 2300 FORMAT(1H1,///10X,19H DESIGN RESULTS AT ,I2.11H ITERATIONS // ALUM0321
ALUM0259 X 15X, 35H *** REQUIRED THRUST AREA ..... , F10.5 // ALUM0322
ALUM0260 X 15X, 35H *** REQUIRED MOM. OR INERTIA ..... , F10.5 // ALUM0323
ALUM0261 X 15X, 35H *** REQUIRED SECTION MODULUS ..... , F10.5 /// ALUM0324
ALUM0262 X 10X, 52H THE FOLLOWING ALUMINUM CORRUGATED SECTIONS MEET THE / ALUM0325
ALUM0263 X 10X, 37H ABOVE REQUIREMENTS WITH MINIMUM AREA // ALUM0326
ALUM0264 X15X,12H CORRUGATION,3X,5H GAGE,3X,12H THRUST AREA,3X,13H MOM. INERALUM0327
ALUM0265 XTIA // ALUM0328
ALUM0266 2400 FORMAT(17X,3A4,3X,22H (NO ADEQUATE SECTION) ) ALUM0329
ALUM0267 2500 FORMAT(17X,3A4,6X,I2,5X,F10.5,5X,F10.5) ALUM0330
ALUM0268 2600 FORMAT(//// 10X,37H NO STANDARD CORRUGATION IS ADEQUATE. / ALUM0331
ALUM0269 X 10X, 45H AN ANALYSIS OF THE REQUIRED SECTION FOLLOWS. ) ALUM0332
ALUM0270 2700 FORMAT(//// 10X, 39H THE CORRUGATION WITH MINIMUM AREA AND / ALUM0333
ALUM0271 X10X,21H MOMENT OF INERTIA IS,3A4,5H GAGE,I3, / ALUM0334
ALUM0272 X 10X, 41H AN ANALYSIS OF THIS CORRUGATION FOLLOWS. ) ALUM0335
ALUM0273 2800 FORMAT(1H1,///10X,32H STRUCTURAL RESPONSES OF CULVERT , ALUM0336
ALUM0274 X 3X, 12H LOAD STEP = ,I2 // ALUM0337
ALUM0275 X 4X,9H X-COORD.,5X,8H X-DISP.,4X,9H N-PRES.,6X,7H MOMENT,6X, ALUM0338
ALUM0276 X 7H THRUST,7X,6H SHEAR /4X,9H Y-COORD.,5X,8H Y-DISP.,5X,8H S-PRES. ALUM0339
ALUM0277 X ,4X,9H M-STRESS,4X,9H T-STRESS,4X,9H S-STRESS , ALUM0340
ALUM0278 X (//F13.2,5E13.3/F13.2,5E13.3) ALUM0341
ALUM0279 2900 FORMAT(1H1,///10X,34H CALCULATED SAFETY FACTORS AT STEP, I3 /// ALUM0342
ALUM0280 X15X,50H THRUST STRESS ..... (YIELD-STRESS/MAX-STRESS) =,F10.2// ALUM0343
ALUM0281 X15X,50H DISPLACEMENT ..... (DIAMETER*0.2/MAX-DISP.) =,F10.2// ALUM0344
ALUM0282 X15X,50H ELASTIC BUCKLING ..... (P-CRITICAL/P-AVERAGE) =,F10.2// ALUM0345
ALUM0283 X15X,50H STRAIN RUPTURE ..... (RUPTURE-STRAIN/MAX-STRAIN) =,F10.2//ALUM0346
ALUM0284 X 10X, 20H PERFORMANCE FACTORS /// ALUM0347
ALUM0285 X15X,50H BENDING STRESS ..... (YIELD-STRESS/MAX-STRESS) =,F10.2 //ALUM0348
ALUM0286 X15X,50H HANDLING REQUIREMENT ..... (FF*EI/D**2) =,F10.2 /)ALUM0349
ALUM0287 2950 FORMAT(/// 10X, 27H PLASTIC HINGING EVALUATION /// ALUM0350
ALUM0288 X 15X,42H RATIO OF LOAD AT INITIAL YIELDING ..... , F10.5 // ALUM0351
ALUM0289 X 15X,42H RATIO OF MAXIMUM HINGE PENETRATION .... , F10.5 // ALUM0352
ALUM0290 X 15X,42H RATIO OF CIRCUMFERENCE IN YIELDING .... , F10.5 // ALUM0353
ALUM0291 X 15X,42H RATIO OF MAX-DISP. DUE TO YIELDING .... , F10.5 // ALUM0354
ALUM0292 X 15X,42H RATIO OF MAX-STRAIN/YIELD-STRAIN ..... , F10.5 /// ALUM0355
ALUM0293 2990 FORMAT(1H1,///10X,40H NONLINEAR CHARACTERISTICS, LOAD STEP =,I3//ALUM0356
ALUM0294 X5X,73H NODE INNER-FIBER OUTER-FIBER STRAIN RATIO ALUM0357
ALUM0295 X FRACTION OF /21X,57HSTRAIN STRAIN MAX-TO-YIELD ALUM0358
ALUM0296 X WALL YIELDED // (5X,I5,2E17.5,7X,F10.5,7X,F10.5)) ALUM0359
ALUM0297 END ALUM0360
ALUM0298 * * * * * ANSP0001
ALUM0299 * CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM ANSP0002
ALUM0300 * VERSION 1.0, MARCH 1990 ANSP0003
ALUM0301 * PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY ANSP0004
ALUM0302 * THE FEDERAL HIGHWAY ADMINISTRATION ANSP0005
ALUM0303 * * * * * ANSP0006

```

```

SUBROUTINE ANISP(C11, C12, C22, C33, PI)
THIS SUBROUTINE TRANSFORMS ORTHOTROPIC PROPERTIES FROM THEIR
PREFERRED DIRECTIONS TO GLOBAL COORDINATES
*
* Uncomment this statement in all program modules for double
*   precision version. SCM 3-10-90
*
* IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
COMMON /MATERL/ CP(3,3),DEN(10),E(10,10),GNU(10,10),H(10,10),
1 ITYPE(10),MON(10)
DIMENSION TH(3,3),RO(3,3)
CP(1,3)=0.0
CP(2,3)=0.0
CP(3,1)=0.0
CP(3,2)=0.0
CP(1,1)=C11
CP(1,2)=C12
CP(2,2)=C22
CP(3,3)=C33
CP(2,1)=C12
IF(PI.EQ.0.0)GO TO 80
PI=PI*0.0174533
SPH=SIN(PI)
CPH=COS(PI)
D1=CPH*CPH
D2=SPH*SPH
C1=SPH*CPH
TH(1,1)=D1
TH(1,2)=D2
TH(1,3)=C1
TH(2,1)=D2
TH(2,2)=D1
TH(2,3)=-C1
TH(3,1)=-2.0*C1
TH(3,2)=2.0*C1
TH(3,3)=D1-D2
DO 67 M=1,3
DO 67 N=1,3
DU=0.0
DO 65 L=1,3
65 DU=DU+TH(L,N)*CP(M,L)
67 RO(M,N)=DU
DO 75 M=1,3
DO 75 N=1,3
DU1=0.0
DO 72 L=1,3
72 DU1=DU1+TH(L,M)*RO(L,N)
75 CP(M,N)=DU1
80 RETURN
END
* * * * *
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
*   VERSION 1.0, MARCH 1990
*   PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
*   THE FEDERAL HIGHWAY ADMINISTRATION
* * * * *
SUBROUTINE BASIC (IA, ICOME, IEXIT, LEVEL, NINC, NPMAT, NPPT, PDIA,
1 PIPMAT, RESULT, XMODE)
GENERAL SUBROUTINE FOR ANALYSING ARBITRARY PIPES. NO EVALUATION
OF STRUCTURAL RESPONSES ARE MADE.

```

```

ANSP0007 C
ANSP0008 *
ANSP0009 *   Uncomment this statement in all program modules for double
ANSP0010 *   precision version. SCM 3-10-90
ANSP0011 *
ANSP0012 *   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
ANSP0013 *
ANSP0014 *   Revise dimensions of PIPMAT and RESULT arrays from 30 to 99.
ANSP0015 *   SCM 4-30-89
ANSP0016 *
ANSP0017 *   DIMENSION PIPMAT(5,99),RESULT(20,99)
ANSP0018 *
ANSP0019 C   GO TO (100,200),ICOME
ANSP0020 C
ANSP0021 C   FIRST ENTRY READS PIPE PROPERTIES.
ANSP0022 100 IF(NPMAT.EQ.0) RETURN
ANSP0023 *
ANSP0024 *   Add NEXTRA parameter to read statement for input of
ANSP0025 *   unconnected elements.   MGK 1-17-89
ANSP0026 *
ANSP0027 *   READ(5,1000) PDIA,NEXTRA
ANSP0028 *   NTOTAL = NPMAT + NEXTRA
ANSP0029 *
ANSP0030 150 READ(5,1500) ISEQ1,ISEQ2,(PIPMAT(J,ISEQ1),J=1,5)
ANSP0031 *   IF(ISEQ1.LE.0) GO TO 400
ANSP0032 *
ANSP0033 *   Revise statements to substitute NTOTAL for NPMAT to
ANSP0034 *   implement unconnected element option.   MGK 1-17-89
ANSP0035 *
ANSP0036 *   IF(ISEQ2.GT.NTOTAL) GO TO 400
ANSP0037 *
ANSP0038 *   IF(ISEQ2.LE.ISEQ1) ISEQ2=ISEQ1
ANSP0039 *   DO 175 I=ISEQ1,ISEQ2,I
ANSP0040 *   DO 175 J=1,5
ANSP0041 175 PIPMAT(J,I) = PIPMAT(J,ISEQ1)
ANSP0042 *
ANSP0043 *   IF(ISEQ2.EQ.NTOTAL) GO TO 190
ANSP0044 *
ANSP0045 *   GO TO 150
ANSP0046 *
ANSP0047 *   Revise output statement to use NTOTAL instead of NPMAT for
ANSP0048 *   unconnected element option.   MGK 1-17-89
ANSP0049 *
ANSP0050 190 WRITE(6,2000) PDIA,(J,(PIPMAT(I,J),I=1,5),J=1,NTOTAL)
ANSP0051 *   IF(NTOTAL.GT.99) GO TO 400
ANSP0052 *
ANSP0053 *   RETURN
ANSP0054 C
ANSP0055 C   SECOND ENTRY PRINTS RESULTS
ANSP0056 C
ANSP0057 200 CONTINUE
ANSP0058 C
ANSP0059 C   CHECK TO SEE IF NONLINEAR SOIL MODEL HAS CONVERGED ...
ANSP0060 C
ANSP0061 C   IF(NINC.GT.0) GO TO 210
ANSP0062 C   IEXIT = -1
ANSP0063 C   RETURN
ANSP0064 C
ANSP0065 210 CONTINUE
ANSP0066 C
ANSP0067 C   OUTPUT FINITE ELEMENT FIELD DATA (IF REQUESTED)
ANSP0068 C
ANSP0069 C   IF (LEVEL.EQ.1) GO TO 350
ANSP0070 C
ANSP0071 C
ANSP0072 C
ANSP0073 C
ANSP0074 C

```

```

BASC0012
BASC0013
BASC0014
BASC0015
BASC0016
BASC0017
BASC0018
BASC0019
BASC0020
BASC0021
BASC0022
BASC0023
BASC0024
BASC0025
BASC0026
BASC0027
BASC0028
BASC0029
BASC0030
BASC0031
BASC0032
BASC0033
BASC0034
BASC0035
BASC0036
BASC0037
BASC0038
BASC0039
BASC0040
BASC0041
BASC0042
BASC0043
BASC0044
BASC0045
BASC0046
BASC0047
BASC0048
BASC0049
BASC0050
BASC0051
BASC0052
BASC0053
BASC0054
BASC0055
BASC0056
BASC0057
BASC0058
BASC0059
BASC0060
BASC0061
BASC0062
BASC0063
BASC0064
BASC0065
BASC0066
BASC0067
BASC0068
BASC0069
BASC0070
BASC0071
BASC0072
BASC0073
BASC0074

```

```

      ICOME = 3
      CALL PRHERO(IA, ICOME, IEXIT, LEVEL, NINC, NPMAT, NPPT, PDIA, PIPMAT,
1      RESULT, SK, SM)
350  ICOME = 2
      CONTINUE
      IEXIT = 1
      IF(IA.LT.NINC) IEXIT = 0
      IF(NPMAT.LE. 0) RETURN
      WRITE(6,2300) (RESULT(1,J), RESULT(3,J), RESULT(8,J), RESULT(5,J),
X      RESULT(6,J), RESULT(7,J), RESULT(2,J), RESULT(4,J), RESULT(9,J)
X      , J=1, NPPT)
      RETURN
C
C      ERROR EXIT
C
400  WRITE(6,3000) ISEQ1, ISEQ2
      STOP 10
C
C      FORMATS
C
*      Revise format statement for reading input of unconnected
+      structures option. MGK 1-17-89
+
1000  FORMAT(F10.0, I5)
*
1500  FORMAT(2I5, 5F10.0)
2000  FORMAT(/// 10X, 35H PIPE PROPERTIES ARE AS FOLLOWS ... //
X      15X, 36H AVERAGE PIPE DIAMETER ..... , E15.5 //
X      1X, 9H MATERIAL , 5X, 7H YOUNGS , 4X, 8H POISSON , 5X, 7H THRUST , 5X,
X      7H MOMENT , 4X, 8H DENSITY / 3X, 7H NUMBER , 4X, 8H MODULUS , 6X,
X      6H RATIO , 7X, 5H AREA , 4X, 8H INERTIA / (5X, 15, SE12.4))
2300  FORMAT(1H1, //10X, 32H STRUCTURAL RESPONSES OF CULVERT //
X      4X, 9H X-COORD., 5X, 8H X-DISP., 4X, 9H N-PRES., 6X, 7H MOMENT, 6X,
X      7H THRUST, 7X, 6H SHEAR / 4X, 9H Y-COORD., 5X, 8H Y-DISP., 5X, 8H S-PRES.
X      , (//F13.2, SE13.3/F13.2, 2E13.3))
3000  FORMAT( //27H FATAL DATA ERROR IN BASIC /
X      44H BEAM SEQUENCE NUMBERS ARE OUT OF BOUNDS ..., 2I5)
      END
* * * * *
+      CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
+      VERSION 1.0, MARCH 1990
+      PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
+      THE FEDERAL HIGHWAY ADMINISTRATION
+ * * * * *
      SUBROUTINE BEAMEL
C
C      THIS SUBROUTINE CALCULATES THE STIFFNESS MATRIX FOR A BENDING ELEM
C
C
*      Uncomment this statement in all program modules for double
*      precision version. SCM 3-10-90
*
*      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
      COMMON /ELEMNT/ AS(3), BS(3), C1(17,17), EST(4,3), RQ(4), RT(3),
1      STORE(17), Z4(12), ZQ(4), ZT(3), ZY(17), J1, RO
      COMMON /MATERL/ CP(3,3), DEN(10), E(10,10), GNU(10,10), H(10,10),
1      ITYPE(10), NON(10)
      CA=RQ(2)-RQ(1)
      SA=ZQ(2)-ZQ(1)
      XL=SQRT(CA*CA+SA*SA)
      CA=CA/XL
      SA=-SA/XL
      STORE(1)=CA

```

```

BASC0075  STORE(2)=-SA
BASC0076  C
BASC0077  C      NEW ADDITION STATEMENT IN 1983 FOR JOINT SLIPPING BEHAVIOR.
BASC0078  C
BASC0079  C      STORE(3)=XL
BASC0080  C
BASC0081  EB = CP(1,1)
BASC0082  GNUB = CP(1,2)
BASC0083  AB = CP(1,3)
BASC0084  XIB = CP(2,1)
BASC0085  EB = EB/(1.0-GNUB*GNUB)
BASC0086  DU=EB*AB/XL
BASC0087  C1(1,1)=DU
BASC0088  C1(4,1)=-DU
BASC0089  C1(1,4)=-DU
BASC0090  C1(4,4)=DU
BASC0091  XL=0.5*XL
BASC0092  RQ(1)=0.0
BASC0093  RQ(3)=0.0
BASC0094  D=.25/XL
BASC0095  RQ(2)=-D
BASC0096  RQ(4)=D
BASC0097  ZQ(2)=D
BASC0098  ZQ(4)=D
BASC0099  D=D/XL
BASC0100  ZQ(1)=D
BASC0101  ZQ(3)=-D
BASC0102  D=8.0*XL*EB*XIB
BASC0103  DO 120 L=1,4
BASC0104  LS=L+1+L/3
BASC0105  DO 120 J=1,4
BASC0106  JS=J+1+J/3
C
C      120 C1(LS,JS)=D*(RQ(J)*RQ(L)+3      *ZQ(J)*ZQ(L))
C
C      LOCAL UNKNOWNNS TRANSFORMED TO GLOBAL UNKNOWNNS
C
      DO 400 NR=1,4,3
      DO 315 JJ=1,6
      D2=C1(NR+1, JJ)
      D1=C1(NR, JJ)
      C1(NR, JJ)=D1*CA+D2*SA
315  C1(NR+1, JJ)=-D1*SA+D2*CA
      DO 318 II=1,6
      D1=C1(II, NR)
      D2=C1(II, NR+1)
      C1(II, NR)=D1*CA+D2*SA
318  C1(II, NR+1)=-D1*SA+D2*CA
400  CONTINUE
C
C      GRAVITY LOAD OF BENDING ELEMENT
C
      ZY(2) = -RO*XL
      ZY(5) = -RO*XL
      RETURN
      END
* * * * *
*      CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
*      VERSION 1.0, MARCH 1990
*      PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
*      THE FEDERAL HIGHWAY ADMINISTRATION
* * * * *
      SUBROUTINE BEAMST (NOD, NO, I, NSELM, NPT, MN)

```

```

BMEL0026
BMEL0027
BMEL0028
BMEL0029
BMEL0030
BMEL0031
BMEL0032
BMEL0033
BMEL0034
BMEL0035
BMEL0036
BMEL0037
BMEL0038
BMEL0039
BMEL0040
BMEL0041
BMEL0042
BMEL0043
BMEL0044
BMEL0045
BMEL0046
BMEL0047
BMEL0048
BMEL0049
BMEL0050
BMEL0051
BMEL0052
BMEL0053
BMEL0054
BMEL0055
BMEL0056
BMEL0057
BMEL0058
BMEL0059
BMEL0060
BMEL0061
BMEL0062
BMEL0063
BMEL0064
BMEL0065
BMEL0066
BMEL0067
BMEL0068
BMEL0069
BMEL0070
BMEL0071
BMEL0072
BMEL0073
BMEL0074
BMEL0075
BMEL0076
BMEL0077
BMEL0078
BMEL0079
BMEL0080
BMEL0081
BMST0001
BMST0002
BMST0003
BMST0004
BMST0005
BMST0006
BMST0007

```

```

C      THIS SUBROUTINE CALCULATES BEAM FORCES, SHEARS AND MOMENTS
C
C      Uncomment this statement in all program modules for double
C      precision version. SCM 3-10-90
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C      DIMENSION NOD(4,NELEM),NQ(NPT)
C      COMMON /ELEMNT/ AS(3),BS(3),CL(17,17),EST(4,3),RQ(4),RT(3),
1      STORE(17),Z4(12),ZQ(4),ZT(3),ZY(17),J1,RO
C      COMMON/SYSTEM/B(240),A(240,120)
C
C      NEW ADDITION COMMON STATEMENT IN 1983 FOR JOINT SLIPPING BEHAVIOR
C
C      Expand array dimension statements from 30 to 99.
C      SCM 5-22-89
C
C      COMMON/XJOINT/DSTRN(99),XLONG(99),JOINT
C
C      STIFFNESS MATRIX TIMES DISPLACEMENT
C
C      COST=STORE(1)
C      SINT=STORE(2)
C
C      NEW ADDITION STATEMENT IN 1983 FOR JOINT SLIPPING BEHAVIOR.
C
C      XL=STORE(3)
C
C      ZY(5)=ZY(4)
C      ZY(4)=ZY(3)
C      N1 = NOD(1,I)
C      ND=NQ(N1)/100+2
C      ZY(3) = B(ND)
C      DO 100 K=1,6
C      DU=0.0
C      DO 90 L=1,6
90  DU=DU+CL(K,L)*ZY(L)
100 STORE(K)=DU
C      RQ(1)=-STORE(1)*COST-STORE(2)*SINT
C      RQ(2)=STORE(1)*SINT-STORE(2)*COST
C      RQ(3) = STORE(3)
C      RQ(4) = STORE(3)
C      ZQ(1)=STORE(4)*COST+STORE(5)*SINT
C      ZQ(2)=-STORE(4)*SINT+STORE(5)*COST
C      ZQ(3) = -STORE(6)
C      ZQ(4) = STORE(2)
C
C      NEW ADDITION STATEMENTS IN 1983 FOR JOINT SLIPPING BEHAVIOR.
C
C      IF(JOINT.EQ.0)RETURN
C      UL1=COST*ZY(1)+SINT*ZY(2)
C      UL2=COST*ZY(4)+SINT*ZY(5)
C      DSTRN(MN)=(UL2-UL1)/XL
C      XLONG(MN)=XL
C      RETURN
C      END

```

 CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
 VERSION 1.0, MARCH 1990
 PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
 THE FEDERAL HIGHWAY ADMINISTRATION

```

BMST0008 ***** CANA0006
BMST0009 SUBROUTINE CANAR1 (LUDATA,KPUTCK,DENSTY)
BMST0010 C
BMST0011 C      ADDED TO CANDE IN 1989.
BMST0012 C
BMST0013 C      GENERATES TWO OR THREE SEGMENT ARCH MESH WITH CURVED OR
BMST0014 C      STRAIGHT SEGMENTS. DEVELOPED BY SYRO STEEL COMPANY.
BMST0015 *
BMST0016 *      Uncomment this statement in all program modules for double
BMST0017 *      precision version. SCM 3-10-90
BMST0018 *
BMST0019 *      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
BMST0020 *
BMST0021 DIMENSION NODMD1(2,107),NODMD2(2,55),NODMDA(2,9),NODMOD(2,171),
X      XYAND1(3,60),XYAND2(3,51),XYANDA(3,9),XYANDS(3,120),
X      LNOD1(11,19),LNOD2(11,37),LNOD3(11,16),LNOD4(11,19),
X      LNODA(11,3),LNOD(11,94),NODBC1(35),NODBCA(6),NODBC(41),
X      X(307),Y(307),TITLE(17),TFMESH(4)
X      DIMENSION R(3),THETA(4),CHORD(3),ARCLEN(3),XCEN(3),YCEN(3),
X      ANG(20),XPN(20),YPN(20),DELTA(3),NDEV(3)
BMST0022 DATA PI/3.14159265/
BMST0023 DATA L1H, IBLANK, PREP, STOP, XMOD/1HL,1H,4HPREP,4HSTOP,4HMOD /
BMST0024 DATA IZERO, IONE, FZERO, XBLANK /0,1,0,0,4H /
BMST0025 DATA TFMESH /4HEMBA,4HTREN,4HHOMO,4H /
BMST0026 DATA INCLEV, INCHMAX, NUMNP, NUMEL, NBPTC / 8,20,244,209,35 /
BMST0027 DATA NODMD1/ 1,0, 4,202, 5,200, 6,200, 8,202,
X      15,202, 30,5, 45,5, 60,5, 75,102, 90,100,
X      127,5, 157,5, 189,5, 223,5, 244,102, 16,300,
X      19,302, 20,300, 21,300, 23,302, 30,3, 31,300,
X      34,302, 35,300, 36,300, 38,302, 45,3, 46,300,
X      49,302, 50,300, 51,300, 53,302, 60,3, 61,300,
X      64,302, 65,300, 66,300, 68,302, 75,3, 76,300,
X      79,302, 80,300, 81,300, 83,302, 90,3, 121,200,
X      127,3, 151,200, 157,3, 183,200, 189,3, 217,200,
X      223,3, 224,300, 237,202, 244,3, 198,0, 200,302,
X      201,5, 202,5, 224,3, 194,0, 196,302, 197,5,
X      203,5, 225,3, 190,0, 192,302, 193,5, 204,5,
X      226,3, 167,0, 169,302, 170,5, 205,5, 227,3,
X      164,0, 166,302, 171,5, 206,5, 228,3, 161,0,
X      163,302, 172,5, 207,5, 229,3, 158,0, 160,302,
X      173,5, 208,5, 230,3, 138,0, 140,302, 174,5,
X      209,5, 231,3, 136,0, 137,5, 141,302, 175,5,
X      210,5, 232,3, 134,0, 135,5, 142,302, 176,5 /
BMST0028 DATA NODMD2 / 211,5, 233,3, 83,1, 96,5, 102,5,
X      108,5, 114,5, 150,5, 182,5, 216,5, 237,3,
X      233,1, 213,5, 214,5, 215,302, 132,0, 133,5,
X      143,302, 177,5, 212,5, 213,3, 130,0, 131,5,
X      144,302, 178,5, 179,5, 214,3, 128,0, 129,5,
X      145,302, 146,5, 180,5, 215,3, 109,0, 111,302,
X      114,3, 115,0, 117,302, 148,5, 149,5, 150,3,
X      118,0, 120,302, 147,5, 181,5, 182,3, 103,0,
X      105,302, 108,3, 97,0, 99,302, 102,3, 91,0,
X      93,302, 96,3 /
BMST0029 DATA NODMDA / 245,0, 258,202, 265,302, 266,0, 279,202,
X      286,302, 287,0, 300,202, 307,302 /
BMST0030 DATA XYAND1/ 3*0.0, 0.0,1.0,0.79, 0.0,1.0,0.0,
X      0.0,1.0,0.0, 0.0,2*1.0, 0.0,1.0,1.26,
X      15.0,0.0,0.79, 15.0,2*0.0, 15.0,0.0,1.0,
X      1.0,30.0,0.0, 4.0,30.0,0.79, 5.0,30.0,0.0,
X      6.0,30.0,0.0, 8.0,30.0,1.0, 2*0.0,1.26,
X      1.0,45.0,0.0, 4.0,45.0,0.79, 5.0,45.0,0.0,
X      6.0,45.0,0.0, 8.0,45.0,1.0, 2*0.0,1.26,
X      1.0,60.0,0.0, 4.0,60.0,0.79, 5.0,60.0,0.0,

```



```

X      6.0,60.0,0.0,      8.0,60.0,1.0,      2*0.0,1.26,      CANA0069
X      1.0,75.0,0.0,      4.0,75.0,0.79,      5.0,75.0,0.0,      CANA0070
X      6.0,75.0,0.0,      8.0,75.0,1.0,      2*0.0,1.26,      CANA0071
X      1.0,90.0,0.0,      4.0,90.0,0.79,      5.0,90.0,0.0,      CANA0072
X      6.0,90.0,0.0,      8.0,90.0,0.0,      2*0.0,1.26,      CANA0073
X      0.0,127.0,0.0,      2*0.0,1.26,      0.0,157.0,0.0,      CANA0074
X      2*0.0,1.26,      0.0,189.0,0.0,      2*0.0,1.26,      CANA0075
X      0.0,223.0,0.0,      2*0.0,1.26,      1.0,244.0,0.0,      CANA0076
X      0.0,244.0,1.0,      2*0.0,1.26,      3*0.0,      CANA0077
X      2*198.0,0.0,      2*0.0,1.00,      3*0.0,      CANA0078
X      2*194.0,0.0,      2*0.0,1.00,      3*0.0,      CANA0079
X      2*190.0,0.0,      2*0.0,1.00,      3*0.0 /      CANA0080
DATA XYAND2/ 2*167.0,0.0,      2*0.0,1.00,      3*0.0,      CANA0081
X      2*164.0,0.0,      2*0.0,1.00,      3*0.0,      CANA0082
X      2*161.0,0.0,      2*0.0,1.00,      3*0.0,      CANA0083
X      2*158.0,0.0,      2*0.0,1.00,      3*0.0,      CANA0084
X      2*138.0,0.0,      2*0.0,1.00,      3*0.0,      CANA0085
X      2*136.0,0.0,      2*0.0,1.00,      3*0.0,      CANA0086
X      2*134.0,0.0,      2*0.0,1.00,      3*0.0,      CANA0087
X      2*0.0,1.0,      3*0.0,      236.0,216.0,1.0, CANA0088
X      3*0.0,      2*132.0,0.0,      2*0.0,1.00,      CANA0089
X      3*0.0,      2*130.0,0.0,      2*0.0,1.00,      CANA0090
X      3*0.0,      2*128.0,0.0,      2*0.0,1.00,      CANA0091
X      3*0.0,      2*109.0,0.0,      2*0.0,1.00,      CANA0092
X      3*0.0,      2*115.0,0.0,      2*0.0,1.00,      CANA0093
X      3*0.0,      2*118.0,0.0,      2*0.0,1.00,      CANA0094
X      3*0.0,      2*103.0,0.0,      2*0.0,1.00,      CANA0095
X      3*0.0,      2*97.0,0.0,      2*0.0,1.00,      CANA0096
X      3*0.0,      2*91.0,0.0,      2*0.0,1.00 /      CANA0097
DATA XYANDA / 3*0.0,      0.0,245.0,1.0,      15.0,245.0,1.26, CANA0098
X      3*0.0,      0.0,266.0,1.0,      15.0,266.0,1.26, CANA0099
X      3*0.0,      0.0,287.0,1.0,      15.0,287.0,1.26, CANA0100
DATA LNOD1 /      CANA0101
X 1,194,198,0,0,1,1,0,0,0,0,      2,190,194,0,0,2,1,0,0,0,0,      CANA0102
X 3,167,190,0,0,3,1,0,0,0,0,      4,164,167,0,0,4,1,0,0,0,0,      CANA0103
X 5,161,164,0,0,5,1,0,0,0,0,      6,158,161,0,0,6,1,0,0,0,0,      CANA0104
X 7,138,158,0,0,7,1,0,0,0,0,      8,136,138,0,0,8,1,0,0,0,0,      CANA0105
X 9,134,136,0,0,9,1,0,0,0,0,      10,132,134,0,0,10,1,0,0,0,0,      CANA0106
X 11,130,132,0,0,11,1,0,0,0,0,      12,128,130,0,0,12,1,0,0,0,0,      CANA0107
X 13,118,128,0,0,13,1,0,0,0,0,      14,115,118,0,0,14,1,0,0,0,0,      CANA0108
X 15,109,115,0,0,15,1,0,0,0,0,      16,103,109,0,0,16,1,0,0,0,0,      CANA0109
X 17,97,103,0,0,17,1,0,0,0,0,      18,91,97,0,0,18,1,0,0,0,0,      CANA0110
X 19,80,91,0,0,19,1,0,0,0,0 /      CANA0111
DATA LNOD2 /      CANA0112
X 20,1,2,17,16,1,1,0,1,1,0,      34,16,17,32,31,1,1,0,1,1,0,      CANA0113
X 48,31,32,47,46,1,1,0,1,1,0,      62,46,47,62,61,1,1,0,1,1,0,      CANA0114
X 76,61,62,77,76,1,1,0,1,1,0,      81,66,67,82,81,1,1,0,1,1,0,      CANA0115
X 79,64,65,80,79,2,1,0,1,1,0,      91,96,84,121,102,3,2,0,0,0,0,      CANA0116
X 90,83,84,96,0,3,2,0,0,0,0,      98,102,121,108,0,3,3,0,0,0,0,      CANA0117
X 92,84,85,122,121,3,2,0,1,1,0,      100,121,122,152,151,3,3,0,1,1,0,      CANA0118
X 99,108,121,151,114,3,3,0,0,0,0,      107,150,151,183,0,3,4,0,0,0,0,      CANA0119
X 106,114,151,150,0,3,4,0,0,0,0,      109,151,152,184,183,3,4,0,1,1,0,      CANA0120
X 108,150,183,182,0,3,4,0,0,0,0,      122,216,217,238,237,3,8,0,1,1,0,      CANA0121
X 115,182,183,217,216,3,7,0,1,1,0,      132,93,94,100,99,3,2,0,1,1,0,      CANA0122
X 129,80,81,94,93,3,2,0,1,1,0,      138,105,106,112,111,3,3,0,1,1,0,      CANA0123
X 135,99,100,106,105,3,3,0,1,1,0,      141,111,112,148,117,3,4,0,0,0,0,      CANA0124
X 141,111,112,148,117,3,4,0,0,0,0,      142,112,113,149,148,3,4,0,1,1,0,      CANA0125
X 144,117,148,147,120,3,4,0,0,0,0,      145,148,149,181,147,3,4,0,0,0,0,      CANA0126
X 146,149,150,182,181,3,4,0,0,0,0,      147,200,196,197,201,3,5,0,0,0,0,      CANA0127
X 148,196,192,193,197,3,5,0,0,0,0,      149,192,169,170,193,3,5,0,0,0,0,      CANA0128
X 150,169,166,171,170,3,5,0,0,0,0,      151,166,163,172,171,3,5,0,0,0,0,      CANA0129
X 152,163,160,173,172,3,5,0,0,0,0,      153,160,140,174,173,3,5,0,0,0,0,      CANA0130
X 154,140,141,175,174,3,5,0,1,1,0,      158,144,145,146,178,3,5,0,0,0,0 / CANA0131

DATA LNOD3 /      CANA0132
X 159,145,120,147,146,3,5,0,0,0,0,      160,201,197,203,202,3,6,0,0,0,0,      CANA0133
X 161,197,193,204,203,3,6,0,0,0,0,      162,193,170,205,204,3,6,0,0,0,0,      CANA0134
X 163,170,171,206,205,3,6,0,1,1,0,      170,177,178,179,212,3,6,0,0,0,0,      CANA0135
X 171,178,146,180,179,3,6,0,0,0,0,      172,146,147,181,180,3,6,0,0,0,0,      CANA0136
X 173,202,203,225,224,3,7,0,1,1,0,      182,211,212,213,233,3,7,0,0,0,0,      CANA0137
X 183,212,179,214,213,3,7,0,0,0,0,      184,179,180,215,214,3,7,0,0,0,0,      CANA0138
X 185,180,181,182,215,3,7,0,0,0,0,      186,182,216,215,0,3,7,0,0,0,0,      CANA0139
X 187,213,234,233,0,3,8,0,0,0,0,      188,213,214,235,234,3,8,0,1,1,0 / CANA0140
DATA LNOD4 /      CANA0141
X 191,198,200,199,0,1,5,1,0,0,0,      192,194,196,195,0,2,5,1,0,0,0,      CANA0142
X 193,190,192,191,0,3,5,1,0,0,0,      194,167,169,168,0,4,5,1,0,0,0,      CANA0143
X 195,164,166,165,0,5,5,1,0,0,0,      196,161,163,162,0,6,5,1,0,0,0,      CANA0144
X 197,158,160,159,0,7,5,1,0,0,0,      198,138,140,139,0,8,5,1,0,0,0,      CANA0145
X 199,136,141,137,0,9,5,1,0,0,0,      200,134,142,135,0,10,5,1,0,0,0,      CANA0146
X 201,132,143,133,0,11,5,1,0,0,0,      202,130,144,131,0,12,5,1,0,0,0,      CANA0147
X 203,128,145,129,0,13,5,1,0,0,0,      204,118,120,119,0,14,4,1,0,0,0,      CANA0148
X 205,115,117,116,0,15,4,1,0,0,0,      206,109,111,110,0,16,3,1,0,0,0,      CANA0149
X 207,103,105,104,0,17,3,1,0,0,0,      208,97,99,98,0,18,2,1,0,0,0,      CANA0150
X 209,91,93,92,0,19,2,1,0,0,0 /      CANA0151
DATA LNOD5 /      CANA0152
X 191,224,225,246,245,3,9,0,1,1,0,      CANA0153
X 211,245,246,267,266,3,10,0,1,1,0,      CANA0154
X 231,266,267,288,287,3,11,0,1,1,0 /      CANA0155
DATA NODBC1 / 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,30,45,60,75,90,      CANA0156
X      127,157,189,223,224,202,201,200,199,198,76,61,46,31,16 /      CANA0157
DATA NODBCA / 245,265,266,286,287,307 /      CANA0158
C      CANA0159
C      CANA0160
C      CANA0161
C      CANA0162
DO 1,J=1,3      CANA0163
      R(J) = 0.0      CANA0164
      THETA(J) = 0.0      CANA0165
      CHORD(J) = 0.0      CANA0166
      ARCLN(J) = 0.0      CANA0167
      XCEN(J) = 0.0      CANA0168
      YCEN(J) = 0.0      CANA0169
1      DELTA(J) = 0.0      CANA0170
      DO 2,J=1,20      CANA0171
        ANG(J) = 0.0      CANA0172
        XPN(J) = 0.0      CANA0173
2      YPN(J) = 0.0      CANA0174
      DO 3,J=1,307      CANA0175
        X(J) = 0.0      CANA0176
        Y(J) = 0.0      CANA0177
3      HPRESS = 0      CANA0178
      INCADD = 0      CANA0179
      PINC = 0      CANA0180
      XDIV = 0      CANA0181
      YDIV = 0      CANA0182
C      CANA0183
C      READ DATA CARDS THAT SPECIFY THE CANNED MESH      CANA0184
C      CANA0185
      READ(5,1000) WORD,TITLE,WORD1      CANA0186
      READ(5,1005) IPILOT,IWRT,MGENPR,NINC,HTCOVER,DENSTY,TRNDEP,TRNWID,      CANA0187
        TRNSLP      CANA0188
      READ(5,1010) RISE,HFSPAN,SDRISE,FTDEP,FTWIDO,FTWIDI,SPCFAC      CANA0189
      READ(5,1020) (R(J),THETA(J),J=1,3),THETA(4),NTN,NCN      CANA0190
C      CANA0191
C      CONVERT INPUT TO PROPER UNITS      CANA0192
C      CANA0193
      HTCOVER = HTCOVER*12      CANA0194
      TRNDEP = TRNDEP*12

```

```

C      TRNWID = TRNWID*12
C      CHECK INPUT PARAMETERS
C
      DO 4,MESHTP=1,3
        IF (WORD.EQ.TPMESH(MESHTP)) GO TO 5
        CONTINUE
      4  WRITE(6,2000) WORD
        STOP 12
      5  IF (WORD1.NE.XMOD .AND. WORD1.NE.XBLANK) THEN
        WRITE(6,2003)
        KPUTCK = KPUTCK + 1
        END IF
        IF (NINC.LT.0 .OR. NINC.GT.INCMAX) THEN
        WRITE(6,2005)
        KPUTCK = KPUTCK + 1
        END IF
        IF (NINC.EQ.0) NPUTCK = 1
        IF (HTCOVER.LT.0.1*RISE) THEN
        WRITE(6,2010)
        END IF
        IF (RISE.LE.0.0) THEN
        WRITE(6,2011)
        KPUTCK = KPUTCK + 1
        END IF
        IF (HFSPAN.LE.0.0) THEN
        WRITE(6,2012)
        KPUTCK = KPUTCK + 1
        END IF
        IF (SDRISE.LT.0.0) THEN
        WRITE(6,2013)
        KPUTCK = KPUTCK + 1
        END IF
        DO 6,I=1,3
          IF (R(I).LT.0.0) THEN
            WRITE(6,2014)
            KPUTCK = KPUTCK + 1
            END IF
        6  CONTINUE
        IF (WORD.EQ.TPMESH(2)) THEN
          IF (TRNWID.LT.0.3*HFSPAN .OR. TRNWID.GT.HFSPAN) THEN
            WRITE(6,2015)
            KPUTCK = KPUTCK + 1
            END IF
          IF (TRNDEP.LT.0.0 .OR. TRNDEP.GT.HTCOVER+RISE) THEN
            WRITE(6,2020)
            KPUTCK = KPUTCK + 1
          ELSE IF (TRNDEP.LT.0.0 .OR. TRNDEP.GT.2.5*RISE) THEN
            WRITE(6,2020)
            KPUTCK = KPUTCK + 1
            END IF
          IF (TRNSLP.LT.0.0 .OR. TRNSLP.GT.1.0) THEN
            WRITE(6,2025)
            KPUTCK = KPUTCK + 1
            END IF
          IF (FTWIDO.LT.0.2*TRNWID .OR. FTWIDO.GT.0.5*TRNWID) THEN
            WRITE(6,2030)
            KPUTCK = KPUTCK + 1
            END IF
        ELSE
          IF (FTWIDO.LT.0.1*(HFSPAN+RISE) .OR.
X      FTWIDO.GT.0.25*(HFSPAN+RISE)) THEN
            WRITE(6,2030)

```

```

CANA0195      KPUTCK = KPUTCK + 1
CANA0196      END IF
CANA0197      END IF
CANA0198      IF (FTWIDI.LT.0.1*HFSPAN .OR. FTWIDI.GT.0.5*HFSPAN) THEN
CANA0199        WRITE(6,2035)
CANA0200        KPUTCK = KPUTCK + 1
CANA0201        END IF
CANA0202        IF (FTDEP.LT.0.05*RISE .OR. FTDEP.GT.0.3*RISE) THEN
CANA0203          WRITE(6,2040)
CANA0204          KPUTCK = KPUTCK + 1
CANA0205          END IF
CANA0206          IF (SPCFAC.LT.0.0 .OR. SPCFAC.GT.1.3) THEN
CANA0207            WRITE(6,2042)
CANA0208            KPUTCK = KPUTCK + 1
CANA0209            END IF
CANA0210            IF (SDRISE.GT.0.0) THEN
CANA0211              IF (NTN.GT.17 .OR. NTN.LT.0) THEN
CANA0212                WRITE(6,2045)
CANA0213                KPUTCK = KPUTCK + 1
CANA0214                END IF
CANA0215                IF (NCN.GT.18 .OR. NCN.LT.0) THEN
CANA0216                  WRITE(6,2050)
CANA0217                  KPUTCK = KPUTCK + 1
CANA0218                  END IF
CANA0219                  ELSE
CANA0220                    IF (NTN.GT.18 .OR. NTN.LT.0) THEN
CANA0221                      WRITE(6,2045)
CANA0222                      KPUTCK = KPUTCK + 1
CANA0223                      END IF
CANA0224                      IF (NCN.GT.19 .OR. NCN.LT.0) THEN
CANA0225                        WRITE(6,2050)
CANA0226                        KPUTCK = KPUTCK + 1
CANA0227                        END IF
CANA0228                        END IF
C      C
C      DETERMINE MESH SIZE BASED ON HTCOVER
C
      IF (HTCOVER.LE.0.4*RISE .AND. HTCOVER.GE.0.1*RISE) THEN
        JNMODA = 0
        JXYA = 0
        JLNODA = 0
        JBKA = 0
        KCOVER = 0
        END IF
      IF (HTCOVER.LE.0.7*RISE .AND. HTCOVER.GT.0.4*RISE) THEN
        JNMODA = 3
        JXYA = 3
        JLNODA = 1
        JBKA = 2
        KCOVER = 1
        INCLEV = 9
        NUMNP = 265
        NUMEL = 229
        NBPTC = 37
        END IF
      IF (HTCOVER.LE.1.0*RISE .AND. HTCOVER.GT.0.7*RISE) THEN
        JNMODA = 6
        JXYA = 6
        JLNODA = 2
        JBKA = 4
        KCOVER = 2
        INCLEV = 10
        NUMNP = 286

```

```

CANA0258
CANA0259
CANA0260
CANA0261
CANA0262
CANA0263
CANA0264
CANA0265
CANA0266
CANA0267
CANA0268
CANA0269
CANA0270
CANA0271
CANA0272
CANA0273
CANA0274
CANA0275
CANA0276
CANA0277
CANA0278
CANA0279
CANA0280
CANA0281
CANA0282
CANA0283
CANA0284
CANA0285
CANA0286
CANA0287
CANA0288
CANA0289
CANA0290
CANA0291
CANA0292
CANA0293
CANA0294
CANA0295
CANA0296
CANA0297
CANA0298
CANA0299
CANA0300
CANA0301
CANA0302
CANA0303
CANA0304
CANA0305
CANA0306
CANA0307
CANA0308
CANA0309
CANA0310
CANA0311
CANA0312
CANA0313
CANA0314
CANA0315
CANA0316
CANA0317
CANA0318
CANA0319
CANA0320

```



```

      NUMEL = 249
      NBPTC = 39
      END IF
      IF (HTCOVER.GT.1.0*RISE) THEN
        JNMODA = 9
        JXYA = 9
        JLNODA = 3
        JBCA = 6
        KCOVER = 3
        INCLEV = 11
        NUMNF = 307
        NUMEL = 269
        NBPTC = 41
      END IF
C
C  CONTATENATE DATA ARRAYS
      DO 10, NUM=1, 107
        DO 10, NUM1=1, 2
          NODMOD(NUM1, NUM) = NODMD1(NUM1, NUM)
      10  DO 20, NUM=1, 55
        DO 20, NUM1=1, 2
          NODMOD(NUM1, NUM+107) = NODMD2(NUM1, NUM)
      20  IF (JNMODA.GT.0) THEN
        DO 25, NUM=1, JNMODA
          DO 25, NUM1=1, 2
            NODMOD(NUM1, NUM+162) = NODMDA(NUM1, NUM)
        25  END IF
        DO 30, NUM=1, 60
          DO 30, NUM1=1, 3
            XYANDS(NUM1, NUM) = XYAND1(NUM1, NUM)
      30  DO 40, NUM=1, 51
        DO 40, NUM1=1, 3
          XYANDS(NUM1, NUM+60) = XYAND2(NUM1, NUM)
      40  IF (JXYA.GT.0) THEN
        DO 45, NUM=1, JXYA
          DO 45, NUM1=1, 3
            XYANDS(NUM1, NUM+111) = XYANDA(NUM1, NUM)
        45  END IF
        DO 50, NUM=1, 19
          DO 50, NUM1=1, 11
            LNOD(NUM1, NUM) = LNOD1(NUM1, NUM)
      50  DO 60, NUM=1, 37
        DO 60, NUM1=1, 11
          LNOD(NUM1, NUM+19) = LNOD2(NUM1, NUM)
      60  DO 70, NUM=1, 16
        DO 70, NUM1=1, 11
          LNOD(NUM1, NUM+56) = LNOD3(NUM1, NUM)
      70  IF (JLNODA.GT.0) THEN
        DO 75, NUM=1, JLNODA
          DO 75, NUM1=1, 11
            LNOD(NUM1, NUM+72) = LNODA(NUM1, NUM)
      75  DO 77, NUM=1, 19
        DO 76, NUM1=1, 11
          LNOD(NUM1, NUM+72+JLNODA) = LNOD4(NUM1, NUM)
      76  LNOD(1, NUM+72+JLNODA) = 190 + JLNODA*20 + NUM
      77  GO TO 85
        END IF
      DO 80, NUM=1, 19
        DO 80, NUM1=1, 11
          LNOD(NUM1, NUM+72) = LNOD4(NUM1, NUM)
      80  DO 86, NUM=1, 35
      85  DO 86, NUM=1, 35
      86  NODBC(NUM) = NODBC1(NUM)

```

```

CANA0321      IF (JBCA.GT.0) THEN
CANA0322      DO 87, NUM=1, JBCA
CANA0323      87  NODBC(NUM+35) = NODBCA(NUM)
CANA0324      END IF
CANA0325  C
CANA0326  C  ESTABLISH SPACING FACTORS AROUND ARCH
CANA0327  C
CANA0328      SPF = SPCFAC
CANA0329      SPC = SPCFAC - 1
CANA0330      SPVERT = 1.5*SPC*(RISE**2/(RISE**2+HFSPAN**2))
CANA0331      SPHORZ = 0.75*SPC*(HFSPAN**2/(HFSPAN**2+RISE**2))
CANA0332      DO 90, I=53, 80, 3
CANA0333      90  XYANDS(3, I) = 1 + SPC
CANA0334      DO 92, I=87, 111, 3
CANA0335      92  XYANDS(3, I) = 1 + SPC
CANA0336      XYANDS(3, 82) = 1 + SPVERT
CANA0337      XYANDS(3, 49) = 1 + SPHORZ
CANA0338      DO 93, I=113, 119, 3
CANA0339      93  XYANDS(3, I) = 1 + SPHORZ
CANA0340  C
CANA0341  C**** BEGIN SOLUTION OF BEAM ELEMENT GEOMETRY *****
CANA0342  C
CANA0343  C  CALCULATE CHORD AND ARC LENGTHS FOR CURVED SEGMENTS
CANA0344  C
CANA0345      DO 120, J=1, 4
CANA0346      120  THETA(J) = THETA(J)*PI/180
CANA0347      DO 150, J=1, 3
CANA0348      IF (ABS(R(J)).GT.0.0) THEN
CANA0349      CHORD(J) = 2*ABS(R(J))*SIN(THETA(J)/2)
CANA0350      ARCLN(J) = ABS(R(J))*THETA(J)
CANA0351      ELSE
CANA0352      DELTA(J) = THETA(J)
CANA0353      THETA(J) = 0.0
CANA0354      END IF
CANA0355      150  CONTINUE
CANA0356  C
CANA0357  C  ASSIGN BEGINNING COORDINATES
CANA0358  C
CANA0359      X1 = 0.0
CANA0360      Y1 = RISE
CANA0361  C
CANA0362  C  ASSIGN END COORDINATES FOR TWO SEGMENT PROBLEM
CANA0363  C
CANA0364      IF (SDRISE.EQ.0.0) THEN
CANA0365      X3 = HFSPAN
CANA0366      Y3 = 0.0
CANA0367      GO TO 180
CANA0368  C
CANA0369  C  ASSIGN END COORDINATES FOR THREE SEGMENT PROBLEM
CANA0370  C
CANA0371      ELSE
CANA0372      X4 = HFSPAN
CANA0373      Y4 = 0.0
CANA0374      Y3 = SDRISE
CANA0375  C
CANA0376  C  SOLVE INTERSECTION OF CURVED SIDE SEGMENT
CANA0377  C
CANA0378      IF (ARCLN(3).GT.0.0) THEN
CANA0379      X3 = X4+CHORD(3)*COS(PI/2+THETA(4)+THETA(3)/2)
CANA0380  C
CANA0381  C  SOLVE INTERSECTION OF STRAIGHT SIDE SEGMENT
CANA0382  C
CANA0383      ELSE
CANA0384
CANA0385
CANA0386
CANA0387
CANA0388
CANA0389
CANA0390
CANA0391
CANA0392
CANA0393
CANA0394
CANA0395
CANA0396
CANA0397
CANA0398
CANA0399
CANA0400
CANA0401
CANA0402
CANA0403
CANA0404
CANA0405
CANA0406
CANA0407
CANA0408
CANA0409
CANA0410
CANA0411
CANA0412
CANA0413
CANA0414
CANA0415
CANA0416
CANA0417
CANA0418
CANA0419
CANA0420
CANA0421
CANA0422
CANA0423
CANA0424
CANA0425
CANA0426
CANA0427
CANA0428
CANA0429
CANA0430
CANA0431
CANA0432
CANA0433
CANA0434
CANA0435
CANA0436
CANA0437
CANA0438
CANA0439
CANA0440
CANA0441
CANA0442
CANA0443
CANA0444
CANA0445
CANA0446

```

```

      X3 = X4 + Y3*TAN(PI/2-DELTA(3))
      CHORD(3) = SQRT((X3-X4)**2+(Y3-Y4)**2)
    ENDIF
  ENDIF
  C
  C SOLVE INTERSECTION OF TOP AND CORNER SEGMENTS
  C
  180 SIDE13 = SQRT((X1-X3)**2 + (Y1-Y3)**2)
    B1 = ATAN(X3/(Y1-Y3))
    B3 = ATAN((Y1-Y3)/X3)
  C
  C OPTION 1 - STRAIGHT SEGMENTS BOTH TOP AND CORNER
  C
  IF(ARCLEN(1).EQ.0.0 .AND. ARCLEN(2).EQ.0.0) THEN
    A1 = PI/2 - B1 + DELTA(1)
    A3 = PI - B3 - DELTA(2)
    A2 = PI - A1 - A3
    CHORD(1) = SIDE13*SIN(A3)/SIN(A2)
    CHORD(2) = SIDE13*SIN(A1)/SIN(A2)
  C
  C OPTION 2 -STRAIGHT SEGMENT TOP, CURVED SEGMENT CORNER
  C
  ELSE IF(ARCLEN(1).EQ.0.0 .AND. ARCLEN(2).GT.0.0) THEN
    A1 = PI/2 - B1 + DELTA(1)
    A2 = ASIN(SIN(A1)*SIDE13/CHORD(2))
    IF(A2.LT.PI/2) A2 = PI - A2
    A3 = PI - A1 - A2
    CHORD(1) = SIDE13*SIN(A3)/SIN(A2)
  C
  C OPTION 3 - CURVED SEGMENT TOP, STRAIGHT SEGMENT CORNER
  C
  ELSE IF(ARCLEN(1).GT.0.0 .AND. ARCLEN(2).EQ.0.0) THEN
    A3 = PI - B3 - DELTA(2)
    A2 = ASIN(SIN(A3)*SIDE13/CHORD(1))
    IF(A2.LT.PI/2) A2 = PI - A2
    A1 = PI - A2 - A3
    CHORD(2) = SIDE13*SIN(A1)/SIN(A2)
  C
  C OPTION 4 - CURVED SEGMENTS BOTH TOP AND CORNER
  C
  ELSE IF(ARCLEN(1).GT.0.0 .AND. ARCLEN(2).GT.0.0) THEN
    C
    C CHECK THAT INPUT GEOMETRY IS WITHIN BOUNDS
    C
    CKANG1 = (CHORD(1)**2+SIDE13**2-CHORD(2)**2)/
      X (2*CHORD(1)*SIDE13)
    CKANG3 = (CHORD(2)**2+SIDE13**2-CHORD(1)**2)/
      X (2*CHORD(2)*SIDE13)
    CKLIM = COS(PI/2-B1)
    IF(CKANG1.GE.1.0 .OR. CKANG1.LT.CKLIM) GO TO 600
    A1 = ACOS(CKANG1)
    IF(CKANG3.GE.1.0 .OR. CKANG3.LT.0.0) GO TO 600
    A3 = ACOS(CKANG3)
    A2 = PI - A1 - A3
  END IF
  X2 = X3 - CHORD(2)*COS(B3+A3)
  Y2 = Y3 + CHORD(2)*SIN(B3+A3)
  C
  C FIND CENTER ARC COORDINATES FOR CURVED SEGMENTS
  C
  IF(ARCLEN(1).GT.0.0) THEN
    XCEN(1) = X1 + ABS(R(1))*COS(A1+B1+THETA(1)/2-PI)
    YCEN(1) = Y1 + ABS(R(1))*SIN(A1+B1+THETA(1)/2-PI)

```

```

CANA0447
CANA0448
CANA0449
CANA0450
CANA0451
CANA0452
CANA0453
CANA0454
CANA0455
CANA0456
CANA0457
CANA0458
CANA0459
CANA0460
CANA0461
CANA0462
CANA0463
CANA0464
CANA0465
CANA0466
CANA0467
CANA0468
CANA0469
CANA0470
CANA0471
CANA0472
CANA0473
CANA0474
CANA0475
CANA0476
CANA0477
CANA0478
CANA0479
CANA0480
CANA0481
CANA0482
CANA0483
CANA0484
CANA0485
CANA0486
CANA0487
CANA0488
CANA0489
CANA0490
CANA0491
CANA0492
CANA0493
CANA0494
CANA0495
CANA0496
CANA0497
CANA0498
CANA0499
CANA0500
CANA0501
CANA0502
CANA0503
CANA0504
CANA0505
CANA0506
CANA0507
CANA0508
CANA0509

END IF
IF(ARCLEN(2).GT.0.0) THEN
  XCEN(2) = X3 + ABS(R(2))*COS(-A3-B3-THETA(2)/2-PI/2)
  YCEN(2) = Y3 + ABS(R(2))*SIN(-A3-B3-THETA(2)/2-PI/2)
END IF
IF(ARCLEN(3).GT.0.0) THEN
  XCEN(3) = X4 - R(3)*COS(THETA(4))
  YCEN(3) = Y4 - R(3)*SIN(THETA(4))
END IF
C
C SET INDEX CONSTANTS FOR TWO OR THREE SEGMENT PROBLEM
C
IF(SDRISE.EQ.0.0) THEN
  IF(NTN.EQ.0) NTN = 13
  NCN = 19
  NSN = 0
  NDEV(1) = NTN-1
  NDEV(2) = NCN-NTN+1
  NDEV(3) = 0
  ELSE
  IF(NTN.EQ.0) NTN = 10
  IF(NCN.EQ.0) NCN = 16
  NSN = 19
  NDEV(1) = NTN-1
  NDEV(2) = NCN-NTN
  NDEV(3) = NSN-NCN+1
  END IF
C
C FIND BEAM COORDINATES - STRAIGHT TOP SEGMENT
C
ANG(1) = PI/2
XPN(1) = X1
YPN(1) = Y1
IF(ARCLEN(1).EQ.0.0) THEN
  DO 200,J=2,NTN
    ANG(J) = DELTA(1) + PI/2
    XPN(J) = X1 + CHORD(1)*COS(DELTA(1))*(J-1)/NTN-1
    YPN(J) = Y1 + CHORD(1)*SIN(DELTA(1))*(J-1)/NTN-1
  200 CONTINUE
C
C FIND BEAM COORDINATES - CURVED TOP SEGMENT
C
ELSE IF(ARCLEN(1).GT.0.0) THEN
  DO 210,J=2,NTN
    IF((X1-XCEN(1)).EQ.0.0) THEN
      ANG(J) = PI/2 - THETA(1)*(J-1)/(NTN-1)
    ELSE
      ANG(J) = ATAN(ABS((Y1-YCEN(1))/(X1-XCEN(1)))) -
        THETA(1)*(J-1)/(NTN-1)
    X
    ENDIF
    XPN(J) = XCEN(1) + ABS(R(1))*COS(ANG(J))
    YPN(J) = YCEN(1) + ABS(R(1))*SIN(ANG(J))
  210 CONTINUE
END IF
C
C FIND BEAM COORDINATES - STRAIGHT CORNER SEGMENT
C
IF(ARCLEN(2).EQ.0.0) THEN
  DO 220,J=NTN+1,NCN
    ANG(J) = DELTA(2) - PI/2
    XPN(J) = X2 - CHORD(2)*COS(DELTA(2))*(J-NTN)/
      (NCN-NTN+1)
    YPN(J) = Y2 - CHORD(2)*SIN(DELTA(2))*(J-NTN)/

```

```

CANA0510
CANA0511
CANA0512
CANA0513
CANA0514
CANA0515
CANA0516
CANA0517
CANA0518
CANA0519
CANA0520
CANA0521
CANA0522
CANA0523
CANA0524
CANA0525
CANA0526
CANA0527
CANA0528
CANA0529
CANA0530
CANA0531
CANA0532
CANA0533
CANA0534
CANA0535
CANA0536
CANA0537
CANA0538
CANA0539
CANA0540
CANA0541
CANA0542
CANA0543
CANA0544
CANA0545
CANA0546
CANA0547
CANA0548
CANA0549
CANA0550
CANA0551
CANA0552
CANA0553
CANA0554
CANA0555
CANA0556
CANA0557
CANA0558
CANA0559
CANA0560
CANA0561
CANA0562
CANA0563
CANA0564
CANA0565
CANA0566
CANA0567
CANA0568
CANA0569
CANA0570
CANA0571
CANA0572

```

```

      X      (NCN-NTN+1)
220      CONTINUE
C      FIND BEAM COORDINATES - CURVED CORNER SEGMENT
C      ELSE IF(ARCLN(2).GT.0.0) THEN
      DO 230,J=NTN+1,NCN
      ANG(J) = ATAN(ABS((Y2-YCN(2))/(X2-XCN(2)))) -
      THETA(2)*(J-NTN)/(NCN-NTN)
      XPN(J) = XCEN(2) + ABS(R(2))*COS(ANG(J))
      YPN(J) = YCN(2) + ABS(R(2))*SIN(ANG(J))
230      CONTINUE
      END IF
C      FIND BEAM COORDINATES - STRAIGHT SIDE SEGMENT
C      IF(ARCLN(3).EQ.0.0 .AND. SDRISE.GT.0.0) THEN
      DO 240,J=NCN+1,NSN
      ANG(J) = DELTA(3) - PI/2
      XPN(J) = X3 - CHORD(3)*COS(DELTA(3))*(J-NCN)/
      (NSN-NCN+1)
      YPN(J) = Y3 - CHORD(3)*SIN(DELTA(3))*(J-NCN)/
      (NSN-NCN+1)
240      CONTINUE
C      FIND BEAM COORDINATES - CURVED SIDE SEGMENT
C      ELSE IF(ARCLN(3).GT.0.0 .AND. SDRISE.GT.0.0) THEN
      DO 250,J=NCN+1,NSN
      ANG(J) = ATAN(ABS((Y3-YCN(3))/(X3-XCN(3)))) -
      THETA(3)*(J-NCN)/(NSN-NCN+1)
      XPN(J) = XCEN(3) + ABS(R(3))*COS(ANG(J))
      YPN(J) = YCN(3) + ABS(R(3))*SIN(ANG(J))
250      CONTINUE
      END IF
C      ***** END SOLUTION OF BEAM ELEMENT GEOMETRY *****
C      ASSIGN COORDINATES TO KEY NODAL POINTS
300 X(198) = XPN(1)
      Y(198) = YPN(1)
      X(194) = XPN(2)
      Y(194) = YPN(2)
      X(190) = XPN(3)
      Y(190) = YPN(3)
      X(167) = XPN(4)
      Y(167) = YPN(4)
      X(164) = XPN(5)
      Y(164) = YPN(5)
      X(161) = XPN(6)
      Y(161) = YPN(6)
      X(158) = XPN(7)
      Y(158) = YPN(7)
      X(138) = XPN(8)
      Y(138) = YPN(8)
      X(136) = XPN(9)
      Y(136) = YPN(9)
      X(134) = XPN(10)
      Y(134) = YPN(10)
      X(132) = XPN(11)
      Y(132) = YPN(11)
      X(130) = XPN(12)

```

```

CANA0573      Y(130) = YPN(12)
CANA0574      X(128) = XPN(13)
CANA0575      Y(128) = YPN(13)
CANA0576      X(118) = XPN(14)
CANA0577      Y(118) = YPN(14)
CANA0578      X(115) = XPN(15)
CANA0579      Y(115) = YPN(15)
CANA0580      X(109) = XPN(16)
CANA0581      Y(109) = YPN(16)
CANA0582      X(103) = XPN(17)
CANA0583      Y(103) = YPN(17)
CANA0584      X(97) = XPN(18)
CANA0585      Y(97) = YPN(18)
CANA0586      X(91) = XPN(19)
CANA0587      Y(91) = YPN(19)
CANA0588
CANA0589      X(15) = 6*HFSPAN
CANA0590      X(8) = HFSPAN + HFSPAN/4 + RISE/4
CANA0591      X(6) = HFSPAN + FTWIDO
CANA0592      X(5) = HFSPAN
CANA0593      X(4) = HFSPAN - FTWIDI
CANA0594      X(1) = 0.0
CANA0595      Y(1) = -1.5*RISE
CANA0596      Y(75) = -FTDEP
CANA0597      Y(90) = 0.0
CANA0598      X(237) = X(8)
CANA0599      X(258) = X(8)
CANA0600      X(279) = X(8)
CANA0601      X(300) = X(8)
CANA0602      XINC = (X(15)-X(8))/(1+1.26+1.26**2+1.26**3+1.26**4+1.26**5+
CANA0603      1.26**6)
CANA0604      X(121) = X(8) + XINC
CANA0605      X(151) = X(121)
CANA0606      X(183) = X(121)
CANA0607      X(217) = X(121)
CANA0608
CANA0609
CANA0610
CANA0611      IF(KCOVER.EQ.0) Y(244) = RISE + HTCVR
CANA0612      IF(KCOVER.EQ.1) THEN
CANA0613          YINC = HTCVR/(1 + SPF + SPF**2 + SPF**3)
CANA0614          Y(244) = RISE + YINC + SPF*YINC + (SPF**2)*YINC
CANA0615          Y(245) = RISE + HTCVR
CANA0616      ELSE IF(KCOVER.EQ.2) THEN
CANA0617          YINC = HTCVR/(1 + SPF + SPF**2 + SPF**3 + SPF**4)
CANA0618          Y(244) = RISE + YINC + SPF*YINC + (SPF**2)*YINC
CANA0619          Y(245) = Y(244) + (SPF**3)*YINC
CANA0620          Y(266) = RISE + HTCVR
CANA0621      ELSE IF(KCOVER.EQ.3) THEN
CANA0622          IF(HTCVR.GT.1.5*RISE) THEN
CANA0623              YINC = 1.5*RISE/(1+SPF+SPF**2+SPF**3+SPF**4+SPF**5)
CANA0624              Y(244) = RISE + YINC + SPF*YINC + (SPF**2)*YINC
CANA0625              Y(245) = Y(244) + (SPF**3)*YINC
CANA0626              Y(266) = Y(245) + (SPF**4)*YINC
CANA0627              Y(287) = RISE + 1.5*RISE
CANA0628              HPRESS = HTCVR - 1.5*RISE
CANA0629              INCADD = NINC - INCLEV
CANA0630              IF(INCADD.LE.0) INCADD = -1
CANA0631              NBPTC = NBPTC + IABS(INCADD)*21
CANA0632              PINC = -(HPRESS*DENSTY)/(1728*IABS(INCADD))
CANA0633          ELSE
CANA0634              YINC = HTCVR/(1+SPF+SPF**2+SPF**3+SPF**4+SPF**5)
CANA0635              Y(244) = RISE + YINC + SPF*YINC + (SPF**2)*YINC
CANA0636              Y(245) = Y(244) + (SPF**3)*YINC
CANA0637              Y(266) = Y(245) + (SPF**4)*YINC
CANA0638
CANA0639
CANA0640
CANA0641
CANA0642
CANA0643
CANA0644
CANA0645
CANA0646
CANA0647
CANA0648
CANA0649
CANA0650
CANA0651
CANA0652
CANA0653
CANA0654
CANA0655
CANA0656
CANA0657
CANA0658
CANA0659
CANA0660
CANA0661
CANA0662
CANA0663
CANA0664
CANA0665
CANA0666
CANA0667
CANA0668
CANA0669
CANA0670
CANA0671
CANA0672
CANA0673
CANA0674
CANA0675
CANA0676
CANA0677
CANA0678
CANA0679
CANA0680
CANA0681
CANA0682
CANA0683
CANA0684
CANA0685
CANA0686
CANA0687
CANA0688
CANA0689
CANA0690
CANA0691
CANA0692
CANA0693
CANA0694
CANA0695
CANA0696
CANA0697
CANA0698

```

```

      Y(287) = RISE + HTCVR
      END IF
END IF

C
C
C
REVISE MESH FOR TRENCH CONDITION

IF(WORD.NE.TPMESH(2)) GO TO 400
X(8) = HFSPAN + TRNWID
X(237) = X(8) + Y(244)*TRNSLP
YDIV = Y(244)/5
XINC = (X(15)-X(8))/(1+1.26+1.26**2+1.26**3+1.26**4+1.26**5+
X      1.26**6)
XINC2 = (X(15)-X(237))/(1+1.26+1.26**2+1.26**3+1.26**4+1.26**5+
X      1.26**6)
X(121) = X(8) + XINC + YDIV*(TRNSLP-(XINC-XINC2)/Y(244))
X(151) = X(8) + XINC + 2*YDIV*(TRNSLP-(XINC-XINC2)/Y(244))
X(183) = X(8) + XINC + 3*YDIV*(TRNSLP-(XINC-XINC2)/Y(244))
X(217) = X(8) + XINC + 4*YDIV*(TRNSLP-(XINC-XINC2)/Y(244))
IF(TRNDEP.LE.YDIV) THEN
  DO 310,I=27,29
    LNOD(6,I) = 1
    LNOD(7,I) = 1
    TRNDEP = YDIV
  ELSE IF(TRNDEP.GT.YDIV .AND. TRNDEP.LE.2*YDIV) THEN
    DO 320,I=27,32
      LNOD(6,I) = 1
      LNOD(7,I) = 1
      TRNDEP = 2*YDIV
    ELSE IF(TRNDEP.GT.2*YDIV .AND. TRNDEP.LE.3*YDIV) THEN
      DO 330,I=27,36
        LNOD(6,I) = 1
        LNOD(7,I) = 1
        TRNDEP = 3*YDIV
      ELSE IF(TRNDEP.GT.3*YDIV .AND. TRNDEP.LE.4*YDIV) THEN
        DO 340,I=27,37
          LNOD(6,I) = 1
          LNOD(7,I) = 1
          TRNDEP = 4*YDIV
        ELSE
          DO 350,I=27,38
            LNOD(6,I) = 1
            LNOD(7,I) = 1
            TRNDEP = 5*YDIV
          END IF
        IF(KCOVER.EQ.1) THEN
          IF(TRNDEP.LE.Y(244)) THEN
            X(258) = X(237)
          ELSE
            LNOD(6,92) = 1
            LNOD(7,92) = 1
            TRNDEP = Y(245)
            X(258) = X(237) + TRNSLP*(Y(245)-Y(244))
          END IF
        ELSE IF(KCOVER.EQ.2) THEN
          IF(TRNDEP.LE.Y(244)) THEN
            X(258) = X(237)
            X(279) = X(237)
          ELSE IF(TRNDEP.GT.Y(244) .AND. TRNDEP.LE.Y(245)) THEN
            LNOD(6,92) = 1
            LNOD(7,92) = 1
            TRNDEP = Y(245)
            X(258) = X(237) + TRNSLP*(Y(245)-Y(244))
            X(279) = X(258)
          END IF
        END IF
      END IF
    END IF
  END IF

```

```

CANA0699
CANA0700
CANA0701
CANA0702
CANA0703
CANA0704
CANA0705
CANA0706
CANA0707
CANA0708
CANA0709
CANA0710
CANA0711
CANA0712
CANA0713
CANA0714
CANA0715
CANA0716
CANA0717
CANA0718
CANA0719
CANA0720
CANA0721
CANA0722
CANA0723
CANA0724
CANA0725
CANA0726
CANA0727
CANA0728
CANA0729
CANA0730
CANA0731
CANA0732
CANA0733
CANA0734
CANA0735
CANA0736
CANA0737
CANA0738
CANA0739
CANA0740
CANA0741
CANA0742
CANA0743
CANA0744
CANA0745
CANA0746
CANA0747
CANA0748
CANA0749
CANA0750
CANA0751
CANA0752
CANA0753
CANA0754
CANA0755
CANA0756
CANA0757
CANA0758
CANA0759
CANA0760
CANA0761

ELSE
  DO 360,I=92,93
    LNOD(6,I) = 1
    LNOD(7,I) = 1
    TRNDEP = Y(266)
    X(258) = X(237) + TRNSLP*(Y(245)-Y(244))
    X(279) = X(258) + TRNSLP*(Y(266)-Y(245))
  END IF
  ELSE IF(KCOVER.EQ.3) THEN
    IF(TRNDEP.LE.Y(244)) THEN
      X(258) = X(237)
      X(279) = X(237)
      X(300) = X(237)
    ELSE IF(TRNDEP.GT.Y(244) .AND. TRNDEP.LE.Y(245)) THEN
      LNOD(6,92) = 1
      LNOD(7,92) = 1
      TRNDEP = Y(245)
      X(258) = X(237) + TRNSLP*(Y(245)-Y(244))
      X(279) = X(258)
      X(300) = X(258)
    ELSE IF(TRNDEP.GT.Y(245) .AND. TRNDEP.LE.Y(266)) THEN
      DO 370,I=92,93
        LNOD(6,I) = 1
        LNOD(7,I) = 1
        TRNDEP = Y(266)
        X(258) = X(237) + TRNSLP*(Y(245)-Y(244))
        X(279) = X(258) + TRNSLP*(Y(266)-Y(245))
        X(300) = X(279)
      ELSE
        DO 380,I=92,94
          LNOD(6,I) = 1
          LNOD(7,I) = 1
          TRNDEP = Y(287)
          X(258) = X(237) + TRNSLP*(Y(245)-Y(244))
          X(279) = X(258) + TRNSLP*(Y(266)-Y(245))
          X(300) = X(279) + TRNSLP*(Y(287)-Y(266))
        END IF
      END IF
    END IF
  REVISE MESH FOR HOMOGENEOUS MATERIAL
  IF(WORD.NE.TPMESH(3)) GO TO 440
  DO 410,I=20,72+JLNODA
    LNOD(6,I) = 1
  END IF
  RECORD MESH DOCUMENTATION PARAMETERS
  440 WRITE(6,4000) WORD
  IF(WORD.EQ.TPMESH(2)) WRITE(6,4010) TRNDEP/12,TRNWID/12,TRNSLP
  WRITE(6,4020) SPCFAC,HTCOVER/12,(HTCOVER-HPRESS)/12,DENSITY,(3*KCOVER)
  WRITE(6,4100) RISE,HFSPAN,SDRISE,FTDEP,FTWID,FTWIDI
  IF(SDRISE.NE.0) THEN
    WRITE(6,4110)
    NSEG = 3
  ELSE
    WRITE(6,4120)
    NSEG = 2
  END IF
  DO 450 N=1,NSEG
    IF(R(N).NE.0.0) THEN
      IF(N.EQ.1) WRITE(6,4200)
      IF(N.EQ.2) WRITE(6,4210)
      IF(N.EQ.3) WRITE(6,4220)
    END IF
  END DO

```

```

CANA0762
CANA0763
CANA0764
CANA0765
CANA0766
CANA0767
CANA0768
CANA0769
CANA0770
CANA0771
CANA0772
CANA0773
CANA0774
CANA0775
CANA0776
CANA0777
CANA0778
CANA0779
CANA0780
CANA0781
CANA0782
CANA0783
CANA0784
CANA0785
CANA0786
CANA0787
CANA0788
CANA0789
CANA0790
CANA0791
CANA0792
CANA0793
CANA0794
CANA0795
CANA0796
CANA0797
CANA0798
CANA0799
CANA0800
CANA0801
CANA0802
CANA0803
CANA0804
CANA0805
CANA0806
CANA0807
CANA0808
CANA0809
CANA0810
CANA0811
CANA0812
CANA0813
CANA0814
CANA0815
CANA0816
CANA0817
CANA0818
CANA0819
CANA0820
CANA0821
CANA0822
CANA0823
CANA0824

```

```

      WRITE(6,4300) R(N),THETA(N)*180/PI,ARCLN(N),NDEV(N)
      IF(N.EQ.3) WRITE(6,4310) THETA(4)*180/PI
      ELSE
      IF(N.EQ.1) WRITE(6,4230)
      IF(N.EQ.2) WRITE(6,4240)
      IF(N.EQ.3) WRITE(6,4250)
      WRITE(6,4320) DELTA(N)*180/PI,CHORD(N),NDEV(N)
      END IF
450    CONTINUE
C
C    SAVE DATA FOR MAIN CONTROL CARDS
C
500  WRITE(LUDATA,6105) PREP,TITLE,WORD1,NINC,MGENPR,NPUTCK,IPLLOT,IWRT
      1      ,NUMNP,NUMEL,NBPTC
C
C    SAVE DATA NEEDED FOR NODE GENERATION
C
      LIMIT = IBLANK
      K = 0
      DO 520,N = 1,162+JNMODA
      NNP = NODMOD(1,N)
      SPACNG = 0.0
      XCOORD = X(NNP)
      YCOORD = Y(NNP)
      IF (NODMOD(2,N) .EQ. 5) GO TO 510
      IF (K .LT. 111+JXYA) K = K + 1
      IF (XYANDS(1,K) .NE. 0.0) XCOORD = XYANDS(1,K)
      IF (XYANDS(2,K) .NE. 0.0) YCOORD = XYANDS(2,K)
      IF (XYANDS(3,K) .NE. 0.0) SPACNG = XYANDS(3,K)
510  IF (N.EQ. 162+JNMODA) LIMIT = LIH
520  WRITE (LUDATA,6110) LIMIT, NNP, NODMOD(2,N), XCOORD, YCOORD,
      1      IZERO, SPACNG, FZERO
C
C    SAVE DATA NEEDED FOR ELEMENT GENERATION
C
      LIMIT = IBLANK
      DO 530,NE = 1,91+JLNODA
      INCR = LNOD(7,NE)
      IF(INCR.GT.NINC) INCR = NINC
      IF (NE.EQ.91+JLNODA) LIMIT = LIH
530  WRITE (LUDATA,6115) LIMIT, (LNOD(K,NE), K = 1,6), INCR,
      X      (LNOD(K,NE),K=8,11)
C
C    SAVE DATA NEEDED FOR INPUT OF DISPLACEMENT BOUNDARY CONDITIONS
C
      K = IONE
      L = IONE
      LIMIT = IBLANK
      DO 540,N = 1,35+JBACA
      IF (N .GT. 15) THEN
      K = IONE
      L = IZERO
      END IF
      IF (N.EQ.29) THEN
      K = IZERO
      L = IONE
      END IF
      IF (N.EQ.35+JBACA .AND. HPRESS.EQ.0) LIMIT = LIH
540  WRITE (LUDATA,6120) LIMIT, NODBC(N), K, FZERO, L, FZERO, FZERO,
      1      IONE, IZERO, IZERO, FZERO, FZERO
C
C    SAVE DATA NEEDED FOR OVERBURDEN PRESSURE BOUNDARY FORCES
C

```

```

CANA0825      IF(HPRESS.GT.0) THEN
CANA0826          LIMIT = IBLANK
CANA0827          DO 550, N=1,IABS(INCADD)
CANA0828              IF(N.EQ.IABS(INCADD)) LIMIT = LIH
CANA0829              IF(INCADD.GT.0) THEN
CANA0830                  IADD = NINC - INCADD + N
CANA0831                  ELSE
CANA0832                      IADD = NINC + INCADD + N
CANA0833                  END IF
550          WRITE (LUDATA,6120) LIMIT,287,IZERO,FZERO,IZERO,FZERO,
      X      FZERO,IADD,307,IONE,PINC,PINC
CANA0834
CANA0835      END IF
CANA0836      REWIND LUDATA
CANA0837      RETURN
CANA0838
CANA0839      C
CANA0840      C    ERROR EXIT
CANA0841      C
CANA0842      600 STOP '***FATAL ERROR IN ARCH GEOMETRY - CHECK INPUT***'
CANA0843      C
CANA0844      C    FORMATS
CANA0845      C
CANA0846      1000 FORMAT (A4,17A4,A3)
CANA0847      1005 FORMAT (4I5,5F10.0)
CANA0848      1010 FORMAT (7F10.0)
CANA0849      1020 FORMAT (7F10.0,2I5)
CANA0850      2000 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,74HTHE CONTROL WORD FOR
CANA0851          X      MESH TYPE MUST BE "EMBA", "TREN", OR "HOMO", BUT IS ",A4,1H"/1H ,
CANA0852          X      X17H(SEE USER MANUAL)/)
CANA0853      2003 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,34HCONTROL WORD1 MUST BECANA0916
CANA0854          X      MOD OR BLANK/1H ,17H(SEE USER MANUAL)/)
CANA0855      2005 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,72HNUMBER OF CONSTRUCTIOCANA0918
CANA0856          X      XN INCREMENTS "NINC" MUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE USCANA0919
CANA0857          X      XER MANUAL)/)
CANA0858      2010 FORMAT (1H0,34HSPESIFIED COVER HEIGHT TO SHALLOW./1H0,34HPROGRAM WCANA0921
CANA0859          X      XILL SET HTCVR = 0.1*RISE/)
CANA0860      2011 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,24HINVALID ENTRY FOR "RICANA0923
CANA0861          X      XSE"/)
CANA0862      2012 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,26HINVALID ENTRY FOR "HFCANA0925
CANA0863          X      XSPAN"/)
CANA0864      2013 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,26HINVALID ENTRY FOR "SDCANA0927
CANA0865          X      XRISE"/)
CANA0866      2014 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,31HINVALID ENTRY FOR RADCANA0929
CANA0867          X      XIUS "R(I)"/)
CANA0868      2015 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,52HTRENCH WIDTH "TRNWD"CANA0931
CANA0869          X      XMUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE USER MANUAL)/)
CANA0870      2020 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,52HTRENCH DEPTH "TRNDEP"CANA0933
CANA0871          X      XMUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE USER MANUAL)/)
CANA0872      2025 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,52HTRENCH SLOPE "TRNSLP"CANA0935
CANA0873          X      XMUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE USER MANUAL)/)
CANA0874      2030 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,62HOUTSIDE FOOTING WIDTHHCANA0937
CANA0875          X      X "FTWIDO" MUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE USER MANUAL)/CANA0938
CANA0876          X      X/)
CANA0877      2035 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,61HINSIDE FOOTING WIDTH
CANA0878          X      X"FTWIDI" MUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE USER MANUAL)/CANA0941
CANA0879          X      X/)
CANA0880      2040 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,53HFOOTING DEPTH "FTDEP"CANA0943
CANA0881          X      XMUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE USER MANUAL)/)
CANA0882      2042 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,55HSPACING FACTOR "SPCFAC"CANA0945
CANA0883          X      XC" MUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE USER MANUAL)/)
CANA0884      2045 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,72HNUMBER OF TOP SEGMENTCANA0947
CANA0885          X      X NODAL POINTS "NTN" MUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE USCANA0948
CANA0886          X      XER MANUAL)/)
CANA0887      2050 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,75HNUMBER OF CORNER SEGMCANA0950

```



```

DX=0.2*R1
DO 40 I=6,NX2
X(I)=X(I-1)+DX
DX=DX+0.2*R1
IF(DX.GT.R1)DX=R1
IF(XTR.GT.X(I)) NTR=I
40 CONTINUE
TRELIM=4.49*R1
IF(XTR.GT.TRELIM) NMOVE=NX2
C
C NMOVE=NX2 IMPLIES EMBANKMENT CONDITION
C
IF(XTR.GT.TRELIM) GO TO 50
C
C ADJUST TRENCH WALL LOCATION
C
C MOVE CLOSEST LINE TO COINCIDE WITH TRENCH LINE LOCATION
C
NMOVE=NTR
IF(ABS(XTR-X(NTR)) .GE. ABS(XTR-X(NTR+1))) NMOVE=NTR+1
X(NMOVE)=XTR
50 CONTINUE
C
C SET MASTER COORDINATES FOR Y
C
C NY1 ARE SPACED EVENLY ALONG VERTICAL WALL OF BOX
C
Y(6)=-R2
DY=R2/3.0
DO 60 I=7,12
Y(I)=Y(I-1)+DY
60 C
C
C SET BOTTOM NODES IN Y-DIRECTION( TO BE ADJUSTED LATER FOR BDEPTH)
C
Y(1)=-4.0*R2
DY=2.0*R2/3.0
DO 90 I=2,5
Y(I)=Y(I-1)+DY
90 C
C
C SET Y-COORDINATES FOR TOP NODES UP TO SPECIFIED COVER HEIGHT
C
C IF COVER HEIGHT IS GREATER THAN 4R2, TRUNCATE AT 4R2 AND APPLY
C
C REMAINING SOIL WITH EQUIVALENT PRESSURE LOAD IN BOUNDARY CONDITION
C
NHC=12
IF(HTCOVER.LT.0.0) GO TO 95
GO TO 96
95 WRITE(6,2070)HTCOVER
STOP
C
C YCOVER=MAXIMUM Y COORDINATE
C
C NHC=MAXIMUM NUMBER OF VERTICAL NODES IF HTCOVER.GE.4R2
C
C IUP=STARTING VERTICAL NUMBER OF ROWS ON PIPE TOP,ASCENDING
C
C IDN=STARTING VERTICAL NUMBER OF ROWS ON PIPE BOTTOM,DESCENDING
C
96 YCOVER=HTCOVER+R2
NHC=17
IUP=12
IDN=6
DO 100 I=1,5
Y(IUP+I)=-Y(IDN-I)
IF(Y(IUP+I).GT.YCOVER) NHC=IUP+I
IF(Y(IDN-I).GT.YCOVER) GO TO 110
100 CONTINUE
GO TO 120

```

```

CANB0079 C
CANB0080 C RESET Y-COORDINATE TO MATCH " YCOVER "
CANB0081 C
CANB0082 110 DYTOP=(Y(NHC)-YCOVER)*0.5
CANB0083 DYBOT=YCOVER-Y(NHC-1)
CANB0084 IF(NHC.LE.14) GO TO 115
CANB0085 IF(DYTOP.GT.DYBOT) NHC=NHC-1
CANB0086 Y(NHC)=YCOVER
CANB0087 GO TO 120
CANB0088 115 NHC=14
CANB0089 Y(NHC)=YCOVER
CANB0090 Y(NHC-1)=(Y(NHC)+Y(NHC-2))*0.5
CANB0091 120 CONTINUE
CANB0092 C
CANB0093 C ADJUST BOTTOM Y-COORDINATE Y(5) TO BE LOCATED AT BOTTOM OF BEDDING
CANB0094 C
CANB0095 IF(BDEPTH.LE.0.0) GO TO 140
CANB0096 C
CANB0097 C LIMITS FOR MAXIMUM AND MINIMUM BEDDING THICKNESS
CANB0098 C
CANB0099 BDEMIN=0.1*R2
CANB0100 BDEMAX=0.667*R2
CANB0101 IF(BDEPTH.LT.BDEMIN) BDEPTH=BDEMIN
CANB0102 IF(BDEPTH.GT.BDEMAX) BDEPTH=BDEMAX
CANB0103 Y(5)=Y(6)-BDEPTH
CANB0104 GO TO 130
CANB0105 140 WRITE(6,2020)
CANB0106 130 CONTINUE
CANB0107 C
CANB0108 C COMPUTE THE NUMBER OF NODES,NUMBER OF ELEMENTS AND ESTIMATE
CANB0109 C
CANB0110 C MAXIMUM NUMBER OF BOUNDARY CONDITIONS(FOR STORAGE SPACE ONLY)
CANB0111 C
CANB0112 NUMNP=11*NHC-5*4
CANB0113 NUMEL=10*(NHC-1)-6*4+14
CANB0114 NBPTC=200
CANB0115 NPUTCK=0
CANB0116 NPMAT=14
CANB0117 C
CANB0118 C WRITE MASTER CONTROL DATA TO UNIT LUDATA TO SIMULATE LEVEL 3 INPUT
CANB0119 C
CANB0120 WRITE(LUDATA,2030)PREP,TITLE,WORD2,NINC,MGENPR,NPUTCK,IPL0T,IWRT,
CANB0121 1 NUMNP,NUMEL,NBPTC
CANB0122 C
CANB0123 C WRITE NODAL POINT DATA TO UNIT LUDATA FOR LEVEL 3 INPUT
CANB0124 C
CANB0125 LIMIT=IBLANK
CANB0126 NODE=0
CANB0127 C
CANB0128 C VERTICAL NODE COUNTER
CANB0129 C
CANB0130 DO 150 J=1,NHC
CANB0131 ISTART=1
CANB0132 IF(J.GT.6 .AND. J.LT.12) ISTART=5
CANB0133 DO 150 I=ISTART,11
CANB0134 NODE=NODE+1
CANB0135 IF(NODE.EQ.NUMNP) LIMIT=L1H
CANB0136 150 WRITE(LUDATA,2040)LIMIT,NODE,IZERO,X(1),Y(J),IZERO,FZERO,FZERO
CANB0137 C
CANB0138 C GENERATE ELEMENT DATA
CANB0139 C
CANB0140 C GET PIPE ELEMENT CONNECTIVITY
CANB0141 C
LIMIT=IBLANK
CANB0142
CANB0143
CANB0144
CANB0145
CANB0146
CANB0147
CANB0148
CANB0149
CANB0150
CANB0151
CANB0152
CANB0153
CANB0154
CANB0155
CANB0156
CANB0157
CANB0158
CANB0159
CANB0160
CANB0161
CANB0162
CANB0163
CANB0164
CANB0165
CANB0166
CANB0167
CANB0168
CANB0169
CANB0170
CANB0171
CANB0172
CANB0173
CANB0174
CANB0175
CANB0176
CANB0177
CANB0178
CANB0179
CANB0180
CANB0181
CANB0182
CANB0183
CANB0184
CANB0185
CANB0186
CANB0187
CANB0188
CANB0189
CANB0190
CANB0191
CANB0192
CANB0193
CANB0194
CANB0195
CANB0196
CANB0197
CANB0198
CANB0199
CANB0200
CANB0201
CANB0202
CANB0203
CANB0204

```



```

NB1=66
NB2=101
IXB=NB2+1
DO 160 I=1,NPMAT
IF(I.GT.4) GO TO 170
IXA=IXB
IXB=IXA+1
GO TO 160
170 IF(I.GT.10) GO TO 180
IXA=IXB
IXB=IXA-7
IF(I.EQ.5) IXB=95
GO TO 160
180 IXA=IXB
IXB=IXA-1
160 WRITE(LUDATA,2050) LIMIT,I,IXB,IXA,IZERO,IZERO,I,IONE,IZERO,IZERO,
1 IZERO,IZERO
C
C NEXT GENERATE SOIL BLOCK # 1 (BOTTOM)
C
INC=1
INTRAL=0
NUMLAY=0
INTERL=0
DO 190 K=1,4
MAT=1
IF(K.EQ.2 .AND. BDEPTH.GT.0.0) MAT=2
IF(K.NE.4) GO TO 190
INTRAL=1
NUMLAY=5
INTERL=11
190 WRITE(LUDATA,2050) LIMIT,(LNOD1(J,K),J=1,5),MAT,INC,IZERO,INTRAL,
1 NUMLAY,INTERL
C
C GENERATE SOIL BLOCK # 2 (SIDE)
C
DO 200 I=1,6
INTRAL=0
NUMLAY=0
INTERL=0
MAT=LNOD2(5,I)
INC=LNOD2(7,I)
IF(NINC.LT.0) INC=1
IF(I.EQ.6) GO TO 210
GO TO 220
210 INTRAL=1
NUMLAY=5
INTERL=7
IF(NMOVE.EQ.11) INC=4
IF(NMOVE.EQ.11) MAT=3
IF(NINC.LT.0) INC=1
220 WRITE(LUDATA,2050) LIMIT,(LNOD2(J,I),J=1,5),MAT,INC,IZERO,INTRAL,
1 NUMLAY,INTERL
IF(I.EQ.6) GO TO 200
IF(NMOVE.EQ.11) GO TO 200
NELI=LNOD2(1,I)+NMOVE-5
MAT=1
INC=1
IF(NELI.EQ.94) GO TO 200
WRITE(LUDATA,2050) LIMIT,NELI,IZERO,IZERO,IZERO,IZERO,MAT,INC,
1 IZERO,INTRAL,NUMLAY,INTERL
200 CONTINUE
C

```

```

CANB0205 C GENERATE SOIL BLOCK # 3 (SIDE)
CANB0206 C
CANB0207 DO 250 I=1,2
CANB0208 INTRAL=0
CANB0209 NUMLAY=0
CANB0210 INTERL=0
CANB0211 MAT=LNOD3(6,I)
CANB0212 INC=LNOD3(7,I)
CANB0213 IF(NINC.LT.0) INC=1
CANB0214 IF(I.EQ.1) GO TO 240
CANB0215 INTRAL=1
CANB0216 NUMLAY=1
CANB0217 IF(NMOVE.EQ.11) INC=4
CANB0218 IF(NINC.LT.0) INC=1
CANB0219 IF(NMOVE.EQ.11) MAT=3
CANB0220 IF(HTCOVR.LE.0.0) LIMIT=L1H
240 WRITE(LUDATA,2050) LIMIT,(LNOD3(J,I),J=1,5),MAT,INC,IZERO,INTRAL,
1 NUMLAY,INTERL
IF(I.EQ.2) GO TO 250
IF(NMOVE.EQ.11) GO TO 250
NELI=LNOD3(1,I)+NMOVE-5
MAT=1
INC=1
IF(NELI.EQ.100) GO TO 250
WRITE(LUDATA,2050) LIMIT,NELI,IZERO,IZERO,IZERO,IZERO,MAT,INC,
1 IZERO,INTRAL,NUMLAY,INTERL
250 CONTINUE
IF(LIMIT.EQ.L1H) GO TO 400
C
C GENERATE SOIL BLOCK # 4 (TOP)
C
LASTEL=(NHC-12)*10+100
DO 260 I=1,6
INTRAL=0
NUMLAY=0
INTERL=0
MAT=LNOD4(6,I)
INC=LNOD4(7,I)
IF(NINC.LT.0) INC=1
IF(I.EQ.1) GO TO 270
IF(LNOD4(1,I).GE.LASTEL) GO TO 280
GO TO 270
280 IF(I.EQ.6) GO TO 290
GO TO 300
290 INTRAL=1
NUMLAY=5
INTERL=11
IF(NMOVE.EQ.11) INC=9
IF(NINC.LT.0) INC=1
IF(NMOVE.EQ.11) MAT=3
LIMIT=L1H
270 WRITE(LUDATA,2050) LIMIT,(LNOD4(J,I),J=1,5),MAT,INC,IZERO,INTRAL,
1 NUMLAY,INTERL
IF(LIMIT.EQ.L1H) GO TO 400
GO TO 310
300 LIMIT=L1H
NODEI=(NHC-12)*11+100
NODEJ=NODEI+1
NODEK=NODEJ+11
NODEL=NODEK-1
MAT=1
INC=1
INTRAL=1

```

```

CANB0268
CANB0269
CANB0270
CANB0271
CANB0272
CANB0273
CANB0274
CANB0275
CANB0276
CANB0277
CANB0278
CANB0279
CANB0280
CANB0281
CANB0282
CANB0283
CANB0284
CANB0285
CANB0286
CANB0287
CANB0288
CANB0289
CANB0290
CANB0291
CANB0292
CANB0293
CANB0294
CANB0295
CANB0296
CANB0297
CANB0298
CANB0299
CANB0300
CANB0301
CANB0302
CANB0303
CANB0304
CANB0305
CANB0306
CANB0307
CANB0308
CANB0309
CANB0310
CANB0311
CANB0312
CANB0313
CANB0314
CANB0315
CANB0316
CANB0317
CANB0318
CANB0319
CANB0320
CANB0321
CANB0322
CANB0323
CANB0324
CANB0325
CANB0326
CANB0327
CANB0328
CANB0329
CANB0330

```

```

NUMLAY=NHC-12
INTERL=11
IF(NMOVE.EQ.11) INC=NHC-8
IF(NINC.LT.0) INC=1
IF(NMOVE.EQ.11) MAT=3
WRITE(LUDATA,2050) LIMIT, LASTEL, NODEI, NODEJ, NODEK, NODEL, MAT, INC,
1 IZERO, INTRAL, NUMLAY, INTERL
GO TO 400
310 IF(I.EQ.6) GO TO 260
IF(NMOVE.EQ.11) GO TO 260
NELI=LNODE4(1,I)+NMOVE-1
MAT=1
INC=1
IF(NELI.EQ.LASTEL) GO TO 260
WRITE(LUDATA,2050) LIMIT, NELI, IZERO, IZERO, IZERO, IZERO, MAT, INC,
1 IZERO, INTRAL, NUMLAY, INTERL
260 CONTINUE
C
C ALL CONNECTIVITY IS DONE
C
C BOUNDARY CONDITION GENERATION
C OVERBURDEN PRESSURE INTO INCREMENTS
C
400 LIMIT=IBLANK
COVLIM=3.0*R2
IF(HTCOVER.LE.COVLIM) GO TO 440
OVERP=(HTCOVER-3.0*R2)*DENSIT
WRITE(6,2110) OVERP
IF(NINC.LE.9) GO TO 420
OVERP=OVERP/FLOAT(NINC-9)
IDINC=NINC-9
DO 430 I=1, IDINC
NFIRST=157
NLASt=167
NEWINC=9+I
PLOAD=-OVERP
WRITE(LUDATA,2060) LIMIT, NFIRST, IZERO, FZERO, IZERO, FZERO, FZERO,
1 NEWINC, NLASt, IONE, PLOAD, PLOAD
430 CONTINUE
GO TO 440
420 IF(NINC.GT.0) GO TO 440
NFIRST=157
NLASt=167
NEWINC=1
PLOAD=-OVERP
WRITE(LUDATA,2060) LIMIT, NFIRST, IZERO, FZERO, IZERO, FZERO, FZERO,
1 NEWINC, NLASt, IONE, PLOAD, PLOAD
C
C BOUNDARY CONDITIONS (DISPLACEMENTS)
C
440 LASTNO=(NHC-12)*11+112
K=IONE
DO 450 I=1,39
IF(I.GT.11) K=IZERO
IF(NODBC(I).EQ.LASTNO) LIMIT=L1H
WRITE(LUDATA,2060) LIMIT, NODBC(I), IONE, FZERO, K, FZERO, FZERO, IONE,
1 IZERO, IZERO, FZERO, FZERO
IF(LIMIT.EQ.L1H) GO TO 460
450 CONTINUE
C
C PRINT MATERIAL NUMBERS
C
460 WRITE(6,2080)

```

```

CANB0331 DO 470 K=1,3
CANB0332 NUMMAT=K
CANB0333 K1=2*K-1
CANB0334 K2=K1+1
CANB0335 470 WRITE(6,2090)MATTYP(K1),MATTYP(K2),NUMMAT
CANB0336 REWIND LUDATA
CANB0337 RETURN
CANB0338 C
CANB0339 C FORMATS
CANB0340 C
CANB0341 1000 FORMAT(A4,17A4,A4)
CANB0342 1010 FORMAT (#I5,6F10.0)
CANB0343 2000 FORMAT(1H0,' FATAL DATA ERROR. THE CONTROL WORD MUST BE * EMBA OR
CANB0344 1 TREN *, BUT IS ',A4,' * ')
CANB0345 2010 FORMAT(1H1,' * * BEGIN GENERATION OF CANNED MESH * *')
CANB0346 1 ' THE DATA TO BE RUN IS ENTITLED ' //1H0,17A4//
CANB0347 2 ' TYPE OF MESH ----- ',3A4//
CANB0348 3 ' PLOTTING DATA SAVED ----- ',I5//
CANB0349 4 ' PRINT SOIL RESPONSES ----- ',I5//
CANB0350 5 ' PRINT CONTROL FOR PREP OUTPUT ----- ',I5//
CANB0351 6 ' NUMBER OF CONSTRUCTION INCREMENTS ----- ',I5//
CANB0352 7 ' SPAN OF BOX ----- ',F9.2//
CANB0353 8 ' HEIGHT OF BOX ----- ',F9.2//
CANB0354 2120 FORMAT(1H0,' TRENCH WIDTH (FT) ----- ',F9.2//)
CANB0355 2100 FORMAT(1H0,' SOIL ABOVE TOP OF BOX (FT) ----- ',F9.2//)
CANB0356 1 ' MESH HEIGHT ABOVE TOP OF BOX (FT) --- ',F9.2//
CANB0357 2 ' SOIL DENSITY ABOVE MESH (PCF)----- ',F9.2//
CANB0358 2110 FORMAT(1H0,' PRESSURE OF TRUNCATED SOIL (PSI)----- ',F9.2)
CANB0359 2020 FORMAT(///,1H,'WE ARE GOING TO ASSUME NO BEDDING MATERIAL'//)
CANB0360 2030 FORMAT(A4,1X,17A4,3X,A4/815)
CANB0361 2040 FORMAT(A1,I4,I5,2F10.3,I5,5X,2F10.3)
CANB0362 2050 FORMAT(A1,I4,10I5)
CANB0363 2060 FORMAT(A1,I4,I5,F10.2,I5,2F10.2,3I5,2F10.2)
CANB0364 2070 FORMAT(1H0,' HTCOVER=',F10.2// ' FATAL DATA ERROR FOR THE HEIGHT
CANB0365 10F COVER OF SOIL'// ' HTCOVER MUST BE GREATER OR EQUAL THAN ZERO')
CANB0366 2080 FORMAT(1H0,' IDENTIFICATION OF MATERIAL ZONE WITH MATERIAL NUMBE
CANB0367 1R'///10X,' MATERIAL-ZONE',3X,' MATERIAL NO.'//)
CANB0368 2090 FORMAT(1H0,16X,2A4,3X,I2)
CANB0369 END
CANB0370 * * * * *
CANB0371 * CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
CANB0372 * VERSION 1.0, MARCH 1990
CANB0373 * PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
CANB0374 * THE FEDERAL HIGHWAY ADMINISTRATION
CANB0375 * * * * *
CANB0376 SUBROUTINE CONCRE(IA, ICOME, IEXIT, LEVEL, NINC, NPMAT, NPPT, PDIA,
CANB0377 1 PIPMAT, RESULT, XMODE)
CANB0378 C
CANB0379 C SUBROUTINE FOR REINFORCED CONCRETE PIPE CULVERTS
CANB0380 C
CANB0381 * Uncomment this statement in all program modules for double
CANB0382 * precision version. SCM 3-10-90
CANB0383 *
CANB0384 * IMPLICIT DOUBLE PRECISION (A-H,O-Z)
CANB0385 *
CANB0386 * Revise pipe dimension statements from 30 to 99.
CANB0387 * SCM 5-22-89
CANB0388 *
CANB0389 DIMENSION ASI(99),ASO(99),PFS(3),PIPMAT(5,99),PPFS(3),PSTRNI(99),
CANB0390 1 PSTRNO(99),RESULT(20,99),SAFE(4),STNMAT(5),TBI(99),
CANB0391 2 TBO(99),WALLA(33),WALLB(33),WALLC(33),FTV(99)
CANB0392 *
CANB0393 C

```

```

CANB0394
CANB0395
CANB0396
CANB0397
CANB0398
CANB0399
CANB0400
CANB0401
CANB0402
CANB0403
CANB0404
CANB0405
CANB0406
CANB0407
CANB0408
CANB0409
CANB0410
CANB0411
CANB0412
CANB0413
CANB0414
CANB0415
CANB0416
CANB0417
CANB0418
CANB0419
CANB0420
CANB0421
CANB0422
CANB0423
CANB0424
CANB0425
CANB0426
CANB0427
CANB0428
CANB0429
CANB0430
CANB0431
CANB0432
CONC0001
CONC0002
CONC0003
CONC0004
CONC0005
CONC0006
CONC0007
CONC0008
CONC0009
CONC0010
CONC0011
CONC0012
CONC0013
CONC0014
CONC0015
CONC0016
CONC0017
CONC0018
CONC0019
CONC0020
CONC0021
CONC0022
CONC0023
CONC0024

```

```

C   TABLES BELOW GIVE STANDARD WALL THICKNESSES DERIVED FROM ASTM C-76CONC0025
C   USERS MAY SUBSTITUTE THEIR OWN TABLES OF WALL THICKNESSES USING CONC0026
C   THE SAME FORMAT AS BELOW. PIPE SIZES FOR WHICH NO STANDARD WALL CONC0027
C   IS AVAILABLE SHOULD HAVE 0. IN THE MATRIX AT THOSE LOCATIONS CONC0028
C
C   FORMAT FOR LOCATING PIPE SIZES IS AS FOLLOWS-- CONC0029
C
C       12   15   18   21   24   27   30
C       33   36   39   42   45   48   51
C       54   57   60   63   66   69   72
C       75   78   81   84   87   90   93
C       96   99  102  105  108
C
C   COMMON /PEDRO/ STSMAX(30,11),STNMAX(30,11),SM1,SM2,SM3,PCE2,ETC, CONC0030
C   X FYC,FCC,PFSY,SM5,PTSN(30,2),PTSS(30,2) CONC0031
C   DATA WALLA/ CONC0032
C   1 1.750, 1.875, 2.000, 2.250, 2.500, 2.375, 2.750, CONC0033
C   2 2.875, 3.000, 3.250, 3.500, 3.750, 4.000, 4.250, CONC0034
C   3 4.500, 4.750, 5.000, 5.250, 5.500, 5.750, 6.000, CONC0035
C   4 6.250, 6.500, 6.750, 7.000, 7.250, 7.500, 7.750, CONC0036
C   5 8.000, 8.250, 8.500, 8.750, 9.000/ CONC0037
C   DATA WALLB/ CONC0038
C   1 2.000, 2.250, 2.500, 2.750, 3.000, 3.250, 3.500, CONC0039
C   2 3.750, 4.000, 4.250, 4.500, 4.750, 5.000, 5.250, CONC0040
C   3 5.500, 5.750, 6.000, 6.250, 6.500, 6.750, 7.000, CONC0041
C   4 7.250, 7.500, 7.750, 8.000, 8.250, 8.500, 8.750, CONC0042
C   5 9.000, 9.250, 9.500, 9.750, 10.000/ CONC0043
C   DATA WALLC/ CONC0044
C   1 2.750, 3.000, 3.250, 3.500, 3.750, 4.000, 4.250, CONC0045
C   2 4.500, 4.750, 5.000, 5.250, 5.500, 5.750, 6.000, CONC0046
C   3 6.250, 6.500, 6.750, 7.000, 7.250, 7.500, 7.750, CONC0047
C   4 8.000, 8.250, 8.500, 8.750, 9.000, 9.250, 9.500, CONC0048
C   5 9.750, 10.000, 10.250, 10.500, 10.750/ CONC0049
C
C   DATA NFS,PPFS/ 3,1.6,2.,2./ CONC0050
C   DATA FPC,XNU,FSY,SE,SNU,SLS/4000.,.17,40000.,29000000.,.3,2.0/ CONC0051
C   DATA ANALYS, ARBI , CIRC , DESIGN, ELLI , FIX , OPT , STD CONC0052
C   1 / 4HANAL, 4HARBI, 4HCIRC, 4HDESI, 4HELLI, 3HFIX, 3HOPT, 3HSTD CONC0053
C   DATA NTMAX,CWO1/20,0.01/ CONC0054
C
C   GO TO (100,200),ICOME CONC0055
C
C   FIRST ENTRY ESTABLISHES PIPE PROPERTIES AND SAFETY FACTORS CONC0056
C
C   100 READ(5,1000) PDIA,PT,RSHAPE,NONLIN,STNMAT(1),STNMAT(2),STNMAT(3), CONC0057
C   X PFPC,PCE,PNU,PDEN,PFSY,PSE,PSNU,SL CONC0058
C   ITER=0 CONC0059
C   IF(PDIA.EQ.0.0) GO TO 950 CONC0060
C   IF(PFPC.EQ.0.) PFPC=PFPC CONC0061
C   PFPT = 5.0*SQRT(PFPC) CONC0062
C   PFVT = 2.0*SQRT(PFPC) CONC0063
C   PDENT = PDEN CONC0064
C   IF(PDENT.LE.0.) PDENT = 150. CONC0065
C   IF(PCE.EQ.0.) PCE = 33.*PDENT+.15*SQRT(PFPC) CONC0066
C   IF(PNU.EQ.0.) PNU=XNU CONC0067
C   IF(PFSY.EQ.0.) PFSY=FSY CONC0068
C   IF(PSE.EQ.0.) PSE=SE CONC0069
C   IF(PSNU.EQ.0.) PSNU=SNU CONC0070
C   IF(SL.EQ.0.0) SL=SLS CONC0071
C   IF(PDIA.LT.0.0) WRITE(6,2001) CONC0072
C   WRITE(6,2000) PDIA,PFPC,PCE,PNU,PDEN,PFSY,PSE,PSNU CONC0073
C
C   PCENU = PCE/(1.-PNU**2) CONC0074
C
C   PSENU = PSE/(1.-PSNU**2) CONC0075
C   PN = PSENU/PCENU CONC0076
C   IF(NONLIN.LE.0) NONLIN = 1 CONC0077
C   IF(NONLIN.GE.3) NONLIN = 3 CONC0078
C   IF(STNMAT(1).LE.0.0) STNMAT(1) = 0.0 CONC0079
C   IF(STNMAT(2).LE.0.0) STNMAT(2) = 0.5*PFPC/PCENU CONC0080
C   IF(STNMAT(3).LE.0.0) STNMAT(3) = 0.002 CONC0081
C   IF(STNMAT(3).LE.STNMAT(2)) STNMAT(3) = 1.001*STNMAT(2) CONC0082
C   PCE2 = (PFPC-PCENU*STNMAT(2))/(STNMAT(3)-STNMAT(2)) CONC0083
C   STNMAT(4) = PCE2/PCENU CONC0084
C   STNMAT(5) = PFSY/PSENU CONC0085
C   SM1=STNMAT(1) CONC0086
C   SM2=STNMAT(2) CONC0087
C   SM3=STNMAT(3) CONC0088
C   SM5=STNMAT(5) CONC0089
C   ETC=SM1*PCENU CONC0090
C   FYC=STNMAT(2)*PCENU CONC0091
C   FCC=PFPC CONC0092
C   WRITE(6,2002) NONLIN,STNMAT(1),STNMAT(2),STNMAT(3),STNMAT(5),SL CONC0093
C
C   READ SECTION PROPERTIES FOR ANALYSIS CONC0094
C
C   IF(PDIA.LT.0.0) GO TO 111 CONC0095
C   IF(RSHAPE.NE.ELLI .AND. RSHAPE.NE.ARBI) RSHAPE= CIRC CONC0096
C   IF(XMODE.EQ.DESIGN) GO TO 120 CONC0097
C   NREAD =1 CONC0098
C   IF(RSHAPE.EQ.ARBI) NREAD = NPPT CONC0099
C   DO 110 N=1,NREAD CONC0100
C   READ(5,1001) ASI(N),ASO(N),TBI(N),TBO(N) CONC0101
C   IF(TBI(N).LE.0.0) TBI(N) = 1.25 CONC0102
C   IF(TBO(N).LE.0.0) TBO(N) = 1.25 CONC0103
C   CONTINUE CONC0104
C
C   WRITE SECTION PROPERTIES TO BE ANALYSED. CONC0105
C
C   WRITE(6,2005) PT,RSHAPE CONC0106
C   IF(RSHAPE.EQ.CIRC) WRITE(6,2010) ASI(1),ASO(1),TBI(1),TBO(1) CONC0107
C   IF(RSHAPE.EQ.ELLI) WRITE(6,2011) ASI(1),TBI(1) CONC0108
C   IF(RSHAPE.EQ.ARBI) WRITE(6,2012) (ASI(N),ASO(N),TBI(N),TBO(N), CONC0109
C   X N=1,NPPT) CONC0110
C   GO TO 165 CONC0111
C
C   READ ARBITRARY CONCRETE SECTION PROPERTIES FOR NEW OPTION OF CONC0112
C   VARIABLE THICKNESS (LEVEL 2-BOX OR 3) CONC0113
C
C   111 IF(LEVEL.EQ.3) GO TO 112 CONC0114
C   NDTOP=5 CONC0115
C   NDSIDE=7 CONC0116
C   NPPT=2*(NDTOP-1)+NDSIDE CONC0117
C   NEMAT=NPPT-1 CONC0118
C   112 IF(RSHAPE.NE.ARBI) GO TO 114 CONC0119
C   DO 113 N=1,NPPT CONC0120
C   READ(5,1001) ASI(N),ASO(N),TBI(N),TBO(N),PTV(N) CONC0121
C   IF(TBI(N).LE.0.0) TBI(N)=1.25 CONC0122
C   IF(TBO(N).LE.0.0) TBO(N)=1.25 CONC0123
C   IF(PTV(N).LE.0.0) PTV(N)=PT CONC0124
C   113 CONTINUE CONC0125
C   GO TO 118 CONC0126
C
C   READ BOX CULVERT SECTION PROPERTIES FOR LEVEL 2-BOX AND RSHAPE=STD CONC0127
C
C   114 IF(LEVEL.NE.2) GO TO 960 CONC0128
C   READ(5,1001) PTT,PTS,PTB,NH,HV CONC0129

```

CONC0150

```

      READ(5,1001) AS1,AS2,AS3,AS4,XL1,TC
      IF(PTT.LE.0.0) PTT=PT
      IF(PTS.LE.0.0) PTS=PT
      IF(PTB.LE.0.0) PTB=PT
      IF(XL1.LE.0.0) XL1=1.0
      IF(TC.LE.0.0) TC=1.25

C
C      SET UP BOX SECTION PROPERTIES FOR LEVEL 2-BOX
C      (FIRST UNIFORM COVER)
C
      DO 115 N=1,NPPT
        TBI(N)=TC
115    TBO(N)=TC
C
C      DETERMINE NODES FOR OUTER STEEL ON TOP AND BOTTOM
C
      NDAS= FLOAT(NDTOP-1)*XL1+1
      NDAS=NDTOP-NDAS
C
C      LOOP THROUGH FOR TOP AND BOTTOM NODE PROPERTIES
C
      LIM1=NDTOP-1
      DO 116 N=1,LIM1
        ASOUT=0.0
        IF(N.GT,NDAS) ASOUT=AS1
        ASO(N)=ASOUT
        ASI(N)=AS2
        PTV(N)=PTT
        NBOT=NPPT+1-N
        ASO(NBOT)=ASOUT
        ASI(NBOT)=AS3
        PTV(NBOT)=PTB
116
C
C      LOOP THROUGH FOR SIDE WALL (NEGLECTING CORNER NODES)
C
      LIM2=NDSIDE+NDTOP-2
      NTOPI=NDTOP+1
      DO 117 N=NTOPI,LIM2
        ASO(N)=AS1
        ASI(N)=AS4
        PTV(N)=PTS
117
C
C      LASTLY CONSIDER CORNER NODES
C
      ASO(NDTOP)=AS1
      ASI(NDTOP)=0.5*(AS2+AS4)
      PTV(NDTOP)=0.5*(PTT+PTS+HH+HV)
      PTV(NDTOP+NDSIDE-1)=0.5*(PTB+PTS+HH+HV)
      ASO(NDTOP+NDSIDE-1)=AS1
      ASI(NDTOP+NDSIDE-1)=0.5*(AS3+AS4)
C
C      WRITE SECTION PROPERTIES
C
118    CONTINUE
      WRITE(6,2009) (N,ASI(N),ASO(N),TBI(N),TBO(N),PTV(N),N=1,NPPT)
      GO TO 165
C
C      READ OR DEFAULT SAFETY FACTORS, ALLOWABLE CRACK WIDTH, AND
C      DESIGN GOAL FOR WALL THICKNESS
C      ** CRACK WIDTH IS A DESIGN CRITERIA ONLY WHEN THE ALLOWABLE
C      VALUE IS SPECIFIED **
C
120    READ(5,1003) (PFS(I),I=1,NFS),ALCW,GOAL,TBI(1),SRATIO

```

```

CONC0151    IF(TBI(1).LE.0.0) TBI(1) = 1.25
CONC0152    IF(SRATIO.LE.0.0) SRATIO = 0.75
CONC0153    DO 130 I=1,NFS
CONC0154    IF(PFS(I).GT.0.0) GO TO 130
CONC0155    PFS(I)=PFS(I)
CONC0156    130 CONTINUE
CONC0157    C
CONC0158    C    NGOAL = 1, IMPLIES OPTIMUM WALL DESIGN
CONC0159    C    NGOAL = 2, IMPLIES STANDARD WALL DESIGN
CONC0160    C    NGOAL = 3, IMPLIES FIXED WALL THICKNESS
CONC0161    C
CONC0162    NGOAL = 0
CONC0163    IF(GOAL.EQ.OPT) NGOAL = 1
CONC0164    IF(GOAL.EQ.FIX .AND. PT.GT.0.0) NGOAL = 3
CONC0165    IF(NGOAL.EQ.0) GOAL = STD
CONC0166    IF(NGOAL.EQ.0) NGOAL = 2
CONC0167    WRITE(6,2100) (PFS(I),I=1,NFS)
CONC0168    IF(ALCW.NE.0.) WRITE(6,2101) ALCW
CONC0169    IF(ALCW.EQ.0.) ALCW = 1000000.
CONC0170    WRITE(6,2102) RSHAPE,TBI(1),GOAL
CONC0171    IF(NGOAL.EQ.3) WRITE(6,2103) PT
CONC0172    IF(RSHAPE.EQ.CIRC) WRITE(6,2104) SRATIO
CONC0173    PFSC = CW01/ALCW
CONC0174    C
CONC0175    C    ASSIGN MINIMUM WALL THICKNESS FOR DESIGN
CONC0176    C    IF THICKNESS IS SPECIFIED IT WILL REMAIN CONSTANT
CONC0177    C    IF IT IS ADEQUATE
CONC0178    C
CONC0179    IF(NGOAL.EQ.3) GO TO 160
CONC0180    141 GO TO (151,152),NGOAL
CONC0181    151 PT = .05*PDIA
CONC0182    GO TO 160
CONC0183    152 CONTINUE
CONC0184    ISIZE = PDIA/3 -3
CONC0185    IF(ISIZE.GE.1.AND.ISIZE.LE.33) GO TO 155
CONC0186    153 NGOAL = 1
CONC0187    WRITE(6,2018)
CONC0188    GO TO 141
CONC0189    155 PT=WALLA(ISIZE)
CONC0190    IF(PT.LE.0.) PT = WALLB(ISIZE)
CONC0191    IF(PT.LE.0.) PT = WALLC(ISIZE)
CONC0192    IF(PT.LE.0.) GO TO 153
CONC0193    160 PTMIN = PT
CONC0194    C
CONC0195    C    ASSIGN MINIMUM STEEL AREA BASED ON HANDLING DEFLECTION,
CONC0196    C    TEMPERATURE STEEL, OR MINIMUM PRACTICAL REINFORCEMENT
CONC0197    C
CONC0198    AS = (.13*PDIA**3 + 130.*PDIA**2) / (.49*PSE*PT**2) - .0184*PT/PN
CONC0199    IF(AS.LT..0014*PT) AS = .0014*PT
CONC0200    IF(AS.LT..00583) AS = .00583
CONC0201    ASMIN = AS
CONC0202    C
CONC0203    C    ESTABLISH RELATIVE DESIGN PROPORTIONS OF STEEL FOR CIRCULAR AND
CONC0204    C    ELLIPTICAL CAGES
CONC0205    C
CONC0206    IF(RSHAPE.EQ.ELLI) GO TO 161
CONC0207    ASI(1) = ASMIN
CONC0208    ASO(1) = SRATIO*ASMIN
CONC0209    TBO(1) = TBI(1)
CONC0210    GO TO 165
CONC0211    161 ASI(1) = ASMIN
CONC0212    ASO(1) = 0.0
CONC0213    TBO(1) = 0.0
CONC0214
CONC0215
CONC0216
CONC0217
CONC0218
CONC0219
CONC0220
CONC0221
CONC0222
CONC0223
CONC0224
CONC0225
CONC0226
CONC0227
CONC0228
CONC0229
CONC0230
CONC0231
CONC0232
CONC0233
CONC0234
CONC0235
CONC0236
CONC0237
CONC0238
CONC0239
CONC0240
CONC0241
CONC0242
CONC0243
CONC0244
CONC0245
CONC0246
CONC0247
CONC0248
CONC0249
CONC0250
CONC0251
CONC0252
CONC0253
CONC0254
CONC0255
CONC0256
CONC0257
CONC0258
CONC0259
CONC0260
CONC0261
CONC0262
CONC0263
CONC0264
CONC0265
CONC0266
CONC0267
CONC0268
CONC0269
CONC0270
CONC0271
CONC0272
CONC0273
CONC0274
CONC0275
CONC0276

```

```

C      DESIGN AND ANALYSIS ... ASSIGN CIRCULAR OR ELLIPTICAL STEEL CAGES. CONCO277
C      NOTE THIS SECTION IS RE-ENTERED FOR EACH TRIAL DESIGN. CONCO278
C      IF(PDIA.LT.0.0) GO TO 181 CONCO279
C      IF(RSHAPE.EQ.ARBI) GO TO 180 CONCO280
C      DELTB = 0.0 CONCO281
C      IF(RSHAPE.NE.ELLI) GO TO 168 CONCO282
C      KPPT = NPPT CONCO283
C      XPMAT = NPMAT CONCO284
C      DELTB = (PT-2.*TBI(1))/(XPMAT/2.) CONCO285
168 CONTINUE CONCO286
C      DO 170 N=2,NPPT CONCO287
C      ASI(N) = ASI(N-1) CONCO288
C      ASO(N) = ASO(N-1) CONCO289
C      SIGN = 1.0 CONCO290
C      IF(RSHAPE.NE.ELLI) GO TO 169 CONCO291
C      XN = N CONCO292
C      TERM = KPPT/2. - XN + 1. CONCO293
C      IF(TERM.EQ.0.0) SIGN = 0.0 CONCO294
C      IF(TERM.LT.0.0) SIGN=-1. CONCO295
169 TBI(N) = TBI(N-1) + DELTB*SIGN CONCO296
C      TBO(N) = TBO(N-1) CONCO297
170 CONTINUE CONCO298
C      CALCULATE INTIAL SECTION PROPERTIES BASED ON UNCRACKED SECTION. CONCO299
C      (GENERALLY THIS WILL PRODUCE MAXIMUM MOMENT AND THRUST) CONCO300
C      IF(ICOME.EQ.1) PDIA = PDIA + PT CONCO301
180 CONTINUE CONCO302
C      DO 190 N=1,NPPT CONCO303
C      IF(PDIA.LT.0.0) PT=PTV(N) CONCO304
C      APNI = ASI(N)*(PN-1.) CONCO305
C      APNO = ASO(N)*(PN-1.) CONCO306
C      YTI = TBI(N) CONCO307
C      YTO = PT - TBO(N) CONCO308
C      PA = PT + APNI + APNO CONCO309
C      YBAR = (0.5*PT*PT + APNI*YTI + APNO*YTO)/PA CONCO310
C      PI = 0.3333*PT**3 - PT*PT*YBAR + PT*YBAR*YBAR CONCO311
C      PI = PI + APNI*(YTI-YBAR)**2 + APNO*(YTO-YBAR)**2 CONCO312
C      RESULT(14,N) = PA CONCO313
C      RESULT(15,N) = PI CONCO314
C      RESULT(16,N) = YBAR CONCO315
C      PSTRNI(N) = 0.0 CONCO316
C      PSTRNO(N) = 0.0 CONCO317
C      INITIAL VALUES FOR MAX. STRAINS AND STRESSES CONCO318
C      DO 188 IL=1,2 CONCO319
C      PTSN(N,IL)=0.0 CONCO320
C      PTSS(N,IL)=0.0 CONCO321
C      DO 189 II=1,11 CONCO322
C      STNMAX(N,II)=-0.000001/PCENU CONCO323
C      STSMAX(N,II)=-0.000001 CONCO324
189 CONTINUE CONCO325
C      AVERAGE AREA AND MOMENT OF INERTIA AT ELEMENT CENTERS FOR INPUT CONCO326
C      INTO SOLUTION THEORIES. CONCO327
C      DO 195 N=1,NPMAT CONCO328
C      N1 = N+1 CONCO329
C      IF(N1.GT.NPPT) N1=1 CONCO330
C      PIPMAT(1,N) = PCE CONCO331
C      PIPMAT(2,N) = PNU CONCO332
C      PIPMAT(3,N) = 0.5*(RESULT(14,N)+RESULT(14,N1)) CONCO333
C      PIPMAT(4,N) = 0.5*(RESULT(15,N)+RESULT(15,N1)) CONCO334
C      PIPMAT(5,N) = PDEN*PT/1728. CONCO335
195 CONTINUE CONCO336
C      NTIMES = 0 CONCO337
C      RETURN CONCO338
C      SECOND ENTRY ... (1) RE-EVALUATE SECTION PROPERTIES BASED ON CONCO339
C      CURRENT VALUES OF THRUST AND MOMENT. (2) EVALUATE STRUCTURAL CONCO340
C      RESPONSES IN TERMS OF DESIGN CRITERIA. (3) IN DESIGN MODE CONCO341
C      DETERMINE NEW STEEL AREA AND/OR CONCRETE THICKNESS AS REQUIRED. CONCO342
C      CALL CONMAT(NPMAT,NPPT,ICON,ASI,ASO,TBI,TBO,RESULT,PIPMAT,STNMAT, CONCO343
C      X PSTRNI,PSTRNO,PT,PN,PCENU,PSENU,NONLIN,PDIA,PTV,NTIMES,SL) CONCO344
C      IF(ICON.EQ.1) GO TO 500 CONCO345
C      IF(NTIMES.GE.NTIMAX) GO TO 500 CONCO346
C      NTIMES = NTIMES + 1 CONCO347
C      IEXIT = -1 CONCO348
C      WRITE(6,*)NTIMES CONCO349
C      RETURN CONCO350
C      CHECK IF NONLINEAR SOIL MODEL HAS CONVERGED CONCO351
C      500 NTIMES = 0 CONCO352
C      IF(NINC.GT.0) GO TO 505 CONCO353
C      IEXIT = -1 CONCO354
C      RETURN CONCO355
C      SAVE INNER AND OUTER PIPE STRAINS OF CONVERGED SOLUTION CONCO356
C      DO 507 N=1,NPPT CONCO357
C      PSTRNI(N) = RESULT(19,N) CONCO358
C      PSTRNO(N) = RESULT(20,N) CONCO359
507 EVALUATE STRUTURAL RESPONSES. (IN DESIGN MODE ONLY ON LAST STEP) CONCO360
C      BSMAX = MAX TENSILE STRESS IN STEEL CONCO361
C      BCMAX = MAX COMPRESSIVE STRESS IN CONCRETE CONCO362
C      VMAX = MAX SHEAR STRESS ON GROSS SECTION CONCO363
C      DISP = MAX RELATIVE DISPLACEMENT CONCO364
C      CRWD = MAX CRACK WIDTH CONCO365
C      IF(XMODE.EQ.ANALYS) GO TO 510 CONCO366
C      IF(IA.EQ.NINC) GO TO 510 CONCO367
C      IEXIT = 0 CONCO368
C      RETURN CONCO369
C      BSMAX = 0.000001*PFSY CONCO370
C      BCMAX = -0.000001*PFPC CONCO371
C      VMAX = 0.000001*PFPC CONCO372
C      UYMAX = RESULT(4,1)+0.000001*PT CONCO373
C      UYMIN = RESULT(4,1) CONCO374
C      CRWD = 0.0 CONCO375
C      DO 510 N=1,NPPT CONCO376
C      IF(ABS(RESULT(10,N)).GT.BSMAX) BSMAX = ABS(RESULT(10,N)) CONCO377
C      IF(ABS(RESULT(11,N)).GT.BCMAX) BCMAX = ABS(RESULT(11,N)) CONCO378
C      IF(RESULT(12,N).LT.BCMAX) BCMAX = RESULT(12,N) CONCO379
C      VN = ABS(RESULT(13,N)) CONCO380
C      IF(VN.LT.VMAX) GO TO 520 CONCO381
C      VMAX = VN CONCO382
520 IF(RESULT(17,N).GT.CRWD) CRWD = RESULT(17,N) CONCO383
C      CONCO384
C      CONCO385
C      CONCO386
C      CONCO387
C      CONCO388
C      CONCO389
C      CONCO390
C      CONCO391
C      CONCO392
C      CONCO393
C      CONCO394
C      CONCO395
C      CONCO396
C      CONCO397
C      CONCO398
C      CONCO399
C      CONCO400
C      CONCO401
C      CONCO402

```



```

WRITE(6,2900) IA,FSST,FSCC,FSV,FSCW,FSDIS,FSBOW
GO TO 581
580 WRITE(6,2901) IA,FSST,FSCC,FSV,FSCW
C
581 IEXIT = 1
IF(IA.NE.NINC) IEXIT = 0
RETURN
C
C FATAL ERROR EXITS
C
950 WRITE(6,2013)
IEXIT = 1
RETURN
960 WRITE(6,2014)
IEXIT = 1
RETURN
C
C FORMATS
C
1000 FORMAT(2F10.0,1X,A4,I5,3F10.0 /8F10.0)
1001 FORMAT(6F10.0)
1003 FORMAT(4F10.0,7X,A3,2F10.0)
2000 FORMAT(1H1/// 10X,35H PIPE PROPERTIES ARE AS FOLLOWS .... /
X 10X,36H (UNITS ARE INCH-POUND SYSTEM) //
X 15X,36H NOMINAL PIPE DIAMETER ..... , F15.4 //
X 15X,36H CONCRETE COMPRESSIVE STRENGTH ..... , F15.4 //
X 15X,36H CONCRETE ELASTIC MODULUS ..... , F15.4 //
X 15X,36H CONCRETE POISSON RATIO ..... , F15.4 //
X 15X,36H DENSITY OF PIPE (PCF) ..... , F15.4 //
X 15X,36H STEEL YIELD STRENGTH ..... , F15.4 //
X 15X,36H STEEL ELASTIC MODULUS ..... , F15.4 //
X 15X,36H STEEL POISSON RATIO ..... , F15.4 //
2001 FORMAT(///10X,88H***NEGATIVE PIPE DIAMETER IMPLIES NEW CANDE OPTIC
XON FOR VARIABLE CONCRETE THICKNESS. *** //10X,75H***OPTION IS RECONC
XSTRICED TO ANALYSIS ONLY WITH LEVEL 2-BOX, OR LEVEL 3. *** )
2002 FORMAT(//15X,36H NONLINEAR CODE (1,2,OR 3) ..... , 10X,I5 //
X 15X,36H CONC. CRACKING STRAIN (1,2,3) ..... , F15.6 //
X 15X,36H CONC. YIELDING STRAIN (2,3) ..... , F15.6 //
X 15X,36H CONC. CRUSHING STRAIN (2,3) ..... , F15.6 //
X 15X,36H STEEL YIELDING STRAIN (3) ..... , F15.6 //
X 15X,36H SPACING LONGITUDINAL REINFORCEMENT , F15.2 //
2005 FORMAT(// 15X, 36H WALL THICKNESS ..... , F15.4 //
X15X,36H SHAPE OF REBAR CAGES ..... ,11X,A4 /)
2009 FORMAT(1H1,5X,'NODE # STEEL AREAS(IN2)',13X,'STEEL COVERS(IN)
X',8X,'THICKNESS(IN)',// ' N ASI(N) ASO(N)
X TBI(N) TBO(N) PTV(N)',/(I6,5F15.4))
2010 FORMAT(15X,36H INNER CAGE STEEL AREA ..... , F15.4 //
X 15X,36H OUTER CAGE STEEL AREA ..... , F15.4 //
X 15X,36H COVER TO CENTER OF INNER REBAR ..... , F15.4 //
X 15X,36H COVER TO CENTER OF OUTER REBAR ..... , F15.4 //
2011 FORMAT(15X,36H ELLIPTICAL CAGE STEEL AREA ..... , F15.4 //
X 15X,36H MIN. COVER TO CENTER OF REBAR ..... , F15.4 //
2012 FORMAT(15X,36H REINFORCEMENT AREA AND COVER DEPTH //
X 15X,36H INNER A. OUTER A. INNERC. OUTER C. /(15X,4F9.4))
2013 FORMAT(52H PIPE DIAMETER WAS NOT SPECIFIED PROGRAM TERMINATED)
2014 FORMAT(1H0,5X,47H WRONG LEVEL WAS SPECIFIED, PROGRAM TERMINATED)
2016 FORMAT(48H WALL THICKNESS GREATER THAN WALL C IS REQUIRED /
X 34H DESIGN WILL PROCEED IN OPTION OPT)
2018 FORMAT(///41H NO STANDARD WALLS FOR THIS PIPE DIAMETER /
X 34H DESIGN WILL PROCEED IN OPTION OPT)
2100 FORMAT(/// 10X,43H DESIRED SAFETY FACTORS FOR PIPE DESIGN ... //
X 15X,36H STEEL YIELDING ..... , F15.4 //
X 15X,36H CONCRETE CRUSHING ..... , F15.4 //
CONC0529 X 15X,36H CONCRETE SHEAR FAILURE ..... , F15.4 )
CONC0530 2101 FORMAT(//15X,36H ALLOWABLE CRACK WIDTH (IN) ..... , F15.4 )
CONC0531 2102 FORMAT(//15X,36H SHAPE OF REBAR CAGE(S) ..... ,11X,A4//
CONC0532 X 15X,36H MIN. COVER TO CENTER OF REBAR ..... , F15.4 //
CONC0533 X 15X,36H DESIGN GOAL FOR WALL THICKNESS ..... ,12X,A3)
CONC0534 2103 FORMAT(//15X, 36H FIXED WALL THICKNESS ..... , F15.4)
CONC0535 2104 FORMAT(//15X,36H OUTER-TO-INNER STEEL RATIO ..... , F15.4)
CONC0536 2200 FORMAT(1H1,/// 10X,50H DESIGN ITERATIVE PROCEDURE DID NOT CONVERGE
CONC0537 X AFTER ,13/ 10X,55H ITERATIONS. THE LAST SOLUTION IS PRINTED AND ECONC
CONC0538 XVALUATED, / 10X,37H WHERE THE SECTION PROPERTIES ARE ... //
CONC0539 X 15X,36H WALL THICKNESS ..... , F15.5 //
CONC0540 X 15X,36H INNER CAGE STEEL AREA (OR ELLIPSE) , F15.5 //
CONC0541 X 15X,36H OUTER CAGE STEEL AREA ..... , F15.5 )
CONC0542 2310 FORMAT(1H1,/// 10X,19H DESIGN RESULTS AT ,12,11H ITERATIONS //
CONC0543 X 15X,36H *** WALL THICKNESS ..... , F10.5 //
CONC0544 X 15X,36H *** INNER CAGE STEEL AREA (OR ELLI) , F10.5 //
CONC0545 X 15X,36H *** OUTER CAGE STEEL AREA ..... , F10.5 )
CONC0546 2900 FORMAT(1H1,///10X,32H STRUCTURAL RESPONSE OF CULVERT ,
CONC0547 X 20H FOR LOAD INCREMENT ,15//
CONC0548 X 58H COORDINATES, DISPLACEMENTS AND CRACK DEPTHS ARE IN INCHES /
CONC0549 X 26H PRESSURES ARE IN LB/IN**2/28H MOMENTS ARE IN IN.*LB/IN. /
CONC0550 X 31H THRUST AND SHEAR ARE IN LB/IN. ///
CONC0551 X 5H NPPT,6X,9H X-COORD.,7X,8H X-DISP.,7X,8H N-PRES.,8X,7H MOMENT,
CONC0552 X 9X,6H SHEAR,11X,9H Y-COORD.,7X,8H Y-DISP.,7X,8H S-PRES.,BX,
CONC0553 X 7H THRUST,3X,12H CRACK DEPTH,/(15,F15.2,4E15.5/5X,F15.2,4E15.5))
CONC0554 2810 FORMAT(1H1,/// 10X,33H STRESSES IN CULVERT WALL (PSI) ,
CONC0555 X 20H FOR LOAD INCREMENT ,15//
CONC0556 X 9X, 11H ELLIP. OR /
CONC0557 X 5H NPPT,4X,11H INNER CAGE ,4X,11H OUTER CAGE ,6X,9H CONCRETE ,
CONC0558 X 9X,6H SHEAR ,14X,6H STEEL ,9X,6H STEEL ,3X,12H COMPRESSION ,
CONC0559 X 8X,7H STRESS //
CONC0560 X (15,4E15.5))
CONC0561 2820 FORMAT(1H0,///12X,'STRAINS IN THE INNER AND OUTER FIBER OFCONC0624
X THE CULVERT WALL',//2X,'(ONLY STRAINS FOR COMPRESSION ZONES HAVCONC0625
XE PHYSICAL MEANING)',///, ' NPPT INNER STRAIN OUTER STRAIN',/CONC0626
X/(15,2E15.5))
CONC0627 2900 FORMAT(1H1,///15X,26H CALCULATED SAFETY FACTORS ,
CONC0628 X 20H FOR LOAD INCREMENT ,15//
CONC0629 X 10X,50H STEEL YIELD STRESS / MAX. STEEL STRESS ..... , F10.3//CONC0630
X 10X,50H CONCRETE STRENGTH / MAX. COMPRESSIVE STRESS ..... , F10.3//CONC0631
X 10X,50H WALL SHEAR CAPACITY / MAX. SHEAR ..... , F10.3//CONC0632
X 5X,20H PERFORMANCE FACTORS //
CONC0633 X 10X,50H 0.01 INCH / MAX. CRACK WIDTH ..... , F10.3//CONC0634
X 10X,50H ALLOWABLE DISPLACEMENT(LUM) / MAX. DISP. .... , F10.3//CONC0635
X 10X,50H TENSILE STRENGTH / STRESS FROM BOW STRING ..... , F10.3 )CONC0636
2901 FORMAT(1H1,///15X,26H CALCULATED SAFETY FACTORS ,
CONC0637 X 20H FOR LOAD INCREMENT ,15//
CONC0638 X 10X,50H STEEL YIELD STRESS / MAX. STEEL STRESS ..... , F10.3//CONC0639
X 10X,50H CONCRETE STRENGTH / MAX. COMPRESSIVE STRESS ..... , F10.3//CONC0640
X 10X,50H WALL SHEAR CAPACITY / MAX. SHEAR ..... , F10.3//CONC0641
X 5X,20H PERFORMANCE FACTORS //
CONC0642 X 10X,50H 0.01 INCH. / MAX. CRACK WIDTH ..... , F10.3 )CONC0643
CONC0644 END
CONC0645 * * * * *
CONC0646 * CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
CONC0647 * VERSION 1.0, MARCH 1990
CONC0648 * PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
CONC0649 * THE FEDERAL HIGHWAY ADMINISTRATION
CONC0650 * * * * *
CONC0651 SUBROUTINE CONMAT(NPMAT,NPPT,ICON,ASI,ASO,TBI,TBO,RESULT,PIPHAT,
CONC0652 X STNMAT,PSTRN1,PSTRN0,PT,PN,PCE,PSE,NONLIN,PDIA,PTV,NOLIT,SL)
CONC0653 C
CONC0654 C THIS ROUTINE IS ENTERED WITH THE TOTAL MOMENT AND THRUST ON EACH
CONC0655 C

```



```

C PIPE NODE. THE OBJECTIVE IS TO DETERMINE THE FOLLOWING RESPONSES CONM0011
C CONSISTANT WITH THIS LOADING ... CONM0012
C CONM0013
C (1) INNER AND OUTER FIBER STAINS ASSUMING LINEAR DISTRIBUTION. CONM0014
C (2) DEPTH OF CRACKING, DISTANCE TO BENDING AXIS. CONM0015
C (3) EFFECTIVE THRUST AREA AND EFFECTIVE MOMENT OF INERTIA. CONM0016
C (THESE ARE AVERAGED INTO PIPMAT VECTOR FOR NEXT SOLUTION) CONM0017
C (4) STEEL STRESSES, CONCRETE COMPRESSIVE STRESS, AND SHEAR STRESS. CONM0018
C (5) CRACK WIDTH ESTIMATE. CONM0019
C CONM0020
C THE ABOVE RESPONSES ARE STORED IN THE VECTOR RESULT FOR EACH NODE CONM0021
C POINT AS DEFINED BELOW... CONM0022
C CONM0023
C RESULT(10,N) = INNER CAGE STEEL STRESS CONM0024
C RESULT(11,N) = OUTER CAGE STEEL STRESS. CONM0025
C RESULT(12,N) = MAX CONCRETE COMPRESSIVE STRESS. CONM0026
C RESULT(13,N) = MAX SHEAR STRESS IN CONCRETE. CONM0027
C RESULT(14,N) = EFFECTIVE AREA OF SECTION CONM0028
C RESULT(15,N) = MOMENT OF INERTIA ABOUT YBAR. CONM0029
C RESULT(16,N) = DISTANCE TO BENDING AXIS (YBAR). CONM0030
C RESULT(17,N) = WIDTH OF CRACK. CONM0031
C RESULT(18,N) = LENGTH OF CRACK. CONM0032
C RESULT(19,N) = MOMENT INCREMENT, OR CURRENT INNER STRAIN CONM0033
C RESULT(20,N) = THRUST INCREMENT, OR CURRENT OUTER STRAIN CONM0034
C PSTRNI(N) = INNER WALL STRAIN OF PREVIOUS CONVERGED STEP. CONM0035
C PSTRNO(N) = OUTER WALL STRAIN OF PREVIOUS CONVERGED STEP. CONM0036
C
C * Uncomment this statement in all program modules for double CONM0037
C * precision version. SCM 3-10-90 CONM0038
C * CONM0039
C * IMPLICIT DOUBLE PRECISION (A-H,O-Z) CONM0040
C * CONM0041
C * CONM0042
C * Revise pipe dimension statements from 30 to 99. CONM0043
C * SCM 5-22-89 CONM0044
C * CONM0045
C *
C DIMENSION ASI(NPPT),ASO(NPPT),TBI(NPPT),TBO(NPPT),RESULT(20,NPPT), CONM0046
C X PIPMAT(5,NPMAT),STNMAT(5),PSTRNI(99),PSTRNO(99),PTV(99),FE(11), CONM0047
C X CMOM(99),CTHR(99) CONM0048
C * CONM0049
C * COMMON /PEDRO/ STSMAX(30,11),STNMAX(30,11),SM1,SM2,SM3,PCE2,ETC, CONM0050
C X FYC,FCC,FSY,SM5,PTSN(30,2),PTSS(30,2) CONM0051
C DATA LIMIT,TOLER1,TOLER2/3,0.01,0.03/ CONM0052
C CONM0053
C SET OUTER LOOP CONVERGENCE CODE, (FOR ITERATIONS WITHIN LOAD STEP) CONM0054
C CONM0055
C ICON = 1 CONM0056
C CONM0057
C BEGIN NODE LOOP TO CONSIDER EACH NODE INDIVIDUALLY CONM0058
C CONM0059
C DO 100 N=1,NPPT CONM0060
C IF(NOLIT.GT.0) GO TO 81 CONM0061
C CDELM=0.0 CONM0062
C CDELN=0.0 CONM0063
C GO TO 82 CONM0064
C CDELM=CMOM(N) CONM0065
C CDELN=CTHR(N) CONM0066
C 81 CONTINUE CONM0067
C 82 IF(PDIA.LT.0.0) PT=PTV(N) CONM0068
C PA = RESULT(14,N) CONM0069
C PT = RESULT(15,N) CONM0070
C YBAR = RESULT(16,N) CONM0071
C DELM = RESULT(19,N) + CDELM CONM0072
C DELN = RESULT(20,N) + CDELN CONM0073
C
C PDENOM = PSTRNO(N)-PSTRNI(N) CONM0074
C CONM0075
C DETERMINE DISTANCE TO CENTER OF REBARS AND SET INNER LOOP COUNTER. CONM0076
C CONM0077
C YTI = TBI(N) CONM0078
C YTO = PT-TBO(N) CONM0079
C NTIMES = 0 CONM0080
C CONM0081
C BEGIN INNER LOOP --- DETERMINE INNER AND OUTER STRAINS OF PIPE. CONM0082
C CONM0083
C THETA = DELM/(PI*PCE) CONM0084
C STNORM = DELN/(PA*PCE) CONM0085
C STRNI = STNORM + THETA*YBAR + PSTRNI(N) CONM0086
C STRNO = STNORM + THETA*(YBAR-PT) + PSTRNO(N) CONM0087
C IF (DELM .EQ. 0.0 .AND. DELN .EQ. 0.0) GO TO 99 CONM0088
C DENOM = STRNO - STRNI CONM0089
C IF(DENOM.EQ.0.0) DENOM=0.00001*STNMAT(2) CONM0090
C STRNEG=STRNI CONM0091
C IF(STRNEG.GT.STRNO)STRNEG=STRNO CONM0092
C CONM0093
C DETERMINE FE(I) AT INTEGRATION POINTS CONM0094
C CONM0095
C NINT=10 CONM0096
C NINTP=11 CONM0097
C DY=PT/FLOAT(NINT) CONM0098
C YY=0.0 CONM0099
C DO 135 I=1,NINTP CONM0100
C PSTAIN=PSTRNI(N)+PDENOM*YY/PT CONM0101
C CONM0102
C SET MAX. VALUE OF STRESS AND STRAIN AT EACH POINT CONM0103
C CONM0104
C SSM=-ABS(STSMAX(N,I)) CONM0105
C SNM=STNMAX(N,I) CONM0106
C IF(NOLIT.GT.0) GO TO 34 CONM0107
C IF(PSTAIN.GT.SNM) GO TO 27 CONM0108
C IF(PSTAIN.LT.SNM) STNMAX(N,I)=PSTAIN CONM0109
C IF(SNM.LT.SM3)GO TO 33 CONM0110
C IF(SNM.LT.SM2)GO TO 32 CONM0111
C PSTESS=(PSTAIN-SNM)*PCE+SSM CONM0112
C IF(PSTESS.GT.FYC) GO TO 31 CONM0113
C PSTESS=(PSTAIN-SM2)*PCE2+(SM2-SNM)*PCE+SSM CONM0114
C IF(PSTESS.GT.FCC) GO TO 31 CONM0115
C GO TO 33 CONM0116
C 32 PSTESS=(PSTAIN-SNM)*PCE2+SSM CONM0117
C IF(PSTESS.GT.FCC)GO TO 31 CONM0118
C 33 PSTESS=FCC+(PSTAIN-SM3)*0.00001 CONM0119
C IF(PSTESS.LT.SSM .AND. STSMAX(N,I).LT.0.0) GO TO 29 CONM0120
C IF(PSTESS.LT.SSM) STSMAX(N,I)=-PSTESS CONM0121
C GO TO 34 CONM0122
C 29 STSMAX(N,I)=PSTESS CONM0123
C GO TO 34 CONM0124
C 27 PSTESS=(PSTAIN-SNM)*PCE + SSM CONM0125
C IF(PSTESS.GT.FTC) STSMAX(N,I)=ABS(STSMAX(N,I)) CONM0126
C IF(PSTESS.GT.0.0 .AND. STSMAX(N,I).GT.0.0) STSMAX(N,I)=ABS(STSMAX( CONM0127
C XN,I)) CONM0128
C 34 FE(I)=1.0 CONM0129
C ICK=1 CONM0130
C IF(STSMAX(N,I).GT.0.0) ICK=0 CONM0131
C CONM0132
C COMPUTE STRAINS OF THE POINT YY CONM0133
C CONM0134
C STAIN=STRNI+DENOM*YY/PT CONM0135
C IF(NONLIN.LT.2) GO TO 43 CONM0136

```

```

C
C MAX, COMPRESSION STRESS AND STRAIN OF THE POINT YY
C
SSM=-ABS(STSMAX(N,I))
SNM=STNMAX(N,I)

C
C CHECK IF THE POINT IS UNLOADED
C
IF(STAIN.LT.PSTAIN) GO TO 36
IF(SSM.LT.FYC) GO TO 37
STESS=(STAIN-SM2)*PCE+FYC
GO TO 38
37 STESS=(STAIN-SNM)*PCE+SSM
38 IF(STESS.GT.0.0 .AND. ICK.EQ.0) FE(I)=0.0
IF(STESS.GT.FTC) FE(I)=0.0
IF(I.EQ.1) STSIN=STESS
IF(I.EQ.NINTP) STSOU=STESS
PSTESS=(PSTAIN-SNM)*PCE + SSM
FS1=PSTESS
IF(FS1.GT.FTC) FS1=0.0
IF(FS1.GT.0.0 .AND. ICK.EQ.0) FS1=0.0
GO TO 35

C
C THE POINT IS LOADED
C
C COMPUTE THE NEW STRESS
36 IF(STAIN.GT.SNM) GO TO 44
IF(SNM.LT.SM3) GO TO 39
IF(SNM.LT.SM2) GO TO 40
STESS=(STAIN-SNM)*PCE+SSM
IF(STESS.GT.FYC) GO TO 41
STESS=(STAIN-SM2)*PCE2+(SM2-SNM)*PCE+SSM
IF(STESS.GT.FCC) GO TO 41
STESS=FCC+(STAIN-SM3)*0.00001
41 PSTESS=(PSTAIN-SNM)*PCE+SSM
GO TO 42
40 STESS=(STAIN-SNM)*PCE2+SSM
IF(STESS.GT.FCC) GO TO 41
STESS=(SM3-SNM)*PCE2+(STAIN-SM3)*0.00001+SSM
GO TO 41
39 STESS=(STAIN-SM3)*0.00001+PCC
GO TO 41
44 STESS=(STAIN-SNM)*PCE+SSM
IF(STESS.GT.0.0 .AND. ICK.EQ.0) GO TO 45
IF(STESS.GT.FTC) GO TO 45
GO TO 41
45 FE(I)=0.0
GO TO 35
42 FS1=PSTESS
IF(I.EQ.1) STSIN=STESS
IF(I.EQ.NINTP) STSOU=STESS
IF(PSTESS.GT.FTC) FS1=0.0
IF(PSTESS.GT.0.0 .AND. ICK.EQ.0) FS1=0.0
FE1=(SNM*PCE-SSM+FS1)/PCE
FE2=STAIN
FS2=STESS
FE(I)=(FS2-FS1)/(FE2-FE1)*(FE2-FE1)/(FE2-PSTAIN)
FE(I)=FE(I)/PCE
IF(FE(I).LT.0.0000001) FE(I)=0.0000001
GO TO 35
43 IF(STAIN.GT.SM1) FE(I)=0.0
IF(STAIN.GT.0.0 .AND. ICK.EQ.0) FE(I)=0.0

```

```

CONM0137
CONM0138 35 IF(FE(I).EQ.0.0) STSMAX(N,I)=ABS(STSMAX(N,I))
CONM0139 C CONTINUE
CONM0140 C DETERMINE CRACK DEPTH USING STRESSES
CONM0141 C
CONM0142 IF(I.GT.1) GO TO 12
CONM0143 C
CONM0144 C * * COMPUTE INITIAL VALUES OF STRESSES **
CONM0145 C
CONM0146 ICRK=0
CONM0147 IF(FE(I).GT.0.0) ICRK=1
CONM0148 KV=0
CONM0149 IF(ICRK.EQ.1) GO TO 10
CONM0150 HI=0.0
CONM0151 STSHI=STESS
CONM0152 GO TO 135
CONM0153 10 HF=0.0
CONM0154 STSHF=STESS
CONM0155 GO TO 135
CONM0156 12 IF(KV.EQ.1) GO TO 135
CONM0157 IF(ICRK.GT.0) GO TO 13
CONM0158 IF(FE(I).EQ.0.0) GO TO 14
CONM0159 HF=(I-1)*DY
CONM0160 STSHF=STESS
CONM0161 SLIM=FTC
CONM0162 IF(STSMAX(N,I).GT.0.0) SLIM=0.0
CONM0163 HK=HI+((SLIM-STSHI)/(STSHF-STSHI))*DY
CONM0164 XKD=HK
CONM0165 Y1=HK
CONM0166 Y2=PT
CONM0167 KV=1
C
C WE KNOW XKD AND HK
C
GO TO 135
14 HI=(I-1)*DY
CONM0172 STSHI=STESS
CONM0173 GO TO 135
CONM0174 13 IF(FE(I).GT.0.0) GO TO 16
CONM0175 HI=(I-1)*DY
CONM0176 STSHI=STESS
CONM0177 SLIM=FTC
CONM0178 IM1=I-1
CONM0179 IF(STSMAX(N,IM1).GT.0.0) SLIM=0.0
CONM0180 HK=HI-((SLIM-STSHI)/(STSHF-STSHI))*DY
CONM0181 XKD=PT-HK
CONM0182 Y1=0.0
CONM0183 Y2=HK
CONM0184 KV=1
C
C WE KNOW XKD AND HK
C
GO TO 135
16 HF=(I-1)*DY
CONM0190 STSHF=STESS
CONM0191 YY=YY+DY
CONM0192 135 IF(KV.EQ.0 .AND. ICRK.EQ.1) GO TO 18
CONM0193 IF(KV.EQ.0 .AND. ICRK.EQ.0) GO TO 19
CONM0194 GO TO 20
CONM0195 19 XKD=PT
CONM0196 HK=PT
CONM0197 Y1=HK
CONM0198 Y2=PT
CONM0199

```

```

CONM0200
CONM0201
CONM0202
CONM0203
CONM0204
CONM0205
CONM0206
CONM0207
CONM0208
CONM0209
CONM0210
CONM0211
CONM0212
CONM0213
CONM0214
CONM0215
CONM0216
CONM0217
CONM0218
CONM0219
CONM0220
CONM0221
CONM0222
CONM0223
CONM0224
CONM0225
CONM0226
CONM0227
CONM0228
CONM0229
CONM0230
CONM0231
CONM0232
CONM0233
CONM0234
CONM0235
CONM0236
CONM0237
CONM0238
CONM0239
CONM0240
CONM0241
CONM0242
CONM0243
CONM0244
CONM0245
CONM0246
CONM0247
CONM0248
CONM0249
CONM0250
CONM0251
CONM0252
CONM0253
CONM0254
CONM0255
CONM0256
CONM0257
CONM0258
CONM0259
CONM0260
CONM0261
CONM0262

```

```

18 GO TO 20
   HKD=0.0
   HK=0.0
   Y1=0.0
   Y2=HK
20 CONTINUE
C
C DETERMINE EFFECTIVE STEEL AREA
C
   IF(NOLIT.GT.0) GO TO 60
C
C COMPUTE PREVIOUS STRESSES AND STRAINS
C
   PTSNI=PSTRNI(N)+PDENOM*YTI/PT
   PTSNO=PSTRNI(N)+PDENOM*YTO/PT
   PTSSI=(PTSNI-PTSN(N,1))*PSE+PTSS(N,1)
   IF(ABS(PTSSI).LE.FSY) GO TO 61
   PTSSI=PTSSI*FSY/ABS(PTSSI)
61 PTSN(N,1)=PTSNI
   PTSS(N,1)=PTSSI
   PTSSO=(PTSNO-PTSN(N,2))*PSE+PTSS(N,2)
   IF(ABS(PTSSO).LE.FSY) GO TO 62
   PTSSO=PTSSO*FSY/ABS(PTSSO)
62 PTSN(N,2)=PTSNO
   PTSS(N,2)=PTSSO
60 PNI=PN
   PNO = PN
   IF(YTI.GT.Y1 .AND. YTI.LT.Y2) PNI = PN-1.
   IF(YTO.GT.Y1 .AND. YTO.LT.Y2) PNO = PN-1.
   WSI = 1.0
   WSO = 1.0
   TSNI = STRNI + DENOM*YTI/PT
   TSNO = STRNI + DENOM*YTO/PT
   IF(NONLIN.LT.3) GO TO 63
C
C COMPUTE STRESSES OF STEEL
C
   TSSI=(TSNI-PTSN(N,1))*PSE+PTSS(N,1)
   TSSO=(TSNO-PTSN(N,2))*PSE+PTSS(N,2)
   IF(ABS(TSSI).LE.FSY) GO TO 64
   TSSI=TSSI*FSY/ABS(TSSI)
64 IF(ABS(TSSO).LE.FSY) GO TO 65
   TSSO=TSSO*FSY/ABS(TSSO)
65 DFSTSI=ABS(TSSI-PTSS(N,1))
   IF(DFSTSI.GT.0.0001 .OR. ABS(TSSI).LT.FSY) GO TO 66
   WSI=0.0
   GO TO 67
66 WSI=(TSSI-PTSS(N,1))/(TSNI-PTSN(N,1))
   WSI=WSI/PSE
67 DFSTSO=ABS(TSSO-PTSS(N,2))
   IF(DFSTSO.GT.0.0001 .OR. ABS(TSSO).LT.FSY) GO TO 68
   WSO=0.0
   GO TO 69
68 WSO=(TSSO-PTSS(N,2))/(TSNO-PTSN(N,2))
   WSO=WSO/PSE
69 CONTINUE
   IF(WSI.LT.0.0000001) WSI=0.0000001
   IF(WSO.LT.0.0000001) WSO=0.0000001
63 CONTINUE
   APNI = ASI(N)*PNI*WSI
   APNO = ASO(N)*PNO*WSO
C
C INTEGRATE BASIC FUNCTIONS

```

```

CONM0263 C
CONM0264 SUM1=0.0
CONM0265 SUM2=0.0
CONM0266 SUM3=0.0
CONM0267 YY=0.0
CONM0268 DO 55 I=1,NINT,2
CONM0269 I2=I+1
CONM0270 I3=I+2
CONM0271 SUM1=FE(I)+4.0*FE(I2)+FE(I3)+SUM1
CONM0272 YY1=YY
CONM0273 YY2=YY+DY
CONM0274 YY3=YY+2.0*DY
CONM0275 SUM2=FE(I)*YY1+4.0*FE(I2)*YY2+FE(I3)*YY3+SUM2
CONM0276 SUM3=FE(I)*YY1**2+4.0*FE(I2)*YY2**2+FE(I3)*YY3**2+SUM3
CONM0277 YY=YY3
55 C
CONM0278 C
CONM0279 C COMPUTE PA, YBAR, PI
CONM0280 C
CONM0281 PAP=PA
CONM0282 PA=DY*SUM1/3.0+APNI*APNO
CONM0283 YBAR=(DY*SUM2/3.0+APNI*YTI+APNO*YTO)/PA
CONM0284 PI=(DY/3.0)*(SUM3-2.0*YBAR*SUM2+YBAR*YBAR*SUM1)
CONM0285 PI=PI+APNI*(YTI-YBAR)**2+APNO*(YTO-YBAR)**2
CONM0286 PAMIN=0.00001*PT
CONM0287 PIMIN=0.00001*PT
CONM0288 IF(PA.LT.PAMIN) PA=PAMIN
CONM0289 IF(PI.LT.PIMIN) PI=PIMIN
C
C CHECK INNER LOOP EXIT CONDITIONS.
C
   NTIMES = NTIMES + 1
   IF(NTIMES.GT.LIMIT) GO TO 50
   ERR = ABS(PA-PAP)/PT
   IF(ERR.LT.TOLER1) GO TO 50
   GO TO 1
C
C END INNER LOOP.
C
C COMPUTE CORRECTIONS OF MOMENTS AND THRUSTS FOR NEXT ITERATION
C
50 YY=0.0
   CORR=0.0
   CORR=0.0
   DO 136 I=1,NINTP
   IRAJP=0
   IRAJ=0
   SSM=-ABS(STSMAX(N,I))
   SNM=STNMAX(N,I)
   PSTAIN=PSTRNI(N)+PDENOM*YY/PT
   STAIN=STRNI+DENOM*YY/PT
   IF(STAIN.LT.PSTAIN) GO TO 136
   IF(SSM.LT.FYC) GO TO 137
   STESS=(STAIN-SM2)*PCE+FYC
   GO TO 138
137 STESS=(STAIN-SNM)*PCE+SSM
138 IF(STESS.GT.0.0 .AND. STSMAX(N,I).GT.0.0) IRAJ=1
   IF(STESS.GT.FTC) IRAJ=1
   PSTESS=(PSTAIN-SNM)*PCE+SSM
   IF(PSTESS.GT.FTC) IRAJP=1
   IF(PSTESS.GT.0.0 .AND. STSMAX(N,I).GT.0.0) IRAJP=1
   IF(IRAJP.EQ.1) GO TO 136
   IF(IRAJP.EQ.0) GO TO 136

```

```

CONM0326
CONM0327
CONM0328
CONM0329
CONM0330
CONM0331
CONM0332
CONM0333
CONM0334
CONM0335
CONM0336
CONM0337
CONM0338
CONM0339
CONM0340
CONM0341
CONM0342
CONM0343
CONM0344
CONM0345
CONM0346
CONM0347
CONM0348
CONM0349
CONM0350
CONM0351
CONM0352
CONM0353
CONM0354
CONM0355
CONM0356
CONM0357
CONM0358
CONM0359
CONM0360
CONM0361
CONM0362
CONM0363
CONM0364
CONM0365
CONM0366
CONM0367
CONM0368
CONM0369
CONM0370
CONM0371
CONM0372
CONM0373
CONM0374
CONM0375
CONM0376
CONM0377
CONM0378
CONM0379
CONM0380
CONM0381
CONM0382
CONM0383
CONM0384
CONM0385
CONM0386
CONM0387
CONM0388

```

```

HDY=DY
IF(I.EQ.1) HDY=DY/2.0
IF(I.EQ.NINTP) HDY=DY/2.0
HYY=YY
IF(I.EQ.1) HYY=HDY
IF(I.EQ.NINTP) HYY=YY-DY/2.0
CORRN=PSTESS*HDY+CORRN
CORRM=PSTESS*HDY*(YBAR-HYY)+CORRM
136 YY=YY+DY
CMOM(N)=CORRM
CTHR(N)=CORRN
C CHECK FOR CONVERGENCE OF LOAD STEP LOOP. THAT IS, FOR ITERATIONS
C WITHIN THE LOAD STEP, THE MOMENT AND THRUSTS WILL CONVERGE WHEN
C THE MATERIAL PROPERTIES HAVE CONVERGED.
C
ERROR = ABS(YBAR-RESULT(16,N))/PT
IF(ERROR.GT.TOLER2) ICON = 0
ERROR = ABS(PA-RESULT(14,N))/PT
IF(ERROR.GT.TOLER2) ICON = 0
IF(NOLIT.EQ.0 .AND. ABS(CMOM(N)).GT.0.0) ICON=0
IF(NOLIT.EQ.0 .AND. ABS(CTHR(N)).GT.0.0) ICON=0
C
C SAVE NEW ESTIMATES OF SECTION PROPERTIES.
C
RESULT(14,N) = PA
RESULT(15,N) = PI
RESULT(16,N) = YBAR
C
C SAVE STEEL STRESSES
C
RESULT(10,N) = TSSI
RESULT(11,N) = TSSO
IF(ASI(N).EQ.0.0) RESULT(10,N) = 0.0
IF(ASO(N).EQ.0.0) RESULT(11,N) = 0.0
C
C COMPUTE CONCRETE MAX COMPRESSION
C
RESULT(12,N) = PCE*STRNEG
IF(NONLIN.LT.2) GO TO 57
RESULT(12,N)=STSIN
IF(STSIN.GT.STSO) RESULT(12,N)=STSO
IF(RESULT(12,N).GT.FTC) RESULT(12,N)=FTC
C
C COMPUTE MAX SHEAR
C
SHAREA= PT-(PT-YTO+YTI)/2.0
57 RESULT(13,N)=RESULT(7,N)/SHAREA
C
C COMPUTE CRACK WIDTH. BASED ON GERGELY AND LUTZ FORMULA
C
TS=0.0
TB=0.0
IF(ASI(N).LE.0.0) GO TO 70
IF(TS.GT.TSSI) GO TO 70
TS=TSSI
TB=TBI(N)
GO TO 72
70 IF(ASO(N).LE.0.0) GO TO 71
72 IF(ASO(N).LE.0.0) GO TO 73
IF(TS.GT.TSSO) GO TO 73
TS=TSSO
TB=TBO(N)
73 CRWD=(.091*1.34*(TS-5000.0)*(2.0*TB*TB*SL)**0.333)/1000000.0

CONM0389 IF(CRWD.LT.0.0) CRWD=0.0
CONM0390 GO TO 74
CONM0391 71 CRWD=0.0
CONM0392 74 RESULT(17,N)=CRWD
CONM0393 C
CONM0394 C SAVE INNER AND OUTER STRAINS AND CRACK LENGTH
CONM0395 C
CONM0396 RESULT(18,N) = XKD
CONM0397 99 CONTINUE
CONM0398 RESULT(19,N) = STRMI
CONM0399 RESULT(20,N) = STRNO
CONM0400 100 CONTINUE
CONM0401 C
CONM0402 C AVERAGE PIPE PROPERTIES AT ELEMENT CENTERS FOR INPUT TO SOLUTIONS
CONM0403 C
CONM0404 DO 150 N=1,NPMAT
CONM0405 N1 = N+1
CONM0406 IF(N1.GT.NPPT) N1 = 1
CONM0407 PIPMAT(3,N) = 0.5*(RESULT(14,N)+RESULT(14,N1))
CONM0408 150 PIPMAT(4,N) = 0.5*(RESULT(15,N)+RESULT(15,N1))
CONM0409 C
CONM0410 RETURN
CONM0411 C
CONM0412 C** FORMATS
CONM0413 C
CONM0414 END
CONM0415 *
CONM0416 * CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
CONM0417 * VERSION 1.0, MARCH 1990
CONM0418 * PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
CONM0419 * THE FEDERAL HIGHWAY ADMINISTRATION
CONM0420 *
CONM0421 * SUBROUTINE CONV(T(D1,D2))
CONM0422 C
CONM0423 C THIS SUBROUTINE EXPRESSES ISOTROPIC PROPERTIES IN ANISOTROPIC FORM
CONM0424 C
CONM0425 * Uncomment this statement in all program modules for double
CONM0426 * precision version. SCM 3-10-90
CONM0427 *
CONM0428 * IMPLICIT DOUBLE PRECISION (A-H,O-Z)
CONM0429 *
CONM0430 COMMON /MATERL/ CP(3,3),DEN(10),E(10,10),GNU(10,10),H(10,10),
CONM0431 1 ITYPE(10),NON(10)
CONM0432 DO 10 I = 1, 3
CONM0433 DO 10 K = 1, 3
CONM0434 10 CP(I,K) = 0.0
CONM0435 IF(D2 .EQ. 0.5) D2=.495
CONM0436 CP(3,3)=D1*0.5/(1.0+D2)
CONM0437 D1=D1/((1.0+D2)*(1.0-2.0*D2))
CONM0438 CP(1,2)=D2*D1
CONM0439 CE(2,1)=CE(1,2)
CONM0440 D1=D1*(1.0-D2)
CONM0441 CP(1,1)=D1
CONM0442 CP(2,2)=D1
CONM0443 RETURN
CONM0444 END
CONM0445 *
CONM0446 * CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
CONM0447 * VERSION 1.0, MARCH 1990
CONM0448 * PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
CONM0449 * THE FEDERAL HIGHWAY ADMINISTRATION
CONM0450 *
CONM0451 SUBROUTINE DUNCAN(ST,STHARD,NEL,ICON,ITER,MN,NELEM,IA)

```

```

THIS SUBROUTINE GENERATES AN INCREMENTAL PLANE STRAIN
CONSTITUTIVE MATRIX BASED ON THE HYPERBOLIC SOIL MODEL
OF DUNCAN

SELIG HYPERBOLIC BULK MODULUS OPTION ADDED IN 1989.

Uncomment this statement in all program modules for double
precision version. SCM 3-10-90

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

DIMENSION ST(6,NELEM),STHARD(6,NELEM)
COMMON /MATERL/ CP(3,3),DEN(10),E(10,10),GKU(10,10),H(10,10),
S ITYPE(10),NON(10)

Add IBULK(10) to COMMON /HYPER/. MGK 9-29-88

COMMON /HYPER/ C(10),PHIO(10),DPHI(10),ZK(10),ZN(10),RF(10),
S VT(10),BK(10),BM(10),RATIO(10),IBULK(10)

Add common statement for output control. SCM 3-18-89

COMMON /OUTPUT/ IWRT,IPLOT,MAXIT,JCON

DATA TOLER,PATM /0.05,14.7/

STHARD(1,NEL)=TANGENT YOUNG'S MODULUS OF PREVIOUS LOAD STEP

STHARD(2,NEL)=TANGENT BULK MODULUS OF PREVIOUS LOAD STEP

STHARD(4,NEL) = Convergence flag. SCM 3-11-89
= -1.0 (Not converged)
= 1.0 (Converged)

STHARD(5,NEL)=TANGENT YOUNG'S MODULUS OF LAST ITERATION

STHARD(6,NEL)=TANGENT BULK MODULUS OF LAST ITERATION

RECOVER CURRENT STRESSES AND CONVERT TO PRINCIPAL

SXX = -ST(4,NEL)
SYY = -ST(5,NEL)
SKY = ST(6,NEL)

CENTR = (SXX + SYY)/2.0
RADIUS = SQRT(SKY**2 + ((SXX - SYY)/2.0)**2)
S1 = CENTR + RADIUS
S3 = CENTR - RADIUS

ON FIRST ITERATION OF EACH STEP UPDATE PARAMETERS

IF(ITER.NE.1) GO TO 5
E2 = STHARD(5,NEL)
E1 = E2
STHARD(1,NEL) = E1

B2=STHARD(6,NEL)
B1=B2
STHARD(2,NEL)=B1

```

```

DUNC0008      C
DUNC0009      C   FOR FIRST ITERATION OF A NEW ELEMENT ENTERING SYSTEM,
DUNC0010      C   ASSIGN INITIAL GUESS-VALUES
DUNC0011      C
DUNC0012      C   IF(E1.NE.0.0) GO TO 5
DUNC0013      C   POIS1 = VT(MN)
DUNC0014      C   IF(POIS1.EQ.0.0) POIS1 = 0.33
DUNC0015      C   S3 = 0.1*PATM
DUNC0016      C   S1 = S3*(1.0-POIS1)/POIS1
DUNC0017      C
DUNC0018      C   RECALL PREVIOUSLY CONVERGED PARAMETERS ON SUBSEQUENT ITERATIONS
DUNC0019      C
DUNC0020      C   5 IF(ITER.EQ.1) GO TO 10
DUNC0021      C   E1 = STHARD(1,NEL)
DUNC0022      C   B1 = STHARD(2,NEL)
DUNC0023      C
DUNC0024      C   SET WEIGHTING RATIOS FOR AVERAGING ENEW FROM E1 AND E2,
DUNC0025      C   AND BNEW FROM B1 AND B2
DUNC0026      C
DUNC0027      C   10 WTE = 0.5
DUNC0028      C   IF(E1.EQ.0.0) WTE = RATIO(MN)
DUNC0029      C
DUNC0030      C   WTP = 0.5
DUNC0031      C   IF(E1.EQ.0.0) WTP = RATIO(MN)
DUNC0032      C
DUNC0033      C   SET LOWER LIMIT FOR S3PA
DUNC0034      C
DUNC0035      C   S3PA=S3/PATM
DUNC0036      C   IF(S3PA.LT.0.1) S3PA=0.1
DUNC0037      C
DUNC0038      C   TENSION FAILURE CHECK
DUNC0039      C
DUNC0040      C   IF(S1.GT.0.0) GO TO 20
DUNC0041      C   E2=0.0025*ZK(MN)*PATM*S3PA**ZN(MN)
DUNC0042      C   POIS=0.40
DUNC0043      C   IF(VT(MN).GT.0.0) POIS=VT(MN)
DUNC0044      C   B2=E2/(3.0-6.0*POIS)
DUNC0045      C   E2P=STHARD(5,NEL)
DUNC0046      C   ETCHEK=(E2-E2P)/E2
DUNC0047      C   STHARD(5,NEL)=E2
DUNC0048      C   STHARD(6,NEL)=B2
DUNC0049      C   GO TO 30
DUNC0050      C   20 CONTINUE
DUNC0051      C
DUNC0052      C   COMPUTE YOUNGS TANGENT MODULUS E2, THEN ETNEW
DUNC0053      C
DUNC0054      C   XS3=PATM*S3PA
DUNC0055      C
DUNC0056      C   *   Revise intrinsic function to generic form. SCM 3-10-90
DUNC0057      C   *
DUNC0058      C   PHI = PHIO(MN) - DPHI(MN) * LOG10(S3PA)
DUNC0059      C   *
DUNC0060      C   RFF = RF(MN) * (1.0 - SIN(PHI))/
DUNC0061      C   1   (2.0 * C(MN) * COS(PHI) + 2.0 + XS3 * SIN(PHI))
DUNC0062      C   DEV = RFF * (S1 - S3)
DUNC0063      C   IF (DEV.LT.0.0) DEV = 0.0
DUNC0064      C   IF(DEV.GT.0.95) DEV = 0.95
DUNC0065      C   EINIT = ZK(MN) * PATM * S3PA**ZN(MN)
DUNC0066      C   E2 = (1.0 - DEV)**2*EINIT
DUNC0067      C   E2A=E2/3.0
DUNC0068      C   E2B=E2*8.0
DUNC0069      C
DUNC0070      C   USE UNDER RELAXATION FOR E2 WHEN ITER .GT. 2

```

DUNC0071
DUNC0072
DUNC0073
DUNC0074
DUNC0075
DUNC0076
DUNC0077
DUNC0078
DUNC0079
DUNC0080
DUNC0081
DUNC0082
DUNC0083
DUNC0084
DUNC0085
DUNC0086
DUNC0087
DUNC0088
DUNC0089
DUNC0090
DUNC0091
DUNC0092
DUNC0093
DUNC0094
DUNC0095
DUNC0096
DUNC0097
DUNC0098
DUNC0099
DUNC0100
DUNC0101
DUNC0102
DUNC0103
DUNC0104
DUNC0105
DUNC0106
DUNC0107
DUNC0108
DUNC0109
DUNC0110
DUNC0111
DUNC0112
DUNC0113
DUNC0114
DUNC0115
DUNC0116
DUNC0117
DUNC0118
DUNC0119
DUNC0120
DUNC0121
DUNC0122
DUNC0123
DUNC0124
DUNC0125
DUNC0126
DUNC0127
DUNC0128
DUNC0129
DUNC0130
DUNC0131
DUNC0132
DUNC0133


```

C      E2P = STHARD(5,NEL)
+
+      Revise relaxation function. SCM 1-15-89
+
+      IF(ITER.GT.2) E2 = 0.50*E2 + 0.50*E2P
+
+      IF(ITER.GT.2) E2 = E2**2/(E2+E2P) + E2P**2/(E2+E2P)
+      IF(ITER.GT.2 .AND. IBULK(MN).EQ.1) E2=0.5*E2+0.5*E2P
+
+
+      STHARD(5,NEL) = E2
+      ETCHEK = (E2 -E2P)/E2
+
+*****
+      Replace algorithm for calculation of tangent
+      bulk modulus with option for hyperbolic form.
+      MGK 9-29-88
+
+      COMPUTE TANGENT BULK MODULUS B2 IF VT=0
+
+      IF(VT(MN).GT.0.0) GO TO 30
+      B2=BK(MN)*PATM*S3PA**BM(MN)
+      IF(B2.LT.E2A) B2=E2A
+      IF(B2.GT.E2B) B2=E2B
+
+      IF POISSONS RATIO = 0, COMPUTE TANGENT BULK MODULUS, B2
+
+      IF (VT(MN) .GT. 0.0) KTYPE = 3
+      IF (VT(MN) .GT. 0.0) GO TO 30
+
+      IF IBULK = 0, COMPUTE B2 BY POWER LAW
+
+      IF (IBULK(MN) .NE. 0) GO TO 25
+      KTYPE = 1
+      B2 = BK(MN)*PATM*S3PA**BM(MN)
+      GO TO 28
+
+      IF IBULK > 0, DETERMINE MEAN STRESS AND COMPUTE B2
+      BY HYPERBOLIC LAW
+
+      25 KTYPE = 2
+      B2P = STHARD(6,NEL)
+      IF (B2P .EQ. 0.0) B2P = BK(MN)*PATM
+      E2P = STHARD(5,NEL)
+      POISX = 0.5 - E2P/(6.0*B2P)
+      SM = (1.0 + POISX)*(S1+S3)/3.0
+      B2 = BK(MN)*PATM*(1.0 + SM/(BK(MN)*PATM*BM(MN)))**2
+
+      SET LIMITS ON B2 SO THAT THE NEXT POISSON RATIO REMAINS
+      BOUNDED BETWEEN 0.0 AND 0.48
+
+      28 IF (B2 .LT. E2A) B2 = E2A
+      IF (B2 .GT. E2B) B2 = E2B
+
+*****
+      USE UNDER RELAXATION FOR B2 WHEN ITER .GT. 2
+
+      B2P = STHARD(6,NEL)
+
+      Revise relaxation function. SCM 1-15-89
DUNC0134
DUNC0135
DUNC0136
DUNC0137
DUNC0138
DUNC0139
DUNC0140
DUNC0141
DUNC0142
DUNC0143
DUNC0144
DUNC0145
DUNC0146
DUNC0147
DUNC0148
DUNC0149
DUNC0150
DUNC0151
DUNC0152
DUNC0153
DUNC0154
DUNC0155
DUNC0156
DUNC0157
DUNC0158
DUNC0159
DUNC0160
DUNC0161
DUNC0162
DUNC0163
DUNC0164
DUNC0165
DUNC0166
DUNC0167
DUNC0168
DUNC0169
DUNC0170
DUNC0171
DUNC0172
DUNC0173
DUNC0174
DUNC0175
DUNC0176
DUNC0177
DUNC0178
DUNC0179
DUNC0180
DUNC0181
DUNC0182
DUNC0183
DUNC0184
DUNC0185
DUNC0186
DUNC0187
DUNC0188
DUNC0189
DUNC0190
DUNC0191
DUNC0192
DUNC0193
DUNC0194
DUNC0195
DUNC0196
+
+      IF(ITER.GT.2) B2 = 0.50*B2 + 0.50*B2P
+
+      IF(ITER.GT.2) B2 = B2**2/(B2+B2P) + B2P**2/(B2+B2P)
+      IF(ITER.GT.2 .AND. IBULK(MN).EQ.1) B2=0.5*B2+0.5*B2P
+
+
+      STHARD(6,NEL) = B2
+
+      AVERAGE LOAD STEP MODULI
+
+      30 ETNEW=(1.0-WTE)*E1+WTE*E2
+      IF(VT(MN).GT.0.0) GO TO 40
+      BNEW=(1.0-WTP)*B1+WTP*B2
+
+      FIND EQUIVALENT POISSON'S RATIO
+
+      PNEW=(3.0*BNEW-ETNEW)/(6.0*BNEW)
+      IF(PNEW.LT.0.0) PNEW=0.0
+      IF(PNEW.GT.0.48) PNEW=0.48
+      GO TO 50
+
+      CHECK FOR CONVERGENCE (ASSUME B2 CONVERGES WHEN E2 DOES)
+
+      40 PNEW=VT(MN)
+      BNEW=ETNEW/(3.0-6.0*PNEW)
+
+      Revise logic to force iteration if arbitrary element has not
+      converged.
+      SCM 3-10-89
+
+      50 IF(ABS(ETCHEK).GT.TOLER) ICON = 0
+
+      50 IF(ABS(ETCHEK).GT.TOLER) THEN
+      STHARD(4,NEL) = -1.0
+      ELSE
+      STHARD(4,NEL) = 1.0
+      END IF
+
+      DO 55, NNEL=1,NELEM
+      IF(STHARD(4,NNEL).LT.0.0) ICON = 0
+      55 CONTINUE
+
+      IF(ITER.EQ.1) ICON = 0
+      JCON = ICON
+      IF(IABS(NON(MN)).GT.MAXIT) MAXIT = IABS(NON(MN))
+      IF(ITER.GE.IABS(NON(MN))) ICON = 1
+
+      Calculate stress level. MGK 2-16-89
+
+      SLEV < -1.0: Tension failure, (Complete)
+
+      SLEV > 1.0: Shear failure
+
+      IF(S1.LT.0.0) SLEV = -1.0 + S1
+      IF(S1.LT.0.0) GO TO 60
+      SHEAR = RF(MN)/RFF
+      SLEV = (S1-S3)/SHEAR
+
+      60 CONTINUE
+
+      CONVERT TO PLANE STRAIN MATERIAL MATRIX
+
+      CMOD = ETNEW
DUNC0197
DUNC0198
DUNC0199
DUNC0200
DUNC0201
DUNC0202
DUNC0203
DUNC0204
DUNC0205
DUNC0206
DUNC0207
DUNC0208
DUNC0209
DUNC0210
DUNC0211
DUNC0212
DUNC0213
DUNC0214
DUNC0215
DUNC0216
DUNC0217
DUNC0218
DUNC0219
DUNC0220
DUNC0221
DUNC0222
DUNC0223
DUNC0224
DUNC0225
DUNC0226
DUNC0227
DUNC0228
DUNC0229
DUNC0230
DUNC0231
DUNC0232
DUNC0233
DUNC0234
DUNC0235
DUNC0236
DUNC0237
DUNC0238
DUNC0239
DUNC0240
DUNC0241
DUNC0242
DUNC0243
DUNC0244
DUNC0245
DUNC0246
DUNC0247
DUNC0248
DUNC0249
DUNC0250
DUNC0251
DUNC0252
DUNC0253
DUNC0254
DUNC0255
DUNC0256
DUNC0257
DUNC0258
DUNC0259

```

```

CALL CONVCT(CMOD,PNEW,PNEW)
*
MATNO = MN
*
C
C
C
*
PRINT-OUT RECORD OF NONLINEAR MODULI (NON(MN).LT.0)
*
Revise unit numbers, variables and formats for soil modulus
output. SCM 1-9-89
*
IF(IWST.LE.1) GO TO 100
IF(ITXX.NE.ITER) WRITE(16,6000) IA,ITER
IF(ITXX.NE.ITER) WRITE(16,6020)
*
ITXX = ITER
*
WRITE(16,6030) NEL,MATNO,KTYPE,JCON,ETCHECK,SLEV,
X
ETNEW,BNEW,PNEW,CMOD
*
100 CONTINUE
RETURN
*
C
C
C
*
FORMATS
*
6000 FORMAT(1HL,/,/,6X,23HCONSTRUCTION INCREMENT ,I2,5X,
X
10HITERATION ,I2)
6020 FORMAT(1H0,15X,47H* * * ELASTIC MODULI AND CONVERGENCE PARAM
X28H FOR DUNCAN SUBROUTINE * * *,/
X22X,69H(CONVERGENCE ERRORS (ERROR-EMOD) ARE RATIOS GIVEN BY,
XOLD)/NEW ),/
X22X,42HK-TYPE = 1 : DUNCAN POWER LAW BULK MODULUS,
X10X,45HSTRESS-LEV <= -1.0 : COMPLETE TENSION FAILURE,/
X22X,42HK-TYPE = 2 : SELIG HYPERBOLIC BULK MODULUS,
X10X,34HSTRESS-LEV >= +1.0 : SHEAR FAILURE,/
X22X,37HK-TYPE = 3 : CONSTANT POISSON'S RATIO,//
X3X,5HELE-N,4X,5HMAT-N,4X,6HR-TYPE,4X,4HICON,4X,
X10HERROR-EMOD,4X,10HSTRESS-LEV,4X,10HYOUNGS-MOD,6X,8HBULK-MOD
X4X,10HPOIS-RATIO,4X,12HCONFINED-MOD,/)
6030 FORMAT(3X,I4,5X,I4,7X,I2,7X,I2,5X,F10.4,4X,F10.4,2X,E12.5,
X
2X,E12.5,7X,F5.3,6X,E12.5)
*
END
*
*****
*
CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
*
VERSION 1.0, MARCH 1990
*
PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
*
THE FEDERAL HIGHWAY ADMINISTRATION
*
*****
*
SUBROUTINE EMOD (IA,ICON,NPMAT,NPPT,PA,PE1,PE2,PESUM,PI,
1
PIPMAT,PNU,PS,EYIELD,RESULT,ICOME)
*
THIS ROUTINE DETERMINES NONLINEAR BENDING PROPERTIES FOR A
HOMOGENOUS MATERIAL WITH A BILINEAR STRESS-STRAIN CURVE. THE
FOLLOWING RESULT MATRIX IS IDENTIFIED ...
*
RESULT(10,N) = BENDING STRESS IN OUTER FIBER
RESULT(11,N) = THRUST STRESS (AVERAGE OVER SECTION)
RESULT(12,N) = SHEAR STRESS (AVERAGE OVER SECTION)
RESULT(13,N) = FRACTION OF SECTION YIELDED
RESULT(14,N) = MODIFIED THRUST AREA (PASTAR)
RESULT(15,N) = MODIFIED MOMENT OF INERTIA (PISTAR)
RESULT(16,N) = DISTANCE TO BENDING AXIS (YBAR)

```

DUNC0260	C	RESULT(17,N) = MAX BENDING STRESS ABOVE YIELD	EMOD0021
DUNC0261	C	RESULT(18,N) = MAXIMUM FIBER STRAIN/ YIELD STRAIN	EMOD0022
DUNC0262	C	RESULT(19,N) = MOMENT INCREMENT, OR CURRENT INNER STRAIN	EMOD0023
DUNC0263	C	RESULT(20,N) = THRUST INCREMENT, OR CURRENT OUTER STRAIN	EMOD0024
DUNC0264	C	ESTRNI(N) = INNER WALL STRAIN OF PREVIOUS CONVERGED STEP	EMOD0025
DUNC0265	C	PSTRNO(N) = OUTER WALL STRAIN OF PREVIOUS CONVERGED STEP	EMOD0026
DUNC0266	C	YEPR(1,N) = DISTANCE TO BOTTOM OF ELASTIC ZONE (PRE. CON. STEP)	EMOD0027
DUNC0267	C	YEPR(2,N) = DISTANCE TO TOP OF ELASTIC ZONE (PRE. CON. STEP)	EMOD0028
DUNC0268	C		EMOD0029
DUNC0269	*	Uncomment this statement in all program modules for double	EMOD0030
DUNC0270	*	precision version. SCM 3-10-90	EMOD0031
DUNC0271	*		EMOD0032
DUNC0272	*	IMPLICIT DOUBLE PRECISION (A-H,O-Z)	EMOD0033
DUNC0273	*		EMOD0034
DUNC0274	*	Add XMOMPR and DELNM arrays to dimension statement.	EMOD0035
DUNC0275	*	MGK 2-19-89	EMOD0036
DUNC0276	*		EMOD0037
DUNC0277	*	Revise dimensions of PSTRNI, PSTRNO, YEPR, YMOMPR and DELNM	EMOD0038
DUNC0278	*	arrays from 30 to 99. SCM 4-30-89	EMOD0039
DUNC0279	*		EMOD0040
DUNC0280		DIMENSION PIEMAT(5,NPMAT),RESULT(20,NPPT),PSTRNI(99),PSTRNO(99),	EMOD0041
DUNC0281	X	YEPR(4,99),XMOMPR(99),DELMN(99)	EMOD0042
DUNC0282	*		EMOD0043
DUNC0283		DATA LIMIT,TOLER/10,0.05/	EMOD0044
DUNC0284		GO TO (1,1,200),ICOME	EMOD0045
DUNC0285	C		EMOD0046
DUNC0286	C	ON FIRST ITERATION OF FIRST LOAD STEP, COMPUTE CONSTANTS	EMOD0047
DUNC0287	C		EMOD0048
DUNC0288	1	IF(IA.GT.1) GO TO 10	EMOD0049
DUNC0289	-DUNC0289	IF(ICON.GT.1) GO TO 10	EMOD0050
DUNC0290		POIS = 1.-PMU*PMU	EMOD0051
DUNC0291		PSE1 = PE1/POIS	EMOD0052
DUNC0292		PSE2 = PE2/POIS	EMOD0053
DUNC0293		IF(PSE2.EQ.0.0) PSE2 = PSE1/(1.E+10)	EMOD0054
DUNC0294		RATIO = PSE2/PSE1	EMOD0055
DUNC0295		EPSY = PYIELD/PSE1	EMOD0056
DUNC0296		YTOP = 2.0*PI/PS	EMOD0057
DUNC0297	C		EMOD0058
DUNC0298	C	INITIALIZE PREVIOUS CONVERGED STEP RESPONSES	EMOD0059
DUNC0299	C		EMOD0060
DUNC0300		DO 5 N=1,NPPT	EMOD0061
DUNC0301	*		EMOD0062
DUNC0302	*	Initialize new array XMOMPR. MGK 2-19-89	EMOD0063
EMOD0001	*		EMOD0064
EMOD0002		XMOMPR(N) = 0.0	EMOD0065
EMOD0003	*		EMOD0066
EMOD0004		YEPR(1,N) = 0.0	EMOD0067
EMOD0005		YEPR(2,N) = YTOP	EMOD0068
EMOD0006		PSTRNI(N) = 0.0	EMOD0069
EMOD0007	S	PSTRNO(N) = 0.0	EMOD0070
EMOD0008	C		EMOD0071
EMOD0009	C	BEGIN OUTER LOOP ITERATION FOR MOMENT AND THRUST CONVERGENCE	EMOD0072
EMOD0010	C		EMOD0073
EMOD0011	10	ERRSUM = 0.0	EMOD0074
EMOD0012		DO 100 N=1,NPPT	EMOD0075
EMOD0013		PASTAR = RESULT(14,N)	EMOD0076
EMOD0014		PISTAR = RESULT(15,N)	EMOD0077
EMOD0015		YBAR = RESULT(16,N)	EMOD0078
EMOD0016		DELM = RESULT(19,N)	EMOD0079
EMOD0017		DELN = RESULT(20,N)	EMOD0080
EMOD0018		PDENOM = PSTRNO(N)-PSTRNI(N)	EMOD0081
EMOD0019		YT1 = YEPR(1,N)	EMOD0082
EMOD0020		YT2 = YEPR(2,N)	EMOD0083


```

YE1 = YT1
YE2 = YT2
NTIMES = 0

*
* Revise moment summation logic. MGK 2-19-89
*
DELM(N) = DELN*(YTOP/2.0 - YBAR)
RESULT(5,N) = RESULT(5,N) + XMOMPR(N) + DELNM(N)

C
C
C BEGIN INNER LOOP FOR DETERMINING PASTAR, PISTAR AND YBAR BASED ON
C CURRENT ASSUMED VALUES OF MOMENT AND THRUST.
C (FIRST DETERMINE TOTAL STRAIN DISTRIBUTION)
20 THETA = DELM/(PISTAR*PSE1)
STNORM = DELN/(PASTAR*PSE1)
STRNI = STNORM + THETA*YBAR + PSTRNI(N)
STRNO = STNORM + THETA*(YBAR-YTOP) + PSTRNO(N)
DENOM = STRNO-STRNI
IF(DELM.EQ.0.0 .AND. DELN.EQ.0.0) GO TO 60

C
C COMPUTE CURRENT ELASTIC ZONE DEFINED BY YE1 AND YE2.
C
IF(DENOM.EQ.0.0) DENOM = 0.000001*STNORM
YE2 = YTOP*(EPSY-STRNI)/DENOM
YE1 = YTOP*(-EPSY-STRNI)/DENOM
IF(YE2.GT.YE1) GO TO 25
TEMP = YE1
YE1 = YE2
YE2 = TEMP
25 CONTINUE
IF(YE1.GT.YTOP) YE1 = YTOP
IF(YE2.GT.YTOP) YE2 = YTOP
IF(YE1.LT.0.0) YE1 = 0.0
IF(YE2.LT.0.0) YE2 = 0.0

C
C
C COMPARE SIZE OF PREVIOUS ELASTIC ZONE WITH CURRENT ELASTIC ZONE.
C IF NEW ZONE IS NOT SMALLER ASSUME ELASTIC RESPONSE.
C (NOTE THIS ALSO IMPLIES ELASTIC UNLOADING)
C
IF(YT2-YT1 .GE. YE2-YE1) GO TO 30
IF (YT2-YT1 .EQ. 0.0) GO TO 30
PASTAR = PA
YBAR = YTOP/2.
PISTAR = PI
YE2 = YTOP
YE1 = 0.0
GO TO 50

C
C
C DETERMINE UPPER AND LOWER TRANSITION ZONE WEIGHTING FACTORS
30 IF(YT2.LT.YE2) YT2 = YE2
IF(YT1.GT.YE1) YT1 = YE1
YAVG1 = (YT1+YE1) /2.0
EPMAX1 = ABS(STRNI+DENOM*YAVG1/YTOP)
EPMIN1 = ABS(PSTRNI(N) + PDENOM*YAVG1/YTOP)
DIFF1 = ABS(EPMAX1-EPMIN1)

*
* Add logic to avoid division by zero. SCM 12-14-89
*
IF(DIFF1.EQ.0.0) THEN
    WT1 = 0.0
ELSE

```

```

EMOD0084
EMOD0085
EMOD0086
EMOD0087
EMOD0088
EMOD0089
EMOD0090
EMOD0091
EMOD0092
EMOD0093
EMOD0094
EMOD0095
EMOD0096
EMOD0097
EMOD0098
EMOD0099
EMOD0100
EMOD0101
EMOD0102
EMOD0103
EMOD0104
EMOD0105
EMOD0106
EMOD0107
EMOD0108
EMOD0109
EMOD0110
EMOD0111
EMOD0112
EMOD0113
EMOD0114
EMOD0115
EMOD0116
EMOD0117
EMOD0118
EMOD0119
EMOD0120
EMOD0121
EMOD0122
EMOD0123
EMOD0124
EMOD0125
EMOD0126
EMOD0127
EMOD0128
EMOD0129
EMOD0130
EMOD0131
EMOD0132
EMOD0133
EMOD0134
EMOD0135
EMOD0136
EMOD0137
EMOD0138
EMOD0139
EMOD0140
EMOD0141
EMOD0142
EMOD0143
EMOD0144
EMOD0145
EMOD0146

WT1 = (EPSY-EPMIN1 + RATIO*(EPMAX1-EPSY))/DIFF1
END IF

IF(WT1.LT.0.0) WT1 = 0.0
IF(WT1.GT.1.0) WT1 = 1.0

YAVG2 = (YE2+YT2)/2.0
EPMAX2 = ABS(STRNI+DENOM*YAVG2/YTOP)
EPMIN2 = ABS(PSTRNI(N)+PDENOM*YAVG2/YTOP)
DIFF2 = ABS(EPMAX2-EPMIN2)

*
* Add logic to avoid division by zero. SCM 12-14-89
*
IF(DIFF2.EQ.0.0) THEN
    WT2 = 0.0
ELSE
    WT2 = (EPSY-EPMIN2+RATIO*(EPMAX2-EPSY))/DIFF2
END IF

IF(WT2.LT.0.0) WT2 = 0.0
IF(WT2.GT.1.0) WT2 = 1.0

C
C DETERMINE PASTAR BASED ON CONTRIBUTIONS FROM ELASTIC ZONE,
C TRANSITION ZONES AND YIELDING ZONES.
C
PAPR = PASTAR
PASTAR = (YE2-YE1) + RATIO*(YTOP-YT2+YT1)
PASTAR = PASTAR + WT1*(YE1-YT1) + WT2*(YT2-YE2)
PASTAR = PASTAR*PA/YTOP

C
C DETERMINE YBAR FROM ALL ZONE CONTRIBUTIONS.
C
IF(PASTAR.LE.0.0) GO TO 40
YBAR = (YE2**2-YE1**2) + RATIO*(YTOP**2-YT2**2 + YT1**2)
YBAR = YBAR + WT1*(YE1**2-YT1**2) + WT2*(YT2**2-YE2**2)
YBAR = YBAR*PA/(2.*YTOP*PASTAR)

C
C DETERMINE PISTAR FROM ALL ZONE CONTRIBUTIONS
C
YEB1 = (YE1-YBAR)**3
YEB2 = (YE2-YBAR)**3
YTB1 = (YT1-YBAR)**3
YTB2 = (YT2-YBAR)**3
YTOPB = (YTOP-YBAR)**3
PISTAR = (YEB2-YEB1) + RATIO*(YTOPB-YTB2+YTB1+ YBAR**3)
PISTAR = PISTAR + WT1*(YEB1-YTB1) + WT2*(YTB2-YEB2)
PISTAR = PISTAR*PA/(3.*YTOP)

C
C PROPERTIES FOR COMPLETE PLASTIC ZONE WHEN PASTAR = 0.0
C
GO TO 45
40 PASTAR = PA*RATIO
YBAR = YTOP/2.0
PISTAR = PI*RATIO
45 CONTINUE

C
C CHECK INNER LOOP EXIT CONDITIONS
C
NTIMES = NTIMES + 1
IF (NTIMES .GT. 10) GO TO 50
ERR = ABS(PASTAR-PAPR)/PASTAR
IF(ERR.LT.TOLER) GO TO 50
GO TO 20

```

```

EMOD0147
EMOD0148
EMOD0149
EMOD0150
EMOD0151
EMOD0152
EMOD0153
EMOD0154
EMOD0155
EMOD0156
EMOD0157
EMOD0158
EMOD0159
EMOD0160
EMOD0161
EMOD0162
EMOD0163
EMOD0164
EMOD0165
EMOD0166
EMOD0167
EMOD0168
EMOD0169
EMOD0170
EMOD0171
EMOD0172
EMOD0173
EMOD0174
EMOD0175
EMOD0176
EMOD0177
EMOD0178
EMOD0179
EMOD0180
EMOD0181
EMOD0182
EMOD0183
EMOD0184
EMOD0185
EMOD0186
EMOD0187
EMOD0188
EMOD0189
EMOD0190
EMOD0191
EMOD0192
EMOD0193
EMOD0194
EMOD0195
EMOD0196
EMOD0197
EMOD0198
EMOD0199
EMOD0200
EMOD0201
EMOD0202
EMOD0203
EMOD0204
EMOD0205
EMOD0206
EMOD0207
EMOD0208
EMOD0209

```

```

C      END INNER LOOP
C
C      SUM ERROR BETWEEN SUCCESSIVE SOLUTIONS, THAT IS, FOR ITERATIONS
C      WITHIN A LOAD STEP, MOMENT AND THRUST WILL CONVERGE WITH YBAR.
C
50  ERROR = ABS(PASTAR-RESULT(14,N))/PASTAR
      ERRSUM = ERRSUM + ERROR
C
C      SAVE NEW ESTIMATES OF SECTION PROPERTIES
C
60  RESULT(14,N) = PASTAR
      RESULT(15,N) = PISTAR
      RESULT(16,N) = YBAR
      RESULT(19,N) = STRNI
      RESULT(20,N) = STRNO
      YEPR(3,N) = YEL
      YEPR(4,N) = YEZ
C
C      COMPUTE NONLINEAR CHARACTERISTICS (PRINTED OUT ON CONVERGENCE)
C      (1) FRACTION OF SECTION YIELDED
C      (2) BENDING STRESS ABOVE YIELD STRESS (AVERAGES)
C      (3) MAXIMUM FIBER STRAIN/YIELD STRAIN
C
      ELAST = (YE2-YE1)/YTOP
      IF(ELAST.GT.0.999) ELAST = 1.0
      RESULT(13,N) = 1.-ELAST
C
      RESULT(17,N) = 0.0
      DSTRN = 0.5*ABS(STRNO-STRNI)-EPSY
      IF(DSTRN.GT.0.0) RESULT(17,N) = DSTRN*PSE2
C
      * Revise intrinsic function to generic form. SCM 3-10-90
      *
      * STNMAX = MAX(ABS(STRNI),ABS(STRNO))
      *
      * RESULT(18,N) = STNMAX/EPSY
C
100  CONTINUE.
C
C      CHECK ON OUTER LOOP CONVERGENCE
C
      IF(ERRSUM.LT.TOLER) ICON=-1
      IF(ICON.GT.LIMIT) WRITE(6,6500) ERRSUM
      IF(ICON.GT.LIMIT .AND. ERRSUM.LE.0.1) ICON = -1
      IF(ICON.GT.LIMIT .AND. ERRSUM.GT.0.1) ICON=-2
C
C      AVERAGE PIPE PROPERTIES AT ELEMENT CENTERS FOR INPUT TO NEXT
C      SOLUTION AND GET EFFECTIVE MODULUS FOR BUCKLING THEORY
C
      PESUM = 0.0
      DO 150 N=1,NPMAT
      N1 = N+1
      IF(N1.GT.NPPT) N1 = 1
      PIPMAT(3,N) = 0.5*(RESULT(14,N)+RESULT(14,N1))
      PIPMAT(4,N) = 0.5*(RESULT(15,N)+RESULT(15,N1))
150  PESUM = PESUM + PIPMAT(4,N)
      XMAT = NPMAT
      PESUM = PESUM*PI/(XMAT*PI)
C
C      RETURN TO PIPE ROUTINE
C
      RETURN

```

```

EMOD0210
EMOD0211 C      SECOND ENTRY, SAVE CONVERGED STRAINS AND ELASTIC ZONE MEASURES EMOD0273
EMOD0212 C EMOD0274
EMOD0213 C 200 DO 175 N=1,NEPT EMOD0275
EMOD0214 + EMOD0276
EMOD0215 * Save accumulated moment correction. MGK 2-19-89 EMOD0277
EMOD0216 * EMOD0278
EMOD0217 * XMOMPR(N) = XMOMPR(N) + DELNM(N) EMOD0279
EMOD0218 * EMOD0280
EMOD0219 * PSTRNI(N) = RESULT(19,N) EMOD0281
EMOD0220 * PSTRNO(N) = RESULT(20,N) EMOD0282
EMOD0221 * YEPR(1,N) = YEPR(3,N) EMOD0283
EMOD0222 175 YEPR(2,N) = YEPR(4,N) EMOD0284
EMOD0223 C EMOD0285
EMOD0224 C RETURN TO PIPE SUBROUTINE EMOD0286
EMOD0225 C EMOD0287
EMOD0226 * RETURN EMOD0288
EMOD0227 C EMOD0289
EMOD0228 * EMOD0290
EMOD0229 * Revise wording of message to include "PIPE WALL" to avoid EMOD0291
EMOD0230 * confusion with other nonlinear models. SCM 3-18-89 EMOD0292
EMOD0231 * EMOD0293
EMOD0232 * EMOD0294
EMOD0233 6500 FORMAT(1H1,///58(2H *)///112H NONLINEAR PIPE WALL MODULUS DID NOTEMOD0295
EMOD0234 X CONVERGE, SOLUTION WILL CONTINUE WITH A RATIO OF ERROR IN THE MODEMOD0296
EMOD0235 XULUS OF ,E15.5///61H IF ERROR RATIO IS LESS THAN 0.1, SOLUTIONEMOD0297
EMOD0236 X MAY BE ACCEPTABLE ///58(2H *)) EMOD0298
EMOD0237 END EMOD0299
EMOD0238 * * * * * ESTB0001
EMOD0239 * CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM ESTB0002
EMOD0240 * VERSION 1.0, MARCH 1990 ESTB0003
EMOD0241 * PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY ESTB0004
EMOD0242 * THE FEDERAL HIGHWAY ADMINISTRATION ESTB0005
EMOD0243 * * * * * ESTB0006
EMOD0244 SUBROUTINE ESTAB ESTB0007
EMOD0245 C ESTB0008
EMOD0246 C TRANSFORMATION RELATIONSHIPS BETWEEN ELEMENT AND SUB-ELEMENT ESTB0009
EMOD0247 C UNKNOWNNS ESTABLISHED ESTB0010
EMOD0248 * ESTB0011
EMOD0249 * Uncomment this statement in all program modules for double ESTB0012
EMOD0250 * precision version. SCM 3-10-90 ESTB0013
EMOD0251 * ESTB0014
EMOD0252 * IMPLICIT DOUBLE PRECISION (A-H,O-Z) ESTB0015
EMOD0253 * ESTB0016
EMOD0254 COMMON /ELEMNT/ AS(3),BS(3),C1(17,17),EST(4,3),RQ(4),RT(3), ESTB0017
EMOD0255 1 STORE(17),Z4(12),ZQ(4),ZT(3),ZY(17),J1,RO ESTB0018
EMOD0256 COMMON /ESTABL/ ITRW(28,2),NETCL(17,2),T(28,2) ESTB0019
EMOD0257 DO 55 I=1,2 ESTB0020
EMOD0258 DO 50 J=1,28 ESTB0021
EMOD0259 ITRW(J,I)=1 ESTB0022
EMOD0260 50 T(J,I)=1.0 ESTB0023
EMOD0261 DO 55 J=1,17 ESTB0024
EMOD0262 55 NETCL(J,I)=1 ESTB0025
EMOD0263 DO 57 I=1,3 ESTB0026
EMOD0264 ITRW(I+ 8,1)=I+6 ESTB0027
EMOD0265 ITRW(I+12,1)=I+9 ESTB0028
EMOD0266 ITRW(I+18,1)=I+6 ESTB0029
EMOD0267 ITRW(I+21,1)=I+9 ESTB0030
EMOD0268 ITRW(I+24,1)=I+6 ESTB0031
EMOD0269 ITRW(I+ 9,2)=I+6 ESTB0032
EMOD0270 ITRW(I+13,2)=I+9 ESTB0033
EMOD0271 ITRW(I+19,2)=I+9 ESTB0034
EMOD0272 ITRW(I+22,2)=I+6 ESTB0035
57 ITRW(I+25,2)=I+9 ESTB0036

```

```

NETCL( 9,1)=3
NETCL(11,1)=3
NETCL(15,1)=6
NETCL(16,1)=3
NETCL(10,2)=3
NETCL(12,2)=3
NETCL(16,2)=3
NETCL(17,2)=6
ITRW(2,1)=4
ITRW(3,1)=2
ITRW(4,1)=5
ITRW(7,1)=3
ITRW(8,1)=6
ITRW(17,1)=7
ITRW(18,1)=10
T(5,1)=0.0
T(6,1)=0.0
T(12,1)=0.0
T(16,1)=0.0
T(28,1)=0.0
ITRW(3,2)=2
ITRW(4,2)=5
ITRW(5,2)=3
ITRW(6,2)=6
ITRW(8,2)=4
ITRW(17,2)=9
ITRW(18,2)=12
T(1,2)=0.0
T(2,2)=0.0
T( 9,2)=0.0
T(13,2)=0.0
T(19,2)=0.0
RETURN
END
* * * * *
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
*      VERSION 1.0, MARCH 1990
*      PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
*      THE FEDERAL HIGHWAY ADMINISTRATION
* * * * *
SUBROUTINE GENEL (NOD, LGEN, NUMNP, NUMEL, MAXEL, KPUTC,
1      MGENPR, MBAND, LSTOP, LUPREP)
*
* Uncomment this statement in all program modules for double
* precision version. SCM 3-10-90
*
* IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
* DIMENSION NOD(LSTOP,1), LGEN(4,1)
* DIMENSION IX(7),KINDLG(3),ETYP(5)
* DATA ETYP/4HQQUAD,4HTRI,4HBEAM,4HFACE,4HBRD /
* DATA L1H, I1H, N1H, IBLANK, M1H/1HL,1HI,1HN,1H ,1HM/
C
C** INITIALIZE PARAMETERS, READ AND CHECK ELEMENT CARDS
C
IF (MGENPR .GE. 2) WRITE (6,6005)
MAXEL = NUMEL
LIMIT = 0
MBAND = 0
NUMEL = 0
MPRIOR = 1
IPRIOR = 1
JPRIOR = 0
ESTB0037      DO 20 NE = 1,MAXEL
ESTB0038      20 LGEN(1,NE) = 0
ESTB0039      40 IF (LIMIT .EQ. L1H) RETURN
ESTB0040      READ (LUPREP,5015) LIMIT, NE, (IX(K), K = 1,7), KINDLG
ESTB0041      IF (NE .LE. NUMEL .OR. NE .GT. MAXEL) GO TO 520
ESTB0042      IF (NUMEL .EQ. 0 .AND. (NE .NE. 1 .OR. IX(1) .EQ. 0)) GO TO 540
ESTB0043      C
ESTB0044      C** STORE DATA RELATING TO MATERIAL OR INCREMENT CHANGES
ESTB0045      C
ESTB0046      IF (IX(1) .NE. 0) GO TO 60
ESTB0047      LGEN(1,NE) = NE
ESTB0048      LGEN(2,NE) = IX(5)
ESTB0049      LGEN(3,NE) = IX(6)
ESTB0050      LGEN(4,NE) = IX(7)
ESTB0051      IF (LIMIT .EQ. L1H) GO TO 560
ESTB0052      GO TO 40
ESTB0053      C
ESTB0054      C** SETUP PARAMETERS FOR ELEMENT GENERATION
ESTB0055      C
ESTB0056      60 INTRAL = KINDLG(1)
ESTB0057      NUMLAY = KINDLG(2)
ESTB0058      INTERL = KINDLG(3)
ESTB0059      IF (INTRAL .EQ. 0) INTRAL = 1
ESTB0060      IF (NUMLAY .LE. 1) GO TO 80
ESTB0061      LINLAY = (NE-LPRIOR +NUMLAY-1)/ NUMLAY
ESTB0062      C
ESTB0063      C** DETERMINE TYPE OF GENERATION TO ACCOMPLISH
ESTB0064      C
ESTB0065      80 NUMEL = NUMEL +1
ESTB0066      IF (NE .GT. NUMEL) GO TO 140
ESTB0067      C
ESTB0068      C** INITIALIZE PARAMETERS FOR LAST CARD READ IX(1) .NE. 0
ESTB0069      C
ESTB0070      KGEN = I1H
ESTB0071      IF (IX(5) + IX(6)+IX(7) .EQ. 0) KGEN = N1H
ESTB0072      IF (IX(5) .EQ. 0) IX(5) = MPRIOR
ESTB0073      IF (IX(6) .EQ. 0) IX(6) = IPRIOR
ESTB0074      IF (IX(7) .EQ. 0) IX(7) = JPRIOR
ESTB0075      MPRIOR = IX(5)
ESTB0076      IPRIOR = IX(6)
ESTB0077      JPRIOR = IX(7)
ESTB0078      LPRIOR = NUMEL
ESTB0079      C
ESTB0080      C** DETERMINE BAND WIDTH
ESTB0081      C
ESTB0082      MB = 0
ESTB0083      DO 120 K = 1,4
ESTB0084      DO 120 L = K,4
ESTB0085      IF (IX(K) .EQ. 0 .OR. IX(L) .EQ. 0) GO TO 120
ESTB0086      M = IABS (IX(K) -IX(L))
ESTB0087      IF (M .GT. MB) MB = M
ESTB0088      120 CONTINUE
ESTB0089      NDF = 2
ESTB0090      IF (IX(3) .EQ. 0) NDF = 3
ESTB0091      MB = NDF*(MB+1)
ESTB0092      IF (MB .GT. MBAND) MBAND = MB
ESTB0093      C
ESTB0094      C** PLACE CONNECTIVITY INPUT INTO NOD ARRAY
ESTB0095      C
ESTB0096      DO 130 K = 1,LSTOP
ESTB0097      130 NOD(K,NE) = IX(K)
ESTB0098      GO TO 170
ESTB0099      C

```

GENL0030
 GENL0031
 GENL0032
 GENL0033
 GENL0034
 GENL0035
 GENL0036
 GENL0037
 GENL0038
 GENL0039
 GENL0040
 GENL0041
 GENL0042
 GENL0043
 GENL0044
 GENL0045
 GENL0046
 GENL0047
 GENL0048
 GENL0049
 GENL0050
 GENL0051
 GENL0052
 GENL0053
 GENL0054
 GENL0055
 GENL0056
 GENL0057
 GENL0058
 GENL0059
 GENL0060
 GENL0061
 GENL0062
 GENL0063
 GENL0064
 GENL0065
 GENL0066
 GENL0067
 GENL0068
 GENL0069
 GENL0070
 GENL0071
 GENL0072
 GENL0073
 GENL0074
 GENL0075
 GENL0076
 GENL0077
 GENL0078
 GENL0079
 GENL0080
 GENL0081
 GENL0082
 GENL0083
 GENL0084
 GENL0085
 GENL0086
 GENL0087
 GENL0088
 GENL0089
 GENL0090
 GENL0091
 GENL0092

```

C** GENERATE AN ELEMENT
C
140 KGEN = IBLANK
DO 150 K = 1,4
  NOD(K,NUMEL) = NOD(K,NUMEL-1) *INTRAL
  IF (NUMLAY .LE. 1) GO TO 145
  IF (MOD(NUMEL-LPRIOR,LINLAY) .NE. 0) GO TO 145
  NOD(K,NUMEL) = MOD(K,LPRIOR) + INTERL*(NUMEL-LPRIOR)/LINLAY
145 CONTINUE
  IF (NOD(K,NUMEL) .LT. 0 .OR. NOD(K,NUMEL) .GT. NUMNP) GO TO 580
150 IF (NOD(K,NUMEL-1) .EQ. 0) NOD(K,NUMEL) = 0
C
C** MAKE MATERIAL CHANGE FOR GENERATED ELEMENTS WHERE SPECIFIED
C
  IF (LGEN(1,NUMEL) .NE. NUMEL) GO TO 160
  IF (LGEN(2,NUMEL) .GT. 0) MPRIOR = LGEN(2,NUMEL)
  IF (LGEN(3,NUMEL) .GT. 0) IPRIOR = LGEN(3,NUMEL)
  IF (LGEN(4,NUMEL) .GT. 0) JPRIOR = LGEN(4,NUMEL)
  KGEN = MIH
160 NOD(5,NUMEL) = MPRIOR
  NOD(6,NUMEL) = IPRIOR
  NOD(7,NUMEL) = JPRIOR
C
C** PRINT ELEMENT DATA
C
170 IF (MGENPR .LT. 2) GO TO 190
  ELM = ETYPE(5)
  IF (NOD(4,NUMEL) .NE. 0) ELM = ETYPE(1)
  IF (NOD(4,NUMEL) .EQ. 0) ELM = ETYPE(2)
  IF (NOD(3,NUMEL) .EQ. 0) ELM = ETYPE(3)
  IF (NOD(7,NUMEL) .NE. 0) ELM = ETYPE(4)
  IF (KGEN .EQ. I1H .OR. KGEN .EQ. N1H) WRITE(6,6015) KGEN,NUMEL,
1  (NOD(K,NUMEL),K=1,6),ELM,MB,(KINDLG(K),K=1,3)
  IF (KGEN .EQ. IBLANK .OR. KGEN .EQ. MIH) WRITE(6,6015) KGEN,NUMEL,
1  (NOD(K,NUMEL),K=1,6),ELM
190 IF (NE .EQ. NUMEL) GO TO 40
  GO TO 80
C
C** WRITE ERROR MESSAGES FOR ELEMENT INPUT
C
520 WRITE (6,6025)
  GO TO 600
540 WRITE (6,6030)
  GO TO 600
560 WRITE (6,6035)
  GO TO 600
580 WRITE (6,6040)
600 MB = 0
  ELM = ETYPE(5)
  WRITE(6,6015) KGEN,NE,(IX(K),K=1,6),ELM,MB,(KINDLG(K),K=1,3)
  KPUTCK = KPUTCK +1
  GO TO 40
5015 FORMAT (A1, I4, 10I5)
6005 FORMAT (122H1ELEMENT NODAL CONNECTIVITY MATERIAL CONSTR.
1  ELEMENT BAND WITHIN LAYER NUMBER OF BETWEEN LAYGENL0147
  2ER/ 120H NUMBER I J K L NUMBER INCR. GENL0148
  3 TYPE WIDTH NODE INCR. ELEMENT LAYERS NODE INCR.) GENL0149
6010 FORMAT ( 71H0WARNING ONLY. THE LAST ELEMENT LAYER OF THE FOLLOWIGENL0150
1NG GENERATION HAS ,I5, 52H ELEMENTS WHILE THE NUMBER OF ELEMENTSGENL0151
2 IN ALAYER IS ,I5/) GENL0152
6015 FORMAT(IX,A1,I4,I10,3I5,I7,I11,7X,A4,I11,I10,2I17) GENL0153
6025 FORMAT ( 78H0FATAL DATA ERROR ON FOLLOWING ELEMENT CARD. ELEMENGENL0154
1T NUMBER IS OUT OF BOUNDS/) GENL0155
GENL0093 5030 FORMAT ( 103H0FATAL DATA ERROR ON FOLLOWING ELEMENT CARD. THE FIGENL0156
GENL0094 1RST ELEMENT CARD MUST HAVE NE = 1 AND IX(1) .NE. 0/) GENL0157
GENL0095 6035 FORMAT ( 94H0FATAL DATA ERROR ON FOLLOWING ELEMENT CARD. IT IS GENL0158
GENL0096 1ILLEGAL TO SPECIFY LIMIT = L IF IX(1) = 0/) GENL0159
GENL0097 6040 FORMAT ( 117H0FATAL DATA ERROR ON FOLLOWING ELEMENT CARD. A NODEGENL0160
GENL0098 1 OF A CONNECTIVITY BEING GENERATED AS SPECIFIED IS OUT OF BOUNDS/) GENL0161
GENL0099 END GENL0162
GENL0100 * * * * * GENN0001
GENL0101 * CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM GENN0002
GENL0102 + VERSION 1.0, MARCH 1990 GENN0003
GENL0103 * PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY GENN0004
GENL0104 * THE FEDERAL HIGHWAY ADMINISTRATION GENN0005
GENL0105 * * * * * GENN0006
GENL0106 SUBROUTINE GENEND (X, Y, NOD, NODB, BIV, KODE, NODTYP, XYGEN, GENN0007
GENL0107 1 NUMNP, NUMEL, NBPTC, MAXBC, KPUTC, MGENPR, NSMAT, LSTOP, GENN0008
GENL0108 1 NUMLI,LUPREP,MAXMAT,NXMAT) GENN0009
GENL0109 + GENN0010
GENL0110 * Uncomment this statement in all program modules for double GENN0011
GENL0111 * precision version. SCM 3-10-90 GENN0012
GENL0112 * GENN0013
GENL0113 * IMPLICIT DOUBLE PRECISION (A-H,O-Z) GENN0014
GENL0114 * GENN0015
GENL0115 DIMENSION X(1), Y(1), NOD(LSTOP,1), NODB(1), BIV(3,1), KODE(1), GENN0016
GENL0116 1 NODTYP(1), XYGEN(3,1) GENN0017
GENL0117 * GENN0018
GENL0118 * Revise array dimension from AN(2) to AN(3) for extended boundary GENN0019
GENL0119 * condition option. SCM 10-25-89 GENN0020
GENL0120 * GENN0021
GENL0121 DIMENSION NAMEG(33),AREA(4),IX(4),ANI(2),AN(3),IIFLG(3),BIVD(3) GENN0022
GENL0122 DIMENSION RBC(2) GENN0023
GENL0123 * GENN0024
GENL0124 COMMON /AFCALC/ XX(4), YY(4) GENN0025
GENL0125 DATA IBLANK ,L1H /1H , 1HL/ GENN0026
GENL0126 DATA NAMEG /4HNOT ,4*4HSTRA,4HLAPL,4HINPU,2*4HSTRA,2*4HCURV , GENN0027
GENL0127 1 4HUSED,4*4HIGHT,4HACE ,4HT ,2*4HIGHT,4HE (A,4HE (E, GENN0028
GENL0128 3 4H****,8*4H ,4HRC) ,4HLIP) / GENN0029
GENL0129 DATA ANI,SAFE /4HF = ,4HD = ,1.E-6/ GENN0030
GENL0130 * GENN0031
GENL0131 * Add data statement for extended boundary condition option. GENN0032
GENL0132 + SCM 10-25-89 GENN0033
GENL0133 * GENN0034
GENL0134 DATA RBC /4H F ,4H D / GENN0035
GENL0135 * GENN0036
GENL0136 C GENN0037
GENL0137 C** GENERATE COORDINATES FOR UNSPECIFIED NODES USING LAPLACE SCHEME GENN0038
GENL0138 C GENN0039
GENL0139 C** INITIALIZE PARAMETERS GENN0040
GENL0140 C GENN0041
GENL0141 NMISNG = 0 GENN0042
GENL0142 DO 20 NNP = 3,NUMNP GENN0043
GENL0143 IF (NODTYP(NNP) .GT. 5) GO TO 20 GENN0044
GENL0144 IF (NODTYP(NNP)/2 .NE. 2) X(NNP) = (X(NNP-2) +X(NNP-1))/ 2.0 GENN0045
GENL0145 IF (NODTYP(NNP)/2 .NE. 1) Y(NNP) = (Y(NNP-2) +Y(NNP-1))/ 2.0 GENN0046
GENL0146 NMISNG = NMISNG +6 GENN0047
GENL0147 20 CONTINUE GENN0048
GENL0148 IF (NMISNG .EQ. 0) GO TO 220 GENN0049
GENL0149 IF (NMISNG .GT. NUMLI) NMISNG = NUMLI GENN0050
GENL0150 IF (KPUTCK .NE. 0) NMISNG = 30 GENN0051
C GENN0052
C** LAPLACE ITERATION GENN0053
C GENN0054
DO 70 NM = 1,NMISNG GENN0055
  FACTOR = 0.0 GENN0056

```

```

DO 30 NNP = 1, NUMNP
DO 30 K = 1, 3
30 XYGEN(K, NNP) = 0.0
C
C** LOOP OVER ELEMENTS TO ESTABLISH MISSING NODAL COORDINATES
C
DO 50 NE = 1, NUMEL
LIMIT = 4
IF (MOD(4, NE) .EQ. 0) LIMIT = 3
IF (MOD(3, NE) .EQ. 0) LIMIT = 2
DO 40 K = 1, LIMIT
NNP = NOD(K, NE)
IF (NODTYP(NNP) .GT. 6 .OR. NODTYP(NNP) .LE. 0) GO TO 40
IF (NODTYP(NNP) .EQ. 1) NODTYP(NNP) = 6
L = K-1
IF (K.EQ.1) L = LIMIT
NNPBAC = NOD(L, NE)
L=K+1
IF (K .EQ. LIMIT) L = 1
NNPFOR = NOD(L, NE)
XYGEN(1, NNP) = XYGEN(1, NNP) + X(NNPFOR) + X(NNPBAC)
XYGEN(2, NNP) = XYGEN(2, NNP) + Y(NNPFOR) + Y(NNPBAC)
XYGEN(3, NNP) = XYGEN(3, NNP) + Z.0
40 CONTINUE
50 CONTINUE
DO 60 NNP = 1, NUMMP
IF (NODTYP(NNP) .GT. 6 .OR. NODTYP(NNP) .EQ. 1) GO TO 60
XPRIOR = X(NNP)
YPRIOR = Y(NNP)
IF (NODTYP(NNP)/2 .NE. 2) X(NNP) = XYGEN(1, NNP)/XYGEN(3, NNP)
IF (NODTYP(NNP)/2 .NE. 1) Y(NNP) = XYGEN(2, NNP)/XYGEN(3, NNP)
BETA = ABS((X(NNP)-XPRIOR)/(ABS(XPRIOR)+1.E-20))
1 +ABS((Y(NNP)-YPRIOR)/(ABS(YPRIOR) +1.E-20))
*
* Revise intrinsic function to generic form. SCM 3-10-90
*
FACTOR = MAX(FACTOR, BETA)
+
60 CONTINUE
IF (FACTOR .LE. 0.0001) GO TO 80
70 CONTINUE
NM = NMISNG
80 WRITE(6, 6005) NM, NMISNG, FACTOR
C
C** SET BOUNDARY CONDITION FOR OMITTED NODES
C
220 DO 240 NNP = 1, NUMNP
IF (NODTYP(NNP) .NE. 1) GO TO 230
X(NNP) = 0.0
Y(NNP) = 0.0
KODE(NNP) = 3
230 CONTINUE
240 CONTINUE
IF (MGENPR .LT. 3) GO TO 260
WRITE(6, 6010)
DO 250 NNP = 1, NUMNE
K = NODTYP(NNP)
250 WRITE(6, 6020) NNP, X(NNP), Y(NNP), NAMEG(K), NAMEG(K+1), NAMEG(K+2)
260 CONTINUE
C
C** COMPUTE ELEMENT AREAS IF TRIANGLE, CHECK AREA AND RESTART LOOP
C
DO 380 NE = 1, NUMEL

```

```

GENN0057      IF (NOD(3,NE) .EQ. 0) GO TO 380
GENN0058      IF (NOD(7,NE),NE.0) GO TO 380
GENN0059      LIMIT = 4
GENN0060      IF (NOD(4,NE) .EQ. 0) LIMIT = 3
GENN0061      DO 320 L = 1,LIMIT
GENN0062      K = MOD(L,NE)
GENN0063      IX(L) = K
GENN0064      XX(L) = X(K)
GENN0065      320 YY(L) = Y(K)
GENN0066      IF (LIMIT .EQ. 4) GO TO 330
GENN0067      AREA(1) = AF(3,1) +AF(2,3) -AF(2,1)
GENN0068      IF (AREA(1).GT.SAFE) GO TO 380
GENN0069      WRITE(6,6025) NE,AREA(1)
GENN0070      KPUTCK = KPUTCK + 1
GENN0071      GO TO 380
GENN0072      C
GENN0073      330 AREA(1) = -AF(1,4) -AF(2,1) +AF(2,4)
GENN0074      AREA(2) = AF(3,4) +AF(2,3) -AF(2,4)
GENN0075      AREA(3) = AF(3,1) +AF(2,3) -AF(2,1)
GENN0076      AREA(4) = -AF(1,4) -AF(3,1) +AF(3,4)
GENN0077      C
GENN0078      C** CHECK FOR NEGATIVE AREAS OF QUAD SUB ELEMENTS
GENN0079      C
GENN0080      SIGN = 1.0
GENN0081      SIGNA = 1.0
GENN0082      IF (AREA(1) .LT. SAFE .OR. AREA(2) .LT. SAFE) SIGN = -SIGN
GENN0083      IF (AREA(3) .LT. SAFE .OR. AREA(4) .LT. SAFE) SIGNA = -SIGNA
GENN0084      IF (SIGN .GT. 0.0 .OR. SIGNA .GT. 0.0) GO TO 340
GENN0085      WRITE (6,6025) NE, AREA
GENN0086      KPUTCK = KPUTCK +1
GENN0087      GO TO 380
GENN0088      C
GENN0089      C** SWITCH ORIENTATION OF DIAGONAL OF QUAD IF PRODUCES BETTER ELEMENT
GENN0090      C
GENN0091      340 L = 0
GENN0092      IF (SIGN + ABS (AREA(1) +AREA(2)) .GT.
GENN0093      1 SIGNA + ABS (AREA(3) +AREA(4)) ) GO TO 360
GENN0094      NOD(L,NE) = IX(4)
GENN0095      DO 350 K = 2,4
GENN0096      350 NOD(K,NE) = IX(K-1)
GENN0097      L = 2
GENN0098      360 IF (AREA(L+1) .GT. 10.*AREA(L+2) .OR.
GENN0099      1 AREA(L+1) .LT. AREA(L+2)/10.) WRITE (6,6030) NE, AREA
GENN0100      380 CONTINUE
GENN0101      C
GENN0102      C** SET FLAGS FOR OTHER THAN QUAD ELEMENTS
GENN0103      C
GENN0104      DO 400 NE = 1,NUMEL
GENN0105      IF (NOD(3,NE) .EQ. 0) NOD(3,NE) = NOD(2,NE)
GENN0106      400 IF (NOD(4,NE) .EQ. 0) NOD(4,NE) = NOD(3,NE)
GENN0107      C
GENN0108      C** CHECK MATERIAL DEFINITIONS, AND DETERMINE NUMBER OF MATERIALS.
GENN0109      C
GENN0110      DO 410 J=1,MAXMAT
GENN0111      410 MODTYP(J) = -1
GENN0112      NSMAT = 0
GENN0113      C
GENN0114      DO 450 NE=1,NUMEL
GENN0115      IF(NOD(2,NE).EQ.NOD(3,NE)) GO TO 450
GENN0116      IF(NOD(7,NE).NE.0) GO TO 450
GENN0117      MAT = NOD(5,NE)
GENN0118      C
GENN0119      DO 440 J=1,MAXMAT

```



```

      IF(MAT.EQ.NODTYP(J)) GO TO 450
440 CONTINUE
      NSMAT = NSMAT + 1
      NODTYP(NSMAT) = MAT
450 CONTINUE
      IF(NSMAT.GT.MAXMAT) WRITE(6,6035) NSMAT,MAXMAT
C
C DETERMINE NUMBER OF INTERFACE MATERIAL-GROUPS.
C
      NXMAT = 0
*
* Revise MAXMX from 30 to 99. SCM 10-25-89
*
      MAXMX = 99
*
      DO 460 J=1,MAXMX
460 NODTYP(J) = -1
C
      DO 470 NE=1,NUMEL
      IF(MOD(7,NE).EQ.0) GO TO 470
      MOD(7,NE) = 1
      MATX = MOD(5,NE)
      IF(MATX.GT.0 .AND. MATX.LE.MAXMX) GO TO 466
      WRITE(6,6050) NE
      KPUTCK = KPUTCK + 1
C
C
466 DO 467 J=1,MAXMX
      IF(MATX.EQ.NODTYP(J)) GO TO 470
467 CONTINUE
      NXMAT = NXMAT + 1
      NODTYP(NXMAT) = MATX
470 CONTINUE
C
C BOUNDARY CONDITIONS
C
      PI = 0.0
      LIMIT = IBLANK
      IF(MGENPR.GE.2) WRITE(6,915)
      I=1
180 IF (LIMIT.EQ.LIH) GO TO 195
      READ (LUPREP,5005) LIMIT, KK, (IIFLG(N), BIVD(N), N = 1,2),
1 TH, IA, KKP, INCR, PJ, PK
      IF (I.GT. MAXBC) GO TO 193
      NMIS=0
      IF(KKP.EQ. 0)GO TO 182
      IF(INCR.EQ. 0) INCR=1
      NM=(KKP-KK)/INCR
      NMIS=IABS(NM)
      INCR=INCR*NM/NMIS
182 IF(IA.EQ. 0) IA=1
      NM=1+NMIS
      DP=PK-PJ
      IF(NMIS.EQ. 0) GO TO 185
      M=KK
      XL=0.0
      DO 183 L=1,NMIS
      MP=M
      M=M+INCR
183 XL=SQRT((X(M)-X(MP))**2+(Y(M)-Y(MP))**2)*XL
      DP=DP/XL
185 DXB=0.0
      DYB=0.0
      DO 190 L=1,NM

```

```

GENN0183 DXF=0.0
GENN0184 DYF=0.0
GENN0185 IF(L.EQ. NM) GO TO 186
GENN0186 DXF=X(KK+INCR)-X(KK)
GENN0187 DYF=Y(KK+INCR)-Y(KK)
GENN0188 PK=PJ+DP*(SQRT(DXF*DXF+DYF*DYF))
GENN0189 186 BIV(1,I)=BIVD(1)-(2.0*(DYF+DYB)+PJ+DYB*PI+DYF*PK)/6.0
GENN0190 BIV(2,I)=BIVD(2)+(2.0*(DXF+DXB)+PJ+DXB*PI+DXF*PK)/6.0
GENN0191 DXB=DXF
GENN0192 DYB=DYF
GENN0193 PI=PJ
GENN0194 PJ=PK
GENN0195 BIV(3,I)=TH
GENN0196 NODS(I)=KK*10000+IIFLG(1)*10+IIFLG(2) +IA*100
GENN0197 DO 187 J=1,2
GENN0198 IF (IIFLG(J).EQ. 0) AN(J) = ANI(1)
GENN0199 IF (IIFLG(J).EQ. 1) AN(J) = ANI(2)
GENN0200
*
* Add extended boundary condition definition. MGK 3-18-89
*
      IF (IIFLG(J).EQ. 2) AN(J) = ANI(2)
      IF (IIFLG(J).EQ. 3) AN(J) = ANI(1)
*
187 CONTINUE
*
* Extend boundary condition output for rotational constraints.
* SCM 10-25-89
*
      IF(IIFLG(1).EQ.0 .AND. IIFLG(2).EQ.0) THEN
        AN(3) = RBC(1)
      ELSE IF(IIFLG(1).EQ.2 .AND. IIFLG(2).EQ.0) THEN
        AN(3) = RBC(1)
      ELSE IF(IIFLG(1).EQ.0 .AND. IIFLG(2).EQ.2) THEN
        AN(3) = RBC(1)
      ELSE IF(IIFLG(1).EQ.2 .AND. IIFLG(2).EQ.2) THEN
        AN(3) = RBC(1)
      ELSE
        AN(3) = RBC(2)
      END IF
      IF(MGENPR.GE.2) WRITE(6,906) KK,IA,AN(1),BIV(1,I),AN(2),BIV(2,I),
X BIV(3,I),AN(3)
*
      KK=KK+INCR
      BIV(3,I)=TH*0.0174533
190 I=I+1
      GO TO 180
193 WRITE (6,6045)
      KPUTCK = KPUTCK +1
195 NBPTC=I-1
C
      RETURN
5005 FORMAT (A1,I4,I5,F10.0,I5,2F10.0,3I5 ,2F10.0)
6005 FORMAT ( 38H1LAPLACE GENERATION HAS BEEN COMPLETED /
1 40HNUMBER OF ITERATIONS -----,I5/
2 40HMAXIMUM NUMBER OF ITERATIONS ALLOWED---,I5/
3 40HOLARGEST LAPLACE ERROR FACTOR-----,F15.6)
6010 FORMAT (61H1NODE X-COORDINATE Y-COORDINATE GENEGENN0302
1RATION /14H NUMBER ,37X,4HTYPE) GENN0303
6020 FORMAT (1X,I4,8X,E20.4,E15.4,3X,3A4) GENN0304
6025 FORMAT ( 131H0FATAL DATA ERROR. NEGATIVE AREA FOR ELEMENT NUMBERGENN0305
1 AREA OF TRIANGLES AS INPUT AREA OF TRIANGLES USING ALTERNGENN0306
2ATE DIAGONAL/ 15I, 8X,2E12.3, 13X,2E12.3) GENN0307
6030 FORMAT ( 75H0WARNING ONLY. THE AREA OF ONE SUB ELEMENT IS MUCH GENN0308

```

```

15 SMALLER THAN THE OTHER/ 97H FOR ELEMENT NUMBER AREA OF TRIANGG
2LES AS INPUT AREA OF TRIANGLES USING ALTERNATE DIAGONAL/
3 117, 8X, 2E12.3, 13X, 2E12.3/)
6035 FORMAT(/30H FATAL DATA ERROR. THERE WERE ,I3, 14H DECLARED BUT,
X ,I3, 12H IS MAXIMUM. )
6045 FORMAT( 65HOFATAL DATA ERROR. NUMBER OF BOUNDARY CARDS CANNOT
1EXCEED MAXBC)
6050 FORMAT(/ 62H FATAL DATA ERROR FOR MATERIAL NUMBER OF INTERFACE
ELEMENT NO. ,I5)
*
* Revise formats for boundary condition output. SCM 10-25-89
*
915 FORMAT(1H1,IX,31H* * * BOUNDARY CONDITIONS * * */2X,
X59H(FORCES = LBS; DISPLACEMENTS = INCHES; ROTATIONS = DEGREES)//
X2X,8HBOUNDARY,6X,4HLOAD,6X,14HX-FORCE OR,6X,14HY-FORCE OR,
X6X,12HX-Y ROTATION,6X,15HBEAM ROTATIONAL/6X,4HNODE,6X,4HSTEP,6X,
X14HX-DISPLACEMENT,6X,14HY-DISPLACEMENT,22X,
X19HBOUNDARY CONDITIONS//)
906 FORMAT(5X,15,5X,15,4X,A4,E12.4,4X,A4,E12.4,4X,E12.4,14X,A4)
END
* * * * *
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
* VERSION 1.0, MARCH 1990
* PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
* THE FEDERAL HIGHWAY ADMINISTRATION
* * * * *
SUBROUTINE GENNOD (X, Y, NODGEN, NODTYP, XYGEN, NUMNP, MAXNP,
I KPUTCK, MGENPR, LUPREP)
*
* Uncomment this statement in all program modules for double
* precision version. SCM 3-10-90
*
* IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
DIMENSION X(1), Y(1), NODGEN(3,1), NODTYP(1), XYGEN(3,1)
DATA PI /3.1415/
DATA L1H /1HL/
C
C** INITIALIZE PARAMETERS
C
MAXNP = NUMNP
LIMIT = 0
NPRIOR = 0
NUMNP = 0
NUMGNP = 0
MODEG = 5
DO 10 NNP = 1,MAXNP
Y(NNP) = 0.0
X(NNP) = 0.0
10 NODTYP(NNP) = 1
C
C** READ CURRENT NODE CARD, CHECK PARAMETERS, AND TRANSFER
C
IF (MGENPR.GE.2) WRITE (6,6005)
20 IF (MGENPR .GE. 3 .AND. MODEG .NE. 5) CALL SAVEG (NODGEN, XYGEN,
1 NUMGNP, NNP, MODEG, X(NNP), Y(NNP))
IF (LIMIT .EQ. L1H ) GO TO 800
READ (LUPREP,5010) LIMIT, NNP, MODEG, XCOORD, YCOORD, NPINCR,
1 SPACNG, RADIUS
IF (MGENPR .GE. 2) WRITE (6,6010) NNP, MODEG, XCOORD, YCOORD,
1 NPINCR, SPACNG, RADIUS
IF (NNP .GT. MAXNP .OR. NNP .LE. 0) GO TO 520
KRELAD = MODEG/100
LGTYPE = MOD (MODEG, 100)/ 10
MODEG = MOD (MODEG, 10)
IF (LGTYPE .NE. 0 .AND. RADIUS .NE. 0.0) GO TO 540
IF ( (MODEG.EQ.1.OR.MODEG.EQ.3) .AND. NODTYP(NNP).EQ.1) GO TO 560
IF ( (MODEG.EQ.0.OR.MODEG.EQ.2) .AND. NODTYP(NNP).NE.1) GO TO 580
IF (MODEG .EQ. 1) GO TO 220
IF (MODEG .EQ. 3) GO TO 50
IF (MODEG .EQ. 5) GO TO 240
C
C** SET UP COORDINATES FOR RELATIVELY ADDRESSED NODES
C
IF (KRELAD .NE. 1 .AND. KRELAD .NE. 3) GO TO 30
*
* Revise intrinsic function to generic form. SCM 3-10-90
*
NN = INT(XCOORD)
IF (NN .GT. NUMNP .OR. NN .LE. 0 .OR. NODTYP(NN) .LT. 7) GO TO 600
XCOORD = X(NN)
30 IF (KRELAD .NE. 2 .AND. KRELAD .NE. 3) GO TO 40
*
* Revise intrinsic function to generic form. SCM 3-10-90
*
NN = INT(YCOORD)
IF (NN .GT. NUMNP .OR. NN .LE. 0 .OR. NODTYP(NN) .LT. 7) GO TO 600
YCOORD = Y(NN)
C
C** SET NODE QUANTITIES FOR CURRENT NODE POINT
C
40 X(NNP) = XCOORD
Y(NNP) = YCOORD
NODTYP(NNP) = 7
NUMNP = MAX0 (NUMNP, NNP)
IF (MODEG .EQ. 0) GO TO 220
C
C** SET GENERATION PARAMETERS
C
50 IF (NPINCR .EQ. 0) NPINCR = 1
IF (SPACNG .EQ. 0.0) SPACNG = 1.0
IF (MPRIOR .EQ. 5) GO TO 60
NMISNG = IABS(NNP-NPRIOR)/ NPINCR
IF ( MOD(NNP-NPRIOR, NPINCR) .EQ. 0) NMISNG = NMISNG -1
IF (NMISNG .LE. 0) GO TO 620
NPINCR = NPINCR*ISIGN (1, NNP-NPRIOR)
60 FACTOR = 1.0/ FLOAT(NMISNG+1)
IF (LGTYPE .EQ. 4) GO TO 160
IF (RADIUS .NE. 0.0) GO TO 130
C
C** GENERATE NODES ALONG A STRAIGHT LINE.
C
IF (SPACNG.NE.1.0) FACTOR = (1.0-SPACNG)/ (1.0-SPACNG** (NMISNG+1))
N = NPRIOR
DX = (X(NNP) -XPRIOR)*FACTOR
DY = (Y(NNP) -YPRIOR)*FACTOR
DO 120 NM = 1,NMISNG
NN = NPRIOR +NPINCR*NM
IF (MPRIOR .EQ. 5) NN = NODGEN(1,NN)
IF (NODTYP(NN) .NE. 1) GO TO 640
X(NN) = X(N) +DX
Y(NN) = Y(N) +DY
IF (MGENPR .GE. 3) CALL SAVEG (NODGEN, XYGEN, NUMGNP,
1 NN, -1, X(NN), Y(NN))

```



```

      IF (SPACNG .EQ. 1.0) NODTYP(NN) = 8
      IF (LGTYPE .EQ. 1) NODTYP(NN) = NODTYP(NN) -6
      IF (SPACNG .NE. 1.0) NODTYP(NN) = 9
      IF (LGTYPE .EQ. 2) NODTYP(NN) = NODTYP(NN) -4
      N = NN
      DX = DX*SPACNG
120 DY = DY*SPACNG
      GO TO 220
C
C** GENERATE NODES ALONG AN ARC
C** START BY FINDING COORDINATES OF CENTER OF CURVATURE
C
130 DX = X(NNP) -XPRIOR
      DY = Y(NNP) -YPRIOR
      DL = SQRT (DX*DX +DY*DY)
      RSIGN = SIGN(1.0, RADIUS)
      RADIUS = ABS(RADIUS*1.000001)
      IF (DL/RADIUS/2.0 .GT. 1.0 .OR. DL .EQ. 0.0) GO TO 660
      ASINFK = DL/RADIUS/2.0
      THETA = 2.0*ATAN(1.0)-ATAN(ASINFK/SQRT(1.0-ASINFK**2))
      PHI = ATAN2 (DY, DX)
      ALPHA = PHI +RSIGN*THETA
      XCENTR = XPRIOR +COS(ALPHA)*RADIUS
      YCENTR = YPRIOR +SIN(ALPHA)*RADIUS
C
C** SETUP GENERATION PARAMETERS AND GENERATE SPECIFIED NODES FOR ARC
C
      OMEGA = FACTOR * (4.0*ATAN(1.0) -2.0*THETA)
      BETA = ATAN2 (YPRIOR-YCENTR, XPRIOR-XCENTR)
      DO 140 NM = 1,NMISNG
      NN = NPRIOR +NPINCR*NM
      IF (MPRIOR .EQ. 5) NN = NODGEN(1,NN)
      IF (NODTYP(NN) .NE. 1) GO TO 640
      BETA = BETA +RSIGN*OMEGA
      X(NN) = XCENTR +COS(BETA)*RADIUS
      Y(NN) = YCENTR +SIN(BETA)*RADIUS
      IF (MGENPR .GE. 3) CALL SAVEG (NODGEN, XYGEN, NUMGNP,
1      NN, -1, X(NN), Y(NN))
140 NODTYP(NN) = 10
      GO TO 220
C
C** GENERATE QUADRANTS OF AN ELLIPSE -ALWAYS GENERATE COUNTERCLOCKWISE
C** START BY FINDING CENTER COORDINATES AND ANGLE ALPHA
C
160 XAXIS = X(NNP) -X(NPRIOR)
      YAXIS = Y(NNP) -Y(NPRIOR)
      XCENTR = X(NPRIOR)
      YCENTR = Y(NNP)
      IF (XAXIS*YAXIS .LT. 0.0) XCENTR = X(NNP)
      IF (XAXIS*YAXIS .LT. 0.0) YCENTR = Y(NPRIOR)
      IF (XAXIS .LT. 0.0 .AND. YAXIS .GT. 0.0) ALPHA = 0.0
      IF (XAXIS .LT. 0.0 .AND. YAXIS .LT. 0.0) ALPHA = PI/2.0
      IF (XAXIS .GT. 0.0 .AND. YAXIS .LT. 0.0) ALPHA = PI
      IF (XAXIS .GT. 0.0 .AND. YAXIS .GT. 0.0) ALPHA = 3.0*PI/ 2.0
C
C** GENERATE NODES ALONG ELLIPSE CIRCUMFERENCE
C
      DO 170 NM = 1,NMISNG
      NN = NPRIOR +NPINCR*NM
      IF (MPRIOR .EQ. 5) NN = NODGEN(1,NN)
      IF (NODTYP(NN) .NE. 1) GO TO 640
      THETA = ALPHA +FLOAT(NM)*FACTOR*PI/ 2.0
      X(NN) = 1.0/ SQRT ( 1.0/XAXIS**2 +(TAN(THETA)/YAXIS)**2) *XCENTR
      Y(NN) = (X(NN)-XCENTR) * TAN(THETA) +YCENTR
      IF (MGENPR .GE. 3) CALL SAVEG (NODGEN, XYGEN, NUMGNP,
1      NN, -1, X(NN), Y(NN))
170 NODTYP(NN) = 11
C
C** SET PARAMETERS FOR PRIOR NODE
C
220 NPRIOR = NNP
      XPRIOR = X(NNP)
      YPRIOR = Y(NNP)
      NMISNG = 0
230 MPRIOR = MODEG
      GO TO 20
C
C** SAVE NODES FOR INTERPOLATION (MODEG = 5 OPTION)
C
240 NMISNG = NMISNG +1
      NODGEN(1,NMISNG) = NNP
      GO TO 230
C
C** WRITE ERROR MESSAGES FOR NODE INPUT AND NODES AS GENERATED LIST
C
520 WRITE (6,6015)
      KPUTCK = KPUTCK +1
      GO TO 220
540 WRITE (6,6020)
      KPUTCK = KPUTCK +1
      GO TO 220
560 WRITE (6,6025)
      KPUTCK = KPUTCK +1
      GO TO 220
580 WRITE (6,6030)
      KPUTCK = KPUTCK +1
      GO TO 220
600 WRITE (6,6135)
      KPUTCK = KPUTCK +1
      GO TO 220
620 WRITE (6,6040)
      KPUTCK = KPUTCK +1
      GO TO 220
640 WRITE (6,6045) NN
      KPUTCK = KPUTCK +1
      GO TO 220
660 WRITE (6,6050)
      KPUTCK = KPUTCK +1
      GO TO 220
800 IF (MGENPR.GE.3) CALL SAVEG (NODGEN, XYGEN, NUMGNP, 0,0,0.0,0.0)
      RETURN
5010 FORMAT ( A1, I4, I5, 2F10.0, I5, 5X,2F10.0)
6005 FORMAT ( 82HINODE GENERATION X-COORDINATE Y-COORDINATE
1      NODE SPACING ARC/ 16H NUMBER MODE,39X ,30HINCREMGEN0220
2ENT PARAMETER RADIUS)
6010 FORMAT (I7, I8, E19.4, E15.4, I9, F16.3,F11.2)
6015 FORMAT ( 74HOFATAL DATA ERROR ON ABOVE NODE CARD. NODE POINT NUGEN0223
1MBER IS OUT OF BOUNDS /)
6020 FORMAT ( 78HOFATAL DATA ERROR ON ABOVE NODE CARD. IF MODEG .GE.
110, THEN RADIUS MUST = 0)
6025 FORMAT ( 125HOFATAL DATA ERROR ON ABOVE NODE CARD. A NODE HAS BEGEN0227
1EN RECALLED (I.E., MODEG = 1 OR 3) WHICH HAS NOT PREVIOUSLY BEEN DGEN0228
2DEFINED /)
6030 FORMAT ( 117HOFATAL DATA ERROR ON ABOVE NODE CARD. A NODE HAS BEGEN0230
1EN PRESCRIBED (MODEG = 0 OR 2) WHICH HAS PREVIOUSLY BEEN DEFINED/)GEN0231
6135 FORMAT ( 83HOFATAL DATA ERROR ON ABOVE NODAL CARD. THE NODE RELGEN0232

```

```

LATIVELY ADDRESSED IS UNDEFINED)
6040 FORMAT ( '109HOFATAL DATA ERROR ON ABOVE NODE CARD. INCORRECT GENGEN00234
GENERATION SPECIFIED (POSSIBLY MODEG OR NPINCR IS IN ERROR) /) GEN00235
6045 FORMAT ( '80HOFATAL DATA ERROR. GENERATION SPECIFIED BY ABOVE NOGEN00236
IDE CARD WILL CAUSE NODE ( ,15, 20H ) TO BE REDEFINED /) GEN00237
6050 FORMAT (109HOFATAL DATA ERROR. THE RADIUS SPECIFIED FOR THE ABOVEGEN00238
1E ARC GENERATION IS LESS THAN HALF THE DISTANCE BETWEEN / GEN00239
2 85H THE CURRENT AND PRIOR NODES OR THE CURRENT AND PRIOR NODESGEN00240
3 HAVE THE SAME COORDINATES /) GEN00241
END GEN00242
* * * * *
CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
* * * * *
VERSION 1.0, MARCH 1990
* * * * *
PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
* * * * *
THE FEDERAL HIGHWAY ADMINISTRATION
* * * * *
SUBROUTINE GEOM(AA,D1,D2,D3)
*
Uncomment this statement in all program modules for double
* * * * *
precision version. SCM 3-10-90
*
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
COMMON /ELEMNT/ AS(3),BS(3),C1(17,17),EST(4,3),RQ(4),RT(3),
1 STORE(17),Z4(12),ZQ(4),ZT(3),ZY(17),J1,RO
COMMON /ESTABL/ ITRW(28,2),NETCL(17,2),T(28,2)
DIMENSION IS(5)
DATA IS /1,2,3,1,2/
C
C
C
GEOMETRIC PROPERTIES OF ELEMENT CALCULATED
AA=0.0
D1=0.0
D2=0.0
D3=0.0
DO 110 I=1,3
J=IS(I+1)
K=IS(I+2)
D4 = 2T(J) - ZT(K)
D5 = RT(K) - RT(J)
AA = AA + RT(I)*D4
D1 = D1 + D4*D4
D2=D2+D4*D5
D3=D3+D5*D5
BS(I)=D4
110 AS(I)=D5
C
C
C
CONSTRAINT EQUATIONS NECESSARY FOR CONVERGENCE FORMED
DU=1.0/AA
IF(J1.EQ. 2) GO TO 117
DO 112 I=1,3
T(I+18,1) = 2T(I) * DU
T(I+24,1) = 2T(I) *DU
112 T(I+21,1) = -RT(I) * DU
GO TO 140
117 DO 121 I=1,3
T(I+19,2) = -RT(I) * DU
T(I+25,2) = -RT(I) * DU
121 T(I+22,2) = 2T(I) * DU
140 RETURN
END
* * * * *
GEN00233
*
CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
*
VERSION 1.0, MARCH 1990
*
PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
*
THE FEDERAL HIGHWAY ADMINISTRATION
*
* * * * *
SUBROUTINE HARDIN (ST,STHARD,NEL,ICON,ITER,MN,NELEM)
C
THIS SUBROUTINE GENERATES AN INCREMENTAL PLAIN STRAIN
C
CONSTITUTIVE MATRIX BASED ON THE EXTENDED-HARDIN SOIL MODEL.
C
REFERENCE ... AFWL-TR-72-201, MARCH 1973, BY B. HARDIN.
C
Uncomment this statement in all program modules for double
* * * * *
precision version. SCM 3-10-90
*
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
DIMENSION ST(6,NELEM),STHARD(6,NELEM)
COMMON /MATERL/ CP(3,3),DEN(10),E(10,10),GNU(10,10),H(10,10),
1 ITYPE(10),NON(10)
DATA TOLER /0.05/
C
THE FOLLOWING VARIABLES ARE IDENTIFIED ...
NEL = ELEMENT NUMBER
MN = MATERIAL NUMBER
ITER = ITERATION NUMBER WITHIN LOAD STEP
ICON = CONVERGENCE CODE, 0 = NO, 1 = YES.
H(MN,1) = HARDIN SOIL PARAMETER, 1230*F
H(MN,2) = HARDIN SOIL PARAMETER, C1
H(MN,3) = HARDIN SOIL PARAMETER, A
H(MN,4) = EXTENDED-HARDIN PARAMETER, MIN POISSON RATIO
H(MN,5) = EXTENDED-HARDIN PARAMETER, MAX POISSON RATIO
H(MN,6) = EXTENDED HARDIN CURVE PARAMETER
ST(1,NEL) = STRAIN STATE. I=1,2,3 (XX,YY,XY)
ST(1,NEL) = STRESS STATE. I=4,5,6 (XX,YY,XY)
STHARD(1,NEL) = MAXIMUM SHEAR AT PREVIOUS LOAD STEP
STHARD(2,NEL) = MAXIMUM SHEAR STRAIN AT PREVIOUS LOAD STEP
STHARD(3,NEL) = AVERAGE PRESSURE AT THE PREVIOUS LOAD STEP
STHARD(4,NEL) = VOLUME CHANGE AT THE PREVIOUS LOAD STEP
STHARD(5,NEL) = SECANT SHEAR MODULUS AT LAST ITERATION STEP
STHARD(6,NEL) = SECANT BULK MODULUS AT LAST ITERATION STEP
C**
RECOVER SYSTEM RESPONSES
AA = ST(1,NEL)*ST(4,NEL)-ST(2,NEL)*ST(5,NEL)
BB = ST(1,NEL)*ST(5,NEL) - ST(2,NEL)*ST(4,NEL)
XNU2= H(MN,4)
IF(BB.NE.0.0) XNU2= BB/(AA+BB)
TEMP = ST(1,NEL) - ST(2,NEL)
GAMMA2 = SQRT(TEMP**2+ST(3,NEL)**2)
TEMP = 0.5* (ST(4,NEL) - ST(5,NEL))
TAU2 = SQRT(TEMP**2 +ST(6,NEL)**2)
SIG0 = -(ST(4,NEL)+ST(5,NEL))*(1.+XNU2)/3.0
IF(SIG0.LE.0.0) SIG0=1.0
C
C**
COMPUTE HARDIN VARIABLES
GMAX = H(MN,1)*SQRT(SIG0)
GAMMAR = GMAX/H(MN,2)
RATIO = GAMMA2/GAMMAR
GAMMAP = H(MN,6)*RATIO
TERM = H(MN,3)/EXP(RATIO**0.4)
GAMMAR = RATIO*(1. + TERM)

```

HARD0002
 HARD0003
 HARD0004
 HARD0005
 HARD0006
 HARD0007
 HARD0008
 HARD0009
 HARD0010
 HARD0011
 HARD0012
 HARD0013
 HARD0014
 HARD0015
 HARD0016
 HARD0017
 HARD0018
 HARD0019
 HARD0020
 HARD0021
 HARD0022
 HARD0023
 HARD0024
 HARD0025
 HARD0026
 HARD0027
 HARD0028
 HARD0029
 HARD0030
 HARD0031
 HARD0032
 HARD0033
 HARD0034
 HARD0035
 HARD0036
 HARD0037
 HARD0038
 HARD0039
 HARD0040
 HARD0041
 HARD0042
 HARD0043
 HARD0044
 HARD0045
 HARD0046
 HARD0047
 HARD0048
 HARD0049
 HARD0050
 HARD0051
 HARD0052
 HARD0053
 HARD0054
 HARD0055
 HARD0056
 HARD0057
 HARD0058
 HARD0059
 HARD0060
 HARD0061
 HARD0062
 HARD0063
 HARD0064

```

      EKK = ST(1,NEL) + ST(2,NEL)
C
C** COMPUTE NEW SECANT SHEAR MODULUS
C
      GNEW = GMAX / (1.+GAMMAH)
      GOLD = STHARD(5,NEL)
      GCHECK = (GNEW-GOLD)/GNEW
      GGUESS = GNEW
      STHARD(5,NEL) = GGUESS
C
C** COMPUTE NEW SECANT BULK MODULUS
C** (BULK MODULUS BASED ON HARDIN POISSON
C** RATIO AND SHEAR MODULUS)
C
      VNEW = (H(MN,4) + GAMMAH*(MN,5)) / (1.+GAMMAH)
      BNEW = 2./3.*GGUESS*(1.+VNEW)/(1.-2.*VNEW)
      BOLD = STHARD(6,NEL)
      BCHECK = (BNEW-BOLD)/BNEW
      BGUESS = BNEW
      STHARD(6,NEL) = BGUESS
C
C** COMPUTE CHORD VALUES FOR SHEAR AND BULK MODULI
C
      GCORD = GGUESS
      DGAMMA = GAMMA2 - STHARD(2,NEL)
      IF (DGAMMA.NE. 0.0) GCORD = (GAMMA2*GGUESS-STHARD(1,NEL))/DGAMMA
      IF (GCORD.LE.0.0) GCORD = GGUESS
      IF (GCORD.GT. GMAX) GCORD = GMAX
      BCORD = BGUESS
      DEKK = EKK - STHARD(4,NEL)
      IF (DEKK.NE. 0.0) BCORD = (EKK*BGUESS-STHARD(3,NEL))/DEKK
      IF (BCORD.LE.0.0) BCORD = BGUESS
C
C** STORE PREVIOUSLY CONVERGED SOLUTION ON THE FIRST ITERATION
C** OF THE FOLLOWING LOAD STEP. (NOTE FIRST ITERATION IS PRIOR TO THE
C** APPLICATION OF LOADING)
C
      IF (ITER.GT. 1) GO TO 100
      STHARD(1,NEL) = TAU2
      STHARD(2,NEL) = GAMMA2
      STHARD(3,NEL) = -SIG0
      IF (TAU2.EQ. 0.0) STHARD(3,NEL) = 0.0
      STHARD(4,NEL) = EKK
      ICON = 0
C
C** CHECK FOR CONVERGENCE
C
100 IF (ABS(BCHECK).GT. TOLER) ICON = 0
    IF (ABS(GCHECK).GT. TOLER) ICON = 0
    IF (ITER.GT. NON(MN)) ICON = 1
C
C** CONSTRUCT NEW INCREMENTAL CONSTITUTIVE MATRIX
C
      VCORD = (3.0*BCORD-2.0*GCORD)/(6.0*BCORD+2.0*GCORD)
      IF (VCORD.LT. H(MN,4)) VCORD = H(MN,4)
      IF (VCORD.GT. H(MN,5)) VCORD = H(MN,5)
      ECORD = 2.*(1.+VCORD)*GCORD
      CALL CONV(ECORD,VCORD)
      RETURN
      END
+
+ *****
+ CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
+ VERSION 1.0, MARCH 1990

```

```

HARD0065 * PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY HERC0004
HARD0066 * THE FEDERAL HIGHWAY ADMINISTRATION HERC0005
HARD0067 * * * * * HERC0006
HARD0068 SUBROUTINE HEROIC (BIV,DENSTY,IA,ICOME,IEXIT,LEVEL,NBPTC,MNO, HERC0007
HARD0069 1 NELEM,NINC,NOD,NODB,NPMAT,NPPT,NFT,NQ,NSMAT, HERC0008
HARD0070 2 NXMX,PIPMAT,RESULT,SK,SM,ST,STHARD,STPIPE, HERC0009
HARD0071 3 U,V,X,Y) HERC0010
HARD0072 C HERC0011
HARD0073 C HERC0012
HARD0074 C INCREMENTAL PLAIN STRAIN ANALYSIS HERC0013
HARD0075 C HEROIC = (HER)MANN (O)N (I)NCREMENTAL (C)ONSTRUCTION. HERC0014
HARD0076 C HERC0015
HARD0077 C HERC0016
HARD0078 * Uncomment this statement in all program modules for double HERC0017
HARD0079 * precision version. SCM 3-10-90 HERC0018
HARD0080 * HERC0019
HARD0081 * IMPLICIT DOUBLE PRECISION (A-H,O-Z) HERC0020
HARD0082 * HERC0021
HARD0083 DIMENSION BIV(3,NBPTC),MNO(NELEM),NOD(4,NELEM),NODB(NBPTC), HERC0022
HARD0084 1 NQ(NPT+1),PIPMAT(5,NPPT),RESULT(20,NPPT),ST(6,NELEM), HERC0023
HARD0085 2 STHARD(6,NELEM),STPIPE(2,NPPT),U(NPT),V(NPT),X(NPT), HERC0024
HARD0086 3 Y(NPT) HERC0025
HARD0087 C HERC0026
HARD0088 COMMON /SIZEPR/ BIVD(3),IX,NINCC HERC0027
HARD0089 COMMON /ELEMNT/ AS(3),BS(3),C1(17,17),EST(4,3),RQ(4),RT(3), HERC0028
HARD0090 1 STORE(17),Z4(12),ZQ(4),ZT(3),ZY(17),J1,RQ HERC0029
HARD0091 COMMON/SYSTEM/B(240),A(240,120) HERC0030
HARD0092 COMMON /MATERL/ CP(3,3),DEN(10),E(10,10),GNU(10,10),H(10,10), HERC0031
HARD0093 1 ITYPE(10),NON(10) HERC0032
HARD0094 * HERC0033
HARD0095 * Expand array dimension statments from 30 to 99. HERC0034
HARD0096 * SCM 5-22-89 HERC0035
HARD0097 * HERC0036
HARD0098 COMMON/XFACE/NXMAT,ANGLE(99),FCOEF(99),TENSIL(99),ITMAX HERC0037
HARD0099 * HERC0038
HARD0100 * Add common block. SCM 10-23-89 HERC0039
HARD0101 * HERC0040
HARD0102 COMMON /OUTPUT/ IWRT,IPLT,MAXIT,JCON HERC0041
HARD0103 * HERC0042
HARD0104 * Revise MAXMAT dimension statement from 30 to 99. HERC0043
HARD0105 * SCM 5-22-89 HERC0044
HARD0106 * HERC0045
HARD0107 DATA LUDATA,MAXMAT /11,99/ HERC0046
HARD0108 * HERC0047
HARD0109 C HERC0048
HARD0110 C GO TO (100,110), ICOME HERC0049
HARD0111 C HERC0050
HARD0112 C FIRST ENTRY ESTABLISHES CONSTANTS AND READS FINITE ELEMENT DATA. HERC0051
HARD0113 C HERC0052
HARD0114 100. CONTINUE HERC0053
HARD0115 NEQ = 1200 HERC0054
HARD0116 NXMAT = NXMX HERC0055
HARD0117 C HERC0056
HARD0118 C HERC0057
HARD0119 C** ALL INPUT VALUES FROM PREF ARE READ HERE FROM TAPE LUDATA HERC0058
HARD0120 C HERC0059
HARD0121 * HERC0060
HARD0122 * Revise I/O unit from LUDATA to Unit 14. HERC0061
HARD0123 * SCM 11-22-88 HERC0062
HARD0124 * HERC0063
HARD0125 * HERC0064
HARD0126 * HERC0065
HARD0127 * HERC0066
HARD0128 * HERC0067
HARD0129 * HERC0068
HARD0130 * HERC0069
HARD0131 * HERC0070
HARD0132 * HERC0071
HARD0133 * HERC0072
HARD0134 * HERC0073
HARD0135 * HERC0074
HARD0136 * HERC0075
HARD0137 * HERC0076
HARD0138 * HERC0077
HARD0139 * HERC0078
HARD0140 * HERC0079
HARD0141 * HERC0080
HARD0142 * HERC0081
HARD0143 * HERC0082
HARD0144 * HERC0083
HARD0145 * HERC0084
HARD0146 * HERC0085
HARD0147 * HERC0086
HARD0148 * HERC0087
HARD0149 * HERC0088
HARD0150 * HERC0089
HARD0151 * HERC0090
HARD0152 * HERC0091
HARD0153 * HERC0092
HARD0154 * HERC0093
HARD0155 * HERC0094
HARD0156 * HERC0095
HARD0157 * HERC0096
HARD0158 * HERC0097
HARD0159 * HERC0098
HARD0160 * HERC0099
HARD0161 * HERC0100
HARD0162 * HERC0101
HARD0163 * HERC0102
HARD0164 * HERC0103
HARD0165 * HERC0104
HARD0166 * HERC0105
HARD0167 * HERC0106
HARD0168 * HERC0107
HARD0169 * HERC0108
HARD0170 * HERC0109
HARD0171 * HERC0110
HARD0172 * HERC0111
HARD0173 * HERC0112
HARD0174 * HERC0113
HARD0175 * HERC0114
HARD0176 * HERC0115
HARD0177 * HERC0116
HARD0178 * HERC0117
HARD0179 * HERC0118
HARD0180 * HERC0119
HARD0181 * HERC0120
HARD0182 * HERC0121
HARD0183 * HERC0122
HARD0184 * HERC0123
HARD0185 * HERC0124
HARD0186 * HERC0125
HARD0187 * HERC0126
HARD0188 * HERC0127
HARD0189 * HERC0128
HARD0190 * HERC0129
HARD0191 * HERC0130
HARD0192 * HERC0131
HARD0193 * HERC0132
HARD0194 * HERC0133
HARD0195 * HERC0134
HARD0196 * HERC0135
HARD0197 * HERC0136
HARD0198 * HERC0137
HARD0199 * HERC0138
HARD0200 * HERC0139
HARD0201 * HERC0140
HARD0202 * HERC0141
HARD0203 * HERC0142
HARD0204 * HERC0143
HARD0205 * HERC0144
HARD0206 * HERC0145
HARD0207 * HERC0146
HARD0208 * HERC0147
HARD0209 * HERC0148
HARD0210 * HERC0149
HARD0211 * HERC0150
HARD0212 * HERC0151
HARD0213 * HERC0152
HARD0214 * HERC0153
HARD0215 * HERC0154
HARD0216 * HERC0155
HARD0217 * HERC0156
HARD0218 * HERC0157
HARD0219 * HERC0158
HARD0220 * HERC0159
HARD0221 * HERC0160
HARD0222 * HERC0161
HARD0223 * HERC0162
HARD0224 * HERC0163
HARD0225 * HERC0164
HARD0226 * HERC0165
HARD0227 * HERC0166
HARD0228 * HERC0167
HARD0229 * HERC0168
HARD0230 * HERC0169
HARD0231 * HERC0170
HARD0232 * HERC0171
HARD0233 * HERC0172
HARD0234 * HERC0173
HARD0235 * HERC0174
HARD0236 * HERC0175
HARD0237 * HERC0176
HARD0238 * HERC0177
HARD0239 * HERC0178
HARD0240 * HERC0179
HARD0241 * HERC0180
HARD0242 * HERC0181
HARD0243 * HERC0182
HARD0244 * HERC0183
HARD0245 * HERC0184
HARD0246 * HERC0185
HARD0247 * HERC0186
HARD0248 * HERC0187
HARD0249 * HERC0188
HARD0250 * HERC0189
HARD0251 * HERC0190
HARD0252 * HERC0191
HARD0253 * HERC0192
HARD0254 * HERC0193
HARD0255 * HERC0194
HARD0256 * HERC0195
HARD0257 * HERC0196
HARD0258 * HERC0197
HARD0259 * HERC0198
HARD0260 * HERC0199
HARD0261 * HERC0200
HARD0262 * HERC0201
HARD0263 * HERC0202
HARD0264 * HERC0203
HARD0265 * HERC0204
HARD0266 * HERC0205
HARD0267 * HERC0206
HARD0268 * HERC0207
HARD0269 * HERC0208
HARD0270 * HERC0209
HARD0271 * HERC0210
HARD0272 * HERC0211
HARD0273 * HERC0212
HARD0274 * HERC0213
HARD0275 * HERC0214
HARD0276 * HERC0215
HARD0277 * HERC0216
HARD0278 * HERC0217
HARD0279 * HERC0218
HARD0280 * HERC0219
HARD0281 * HERC0220
HARD0282 * HERC0221
HARD0283 * HERC0222
HARD0284 * HERC0223
HARD0285 * HERC0224
HARD0286 * HERC0225
HARD0287 * HERC0226
HARD0288 * HERC0227
HARD0289 * HERC0228
HARD0290 * HERC0229
HARD0291 * HERC0230
HARD0292 * HERC0231
HARD0293 * HERC0232
HARD0294 * HERC0233
HARD0295 * HERC0234
HARD0296 * HERC0235
HARD0297 * HERC0236
HARD0298 * HERC0237
HARD0299 * HERC0238
HARD0300 * HERC0239
HARD0301 * HERC0240
HARD0302 * HERC0241
HARD0303 * HERC0242
HARD0304 * HERC0243
HARD0305 * HERC0244
HARD0306 * HERC0245
HARD0307 * HERC0246
HARD0308 * HERC0247
HARD0309 * HERC0248
HARD0310 * HERC0249
HARD0311 * HERC0250
HARD0312 * HERC0251
HARD0313 * HERC0252
HARD0314 * HERC0253
HARD0315 * HERC0254
HARD0316 * HERC0255
HARD0317 * HERC0256
HARD0318 * HERC0257
HARD0319 * HERC0258
HARD0320 * HERC0259
HARD0321 * HERC0260
HARD0322 * HERC0261
HARD0323 * HERC0262
HARD0324 * HERC0263
HARD0325 * HERC0264
HARD0326 * HERC0265
HARD0327 * HERC0266
HARD0328 * HERC0267
HARD0329 * HERC0268
HARD0330 * HERC0269
HARD0331 * HERC0270
HARD0332 * HERC0271
HARD0333 * HERC0272
HARD0334 * HERC0273
HARD0335 * HERC0274
HARD0336 * HERC0275
HARD0337 * HERC0276
HARD0338 * HERC0277
HARD0339 * HERC0278
HARD0340 * HERC0279
HARD0341 * HERC0280
HARD0342 * HERC0281
HARD0343 * HERC0282
HARD0344 * HERC0283
HARD0345 * HERC0284
HARD0346 * HERC0285
HARD0347 * HERC0286
HARD0348 * HERC0287
HARD0349 * HERC0288
HARD0350 * HERC0289
HARD0351 * HERC0290
HARD0352 * HERC0291
HARD0353 * HERC0292
HARD0354 * HERC0293
HARD0355 * HERC0294
HARD0356 * HERC0295
HARD0357 * HERC0296
HARD0358 * HERC0297
HARD0359 * HERC0298
HARD0360 * HERC0299
HARD0361 * HERC0300
HARD0362 * HERC0301
HARD0363 * HERC0302
HARD0364 * HERC0303
HARD0365 * HERC0304
HARD0366 * HERC0305
HARD0367 * HERC0306
HARD0368 * HERC0307
HARD0369 * HERC0308
HARD0370 * HERC0309
HARD0371 * HERC0310
HARD0372 * HERC0311
HARD0373 * HERC0312
HARD0374 * HERC0313
HARD0375 * HERC0314
HARD0376 * HERC0315
HARD0377 * HERC0316
HARD0378 * HERC0317
HARD0379 * HERC0318
HARD0380 * HERC0319
HARD0381 * HERC0320
HARD0382 * HERC0321
HARD0383 * HERC0322
HARD0384 * HERC0323
HARD0385 * HERC0324
HARD0386 * HERC0325
HARD0387 * HERC0326
HARD0388 * HERC0327
HARD0389 * HERC0328
HARD0390 * HERC0329
HARD0391 * HERC0330
HARD0392 * HERC0331
HARD0393 * HERC0332
HARD0394 * HERC0333
HARD0395 * HERC0334
HARD0396 * HERC0335
HARD0397 * HERC0336
HARD0398 * HERC0337
HARD0399 * HERC0338
HARD0400 * HERC0339
HARD0401 * HERC0340
HARD0402 * HERC0341
HARD0403 * HERC0342
HARD0404 * HERC0343
HARD0405 * HERC0344
HARD0406 * HERC0345
HARD0407 * HERC0346
HARD0408 * HERC0347
HARD0409 * HERC0348
HARD0410 * HERC0349
HARD0411 * HERC0350
HARD0412 * HERC0351
HARD0413 * HERC0352
HARD0414 * HERC0353
HARD0415 * HERC0354
HARD0416 * HERC0355
HARD0417 * HERC0356
HARD0418 * HERC0357
HARD0419 * HERC0358
HARD0420 * HERC0359
HARD0421 * HERC0360
HARD0422 * HERC0361
HARD0423 * HERC0362
HARD0424 * HERC0363
HARD0425 * HERC0364
HARD0426 * HERC0365
HARD0427 * HERC0366
HARD0428 * HERC0367
HARD0429 * HERC0368
HARD0430 * HERC0369
HARD0431 * HERC0370
HARD0432 * HERC0371
HARD0433 * HERC0372
HARD0434 * HERC0373
HARD0435 * HERC0374
HARD0436 * HERC0375
HARD0437 * HERC0376
HARD0438 * HERC0377
HARD0439 * HERC0378
HARD0440 * HERC0379
HARD0441 * HERC0380
HARD0442 * HERC0381
HARD0443 * HERC0382
HARD0444 * HERC0383
HARD0445 * HERC0384
HARD0446 * HERC0385
HARD0447 * HERC0386
HARD0448 * HERC0387
HARD0449 * HERC0388
HARD0450 * HERC0389
HARD0451 * HERC0390
HARD0452 * HERC0391
HARD0453 * HERC0392
HARD0454 * HERC0393
HARD0455 * HERC0394
HARD0456 * HERC0395
HARD0457 * HERC0396
HARD0458 * HERC0397
HARD0459 * HERC0398
HARD0460 * HERC0399
HARD0461 * HERC0400
HARD0462 * HERC0401
HARD0463 * HERC0402
HARD0464 * HERC0403
HARD0465 * HERC0404
HARD0466 * HERC0405
HARD0467 * HERC0406
HARD0468 * HERC0407
HARD0469 * HERC0408
HARD0470 * HERC0409
HARD0471 * HERC0410
HARD0472 * HERC0411
HARD0473 * HERC0412
HARD0474 * HERC0413
HARD0475 * HERC0414
HARD0476 * HERC0415
HARD0477 * HERC0416
HARD0478 * HERC0417
HARD0479 * HERC0418
HARD0480 * HERC0419
HARD0481 * HERC0420
HARD0482 * HERC0421
HARD0483 * HERC0422
HARD0484 * HERC0423
HARD0485 * HERC0424
HARD0486 * HERC0425
HARD0487 * HERC0426
HARD0488 * HERC0427
HARD0489 * HERC0428
HARD0490 * HERC0429
HARD0491 * HERC0430
HARD0492 * HERC0431
HARD0493 * HERC0432
HARD0494 * HERC0433
HARD0495 * HERC0434
HARD0496 * HERC0435
HARD0497 * HERC0436
HARD0498 * HERC0437
HARD0499 * HERC0438
HARD0500 * HERC0439
HARD0501 * HERC0440
HARD0502 * HERC0441
HARD0503 * HERC0442
HARD0504 * HERC0443
HARD0505 * HERC0444
HARD0506 * HERC0445
HARD0507 * HERC0446
HARD0508 * HERC0447
HARD0509 * HERC0448
HARD0510 * HERC0449
HARD0511 * HERC0450
HARD0512 * HERC0451
HARD0513 * HERC0452
HARD0514 * HERC0453
HARD0515 * HERC0454
HARD0516 * HERC0455
HARD0517 * HERC0456
HARD0518 * HERC0457
HARD0519 * HERC0458
HARD0520 * HERC0459
HARD0521 * HERC0460
HARD0522 * HERC0461
HARD0523 * HERC0462
HARD0524 * HERC0463
HARD0525 * HERC0464
HARD0526 * HERC0465
HARD0527 * HERC0466
HARD0528 * HERC0467
HARD0529 * HERC0468
HARD0530 * HERC0469
HARD0531 * HERC0470
HARD0532 * HERC0471
HARD0533 * HERC0472
HARD0534 * HERC0473
HARD0535 * HERC0474
HARD0536 * HERC0475
HARD0537 * HERC0476
HARD0538 * HERC0477
HARD0539 * HERC0478
HARD0540 * HERC0479
HARD0541 * HERC0480
HARD0542 * HERC0481
HARD0543 * HERC0482
HARD0544 * HERC0483
HARD0545 * HERC0484
HARD0546 * HERC0485
HARD0547 * HERC0486
HARD0548 * HERC0487
HARD0549 * HERC0488
HARD0550 * HERC0489
HARD0551 * HERC0490
HARD0552 * HERC0491
HARD0553 * HERC0492
HARD0554 * HERC0493
HARD0555 * HERC0494
HARD0556 * HERC0495
HARD0557 * HERC0496
HARD0558 * HERC0497
HARD0559 * HERC0498
HARD0560 * HERC0499
HARD0561 * HERC0500
HARD0562 * HERC0501
HARD0563 * HERC0502
HARD0564 * HERC0503
HARD0565 * HERC0504
HARD0566 * HERC0505
HARD0567 * HERC0506
HARD0568 * HERC0507
HARD0569 * HERC0508
HARD0570 * HERC0509
HARD0571 * HERC0510
HARD0572 * HERC0511
HARD0573 * HERC0512
HARD0574 * HERC0513
HARD0575 * HERC0514
HARD0576 * HERC0515
HARD0577 * HERC0516
HARD0578 * HERC0517
HARD0579 * HERC0518
HARD0580 * HERC0519
HARD0581 * HERC0520
HARD0582 * HERC0521
HARD0583 * HERC0522
HARD0584 * HERC0523
HARD0585 * HERC0524
HARD0586 * HERC0525
HARD0587 * HERC0526
HARD0588 * HERC0527
HARD0589 * HERC0528
HARD0590 * HERC0529
HARD0591 * HERC0530
HARD0592 * HERC0531
HARD0593 * HERC0532
HARD0594 * HERC0533
HARD0595 * HERC0534
HARD0596 * HERC0535
HARD0597 * HERC0536
HARD0598 * HERC0537
HARD0599 * HERC0538
HARD0600 * HERC0539
HARD0601 * HERC0540
HARD0602 * HERC0541
HARD0603 * HERC0542
HARD0604 * HERC0543
HARD0605 * HERC0544
HARD0606 * HERC0545
HARD0607 * HERC0546
HARD0608 * HERC0547
HARD0609 * HERC0548
HARD0610 * HERC0549
HARD0611 * HERC0550
HARD0612 * HERC0551
HARD0613 * HERC0552
HARD0614 * HERC0553
HARD0615 * HERC0554
HARD0616 * HERC0555
HARD0617 * HERC0556
HARD0618 * HERC0557
HARD0619 * HERC0558
HARD0620 * HERC0559
HARD0621 * HERC0560
HARD0622 * HERC0561
HARD0623 * HERC0562
HARD0624 * HERC0563
HARD0625 * HERC0564
HARD0626 * HERC0565
HARD0627 * HERC0566
HARD0628 * HERC0567
HARD0629 * HERC0568
HARD0630 * HERC0569
HARD0631 * HERC0570
HARD0632 * HERC0571
HARD0633 * HERC0572
HARD0634 * HERC0573
HARD0635 * HERC0574
HARD0636 * HERC0575
HARD0637 * HERC0576
HARD0638 * HERC0577
HARD0639 * HERC0578
HARD0640 * HERC0579
HARD0641 * HERC0580
HARD0642 * HERC0581
HARD0643 * HERC0582
HARD0644 * HERC0583
HARD0645 * HERC0584
HARD0646 * HERC0585
HARD0647 * HERC0586
HARD0648 * HERC0587
HARD0649 * HERC0588
HARD0650 * HERC0589
HARD0651 * HERC0590
HARD0652 * HERC0591
HARD0653 * HERC0592
HARD0654 * HERC0593
HARD0655 * HERC0594
HARD0656 * HERC0595
HARD0657 * HERC0596
HARD0658 * HERC0597
HARD0659 * HERC0598
HARD0660 * HERC0599
HARD0661 * HERC0600
HARD0662 * HERC0601
HARD0663 * HERC0602
HARD0664 * HERC0603
HARD0665 * HERC0604
HARD0666 * HERC0605
HARD0667 * HERC0606
HARD0668 * HERC0607
HARD0669 * HERC0608
HARD0670 * HERC0609
HARD0671 * HERC0610
HARD0672 * HERC0611
HARD0673 * HERC0612
HARD0674 * HERC0613
HARD0675 * HERC0614
HARD0676 * HERC0615
HARD0677 * HERC0616
HARD0678 * HERC0617
HARD0679 * HERC0618
HARD0680 * HERC0619
HARD0681 * HERC0620
HARD0682 * HERC0621
HARD0683 * HERC0622
HARD0684 * HERC0623
HARD0685 * HERC0624
HARD0686 * HERC0625
HARD0687 * HERC0626
HARD0688 * HERC0627
HARD0689 * HERC0628
HARD0690 * HERC0629
HARD0691 * HERC0630
HARD0692 * HERC0631
HARD0693 * HERC0632
HARD0694 * HERC0633
HARD0695 * HERC0634
HARD0696 * HERC0635
HARD0697 * HERC0636
HARD0698 * HERC0637
HARD0699 * HERC0638
HARD0700 * HERC0639
HARD0701 * HERC0640
HARD0702 * HERC0641
HARD0703 * HERC0642
HARD0704 * HERC0643
HARD0705 * HERC0644
HARD0706 * HERC0645
HARD0707 * HERC0646
HARD0708 * HERC0647
HARD0709 * HERC0648
HARD0710 * HERC0649
HARD0711 * HERC0650
HARD0712 * HERC0651
HARD0713 * HERC0652
HARD0714 * HERC0653
HARD0715 * HERC0654
HARD0716 * HERC0655
HARD0717 * HERC0656
HARD0718 * HERC0657
HARD0719 * HERC0658
HARD0720 * HERC0659
HARD0721 * HERC0660
HARD0722 * HERC0661
HARD0723 * HERC0662
HARD0724 * HERC0663
HARD0725 * HERC0664
HARD0726 * HERC0665
HARD0727 * HERC0666
HARD0728 * HERC0667
HARD0729 * HERC0668
HARD0730 * HERC0669
HARD0731 * HERC0670
HARD0732 * HERC0671
HARD0733 * HERC0672
HARD0734 * HERC0673
HARD0735 * HERC0674
HARD0736 * HERC0675
HARD0737 * HERC0676
HARD0738 * HERC0677
HARD0739 * HERC0678
HARD0740 * HERC0679
HARD0741 * HERC0680
HARD0742 * HERC0681
HARD0743 * HERC0682
HARD0744 * HERC0683
HARD0745 * HERC0684
HARD0746 * HERC0685
HARD0747 * HERC0686
HARD0748 * HERC0687
HARD0749 * HERC0688
HARD0750 * HERC0689
HARD0751 * HERC0690
HARD0752 * HERC0691
HARD0753 * HERC0692
HARD0754 * HERC0693
HARD0755 * HERC0694
HARD0756 * HERC0695
HARD0757 * HERC0696
HARD0758 * HERC0697
HARD0759 * HERC0698
HARD0760 * HERC0699
HARD0761 * HERC0700
HARD0762 * HERC0701
HARD0763 * HERC0702
HARD0764 * HERC0703
HARD0765 * HERC0704
HARD0766 * HERC0705
HARD0767 * HERC0706
HARD0768 * HERC0707
HARD0769 * HERC0708
HARD0770 * HERC0709
HARD0771 * HERC0710
HARD0772 * HERC0711
HARD0773 * HERC0712
HARD0774 * HERC0713
HARD0775 * HERC0714
HARD0776 * HERC0715
HARD0777 * HERC0716
HARD0778 * HERC0717
HARD0779 * HERC0718
HARD0780 * HERC0719
HARD0781 * HERC0720
HARD0782 * HERC0721
HARD0783 * HERC0722
HARD0784 * HERC0723
HARD0785 * HERC0724
HARD0786 * HERC0725
HARD0787 * HERC0726
HARD0788 * HERC0727
HARD0789 * HERC0728
HARD0790 * HERC0729
HARD0791 * HERC0730
HARD0792 * HERC0731
HARD0793 * HERC0732
HARD0794 * HERC0733
HARD0795 * HERC0734
HARD0796 * HERC0735
HARD0797 * HERC0736
HARD0798 * HERC0737
HARD0799 * HERC0738
HARD0800 * HERC0739
HARD0801 * HERC0740
HARD0802 * HERC0741
HARD0803 * HERC0742
HARD0804 * HERC0743
HARD0805 * HERC0744
HARD0806 * HERC0745
HARD0807 * HERC0746
HARD0808 * HERC0747
HARD0809 * HERC0748
HARD0810 * HERC0749
HARD0811 * HERC0750
HARD0812 * HERC0751
HARD0813 * HERC0752
HARD0814 * HERC0753
HARD0815 * HERC0754
HARD0816 * HERC0755
HARD0817 * HERC0756
HARD0818 * HERC0757
HARD0819 * HERC0758
HARD0820 * HERC0759
HARD0821 * HERC0760
HARD0822 * HERC0761
HARD0823 * HERC0762
HARD0824 * HERC0763
HARD0825 * HERC0764
HARD0826 * HERC0765
HARD0827 * HERC0766
HARD0828 * HERC0767
HARD0829 * HERC0768
HARD0830 * HERC0769
HARD0831 * HERC0770
HARD0832 * HERC0771
HARD0833 * HERC0772
HARD0834 * HERC0773
HARD0835 * HERC0774
HARD0836 * HERC0775
HARD0837 * HERC0776
HARD0838 * HERC0777

```

```

1      N=1,NELEM), (NODB(N), (BIV(K,N),K=1,3),N=1,NBPTC)
*      IF (NINC .LE. 0) NINC = 1
C
C
C**  ESTABLISH ORDERING OF PIPE ELEMENTS.
C      CALL RESPIP (MNO,NOD,RESULT,ST,STPIPE,U,V,X,Y,NELEM,NPMAT,
1      NPPT,NPT,ITER,1)
C
C**  STOP PROGRAM IF IN DATA CHECK MODE.
C
*      Revise NPUTCK flag for both data check and beam only problems.
*
      IF (NPUTCK .GT. 0) STOP ' DATA CHECK STOP'
      IF (NPUTCK .LT. 0) GO TO 130
+
C
C**  READ MATERIAL PROPERTIES FOR CONTINUUM ELEMENTS. (SOIL-SYSTEM)
C
C      CALL READM(NSMAT,MAXMAT,DENSTY,LEVEL)
C
C      INITIALIZATION OF STORAGE ARRAYS
C
130  DO 140 I=1,NPT
      U(I)=0.0
140  V(I)=0.0
      DO 141 I = 1, NPMAT
        STPIPE(I,I) = 0.0
141  STPIPE(2,I) = 0.0
      DO 142 I=1,NELEM
        DO 142 J = 1, 6
142  ST(J,I) = 0.0
      DO 143 I=1,NPPT
        DO 143 J=1,9
143  RESULT(J,I) = 0.0
      NINCC = 1ABS(NINC)
      IX=NEQ
      IA=0
      IEXIT =0
      IF(ICOME,EQ.1) RETURN
C
C      SECOND ENTRY SOLVES FOR A SINGLE CONSTRUCTION INCREMENT.
C      THE CONTROL VARIABLE, IEXIT, DEFINED IN THE PIPE LIBRARY CONTROLS
C      ADVANCEMENT OF THE CONSTRUCTION INCREMENT.
C      IF IEXIT .EQ. 0, ADVANCE LOAD STEP
C      IF IEXIT .EQ. -1, REPEAT LOAD STEP
C      IF IEXIT .EQ. -2, REPEAT ALL LOAD STEPS.
C
C
110  IF(IEXIT.EQ.-2) GO TO 130
      NINC = NINCC
      ICON = 1
+
*      Add JCON convergence flag for DUNCAN subroutine. SCM 1-15-89
*
      JCON = 1
+
      IF(IEXIT.NE.-1) GO TO 154
      ITER = ITER + 1
      GO TO 155
154  IA = IA + 1

```

```

HERC0067      ITER = 1
HERC0068
HERC0069 155 CONTINUE
HERC0070
HERC0071 C      THE EQUATION POSITIONING MATRIX FOR THE SYSTEM MATRIX IS FORMED
HERC0072 C      AND THE ELEVATION OF THE TOP OF THE CONSTRUCTION INCREMENT ESTABLISHED
HERC0073 C
      DO 160 I=1,NPT
HERC0074      NQ(I+1)=0
HERC0075 160 DO 198 M=1,NELEM
HERC0076      IAC=MOD(MNO(M),1000)/10
HERC0077      IF(IAC.GT.IA) GO TO 198
HERC0078      LS=200
HERC0079      IF (NOD(2,M) .EQ. NOD(3,M)) LS = 300
HERC0080      DO 180 J=1,4
HERC0081      I = NOD(J,M) + 1
HERC0082      180 IF(NQ(I).LT.300)NQ(I)=LS
HERC0083      198 CONTINUE
HERC0084      NQ(I)=100
HERC0085      DO 200 I=1,NPT
HERC0086      200 NQ(I+1)=NQ(I)+NQ(I+1)
HERC0087
HERC0088 C      THE BAND WIDTH-NPC- IS COMPUTED
HERC0089 C
      NPC=0
HERC0090
      DO 218 I=1,NELEM
HERC0091      IAC = MOD(MNO(I),1000)/10
HERC0092      IF(IAC.GT.IA) GO TO 218
HERC0093      DO 202 J=1,4
HERC0094      JJ = NOD(J,I)
HERC0095      NQ(JJ)=NQ(JJ)+1
HERC0096      DO 202 K=1,4
HERC0097      KK = NOD(K,I) + 1
HERC0098      IV=NQ(KK)/100-NQ(JJ)/100
HERC0099      IF (NPC .LT. IV) NPC=IV
HERC0100      202 CONTINUE
HERC0101      218 CONTINUE
HERC0102
HERC0103 C      COMPUTE MATRIX SPECIFICATIONS
HERC0104 C
      IF (NPC .LE. NEQ) GO TO 220
HERC0105      WRITE(6,6010) NPC,NEQ
HERC0106      RETURN
HERC0107
HERC0108 220 NCOL = NPC
HERC0109      NROW=NQ(NPT +1)/100-1
HERC0110
HERC0111 C      INITIALIZE SYSTEM MATRIX
HERC0112 C
      MN=2*NEQ
HERC0113      DO 223 N=1,MN
HERC0114      S(N) = 0.0
HERC0115      DO 223 J=1,NEQ
HERC0116      223 A(N,J) = 0.0
HERC0117
HERC0118 C      THE STIFFNESS MATRIX IS GENERATED IN BLOCKS AND STORED ON TAPE
HERC0119 C
*      Revise I/O unit from LUDATA to Unit 14.
HERC0120      SCM 11-22-88
HERC0121
HERC0122
HERC0123
HERC0124
HERC0125
HERC0126
HERC0127 REWIND 14
HERC0128
HERC0129 LB=0

```

```

HERC0130
HERC0131
HERC0132
HERC0133
HERC0134
HERC0135
HERC0136
HERC0137
HERC0138
HERC0139
HERC0140
HERC0141
HERC0142
HERC0143
HERC0144
HERC0145
HERC0146
HERC0147
HERC0148
HERC0149
HERC0150
HERC0151
HERC0152
HERC0153
HERC0154
HERC0155
HERC0156
HERC0157
HERC0158
HERC0159
HERC0160
HERC0161
HERC0162
HERC0163
HERC0164
HERC0165
HERC0166
HERC0167
HERC0168
HERC0169
HERC0170
HERC0171
HERC0172
HERC0173
HERC0174
HERC0175
HERC0176
HERC0177
HERC0178
HERC0179
HERC0180
HERC0181
HERC0182
HERC0183
HERC0184
HERC0185
HERC0186
HERC0187
HERC0188
HERC0189
HERC0190
HERC0191
HERC0192

```

```

      LBI=0
      NELEMT=0
288  LB=LB+1
      MB=LB*NEQ
C
C      EACH ELEMENT IS EXAMINED TO DETERMINE IF IT CONTRIBUTES TO THE
C      BLOCK
C
      DO 355 I=1,NELEM
      IX=I
      IF (MOD(I,I) .LT. 0) GO TO 355
      IAC=MOD(MNO(I),1000)/10
      MN=MNO(I)/1000
      IBEND = MOD(2,I) -MOD(3,I)
      MNX = MNO(I) - MN*1000 - IAC*10
      IF (IBEND.EQ.0) RO = PIPMAT(5,MN)
      IF (IBEND.NE.0) RO = DEN(MN)
      IF (MNX.GT.0) RO = 0.0
      IF (IAC-IA) 290,292,355
290  RO=0.0
292  DO 300 L=1,4
      KK = MOD(L,I)
      KI=NQ(KK)/100
      IF (KI .LE. MB) GO TO 305
300  CONTINUE
      GO TO 355
C
C      CHOOSE APPROPRIATE CONSTITUTIVE LAW FOR ELEMENT
C
305  IF (IBEND.EQ.0) GO TO 320
      IF (MNX.GT.0) GO TO 349
      ITYP = ITYPE(MN)
      GO TO (306,306,308,309,310,ITYP)
306  L=0
      DO 307 J=1,3
      DO 307 K=J,3
      L=L+1
      CP(J,K) = H(MN,L)
307  CP(K,J)=CP(J,K)
      GO TO 349
C
C      *** INSERT NEW SOIL MODELS HERE ***
C
308  CALL DUNCAN(ST,STHARD,I,ICON,ITER,MN,NELEM,IA)
      GO TO 349
C
309  CALL INTP1(NOD,ST,X,Y,MNO,EI,GNUI,IX,MN,IA,NELEM,NPT)
      CALL CONVT(EI,GNUI)
      GO TO 349
C
310  CALL HARDIN (ST,STHARD,I,ICON,ITER,MN,NELEM)
      GO TO 349
C
320  CONTINUE
      DO 321 J=1,3
      CP(1,J) = PIPMAT(J,MN)
      CP(2,J) = 0.0
321  CP(3,J) = 0.0
      CP(2,1)= PIPMAT(4,MN)
C
C      CALCULATE THAT PORTION OF THE STIFFNESS MATRIX GIVEN BY A
C      CONSIDERATION OF ELEMENT I
C

```

```

HERC0193 349 CALL STIFNS(BIV,NOD,NODS,NQ,X,Y,MNO,U,V,ST,IA,LBI,MN,IAC,MNX,
HERC0194 X ITER,ICON,NEQ,NBPTC,NELEM,NPT)
HERC0195 NELEMT=NELEMT+1
HERC0196 NOD(1,I) = -NOD(1,I)
HERC0197 355 CONTINUE
HERC0198 C
HERC0199 C IF HAVE LESS THAN 1 BLOCK OF EQUATIONS LEFT NO ADDITIONAL TAPE
HERC0200 C STORAGE IS USED
HERC0201 C
HERC0202 IF (NROW .LE. MB ) GO TO 560
HERC0203 LBI=LBI+1
HERC0204 C
HERC0205 THE BLOCK OF EQUATIONS IS REDUCED AND PUT ON TAPE
HERC0206 C
HERC0207 CALL REDUCI(NEQ,NCOL)
HERC0208 *
HERC0209 * Revise I/O unit from LUDATA to Unit 14.
HERC0210 * SCM 11-22-88
HERC0211 *
HERC0212 WRITE(14) (B(N), (A(N,M),M=1,NCOL),N=1,NEQ)
HERC0213 *
HERC0214 C
HERC0215 THE NUMBER TWO BLOCK OF EQUATIONS IS SHIFTED INTO THE NUMBER ONE
HERC0216 C POSITION
HERC0217 C
HERC0218 DO 555 L=1,NEQ
HERC0219 LRH=L+NEQ
HERC0220 B(L) = B(LRH)
HERC0221 B(LRH) = 0.0
HERC0222 DO 555 M=1,NCOL
HERC0223 A(L,M) = A(LRH,M)
HERC0224 555 A(LRH,M) = 0.0
HERC0225 GO TO 288
HERC0226 560 NT=NROW-LBI*NEQ
HERC0227 C
HERC0228 C THE FINAL TWO BLOCKS OF EQUATIONS ARE REDUCED AND BACK-SUBSTITUTED
HERC0229 C
HERC0230 CALL REDUCI(NT,NCOL)
HERC0231 CALL BAKSUB(NT,NCOL,LB,NEQ)
HERC0232 IF (LBI .EQ. 0) GO TO 564
HERC0233 DO 563 J=1,LBI
HERC0234 DO 562 L=1, NEQ
HERC0235 LL=L+NEQ
HERC0236 562 B(LL) = B(L)
HERC0237 L=LBI-J+1
HERC0238 C
HERC0239 ANY BLOCKS ON THE TAPE ARE RECALLED AND BACK-SUBSTITUTED
HERC0240 C
HERC0241 *
HERC0242 * Revise I/O unit from LUDATA to Unit 14.
HERC0243 * SCM 11-22-88
HERC0244 *
HERC0245 BACKSPACE 14
HERC0246 READ(14) (B(N), (A(N,M),M=1,NCOL),N=1,NEQ)
HERC0247 BACKSPACE 14
HERC0248 *
HERC0249 CALL BAKSUB(NEQ,NCOL,L,NEQ)
HERC0250 563 CONTINUE
HERC0251 564 L=0
HERC0252 DO 566 N=1,LB
HERC0253 K=NEQ
HERC0254 DO 566 KK=1,NEQ
HERC0255 K=K+1

```

```

HERC0256
HERC0257
HERC0258
HERC0259
HERC0260
HERC0261
HERC0262
HERC0263
HERC0264
HERC0265
HERC0266
HERC0267
HERC0268
HERC0269
HERC0270
HERC0271
HERC0272
HERC0273
HERC0274
HERC0275
HERC0276
HERC0277
HERC0278
HERC0279
HERC0280
HERC0281
HERC0282
HERC0283
HERC0284
HERC0285
HERC0286
HERC0287
HERC0288
HERC0289
HERC0290
HERC0291
HERC0292
HERC0293
HERC0294
HERC0295
HERC0296
HERC0297
HERC0298
HERC0299
HERC0300
HERC0301
HERC0302
HERC0303
HERC0304
HERC0305
HERC0306
HERC0307
HERC0308
HERC0309
HERC0310
HERC0311
HERC0312
HERC0313
HERC0314
HERC0315
HERC0316
HERC0317
HERC0318

```



```

      L=L+1
566 B(L) = A(K,N)
C
C RECORD THE STRUCTURAL RESPONSES FOR CONVERGED SOLUTIONS
C
      IF(IEXIT.EQ.-1) GO TO 580
      REWIND 13
      WRITE(13) (U(I),V(I),I=1,NPT),((ST(J,I),I=1,NELEM),J=1,6)
      1 , ((STPIPE(J,I),I=1,NPMAT),J=1,2)
      GO TO 590
C
C READ PREVIOUSLY CONVERGED STRUCTURAL RESPONSES IF IEXIT = -1
C
580 REWIND 13
      READ(13) (U(I),V(I),I=1,NPT),((ST(J,I),I=1,NELEM),J=1,6)
      1 , ((STPIPE(J,I),I=1,NPMAT),J=1,2)
C
590 CONTINUE
C
C STRESSES AND STRAINS COMPUTED
C
      CALL STRESS (BIV,MNO,NOD,MODB,NQ,ST,STPIPE,U,V,IA,NBPTC,
      1 NELEM,NELEMT,NPMAT,NPPT,NPT,SK,SM)
C
C PIPE RESULTS COLLECTED
C
      CALL RESPIF (MNO,NOD,RESULT,ST,STPIPE,U,V,X,Y,NELEM,NPMAT,
      1 NPPT,NPT,IEXIT,2)
C
      IF(ICON.NE.1) NINC = -NINCC
*
* Revise convergence limit and add JCON convergence flag for Duncan
* subroutine. SCM 1-15-89
*
* IF(ITER.LE.20) RETURN
*
* IF(ITER.LT.MAXIT) RETURN
* IF(JCON.LT.1) WRITE(6,5000) IA,ITER
*
      RETURN
C
C** FORMATS
C
6010 FORMAT(1H1,41HFATAL ERROR BANDWIDTH IS TOO LARGE /
      1 30HBANDWIDTH IS ..... ,I5 /
      2 30HMAXIMUM ALLOWABLE BANDWIDTH IS ..... ,I5 )
5000 FORMAT(/23H *** WARNING, LOAD STEP,I3,23H DID NOT CONVERGE AFTER,
      X I3,12H ITERATIONS.,23H PROGRAM WILL CONTINUE. )
      NINC = NINCC
      END
*
* *****
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
* VERSION 1.0, MARCH 1990
* PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
* THE FEDERAL HIGHWAY ADMINISTRATION
* *****
      SUBROUTINE HINGE(PIPMAT,NPMAT,PS,PDIA,YIELD,SM,SK,RESULT,NPPT,
      X PHINGE)
C
C THIS ROUTINE CALCULATES A CORRECTIONAL SET OF STRUCTURAL RESPONSES
C TO ACCOUNT FOR PLASTIC YIELDING AT THE QUARTER POINTS OF A
C CIRCULAR CULVERT.
      HING0019 C THE FOLLOWING HINGING EVALUATIONS ARE CALCULATED ...
      HING0020 C
      HING0021 C PHINGE(1) = RATIO OF ELASTIC MOMENT TO YIELD MOMENT.
      HING0022 C PHINGE(2) = RATIO OF YIELDED SECTION TO TOTAL SECTION.
      HING0023 C PHINGE(3) = RATIO OF YIELDED CIRCUMFERENCE TO TOTAL.
      HING0024 C PHINGE(4) = RATIO OF PLASTIC DISPLACEMENT TO TOTAL.
      HING0025 C PHINGE(5) = RATIO OF TOTAL STRAIN TO YIELD STRAIN.
      HING0026 C
      HING0027 C Uncomment this statement in all program modules for double
      HING0028 C precision version. SCM 3-10-90
      HING0029 C
      HING0030 C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      HING0031 C
      HING0032 C Revise dimensions of RESULT, DESULT and PIPMAT arrays from
      HING0033 C 30 to 99. SCM 4-30-89
      HING0034 C
      HING0035 C DIMENSION RESULT(20,99),XK(6,6),A(6),U(6,6),DESULT(8,99),FW(7)
      HING0036 C X , PHINGE(5),PIPMAT(5,99)
      HING0037 C
      HING0038 C SET CONSTANTS
      HING0039 C
      HING0040 C DO 50 I=1,NPPT
      HING0041 C DO 50 J=1,7
      HING0042 C DESULT(J,I) = 0.0
      HING0043 C DO 55 I=1,5
      HING0044 C PHINGE(I) = 0.0
      HING0045 C
      HING0046 C PE = PIPMAT(1,NPMAT)
      HING0047 C PNU = PIPMAT(2,NPMAT)
      HING0048 C PA = PIPMAT(3,NPMAT)
      HING0049 C PI = PIPMAT(4,NPMAT)
      HING0050 C PRAD = PDIA/2.
      HING0051 C PC = PI/PS
      HING0052 C NP0 = (NPPT+1)/2
      HING0053 C NP1 = NP0-1
      HING0054 C NP2 = NP0-2
      HING0055 C IF(RESULT(5,NP0).EQ.0.0) RETURN
      HING0056 C SIGN = RESULT(5,NP0)/ABS(RESULT(5,NP0))
      HING0057 C XME0 = RESULT(5,NP0)*SIGN
      HING0058 C XME1 = RESULT(5,NP1)*SIGN
      HING0059 C XME2 = RESULT(5,NP2)*SIGN
      HING0060 C XMYLD = YIELD*PI/PC
      HING0061 C XMULT = 1.5*XMYLD
      HING0062 C PHINGE(1) = XMYLD/XME0
      HING0063 C PHINGE(5) = XME0/(PS*YIELD)
      HING0064 C IF(XME0.LT.1.01*XMYLD) RETURN
      HING0065 C SPK = SM*(1.-SK)/PRAD
      HING0066 C ALPHA = PRAD**4*SPK*(1.-PNU**2)/(PE*PI)
      HING0067 C SQRAL = SQRT(ALPHA)
      HING0068 C XP = 0.5*SQRT(2.*SQRAL-1.)
      HING0069 C XQ = 0.5*SQRT(2.*SQRAL+1.)
      HING0070 C C1 = 1.+SQRAL
      HING0071 C C2 = -1.+SQRAL
      HING0072 C C3 = SQRT(ALPHA-.25)
      HING0073 C C4 = 0.5-ALPHA
      HING0074 C F1 = PI/(PA*PRAD**2)
      HING0075 C F2 = PRAD**2*SPK*(1.-PNU**2)/(PE*PA) + 1.
      HING0076 C F3 = PE*PI/(PRAD**2*(1.-PNU**2))
      HING0077 C F4 = YIELD*PRAD*(1.-PNU**2)/(PE*PC)
      HING0078 C HPI = 2.*ATAN(1.0)
      HING0079 C DTHETA = 2.*HPI/(NPPT-1.)
      HING0080 C BE = (-3.*XME0+4.*XME1-XME2)/(2.*DTHETA)

```

```

C      CE = (XME0-2.*XME1+XME2)/(2.*DTHETA*DTHETA)
C
C      ESTABLISH MATRIX, XK, FOR SOLVING SIX UNKNOWN COEFFICIENTS OF
C      END-MOMENT BOUNDARY VALUE PROBLEM.
C
C      CALL SETU(0.0,C1,C2,C3,C4,SQRAL,XP,XQ,U)
C      DO 100 I=1,4
C      XK(1,I) = U(5,I)
C      XK(3,I) = U(4,I)
100    XK(5,I) = F1*U(5,I) + F2*U(1,I)
C      CALL SETU(HPI,C1,C2,C3,C4,SQRAL,XP,XQ,U)
C      DO 110 I=1,4
C      XK(2,I) = U(5,I)
C      XK(4,I) = U(4,I)
110    XK(6,I) = F1*U(5,I) + F2*U(1,I)
C
C      SOLVE FOR THE FIRST FOUR UNKNOWN COEFFICIENTS.
C
C      CALL INVER(XK,4)
C      DO 120 I=1,4
120    A(I) = (XK(I,3)-XK(I,4))/F3
C
C      SOLVE FOR THE LAST TWO UNKNOWN COEFFICIENTS.
C
C      T1 = 0.0
C      T2 = 0.0
C      DO 130 I=1,4
130    T1 = T1 + XK(5,I)*A(I)
C      T2 = T2 + XK(6,I)*A(I)
C      A(5) = (T1-T2)/(F1*HPI)
C      A(6) = T1
C
C      COMPUTE STRUCTURAL RESPONSES FOR A UNIT APPLIED END MOMENTS.
C
C      THETA = HPI
C      ROTAT = 0.0
C      DO 150 N=1,NP0
C      DESULT(3,N) = A(5)
C      DESULT(4,N) = -A(5)*THETA*F2 + A(6)
C      DESULT(5,N) = 0.0
C      DESULT(6,N) = -A(5)*PRAD*SPK
C      DESULT(7,N) = 0.0
C      DESULT(8,N) = -A(5)*SPK
C      CALL SETU(THETA,C1,C2,C3,C4,SQRAL,XP,XQ,U)
C      DO 140 I=1,4
C      IF(N.EQ.NP0) ROTAT = ROTAT + U(3,I)*A(I)/PRAD
C      DESULT(3,N) = DESULT(3,N) + A(I)*U(2,I)
C      DESULT(4,N) = DESULT(4,N) - A(I)*(F1*U(5,I)+F2*U(1,I))
C      DESULT(5,N) = DESULT(5,N) + A(I)*F3*U(4,I)
C      DESULT(6,N) = DESULT(6,N) - A(I)*(F3/PRAD*U(6,I)+PRAD*SPK*U(2,I))
C      DESULT(7,N) = DESULT(7,N) + A(I)*F3/PRAD*U(5,I)
140    DESULT(8,N) = DESULT(8,N) - SPK*A(I)*U(2,I)
C      THETA = THETA-DTHETA
150    CONTINUE
C
C      COMPUTE PLASTIC HINGE ROTATION FOR THE ASSUMED END MOMENT, DMO.
C
C      DMMAX = XME0-XMYLD
C      DMMIN = XME0-XMULT
C      IF(DMMIN.LT.0.0) DMMIN = 0.0
C      DMHIGH = DMMAX
C      DMLOW = DMMIN
C      KOUNT = 0
C
C      HING0077 C
C      HING0078 200 DMO = (DMHIGH+DMLOW)/2.
C      HING0079 XMD0 =-DM0*DESULT(5,NP0)
C      HING0080 XMD1 =-DM0*DESULT(5,NP1)
C      HING0081 XMD2 =-DM0*DESULT(5,NP2)
C      HING0082 BB = (-3.*(XME0+XMD0)+4.*(XME1+XMD1)-(XME2+XMD2))/(2.*DTHETA)
C      HING0083 CC = (XME0+XMD0-2.*(XME1+XMD1)+XME2+XMD2)/(2.*DTHETA**2)
C      HING0084 C
C      HING0085 C COMPUTE LENGTH OF PLASTIC HINGE
C      HING0086 C
C      HING0087 W1 = (-BB/CC+ SQRT((BB/CC)**2 -4.*(XME0+XMD0-XMYLD)/CC))/2.0
C      HING0088 W2 = (-BB/CC- SQRT((BB/CC)**2 -4.*(XME0+XMD0-XMYLD)/CC))/2.0
C      HING0089 WP = W1
C      HING0090 IF(W2.GT.0.0) WP = W2
C      HING0091 C
C      HING0092 C DETERMINE PLASTIC ROTATION OVER HINGE LENGTH WITH 7 POINT SIMPSON
C      HING0093 C INTEGRATION FOR PART1, AND EXACT INTEGRATION FOR PART2.
C      HING0094 C
C      HING0095 NWP = 7
C      HING0096 NWP1 =NWP-1
C      HING0097 DW = WP/NWP1
C      HING0098 WW = 0.0
C      HING0099 DO 250 I=1,NWP
C      HING0100 ARG = 1.-(XME0+XMD0+BB*WW+CC*WW*WW)/XMULT
C      HING0101 IF(ARG.LE.0.0) ARG = 0.000001
C      HING0102 FW(I) = 1./SQRT(3.*ARG)
C      HING0103 WW = WW + DW
250    SIMP = 0.0
C      HING0104 DO 260 I=1,NWP1,2
C      HING0105 SIMP = SIMP + FW(I) +4.*FW(I+1) + FW(I+2)
260    PART1 = SIMP*DW/3.
C      HING0106 PART2 = WP/XMYLD*(XME0+BE*WP/2.+CE*WP*WP/3.)
C      HING0107 ROTAP = F4*(PART1-PART2)
C      HING0108 C
C      HING0109 C COMPARE ROTATION FROM ELASTIC SOLUTION,ROTAE , WITH THE ROTATION
C      HING0110 C OF THE PLASTIC HINGE, ROTAP.
C      HING0111 C
C      HING0112 ROTAE = -ROTAT*DM0
C      HING0113 RATIO = ROTAE/ROTAP
C      HING0114 IF(ABS(RATIO-1.) .LT.0.01) GO TO 300
C      HING0115 IF(ROTAE.LT.ROTAP) DMLOW = DMO
C      HING0116 IF(ROTAE.GT.ROTAP) DMHIGH = DMO
C      HING0117 KOUNT = KOUNT + 1
C      HING0118 IF(KOUNT.GT.30) GO TO 400
C      HING0119 GO TO 200
C      HING0120 C
C      HING0121 C COLLECT CONVERGED SOLUTION AND ADD TO ELASTIC SOLUTION
C      HING0122 C
C      HING0123 C
C      HING0124 C
C      HING0125 300 CONTINUE
C      HING0126 K = NPPT
C      HING0127 THETA = HPI
C      HING0128 DO 350 N=1,NP0
C      HING0129 RDISP = DESULT(3,N)*DM0
C      HING0130 TDISP = DESULT(4,N)*DM0
C      HING0131 DESULT(3,N) = RDISP*COS(THETA) - TDISP*SIN(THETA)
C      HING0132 DESULT(4,N) = RDISP*SIN(THETA) + TDISP*COS(THETA)
C      HING0133 DESULT(3,K) = DESULT(3,N)
C      HING0134 DESULT(4,K) = -DESULT(4,N)
C      HING0135 THETA = THETA-DTHETA
C      HING0136 DO 325 J=5,8
C      HING0137 DESULT(J,N) = DESULT(J,N)*DM0
C      HING0138 DESULT(J,K) = DESULT(J,N)
325    CONTINUE
C      HING0139

```

HING0140
 HING0141
 HING0142
 HING0143
 HING0144
 HING0145
 HING0146
 HING0147
 HING0148
 HING0149
 HING0150
 HING0151
 HING0152
 HING0153
 HING0154
 HING0155
 HING0156
 HING0157
 HING0158
 HING0159
 HING0160
 HING0161
 HING0162
 HING0163
 HING0164
 HING0165
 HING0166
 HING0167
 HING0168
 HING0169
 HING0170
 HING0171
 HING0172
 HING0173
 HING0174
 HING0175
 HING0176
 HING0177
 HING0178
 HING0179
 HING0180
 HING0181
 HING0182
 HING0183
 HING0184
 HING0185
 HING0186
 HING0187
 HING0188
 HING0189
 HING0190
 HING0191
 HING0192
 HING0193
 HING0194
 HING0195
 HING0196
 HING0197
 HING0198
 HING0199
 HING0200
 HING0201
 HING0202


```

      DESULT(7,K) = -DESULT(7,K)
350 K = K-1
C
      DO 360 N=1,NPPT
      DO 360 J=3,8
360 RESULT(J,N) = RESULT(J,N) + DESULT(J,N)
C
C      EVALUATE HINGE CHARACTERISTICS
C
      HLONG = WP*PRAD
      HDEEP = PC*(1.-SQRT(3.*(1.-(XME0+XMD0)/XMULT)))
      PHINGE(2) = HDEEP/(PI/PS)
      PHINGE(3) = 2.*HLONG/(HPI*PRAD)
      PHINGE(4) = 2.*DESULT(4,1)/(RESULT(4,1)-RESULT(4,NPPT))
      PHINGE(5) = 1./SQRT(3.*(1.-ABS(RESULT(5,NP0))/XMULT))
      RETURN
C
C      EXIT FOR NO SOLUTION, FULL PENETRATION
C
400 PHINGE(1) = -PHINGE(1)
      WRITE(6,2000) PA,PI,PS
      RETURN
C
2000 FORMAT(1H1,10X,46H FULL PLASTIC PENETRATION OF HINGE HAS FORMED, /
X 10X,48H AND A CORRECTIONAL SOLUTION CANNOT BE OBTAINED. /
X 10X, 48H (IT IS SUGGESTED TO USE STIFFER SOIL MATERIAL.) /
X 10X, 53H THE FOLLOWING RESULTS ARE FROM THE ELASTIC SOLUTION, /
X 10X, 42H WHERE THE PIPE SECTION PROPERTIES ARE ... //
X 15X,36H THRUST AREA PER UNIT LENGTH ..... , E15.5 //
X 15X,36H MOM. OF INERTIA PER UNIT LENGTH ... , E15.5 //
X 15X,36H SECTION MODULUS PER UNIT LENGTH ... , E15.5 //)
      END
* * * * *
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
* VERSION 1.0, MARCH 1990
* PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
* THE FEDERAL HIGHWAY ADMINISTRATION
* * * * *
SUBROUTINE INTPl(NOD,ST,X,Y,MNO,EI,GNUI,NEL,MN,IA,NELEM,NPT)
C
C THIS ROUTINE CALCULATES YOUNGS MODULUS AND POISSONS RATIO AS A
C FUNCTION OF OVERBURDEN PRESSURE, I.E. Y-STRESS.
C
* Uncomment this statement in all program modules for double
* precision version. SCM 3-10-90
*
* IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
      DIMENSION NOD(4,NELEM),ST(6,NELEM),X(NPT),Y(NPT),MNO(NELEM)
      COMMON/MATERL/ CP(3,3),DEN(10),E(10,10),GNU(10,10),H(10,10),
1      ITYPE(10),NON(10)
C
C FOR THE ELEMENT NEL FIND COORDINATES OF CENTER, (XNEL,YNEL)
C
      NOMI = NON(MN)
      XNEL = 0.0
      YNEL = 0.0
      DO 100 J=1,4
      ND = NOD(J,NEL)
      XNEL = XNEL + 0.25*X(ND)
100 YNEL = YNEL + 0.25*Y(ND)
C
C FIND THOSE ELEMENTS IN CURRENT CONSTRUCTION INCREMENT BEING
      HINGO203 C PLACED ON TOP OF ELEMENT NEL.
      HINGO204 C
      HINGO205 DPRES = 0.0
      HINGO206 DO 200 I=1,NELEM
      HINGO207 IAC = MOD(MNO(I),1000)/10
      HINGO208 IF(IAC.NE.IA) GO TO 200
      HINGO209 MAT = MNO(I)/1000
      HINGO210 MNX = MNO(I)-MAT*1000-IAC*10
      HINGO211 IF(MNX.NE.0) GO TO 200
      HINGO212 ND1 = IABS(NOD(1,I))
      HINGO213 ND1 = IABS(NOD(1,I))
      HINGO214 ND2 = NOD(2,I)
      HINGO215 ND3 = NOD(3,I)
      HINGO216 ND4 = NOD(4,I)
      HINGO217 IF(ND2.EQ.ND3) GO TO 200
      HINGO218 *
      HINGO219 * Revise intrinsic function to generic form. SCM 3-10-90
      HINGO220 *
      HINGO221 XMIN = MIN(X(ND1),X(ND2),X(ND3),X(ND4))
      HINGO222 XMAX = MAX(X(ND1),X(ND2),X(ND3),X(ND4))
      HINGO223 *
      HINGO224 IF(XNEL.LT.XMIN) GO TO 200
      HINGO225 IF(XNEL.GE.XMAX) GO TO 200
      HINGO226 *
      HINGO227 YMAX = MAX(Y(ND1),Y(ND2),Y(ND3),Y(ND4))
      HINGO228 *
      HINGO229 IF(YNEL.GT.YMAX) GO TO 200
      HINGO230 C
      HINGO231 C SUM PRESSURE CONTRIBUTIONS OF EACH NEW ELEMENT ABOVE NEL.
      HINGO232 C
      HINGO233 YDEPTH = 0.5*(ABS(Y(ND1)-Y(ND3)) + ABS(Y(ND2)-Y(ND4)))
      HINGO234 IF(I.EQ.NEL) YDEPTH = 0.5*YDEPTH
      INTF0001 GAM = DEN(MAT)
      INTF0002 DPRES = DPRES + GAM*YDEPTH
      INTF0003 200 CONTINUE
      INTF0004 C
      INTF0005 C TOTAL PRESSURE ON NEL DURING THIS INCREMENT IS PREVIOUS PRESSURE
      INTF0006 C PLUS ONE-HALF OF CURRENT OVERBURDEN INCREMENT.
      INTF0007 C
      INTF0008 PNEL = ABS(ST(5,NEL)) + 0.5*DPRES
      INTF0009 C
      INTF0010 C FIND OVERBURDEN PRESSURE DATA POINTS ON EITHER SIDE OF PNEL.
      INTF0011 C
      INTF0012 K1 = 1
      INTF0013 K2 = 2
      INTF0014 DO 300 J=2,NOMI
      INTF0015 J1 = J-1
      INTF0016 IF(PNEL.LT.H(MN,J1)) GO TO 300
      INTF0017 IF(PNEL.GE.H(MN,J)) GO TO 300
      INTF0018 K1 = J1
      INTF0019 K2 = J
      INTF0020 GO TO 350
      INTF0021 300 CONTINUE
      INTF0022 IF(PNEL.LT.H(MN,NOMI)) GO TO 350
      INTF0023 K1 = NOMI-1
      INTF0024 K2 = NOMI
      INTF0025 350 DH = H(MN,K2)-H(MN,K1)
      INTF0026 C
      INTF0027 C COMPUTE CHORD MODULUS AND AVERAGE POISSON RATIO
      INTF0028 C
      INTF0029 DHB = H(MN,K2) - H(MN,K1)*E(MN,K2)/E(MN,K1)
      INTF0030 IF(DHB.LE.0.0) DHB = 0.01*DH
      INTF0031 EI = E(MN,K2)*DH/DHB
      INTF0032
      INTF0033
      INTF0034
      INTF0035
      INTF0036
      INTF0037
      INTF0038
      INTF0039
      INTF0040
      INTF0041
      INTF0042
      INTF0043
      INTF0044
      INTF0045
      INTF0046
      INTF0047
      INTF0048
      INTF0049
      INTF0050
      INTF0051
      INTF0052
      INTF0053
      INTF0054
      INTF0055
      INTF0056
      INTF0057
      INTF0058
      INTF0059
      INTF0060
      INTF0061
      INTF0062
      INTF0063
      INTF0064
      INTF0065
      INTF0066
      INTF0067
      INTF0068
      INTF0069
      INTF0070
      INTF0071
      INTF0072
      INTF0073
      INTF0074
      INTF0075
      INTF0076
      INTF0077
      INTF0078
      INTF0079
      INTF0080
      INTF0081
      INTF0082
      INTF0083
      INTF0084
      INTF0085
      INTF0086
      INTF0087
      INTF0088
      INTF0089
      INTF0090
      INTF0091
      INTF0092
      INTF0093
      INTF0094

```

```

      GNUI = 0.5*(GNU(MN,K2) + GNU(MN,K1))
      RETURN
      END
* * * * *
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
*   VERSION 1.0, MARCH 1990
*   PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
*   THE FEDERAL HIGHWAY ADMINISTRATION
* * * * *
      SUBROUTINE INVER(A,NMAX)
C
C   INVERT MATRIX A
C
*   Uncomment this statement in all program modules for double
*   precision version. SCM 3-10-90
*
*   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
      DIMENSION A(6,6)
      DO 200 N=1,NMAX
        D=A(N,N)
        DO 100 J =1,NMAX
          A(N,J) = -A(N,J)/D
          DO 150 I=1,NMAX
            IF(N-I) 110,150,110
          110 DO 140 J=1,NMAX
            IF(N-J) 120,140,120
          120 A(I,J)=A(I,J)+A(I,N)*A(N,J)
          140 CONTINUE
          150 A(I,N)=A(I,N)/D
          A(N,N) =1.0/D
        200 CONTINUE
      RETURN
      END
* * * * *
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
*   VERSION 1.0, MARCH 1990
*   PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
*   THE FEDERAL HIGHWAY ADMINISTRATION
* * * * *
C
C   * * * * * SUBROUTINE JMOD * * * * *
C   NEW ADDITION SUBROUTINE IN 1983 FOR JOINT SLIPPING BEHAVIOR
C
      SUBROUTINE JMOD(STRN,STRNPR,PE,PES,PEP,PEF,STRES,STRESP,PEOLD
*,PENEW,PNU,IFLAG,JCON,NJ)
C
C   THE OBJECTIVE OF THIS SUBROUTINE IS TO COMPUTE PENEW AND IFLAG,
C   WHERE PENEW IS THE CURRENT EFFECTIVE YOUNGS MODULUS FOR AXIAL
C   STRAIN, AND IFLAG IS A CODE FOR IDENTIFYING WHAT REGION CONTROLS
C   THE CALCULATIONS OF PENEW
C
C   IFLAG=0 NO REGION, (ERROR SOMEHOW)
C   IFLAG=1 ELASTIC REGION, (PRE SLIP)
C   IFLAG=2 TRANSITION, (ELASTIC-SLIP REGION)
C   IFLAG=3 SLIP REGION
C   IFLAG=4 TRANSITION, (SLIP-POSTSLIP REGION)
C   IFLAG=5 POST SLIP REGION
C   IFLAG=6 TRANSITION, (POSTSLIP-FAILURE REGION)
C   IFLAG=7 FAILURE REGION
C   IFLAG=-1 UNLOADING,ELASTIC REGION
C
      START BY SETTING ABSOLUTE VALUES TO CURRENT STRAIN OF
      INTP0095 C PREVIOUS CONVERGED LOAD STEP JMOD0030
      INTP0096 C JMOD0031
      INTP0097 * Uncomment this statement in all program modules for double JMOD0032
      INVR0001 * precision version. SCM 3-10-90 JMOD0033
      INVR0002 * JMOD0034
      INVR0003 * IMPLICIT DOUBLE PRECISION (A-H,O-Z) JMOD0035
      INVR0004 * JMOD0036
      INVR0005 DIMENSION ANDA1(15),JFLAG(15),ANDA2(15) JMOD0037
      INVR0006 COMMON/LIMITS/PSLIP,SSTRN,PFAIL JMOD0038
      INVR0007 ASTRN=ABS(STRN) JMOD0039
      INVR0008 ASTRNP=ABS(STRNPR) JMOD0040
      INVR0009 ASTRS=ABS(STRES) JMOD0041
      INVR0010 ASTRSP=ABS(STRESP) JMOD0042
      INVR0011 IFLAG=0 JMOD0043
      INVR0012 C JMOD0044
      INVR0013 C CHECK IF UNLOADING IS OCCURING. IF SO, USE ELASTIC MODULUS JMOD0045
      INVR0014 C JMOD0046
      INVR0015 IF(ASTRN.LT.ASTRNP)IFLAG=-1 JMOD0047
      INVR0016 IF(IFLAG.EQ.0)GO TO 100 JMOD0048
      INVR0017 PENEW=PE JMOD0049
      INVR0018 RETURN JMOD0050
      INVR0019 C JMOD0051
      INVR0020 C CHECK IF LOADING IN ELASTIC REGION. IF SO, USE ELASTIC MODULUS. JMOD0052
      INVR0021 C JMOD0053
      INVR0022 100 IF(ASTRS.LT.PSLIP) IFLAG=1 JMOD0054
      INVR0023 IF(IFLAG.EQ.0) GO TO 200 JMOD0055
      INVR0024 IF(JFLAG(NJ).EQ.2)STRES=STRES/ASTRS*PSLIP JMOD0056
      INVR0025 IF(JFLAG(NJ).EQ.2)ASTRS=ABS(STRES) JMOD0057
      INVR0026 IF(JFLAG(NJ).EQ.2) GO TO 200 JMOD0058
      INVR0027 JFLAG(NJ)=IFLAG JMOD0059
      INVR0028 PENEW=PE JMOD0060
      INVR0029 RETURN JMOD0061
      INVR0030 C JMOD0062
      INVR0031 C CHECK IF LOADING IN TRANSITION REGION ELASTIC-SLIP. JMOD0063
      JMOD0001 C IF SO, USE WEIGHTED TRANSITION MODULUS. JMOD0064
      JMOD0002 C WEIGHT IS DETERMINED ON FIRST ITERATION SUCH THAT THE JMOD0065
      JMOD0003 C FIRST GUESS FOR PENEW IS 1.01*PES (THIS WORKS BEST JMOD0066
      JMOD0004 C FOR CONVERGENCE) JMOD0067
      JMOD0005 C JMOD0068
      JMOD0006 200 IF(ASTRS.GE.PSLIP .AND. PSLIP.GT.ASTRSP)IFLAG=2 JMOD0069
      JMOD0007 IF(IFLAG.EQ.0) GO TO 300 JMOD0070
      JMOD0008 IF(ASTRN.GT.SSTRN .AND. PSLIP.GT.ASTRSP) GO TO 900 JMOD0071
      JMOD0009 PENEW=(PE-PES)*(PSLIP-ASTRSP)*(1.-PNU**2)/PE/(ASTRN-ASTRNP)+PES JMOD0072
      JMOD0010 IF(JCON.NE.1) GO TO 250 JMOD0073
      JMOD0011 ANDA=(PEOLD-1.01*PES)/(PEOLD-PENEW) JMOD0074
      JMOD0012 IF(ANDA.LT.1.0)ANDA=1.0 JMOD0075
      JMOD0013 ANDA1(NJ)=ANDA JMOD0076
      JMOD0014 250 PENEW=ANDA1(NJ)*PENEW+(1.0-ANDA1(NJ))*PEOLD JMOD0077
      JMOD0015 IF(PENEW.GT.PE)PENEW=(PE+PEOLD)/2. JMOD0078
      JMOD0016 IF(PENEW.LT.PES.AND. JCON.NE.1)PENEW=(PES+PEOLD)/2. JMOD0079
      JMOD0017 IF(PENEW.LT.PES.AND. JCON.EQ.1)PENEW=1.01*PES JMOD0080
      JMOD0018 JFLAG(NJ)=IFLAG JMOD0081
      JMOD0019 RETURN JMOD0082
      JMOD0020 C JMOD0083
      JMOD0021 C CHECK IF LOADING IN PURE SLIPPING REGION. IF SO,USE SLIPPING JMOD0084
      JMOD0022 C MODULUS PES JMOD0085
      JMOD0023 C JMOD0086
      JMOD0024 300 IF(SSTRN.GT.ASTRN .AND. ASTRSP.GE.PSLIP) IFLAG=3 JMOD0087
      JMOD0025 IF(IFLAG.EQ.0) GO TO 400 JMOD0088
      JMOD0026 IF(JFLAG(NJ).EQ.4 .AND. ASTRN.LT.SSTRN) STRN=STRN/ASTRN*SSTRN JMOD0089
      JMOD0027 IF(JFLAG(NJ).EQ.4 .AND. ASTRN.LT.SSTRN) ASTRN=ABS(STRN) JMOD0090
      JMOD0028 IF(JFLAG(NJ).EQ.4) GO TO 400 JMOD0091
      JMOD0029 PENEW=PES JMOD0092

```

```

      RETURN
C
C CHECK IF LOADING IN TRANSITION REGION SLIP-POSTSLIP. IF SO,
C USE WEIGHTED TRANSITION MODULUS
C
400 IF(ASTRN.GE.SSTRN .AND. SSTRN.GT.ASTRNP) IFLAG=4
   IF(IFLAG.EQ.0) GO TO 500
   JFLAG(NJ)=IFLAG
   PENEW=PEOLD*(ASTRN-ASTRNP)/((SSTRN-ASTRNP)*(1.-PES/PEP)
   * +(ASTRN-ASTRNP)*PEOLD/PEP)
*
* Revise intrinsic function to generic form. SCM 3-10-90
*
   IF(JCON.EQ.1 .AND. PEOLD.EQ.PES) SRATIO=LOG10(PEP/PES)/3.*
   * (SSTRN-ASTRNP)/(ASTRN-SSTRN)
*
   IF(PENEW.LT.PEOLD) GO TO 450
   ANDA=2.*PENEW/PEOLD*SRATIO
   IF(ANDA.LT.1.2) ANDA=1.2
   IF((PENEW/PEOLD).LT.1.01) ANDA=1.0
   IF(PES.GT.0.0003*PEP) ANDA=1.0
   PENEW=ANDA*PENEW+(1.-ANDA)*PEOLD
   RETURN
450 ANDA=2.*PEOLD/PENEW
   IF((PEOLD/PENEW).LT.1.01) ANDA=1.0
   PENEW=ANDA*PENEW+(1.-ANDA)*PEOLD
   RETURN
C
C CHECK IF LOADING IN POSTSLIP REGION. IF SO, USE POSTSLIP MODULUS
C
500 IF(PFAIL.GT.ASTRS .AND. ASTRNP.GE.SSTRN) IFLAG=5
   IF(IFLAG.EQ.0) GO TO 600
   IF(JFLAG(NJ).EQ.6) STRES=STRES/ASTRS*PFAIL
   IF(JFLAG(NJ).EQ.6) ASTRS=ABS(STRES)
   IF(JFLAG(NJ).EQ.6) GO TO 600
   JFLAG(NJ)=IFLAG
   PENEW=PEP
   RETURN
C
C CHECK IF LOADING IN TRANSITION REGION POSTSLIP-FAILURE. IF SO,
C USE WEIGHTED TRANSITION MODULUS.
C
600 IF(ASTRS.GE.PFAIL .AND. ASTRSP.LT.PFAIL) IFLAG=6
   IF(IFLAG.EQ.0) GO TO 700
   IF(JFLAG(NJ).EQ.4) GO TO 660
   PENEW1=(PEP-PEF)*(PFAIL-ASTRSP)*(1.-PNU**2)/PEP/(ASTRN-ASTRNP)+PEF
   IF(JCON.NE.1) GO TO 650
   ANDA=(PEOLD-1.01*PEF)/(PEOLD-PENEW1)
   IF(ANDA.LT.1.0) ANDA=1.0
   ANDA2(NJ)=ANDA
650 PENEW=ANDA2(NJ)*PENEW1+(1.-ANDA2(NJ))*PEOLD
   IF(PENEW.LT.PEF .AND. JCON.NE.1) PENEW=(PEF+PEOLD)/2.
   IF(PENEW.LT.PEF .AND. JCON.EQ.1) PENEW=1.01*PEF
   IF(PENEW.GT.PEF) PENEW=(PEP+PEOLD)/2.
   JFLAG(NJ)=IFLAG
   RETURN
C
660 IFLAG=5
   JFLAG(NJ)=IFLAG
   PENEW=PEP
   RETURN
C
C CHECK IF LOADING IN FAILURE REGION. IF SO, USE FAILURE MODULUS.

```

```

JMOD0093
JMOD0094
JMOD0095
JMOD0096
JMOD0097
JMOD0098
JMOD0099
JMOD0100
JMOD0101
JMOD0102
JMOD0103
JMOD0104
JMOD0105
JMOD0106
JMOD0107
JMOD0108
JMOD0109
JMOD0110
JMOD0111
JMOD0112
JMOD0113
JMOD0114
JMOD0115
JMOD0116
JMOD0117
JMOD0118
JMOD0119
JMOD0120
JMOD0121
JMOD0122
JMOD0123
JMOD0124
JMOD0125
JMOD0126
JMOD0127
JMOD0128
JMOD0129
JMOD0130
JMOD0131
JMOD0132
JMOD0133
JMOD0134
JMOD0135
JMOD0136
JMOD0137
JMOD0138
JMOD0139
JMOD0140
JMOD0141
JMOD0142
JMOD0143
JMOD0144
JMOD0145
JMOD0146
JMOD0147
JMOD0148
JMOD0149
JMOD0150
JMOD0151
JMOD0152
JMOD0153
JMOD0154
JMOD0155
C
700 IF(ASTRS.GT.PFAIL .AND. ASTRSP.GE.PFAIL) IFLAG=7
   IF(IFLAG.EQ.0) GO TO 800
   IF(JFLAG(NJ).EQ.4) GO TO 900
   PENEW=PEF
   RETURN
C
C CHECK IF IFLAG STILL EQUAL ZERO. IF SO, WE HAVE AN ERROR,
C I.E., LOAD STEP IS TOO LARGE
C
800 WRITE(6,2000)
   GO TO 950
900 WRITE(6,2100)
950 STOP
2000 FORMAT(////,44H **FATAL ERROR: I.E.OR TOO LARGE LOAD STEP**)
2100 FORMAT(////,20H TOO LARGE LOAD STEP)
C*****JMOD0172
      END
      JMOD0173
      * * * * *
      * CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
      *
      * VERSION 1.0, MARCH 1990
      *
      * PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
      * THE FEDERAL HIGHWAY ADMINISTRATION
      * * * * *
      * SUBROUTINE MODMSH (BIV,NOD,NODB,X,Y,NBPTC,MAXBC,NELEM,NPT)
      *
      * ROUTINE ALLOWS THE USER TO MODIFY THE LEVEL 2 MESH
      *
      * Uncomment this statement in all program modules for double
      * precision version. SCM 3-10-90
      *
      * IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      *
      * DIMENSION BIV(3,NBPTC), NOD(7,NPT), NODB(NBPTC),X(NPT),Y(NPT)
      *
      * Revise array dimension from AN(2) to AN(3) for generalized
      * boundary condition option. SCM 10-25-89
      *
      * DIMENSION AN(3),ANI(2),BV(3),IFLAG(2),ND(7)
      * DIMENSION RBC(2)
      *
      * DATA ANI /4HF = ,4HD = /
      *
      * Add data statement for generalized boundary condition option,
      * SCM 10-25-89
      *
      * DATA RBC /4H F ,4H D /
      *
      * READ A CONTROL CARD TO DETERMINE WHAT PARAMETERS ARE TO BE CHANGED
      *
      * READ (5,5010) NEWXY, NEWEL, NEWBD
      * WRITE(6,6010) NEWXY, NEWEL, NEWBD
      * IF (NEWXY + NEWEL+ NEWBD .EQ. 0) GO TO 120
      * IF (NEWXY .EQ. 0) GO TO 30
      * WRITE( 6,6020)
      *
      * CHANGE THE NODE COORDINATES
      *
      * DO 20 N = 1, NEWXY
      * READ(5,5015) NEWNOD, XCOORD, YCOORD
      * IF (NEWNOD .GT. 0 .AND. NEWNOD .LT. NPT) GO TO 10
      * WRITE(6,6030) NEWNOD

```

```

      GO TO 20
10  WRITE(6,6040) NEWNOD, X(NEWNOD),Y(NEWNOD),XCOORD,YCOORD
      X(NEWNOD) = XCOORD
      Y(NEWNOD) = YCOORD
20  CONTINUE
C
C**  READ IN MODIFIED ELEMENTS
C
30  CONTINUE
      IF (NEWEL .EQ. 0) GO TO 70
      WRITE(6,6050)
      DO 60 N = 1, NEWEL
      READ(5,5030) NE, (ND(I),I=1,7)
      IF (NE.GT. 0 .AND. NE .LE. NELEM) GO TO 40
      WRITE(6,6060) NE
      GO TO 60
40  CONTINUE
      DO 50 K = 1, 7
      IF (ND(K) .EQ. 0) ND(K) = NOD(K,NE)
      IF (ND(K) .EQ. -1) ND(K) = 0
50  CONTINUE
      WRITE(6,6070) NE, (NOD(I,NE),I=1,7), (ND(I),I=1,7)
      DO 55 K = 1, 7
      55 NOD(K,NE) = ND(K)
60  CONTINUE
70  CONTINUE
C
C**  DEFINE NEW BOUNDARY CONDITIONS
C
      IF (NEWBD .EQ. 0) RETURN
      WRITE(6,6090)
      DO 110 N = 1, NEWBD
      READ(5,5040) NU, IFLAG(1), BV(1), IFLAG(2), BV(2), BV(3), IA
      NBPTC = NBPTC + 1
      IF (NBPTC .LE. MAXBC) GO TO 75
      WRITE(6,6120) MAXBC, NBPTC
      GO TO 110
75  IF (NU .GT. 0 .AND. NU .LE. NPT) GO TO 80
      WRITE(6,6030) NU
      STOP 76
80  CONTINUE
      DO 90 J = 1, 2
      IF (IFLAG(J) .EQ. 0) AN(J) = ANI(1)
      IF (IFLAG(J) .EQ. 1) AN(J) = ANI(2)
*
*  Add extended boundary condition definition.  MGK 3-18-89
*
      IF (IFLAG(J) .EQ. 2) AN(J) = ANI(2)
      IF (IFLAG(J) .EQ. 3) AN(J) = ANI(1)
*
90  CONTINUE
*
*  Extend boundary condition output for rotational constraints.
*  SCM 10-25-89
*
      IF (IFLAG(1) .EQ. 0 .AND. IFLAG(2) .EQ. 0) THEN
        AN(3) = RBC(1)
      ELSE IF (IFLAG(1) .EQ. 2 .AND. IFLAG(2) .EQ. 0) THEN
        AN(3) = RBC(1)
      ELSE IF (IFLAG(1) .EQ. 0 .AND. IFLAG(2) .EQ. 2) THEN
        AN(3) = RBC(1)
      ELSE IF (IFLAG(1) .EQ. 2 .AND. IFLAG(2) .EQ. 2) THEN
        AN(3) = RBC(1)

```

```

MODM0046      ELSE
MODM0047        AN(3) = RBC(2)
MODM0048      END IF
MODM0049      WRITE(6,6080) NU, IA, AN(1), BV(1), AN(2), BV(2),
MODM0050      X          BV(3), AN(3)
MODM0051      BV(3) = BV(3) * 0.0174533
MODM0052  *
MODM0053      DO 100 I = 1, 3
MODM0054      100 BIV(I, NBPTC) = BV(I)
MODM0055      NODB(NBPTC) = NU * 10000 + IA * 100 + IFLAG(1) * 10 + IFLAG(2)
MODM0056      110 CONTINUE
MODM0057      120 CONTINUE
MODM0058      RETURN
MODM0059      5010 FORMAT (3I5)
MODM0060      5015 FORMAT (15, 5X, 2F10.0)
MODM0061      5030 FORMAT (8I5)
MODM0062      5040 FORMAT (2I5, F10.0, 15, F10.0, F10.0, 15)
MODM0063      6010 FORMAT (1H1, 10X, 47H * * * CHANGES TO STANDARD LEVEL 2 MESH * * *
MODM0064      1          /10X, 40H * NUMBER OF NODES TO BE CHANGED -----, 17,
MODM0065      1          1H* /10X, 40H * NUMBER OF ELEMENTS TO BE CHANGED ----, 17,
MODM0066      1          1H* /10X, 40H * ADDITIONAL BOUNDARY CONDITIONS -----, 17,
MODM0067      4          1H* /10X, 47H * * * * *
MODM0068      5          ///)
MODM0069      6020 FORMAT (1H , 13X, 11HNODE NUMBER, 13X, 15HOLD COORDINATES, 20X,
MODM0070      1          15HNEW COORDINATES /
MODM0071      2          40X, 1HX, 10X, 1HY, 25X, 1HX, 10X, 1HY)
MODM0072      6030 FORMAT (///103H * * WARNING NODE NUMBER IS OUTSIDE OF ALLOWABLE RANGE
MODM0073      1          1GE (0 TO NPT) AND CARD WILL BE SKIPPED NODE READ = , 15 ///)
MODM0074      6040 FORMAT (15X, 15, 10X, F10.2, 10X, F10.2, 15X, F10.2, 10X, F10.2)
MODM0075      6050 FORMAT (///10X, 15H ELEMENT NUMBER, 10X, 19H ELEMENT CONNECTIVE,
MODM0076      1          5X, 16H MATERIAL NUMBER, 2X, 19H CONSTRUCTION INCR. , 5X,
MODM0077      2          4HX(7) / 33X, 1HI, 7X, 1HJ, 7X, 1HK, 7X, 1HL/)
MODM0078      6060 FORMAT (///10X, 46H * * WARNING ELEMENT NUMBER IS OUT OF BOUNDS.
MODM0079      1          , 40H (1 TO NELEM) ELEMENT NUMBER IS ***** , 15///)
MODM0080      6070 FORMAT(14X, 15, 7X, 4(4X, 14), 7X, 14, 15X, 15, 12X, 14, 5X,
MODM0081      1          22HOLD ELEMENT PARAMETERS/26X, 4(4X, 14), 7X, 14, 15X, 15,
MODM0082      2          12X, 14, 5X, 22HNEW ELEMENT PARAMETERS /)
MODM0083  *
MODM0084  *  Revise formats for generalized boundary condition option.
MODM0085  *  SCM 10-25-89
MODM0086  *
MODM0087      6090 FORMAT(1H1, 1X, 31H* * * BOUNDARY CONDITIONS * * */2X,
MODM0088      X59H(FORCES = LBS; DISPLACEMENTS = INCHES; ROTATIONS = DEGREES)//
MODM0089      X2X, 8HBOUNDARY, 6X, 4HLOAD, 6X, 14HX-FORCE      OR, 6X, 14HY-FORCE      OR,
MODM0090      X6X, 12HX-Y ROTATION, 6X, 15HBEAM ROTATIONAL/6X, 4HNODE, 6X, 4HSTEP, 6X,
MODM0091      X14HX-DISPLACEMENT, 6X, 14HY-DISPLACEMENT, 22X,
MODM0092      X19HBOUNDARY CONDITIONS//)
MODM0093      6080 FORMAT(5X, 15, 5X, 15, 4X, A4, E12.4, 4X, A4, E12.4, 4X, E12.4, 14X, A4)
MODM0094  *
MODM0095      6120 FORMAT (1H1, 50H FATAL ERROR MAX. NUMBER OF BOUNDARY POINTS IS , 15/MODM0158
MODM0096      1          50H NUMBER OF BOUNDARY POINTS REQUIRED BY INPUT , 15/MODM0159
MODM0097      2          50H TO RUN THIS PROBLEM INCREASE SIZE OF BLANK COMMON /MODM0160
MODM0098      3          50H IN SUBROUTINE PREHERO AND RECOMPILE THE ROUTINE )MODM0161
MODM0099      END
MODM0100  *
MODM0101  *  CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
MODM0102  *  VERSION 1.0, MARCH 1990
MODM0103  *  PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
MODM0104  *  THE FEDERAL HIGHWAY ADMINISTRATION
MODM0105  *  *
MODM0106      SUBROUTINE PLASTI(IA, ICOME, IEXIT, LEVEL, NINC, NPMAT, NPPT, PDIA,
MODM0107      1          PTPMAT, RESULT, SK, SM, XMODE)
MODM0108  C  SUBROUTINE FOR PLASTIC PIPE CULVERTS.
MODM0109
MODM0110
MODM0111
MODM0112
MODM0113
MODM0114
MODM0115
MODM0116
MODM0117
MODM0118
MODM0119
MODM0120
MODM0121
MODM0122
MODM0123
MODM0124
MODM0125
MODM0126
MODM0127
MODM0128
MODM0129
MODM0130
MODM0131
MODM0132
MODM0133
MODM0134
MODM0135
MODM0136
MODM0137
MODM0138
MODM0139
MODM0140
MODM0141
MODM0142
MODM0143
MODM0144
MODM0145
MODM0146
MODM0147
MODM0148
MODM0149
MODM0150
MODM0151
MODM0152
MODM0153
MODM0154
MODM0155
MODM0156
MODM0157
MODM0158
MODM0159
MODM0160
MODM0161
MODM0162
MODM0001
MODM0002
MODM0003
MODM0004
MODM0005
MODM0006
MODM0007
MODM0008
MODM0009

```

```

C      SUBROUTINE FOR CORRUGATED STEEL CULVERTS.
C
C      THE FOLLOWING ADDITIONS TO THE * RESULT * ARRAY ARE CALCULATED
C      IN THIS SUBROUTINE.
C
C          RESULT(10,N) = BENDING STRESSES IN OUTER FIBERS.
C          RESULT(11,N) = THRUST STRESS (AVERAGE OVER SECTION)
C          RESULT(12,N) = SHEAR STRESS (AVERAGE OVER SECTION)
C
C      Uncomment this statement in all program modules for double
C      precision version. SCM 3-10-90
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C      Revise pipe dimension statements from 30 to 99.
C      SCM 5-22-89
C
C      DIMENSION FS(3),PFS(3),PIPMAT(5,99),RESULT(20,99)
C
C      DATA E,XNU,YIELD/1600000.,0.3,25000./
C      DATA NFS,FS/3,4.,3.,3./
C      DATA FF /0.333/
C      DATA ANALYS, DESIGN /4HANAL,4HDES1/
C
C      GO TO (100,200),ICOME
C
C      FIRST ENTRY ESTABLISHES PIPE PROPERTIES, AND SAFETY FACTORS
C
100  READ(5,1000) PID ,PE,PNU,PYIELD,PDEN
      IF(PE.LE.0.0) PE = E
      IF(PNU.LE. 0.0) PNU = XNU
      IF(PYIELD.LE.0.0) PYIELD = YIELD
      PTMIN = (12.0 * PID ** 2 / (FF * PE)) ** 0.333
C
      WRITE(6,2000) PID ,PE,PNU,PYIELD,PDEN
      IF(XMODE.EQ.DESIGN) GO TO 110
      READ(5,1000) PT
      IF(PT .LT. PTMIN) WRITE(6, 2001) PTMIN
      PA = PT
      PI = PT ** 3 / 12.0
      PS = PT ** 2 / 6.0
      WRITE(6,2010) PT, PA, PI, PS
      GO TO 130
C
110  READ(5,1000) (PFS(I),I=1,NFS)
      DO 120 I=1,NFS
      IF(PFS(I).GT.0.0) GO TO 120
      PFS(I) = FS(I)
120  CONTINUE
      WRITE(6,2100) (PFS(I),I=1,NFS)
C
      PT = PTMIN
      PA = PTMIN
      PI = PTMIN ** 3 / 12.0
      PS = PTMIN ** 2 / 6.0
      ITER = 0
C
C      INITIALIZE PIPE PROPERTIES ALL AROUND PIPE TO BE THE SAME
C
130  DO 140 N=1,NPMAT
      PIPMAT(1,N) = PE
      PIPMAT(2,N) = PNU
      PIPMAT(3,N) = PA
      PIPMAT(4,N) = PI
140  PIPMAT(5,N) = PDEN*PA
      PDIA = PID + PT
      RETURN
C
C      SECOND ENTRY DETERMINES MAXIMUM STRUCTURAL RESPONSES FOR
C      EVALUATING THE PIPE IN TERMS OF FAILURE CRITERIA. IN THE DESIGN
C      MODE, THE PROGRAM ITERATES TO FIND ADEQUATE SECTION PROPERTIES.
C
C      IEXIT = 0
C
C      CHECK FOR CONVERGENCE OF NONLINEAR SOIL LAW
C
C      IF(NINC.GT.0) GO TO 205
      IEXIT = -1
      RETURN
C
C      EVALUATE CURRENT MAXIMUM STRESSES, DISPLACEMENTS, AND PRESSURES.
C
205  TMAX = 0.0
      BMAX = 0.0
      PAVG = 0.0
      UYMAX = RESULT(4,1)
      UYMIN = RESULT(4,1)
C
      DO 210 N=1,NPFT
      RESULT(10,N) = RESULT(5,N)/PS
      RESULT(11,N) = RESULT(6,N)/PA
      RESULT(12,N) = RESULT(7,N)/PA
      IF(ABS(RESULT(10,N)).GT.BMAX) BMAX = ABS(RESULT(10,N))
      IF(ABS(RESULT(11,N)).GT.TMAX) TMAX = ABS(RESULT(11,N))
      IF(RESULT(4,N).GT.UYMAX) UYMAX = RESULT(4,N)
      IF(RESULT(4,N).LT.UYMIN) UYMIN = RESULT(4,N)
      PAVG = PAVG + RESULT(8,N)
210  CONTINUE
      PAVG = ABS(PAVG/NPFT)
      DISP = UYMAX-UYMIN
      IF (SM .EQ. 0.0) SM = 1.0
      PBUCK = PDIA**3/(36.*PE*SM*(1.-SK))*(1.-PNU**2)
      PCRIT = SQRT(PI/PBUCK)
C
C      DETERMINE THE ACTUAL SAFETY FACTORS
C
      DISPSF = 0.2 * PDIA / DISP
      OUTFSF = PYIELD / (TMAX + BMAX)
      BUCKSF = PCRIT / PAVG
      IF(XMODE.EQ.ANALYS) GO TO 300
C
C      ARE ACTUAL SAFETY FACTORS ACCEPTABLE
C
      IF(IA.LT.NINC) RETURN
      ITER = ITER + 1
      DISPR = PFS(1) / DISPSF
      OUTFR = PFS(2) / OUTFSF
      BUCKR = PFS(3) / BUCKSF
C
C      Revise intrinsic function to generic form. SCM 3-10-90
C
      RMAX = MAX(DISPR, OUTFR, BUCKR)
      IF(ABS(1.0 - RMAX) .LE. 0.05) GO TO 250
      PLAS0010
      PLAS0011
      PLAS0012
      PLAS0013
      PLAS0014
      PLAS0015
      PLAS0016
      PLAS0017
      PLAS0018
      PLAS0019
      PLAS0020
      PLAS0021
      PLAS0022
      PLAS0023
      PLAS0024
      PLAS0025
      PLAS0026
      PLAS0027
      PLAS0028
      PLAS0029
      PLAS0030
      PLAS0031
      PLAS0032
      PLAS0033
      PLAS0034
      PLAS0035
      PLAS0036
      PLAS0037
      PLAS0038
      PLAS0039
      PLAS0040
      PLAS0041
      PLAS0042
      PLAS0043
      PLAS0044
      PLAS0045
      PLAS0046
      PLAS0047
      PLAS0048
      PLAS0049
      PLAS0050
      PLAS0051
      PLAS0052
      PLAS0053
      PLAS0054
      PLAS0055
      PLAS0056
      PLAS0057
      PLAS0058
      PLAS0059
      PLAS0060
      PLAS0061
      PLAS0062
      PLAS0063
      PLAS0064
      PLAS0065
      PLAS0066
      PLAS0067
      PLAS0068
      PLAS0069
      PLAS0070
      PLAS0071
      PLAS0072
      PLAS0073
      PLAS0074
      PLAS0075
      PLAS0076
      PLAS0077
      PLAS0078
      PLAS0079
      PLAS0080
      PLAS0081
      PLAS0082
      PLAS0083
      PLAS0084
      PLAS0085
      PLAS0086
      PLAS0087
      PLAS0088
      PLAS0089
      PLAS0090
      PLAS0091
      PLAS0092
      PLAS0093
      PLAS0094
      PLAS0095
      PLAS0096
      PLAS0097
      PLAS0098
      PLAS0099
      PLAS0100
      PLAS0101
      PLAS0102
      PLAS0103
      PLAS0104
      PLAS0105
      PLAS0106
      PLAS0107
      PLAS0108
      PLAS0109
      PLAS0110
      PLAS0111
      PLAS0112
      PLAS0113
      PLAS0114
      PLAS0115
      PLAS0116
      PLAS0117
      PLAS0118
      PLAS0119
      PLAS0120
      PLAS0121
      PLAS0122
      PLAS0123
      PLAS0124
      PLAS0125
      PLAS0126
      PLAS0127
      PLAS0128
      PLAS0129
      PLAS0130
      PLAS0131
      PLAS0132
      PLAS0133
      PLAS0134
      PLAS0135

```



```

C
C DESIGN... REVISE SECTION PROPERTIES AS REQUIRED
C
PTNEW = PT * RMAX
IF( PTNEW .LT. PTMIN) GO TO 250
PT = PTNEW
PA = PT
PI = PT ** 3 / 12.0
PS = PT ** 2 / 6.0
C
C UPDATE SECTION PROPERTIES AND CHECK FOR CONVERGENCE.
C
IF(ITER.GT.5) WRITE(6,2200) ITER,PT,PA,PI,PS
IF(ITER.GT.5) GO TO 300
IEXIT = -2
GO TO 130
250 WRITE(6,2300) ITER,PT,PA,PI,PS
C
C PRINTOUT STRUCTURAL RESULTS AND EVALUATION.
C
300 CONTINUE
C
C PRINT FINITE ELEMENT FIELD RESPONSES.
C
IF (LEVEL .EQ. 1) GO TO 350
ICOME = 3
CALL PRHERO(IA, ICOME, IEXIT, LEVEL, NINC, NPMAT, NPPT, PDIA, PIPMAT,
1 RESULT, SK, SM)
ICOME = 2
350 CONTINUE
WRITE(6,2800) IA, (RESULT(1,J), RESULT(3,J), RESULT(8,J), RESULT(5,J),
X RESULT(6,J), RESULT(7,J), RESULT(2,J), RESULT(4,J), RESULT(9,J),
X RESULT(10,J), RESULT(11,J), RESULT(12,J), J=1, NPPT)
C
C
PERFAC = (FF * PE * PI) / PDIA ** 2
WRITE(6, 2900) IA, DISPEF, OUTFSF, RUCKSF, PERFAC
IEXIT = 1
IF(IA.LT.NINC) IEXIT= 0
RETURN
C
C FORMATS
C
1000 FORMAT(6F10.0)
2000 FORMAT(/// 10X,35H PIPE PROPERTIES ARE AS FOLLOWS ... //
X 15X,36H AVERAGE PIPE DIAMETER (IN.) ..... , F11.2 //
X 15X,36H YOUNGS MODULUS OF PIPE (PSI)..... , F11.2 //
X 15X,36H POISSONS RATIO OF PIPE (-) ..... , F11.2 //
X 15X,36H YIELD STRESS OF PIPE (PSI)..... , F11.2 //
X 15X,36H DENSITY OF PIPE (PCI)..... , F11.2 //
2001 FORMAT(/ 10X,50H*** WARNING *** PIPE WALL THICKNESS IS LESS THAN
X ,8H ALLOWED , F5.2, 4H IN.)
2010 FORMAT(// 10X, 31H SECTION PROPERTIES OF PIPE ... //
X 15X, 36H THICKNESS OF PIPEWALL (IN)..... , F10.5 //
X 15X, 36H THRUST AREA (IN**2/IN) ..... , F10.5 //
X 15X, 36H MOM. OF INERTIA (IN**4/IN) ..... , F10.5 //
X 15X, 36H SECTION MODULUS (IN**3/IN) ..... , F10.5 //
2100 FORMAT(/// 10X,43H DESIRED SAFETY FACTORS FOR PIPE DESIGN ... ///
X 15X, 36H DEFLECTION COLLAPSE, (0.2*D) ..... , F10.5 //
X 15X, 36H OUTER-FIBER ULTIMATE STRESS..... , F10.5 //
X 15X, 36H ELASTIC BUCKLING ..... , F10.5 / )
2200 FORMAT(1H1,/// 10X,50H DESIGN ITERATIVE PROCEDURE DID NOT CONVERGE
X AFTER ,I3/ 10X,55H ITERATIONS. THE LAST SOLUTION IS PRINTED AND
PLAS0136 XVALUATED, / 10X,37H WHERE THE SECTION PROPERTIES ARE ... //
PLAS0137 X 15X, 36H THICKNESS OF PIPEWALL (IN)..... , F10.5 //
PLAS0138 X 15X,36H THRUST AREA PER UNIT LENGTH ..... , E15.5 //
PLAS0139 X 15X,36H MOM. OF INERTIA PER UNIT LENGTH ... , E15.5 //
PLAS0140 X 15X,36H SECTION MODULUS PER UNIT LENGTH ... , E15.5 //)
PLAS0141 2300 FORMAT(1H1,///10X,19H DESIGN RESULTS AT ,I2,11H ITERATIONS //
PLAS0142 X 15X, 35H *** REQUIRED WALL THICKNESS..... , F10.5 //
PLAS0143 X 15X, 35H *** REQUIRED THRUST AREA ..... , F10.5 //
PLAS0144 X 15X, 35H *** REQUIRED MOM. OR INERTIA ..... , F10.5 //
PLAS0145 X 15X, 35H *** REQUIRED SECTION MODULUS ..... , F10.5 ///)
PLAS0146 2800 FORMAT(1H1,///10X,32H STRUCTURAL RESPONSES OF CULVERT ,
PLAS0147 X 3X, 12H LOAD STEP = ,I2 //
PLAS0148 X 4X,9H X-COORD.,5X,8H X-DISP.,4X,9H N-PRES.,6X,7H MOMENT,6X,
PLAS0149 X 7H THRUST,7X,6H SHEAR /4X,9H Y-COORD.,5X,8H Y-DISP.,5X,8H S-PRES.
PLAS0150 X ,4X,9H M-STRESS,4X,9H T-STRESS,4X,9H S-STRESS ,
PLAS0151 X (//F13.2,5E13.3/F13.2,5E13.3)
PLAS0152 2900 FORMAT(1H1,///10X,34H CALCULATED SAFETY FACTORS AT STEP, I3 ///
PLAS0153 X15X, 50H DISPLACEMENT ..... (DIAMETER*0.2/MAX-DISP.) =,F10.2//PLAS0216
PLAS0154 X15X, 50H OUTER-FIBER STRESS ... (ULTIMATE / MAXIMUM) =,F10.2//PLAS0217
PLAS0155 X15X, 50H ELASTIC BUCKLING ..... (P-CRITICAL/P-AVERAGE) =,F10.2//PLAS0218
PLAS0156 X 10X, 21H PERFORMANCE FACTORS //
PLAS0157 X15X, 50H HANDELING ..... (FF*B*I/D**2) =,F10.2)
PLAS0158 END
PLAS0159 * * * * * PREP0001
PLAS0160 * CANOE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM PREP0002
PLAS0161 * VERSION 1.0, MARCH 1990 PREP0003
PLAS0162 * PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY PREP0004
PLAS0163 * THE FEDERAL HIGHWAY ADMINISTRATION PREP0005
PLAS0164 * * * * * PREP0006
PLAS0165 * PREP0007
PLAS0166 * Add RESULT array and NPPT variable to subroutine parameter list. PREP0008
PLAS0167 * SCM 5-22-89 PREP0009
PLAS0168 * PREP0010
PLAS0169 * SUBROUTINE PREP(FEDATA, PIPMAT, DENSITY, ISIZE, KPUTCH, LCODE, LUDATA, PREP0011
PLAS0170 1 LUPLUT, MAXBO, MAXEL, MAXMAT, MAXNP, NPMAT, NPT, NSMAT, PREP0012
PLAS0171 2 NXMAT, NBPTC, NELEM, PDIA, RESULT, NPPT) PREP0013
PLAS0172 * PREP0014
PLAS0173 * Uncomment this statement in all program modules for double PREP0015
PLAS0174 * precision version. SCM 3-10-90 PREP0016
PLAS0175 * PREP0017
PLAS0176 * IMPLICIT DOUBLE PRECISION (A-H,O-Z) PREP0018
PLAS0177 * PREP0019
PLAS0178 * Add RESULT array to dimension statement. PREP0020
PLAS0179 * SCM 5-22-89 PREP0021
PLAS0180 * PREP0022
PLAS0181 * DIMENSION FEDATA(ISIZE), PIPMAT(5, NPMAT), RESULT(20, NPPT), TITLE(17) PREP0023
PLAS0182 * PREP0024
PLAS0183 * Add common statement for output control. SCM 3-18-89 PREP0025
PLAS0184 * PREP0026
PLAS0185 * COMMON /OUTPUT/ IWRT, IPLOT, MAXIT, JCON PREP0027
PLAS0186 * PREP0028
PLAS0187 * DATA LSTOP, NUMLI /7,200/ PREP0029
PLAS0188 * DATA XPREF, XMOD /4HEREP, 4HMOD / PREP0030
PLAS0189 * PREP0031
PLAS0190 * Add logic for evaluating LCODE parameter. SCM 3-18-89 PREP0032
PLAS0191 * PREP0033
PLAS0192 * IF(LCODE.GT.0) THEN PREP0034
PLAS0193 * LEVEL = LCODE PREP0035
PLAS0194 * ELSE PREP0036
PLAS0195 * LEVEL = IABS(LCODE/100) PREP0037
PLAS0196 * NCAN = IABS(LCODE + LEVEL*100) PREP0038
PLAS0197 * ENDIF PREP0039
PLAS0198 * PREP0040

```

```

C
C** GENERATE THE CANNED MESH
C
KPUTCK = 0
LUPREP = 5
IF (LEVEL.EQ.2) LUPREP = LUDATA
*
* Revise control statements for canned mesh additions.
*   SCM 3-18-89
*
IF (LEVEL.EQ.2) THEN
  IF (NCAN.EQ.0 .OR. NCAN.EQ.1) CALL CAN1 (LUDATA,KPUTCK,PDIA,
X    DENSTY)
  IF (NCAN.EQ.2) CALL CANBOX (LUDATA,KPUTCK,DENSTY)
  IF (NCAN.EQ.3) CALL CANAR1 (LUDATA,KPUTCK,DENSTY)
ENDIF
*
C
C** READ/WRITE MAIN CONTROL CARDS AND SET DEFAULT VALUES
C
READ (LUPREP,5010) WORD,TITLE,WORD2,NINC,MGENPR,NPUTCK,IPL0T,
1    IWRT,NPT,NELEM,NBPTC
IF (WORD.EQ.XPREP) GO TO 50
WRITE (6,6005) WORD
STOP 14
C
50 WRITE (6,6010) TITLE,NINC,MGENPR,NPUTCK,IPL0T,IWRT
1    ,NPT,NELEM,NBPTC
MAXBC = MAXBD
IF (LEVEL.EQ.3) MAXBC = NBPTC
IF (MGENPR.EQ.0) MGENPR = 3
NUMBC = NBPTC
IF (WORD2.EQ.XMOD) NUMBC = MAXBD
C
C-----
C** SET UP STORAGE REQUIREMENTS FOR MESH GENERATOR
C
NDNEL = MAX0(NPT,NELEM)
LOCK = 1
LOCY = LOCK + NPT
LOCNGN = LOCY + NPT
LOCNTP = LOCNGN + 7*NDNEL
LOCKYG = LOCNTP + NPT
C
C** STORAGE FOR GENEL
C
LOCNOD = LOCNGN
LOCLGN = LOCKYG
C
C** STORAGE FOR SAVED
C
LOCNDB = LOCKYG + 4*NDNEL
LOCBIV = LOCNDB + NUMBC
LOCKDE = LOCBIV + 3*NUMBC
ISZTST = LOCKDE + NPT
IF (ISZTST.LE.ISIZE) GO TO 60
WRITE(6,6015) ISZTST, ISIZE
STOP 15
C
C-----
60 CALL GENNOD (FEDATA(LOCK), FEDATA(LOCY), FEDATA(LOCNGN),

```

```

PREP0041 1 FEDATA(LOCNTP), FEDATA(LOCKYG), NPT, MAXNP, KPUTCK, PREP0104
PREP0042 2 MGENPR, LUPREP) PREP0105
PREP0043 C PREP0106
PREP0044 CALL GENEL(FEDATA(LOCNOD), FEDATA(LOCLGN), NPT, NELEM, MAXEL, KPUTCK, PREP0107
PREP0045 1 MGENPR, MBAND, LSTOP, LUPREP) PREP0108
PREP0046 C PREP0109
PREP0047 CALL GENEND(FEDATA(LOCK), FEDATA(LOCY), FEDATA(LOCNOD), PREP0110
PREP0048 1 FEDATA(LOCNDB), FEDATA(LOCBIV), FEDATA(LOCKDE), PREP0111
PREP0049 2 FEDATA(LOCNTP), FEDATA(LOCKYG), NPT, NELEM, NBPTCU, MAXBC, PREP0112
PREP0050 3 KPUTCK, MGENPR, NSMAT, LSTOP, NUMLI, LUPREP, MAXMAT, NXMAT) PREP0113
PREP0051 C PREP0114
PREP0052 C** ROUTINE MODMSH CHANGES THE CAN MESH TO USER SPECS. PREP0115
PREP0053 C PREP0116
PREP0054 IF (WORD2.EQ.XMOD) CALL MODMSH(FEDATA(LOCBIV), FEDATA(LOCNOD), PREP0117
PREP0055 1 FEDATA(LOCNDB), FEDATA(LOCK), FEDATA(LOCY), PREP0118
PREP0056 2 NBPTCU, MAXBC, NELEM, NPT) PREP0119
PREP0057 C PREP0120
PREP0058 C** CHECK NUMBER OF BOUNDARY CONDITIONS PREP0121
PREP0059 C PREP0122
PREP0060 IF (WORD2.EQ.XMOD) NBPTC = NUMBC PREP0123
PREP0061 IF (NBPTCU.LE.NBPTC) GO TO 70 PREP0124
PREP0062 WRITE(6,6018) NBPTCU, NBPTC PREP0125
PREP0063 KPUTCK = KPUTCK + 1 PREP0126
PREP0064 70 NBPTC = NBPTCU PREP0127
PREP0065 C PREP0128
PREP0066 C** SAVE GENERATED DATA PREP0129
PREP0067 C PREP0130
PREP0068 WRITE (6,6020) TITLE, KPUTCK, NSMAT, NPMAT, NXMAT, MBAND PREP0131
PREP0069 C PREP0132
PREP0070 * Add NPPT parameter to subroutine call. SCM 3-10-90 PREP0133
PREP0071 * PREP0134
PREP0072 CALL SAVED(FEDATA(LOCK), FEDATA(LOCY), FEDATA(LOCNOD), FEDATA(LOCNDB) PREP0135
PREP0073 1 , FEDATA(LOCBIV), FEDATA(LOCKDE), FEDATA(LOCKYG), TITLE, PREP0136
PREP0074 2 NPT, NELEM, IPL0T, NBPTC, NPUTCK, KPUTCK, LSTOP, NINC, PREP0137
PREP0075 3 LUDATA, LUPLOT, NSMAT, NPMAT, NPPT) PREP0138
PREP0076 * PREP0139
PREP0077 C PREP0140
PREP0078 C** ESTABLISH RESPONSE OUTPUT AND PLOTTING CONTROLS. PREP0141
PREP0079 C PREP0142
PREP0080 * PREP0143
PREP0081 * Revise ranges of output and plotting controls. SCM 3-01-89 PREP0144
PREP0082 * PREP0145
PREP0083 IF (IPL0T.LT.0 .OR. IPL0T.GT.3) IPL0T = 1 PREP0146
PREP0084 IF (IWRT.LT.0 .OR. IWRT.GT.3) IWRT = 1 PREP0147
PREP0085 * PREP0148
PREP0086 IOUT = -(LUPLOT*100 + IPL0T*10 + IWRT) PREP0149
PREP0087 * PREP0150
PREP0088 * Add RESULT and NPPT parameters to subroutine call. PREP0151
PREP0089 * SCM 3-4-89 PREP0152
PREP0090 * PREP0153
PREP0091 CALL RESOUT (FEDATA(1), FEDATA(1), FEDATA(1), FEDATA(1), FEDATA(1), PREP0154
PREP0092 1 FEDATA(1), FEDATA(1), IOUT, NELEM, NPT, RESULT, NPPT) PREP0155
PREP0093 * PREP0156
PREP0094 C PREP0157
PREP0095 RETURN PREP0158
PREP0096 5010 FORMAT (A4, I4, 17A4, 3X, A4/815) PREP0159
PREP0097 6005 FORMAT ( 63HOFATAL DATA ERROR. THE CONTROL WORD MUST BE *PREP * PREP0160
PREP0098 1, BUT IS *, A5, 1H*) PREP0161
PREP0099 6010 FORMAT ( 43H1* * BEGIN PREP OF FINITE ELEMENT INPUT + */ PREP0162
PREP0100 1 //31H0THE DATA TO BE RUN IS ENTITLED //1H0, 17A4/// PREP0163
PREP0101 2 40H0NUMBER OF CONSTRUCTION INCREMENTS-----, 15/ PREP0164
PREP0102 3 40H0PRINT CONTROL FOR PREP OUTPUT-----, 15/ PREP0165
PREP0103 4 40H0INPUT DATA CHECK-----, 15/ PREP0166

```



```

5      40HO PLOT TAPE GENERATION-----,15/
6      40HO ENTIRE FINITE ELEMENT RESULTS OUTPUT---,15/
7      40HO THE NUMBER OF NODES IS-----,15/
8      40HO THE NUMBER OF ELEMENTS IS-----,15/
9      40HO THE NUMBER OF BOUNDARY CONDITIONS IS---,15/
6015 FORMAT (1H1,40H *ERROR DETECTED IN ROUTINE PREP * ,//
1      1H ,40H *BLANK COMMON IS TOO SMALL TO SOLVE * ,//
2      1H ,40H *CURRENT PROBLEM * ,//
3      1H ,23H *PRESENT SIZE (WORDS) ,15,10X, 1H* ,//
4      1H ,23H *REQUIRED SIZE (WORDS) ,15, 10X, 1H* ,//
5      1H ,40H *PLEASE INCREASE SIZE OF BLANK * ,//
6      1H ,40H *COMMON IN PREHERO * ,//
7      1H ,40H *FOR DETAILS SEE SYSTEM MANUAL * ,//
8      1H ,40H *CANDE WILL NOW STOP * )
6018 FORMAT (40H * * * FATAL DATA ERROR * * * /
1      1 40H * * * NUMBER OF BOUNDARY CONDITIONS * ,15/
2      2 40H * * * MAXIMUM NUMBER ALLOWED * * * ,15/
3      3 40H * * * (SPECIFIED ON CARD C2. COL 36 -40 )
6020 FORMAT ( 47H1* * * * * FOR MESH GENERATION OF * * * * */
1      1 // 1H0,17A4 ///
2      2 40HO THE NUMBER OF DATA ERRORS IS-----,15/
3      3 40HO THE NUMBER OF SOIL MATERIALS IS-----,15/
4      4 40HO THE NUMBER OF PIPE MATERIALS IS-----,15/
5      5 40HO THE NUMBER OF INTERFACE MATERIALS IS---,15/
6      6 40HO BAND WIDTH ESTIMATE (MAX)-----,15 )
END
* * * * *
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
* VERSION 1.0, MARCH 1990
* PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
* THE FEDERAL HIGHWAY ADMINISTRATION
* * * * *
* Change LEVEL to LCODE parameter in subroutine statement for
* passing of output parameters. SCM 3-18-89
*
* SUBROUTINE PRHERO (IA, ICOME, IEXIT, LCODE, NINC, NPMAT, NPPT, PDIA,
1 PIPMAT, RESULT, SK, SM)
*
* C
* C** SWITCHING ROUTINE FOR LEVEL 2 AND 2 SOLUTIONS
* C
* Uncomment this statement in all program modules for double
* precision version. SCM 3-10-90
*
* IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
* DIMENSION PIPMAT(5,NPMAT),RESULT(20,NPPT)
*
* Revise FEDATA and ISIZE parameters from 6000 to 8000.
* SCM 5-30-89
*
* CC
* C
* CC Revise FEDATA and ISIZE parameters from 8000 to 10000. by KS
*
* COMMON FEDATA(10000)
* DATA ISIZE/10000/
*
* DATA LUPLOT,LUDATA /10,11/
* DATA MAXNP,MAXEL,MAXBC,MAXMAT /200,100,400,10/
*
* Add logic for evaluating LCODE parameter. SCM 3-18-89

```

```

PREP0167 IF(LCODE.GT.0) THEN
PREP0168 LEVEL = LCODE
PREP0169 ELSE
PREP0170 LEVEL = IABS(LCODE/100)
PREP0171 NCAN = IABS(LCODE + LEVEL*100)
PREP0172 ENDIF
PREP0173 *
PREP0174 C
PREP0175 GO TO (100,110,140),ICOME
PREP0176 C
PREP0177 100 CONTINUE
PREP0178 C
PREP0179 C** PREPARE INPUT DATA FOR FINITE ELEMENT METHOD
PREP0180 C
PREP0181 CALL ESTAB
PREP0182 REWIND LUDATA
PREP0183 REWIND LUPLOT
PREP0184 *
PREP0185 * Add RESULT array and NPPT variable to subroutine parameter list.
PREP0186 * SCM 5-22-89
PREP0187 *
PREP0188 CALL PREP (FEDATA, PIPMAT, DENSTY, ISIZE, KPITCH, LCODE, LUDATA,
1 LUPLOT, MAXBC, MAXEL, MAXMAT, MAXNE, NPMAT, NPT, NSMAT,
2 NXMAT, NBPTC, NELEM, PDIA, RESULT, NPPT)
PREP0191 *
PREP0192 C
PREP0193 C** DEFINE BLANK COMMON STORAGE FOR HEROIC
PREP0194 C
PREP0195 LOCX = 1
PREP0196 LOCY = LOCX + NPT
PREP0197 LOCNOB = LOCY + NPT
PREP0198 LOCNO = LOCNOB + NELEM * 4
PREP0199 LOCNOB = LOCNO + NELEM
PREP0200 LOCBI = LOCNOB + NBPTC
PREP0201 LOCNO = LOCBI + NBPTC * 3
PREP0202 LOCY = LOCNO + NPT + 1
PREP0203 LOCV = LOCY + NPT
PREP0204 LOCST = LOCV + NPT
PREP0205 LOCSTP = LOCST + NELEM*6
PREP0206 LOCSTH = LOCSTP + NPMAT*2 + 2
PREP0207 ISZTST = LOCSTH + NELEM * 6
PREP0208 IF (ISZTST .LE. ISIZE) GO TO 105
PREP0209 *
PREP0210 * Reverse order of ISZTST and ISIZE in WRITE statement below.
PREP0211 * SCM 10-23-89
PREP0212 *
PREP0213 WRITE(6,6020) ISIZE, ISZTST
PREP0214 *
PREP0215 STOP 11
PREP0216 C
PREP0217 C** ZERO BLANK COMMON
PREP0218 C
PREP0219 105 DO 120 NZERO = 1, ISIZE
PREP0220 120 FEDATA(NZERO) = 0.0
PREP0221 110 CONTINUE
PREP0222 C
PREP0223 C** HEROIC PERFORMS 2 AND 3 SOLUTIONS (THE FINITE ELEMENT METHOD)
PREP0224 C
PREP0225 CALL HEROIC (FEDATA(LOCBI), DENSTY, IA, ICOME, IEXIT, LEVEL, NBPTC,
1 FEDATA(LOCNO), NELEM, NINC, FEDATA(LOCNOB),
2 FEDATA(LOCNOB), NPMAT, NPPT, NPT, FEDATA(LOCNQ), NSMAT,
3 NXMAT, PIPMAT, RESULT, SK, SM, FEDATA(LOCST),
4 FEDATA(LOCSTH), FEDATA(LOCSTP), FEDATA(LOCY),

```

```

PRHR0035
PRHR0036
PRHR0037
PRHR0038
PRHR0039
PRHR0040
PRHR0041
PRHR0042
PRHR0043
PRHR0044
PRHR0045
PRHR0046
PRHR0047
PRHR0048
PRHR0049
PRHR0050
PRHR0051
PRHR0052
PRHR0053
PRHR0054
PRHR0055
PRHR0056
PRHR0057
PRHR0058
PRHR0059
PRHR0060
PRHR0061
PRHR0062
PRHR0063
PRHR0064
PRHR0065
PRHR0066
PRHR0067
PRHR0068
PRHR0069
PRHR0070
PRHR0071
PRHR0072
PRHR0073
PRHR0074
PRHR0075
PRHR0076
PRHR0077
PRHR0078
PRHR0079
PRHR0080
PRHR0081
PRHR0082
PRHR0083
PRHR0084
PRHR0085
PRHR0086
PRHR0087
PRHR0088
PRHR0089
PRHR0090
PRHR0091
PRHR0092
PRHR0093
PRHR0094
PRHR0095
PRHR0096
PRHR0097

```

```

5      FEDATA(LOCV),FEDATA(LOCK),FEDATA(LOCY))
      RETURN
140 CONTINUE
C
C** RESOUT PRINTS OUT FINITE ELEMENT RESPONSES
C
*
* Add RESULT and NPPT parameters to subroutine call
*   SCM 3-01-89
*
      CALL RESOUT(FEDATA(LOCMNO),FEDATA(LOCNOD),FEDATA(LOCST),
1      FEDATA(LOCV),FEDATA(LOCV),FEDATA(LOCK),FEDATA(LOCY),
2      IA,NELEM,NPT,RESULT,NPPT)
*
6020 FORMAT (1H0,40H * * * * * ,//
1      1H ,40H * * * * * ERROR * * * * * ,//
2      1H ,40H *BLANK COMMON NEEDS TO BE INCREASED * * ,//
3      1H ,21H *BLANK COMMON IS NOW,I5,SHWORDS,
4      1H ,32H * REQUIRED SIZE OF BLANK COMMON,I5,SHWORDS,//
5      1H ,47H *CHANGE BLANK COMMON SIZE IN SUBROUTINE PRHERO)
      RETURN
      END
* * * * *
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
*   VERSION 1.0, MARCH 1990
*   PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
*   THE FEDERAL HIGHWAY ADMINISTRATION
* * * * *
SUBROUTINE PRINC (XM,XML)
C
C THIS SUBROUTINE FINDS THE PRINCIPAL COMPONENTS OF A SECOND RANK
C TENSOR IN TWO-DIMENSIONS
C
*
* Uncomment this statement in all program modules for double
*   precision version. SCM 3-10-90
*
* IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
      DIMENSION XM(4), XML(4)
      XML(3)=0.0
      DU1=(XM(1)+XM(2))/2.0
      DU2=SQRT(((XM(1)-XM(2))/2.0)**2+XM(3)**2)
      XML(1)=DU1+DU2
      XML(2)=DU1-DU2
      IF (XM(1) .LE. XM(2)) GO TO 705
      DU3=XML(1)-XM(2)
      IF(DU3.EQ.0.0) GO TO 718
      DU2=XM(3)/DU3
      DU1=SQRT(1.0/(1.0+DU2**2))
      DU2=DU2*DU1
      GO TO 716
705 DU3=XML(1)-XM(1)
      IF(DU3.EQ.0.0) GO TO 718
      DU1=XM(3)/DU3
      DU2=SQRT(1.0/(1.0+DU1**2))
      DU1=DU1*DU2
716 XML(3)=ATAN2(DU2,DU1)/0.01745329
718 RETURN
      END
* * * * *
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
*   VERSION 1.0, MARCH 1990
*   PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY

```

```

PRHR0098 * THE FEDERAL HIGHWAY ADMINISTRATION REDM00005
PRHR0099 * * * * * REDM00006
PRHR0100 SUBROUTINE READM( NSMAT,MAXMAT,DENSTY,LEVEL) REDM00007
PRHR0101 * REDM00008
PRHR0102 * Uncomment this statement in all program modules for double REDM00009
PRHR0103 * precision version. SCM 3-10-90 REDM00010
PRHR0104 * REDM00011
PRHR0105 * IMPLICIT DOUBLE PRECISION (A-H,O-Z) REDM00012
PRHR0106 * REDM00013
PRHR0107 COMMON /MATERL/ CP(3,3),DEN(10),E(10,10),GNU(10,10),H(10,10), REDM00014
PRHR0108 1 ITYPE(10),NON(10) REDM00015
PRHR0109 * REDM00016
PRHR0110 * Expand array dimension statements from 30 to 99. REDM00017
PRHR0111 * SCM 5-22-89 REDM00018
PRHR0112 * REDM00019
PRHR0113 COMMON/XFACE/ NXMAT,ANGLE(99),FCOEF(99),TENSIL(99),ITMAX REDM00020
PRHR0114 * REDM00021
PRHR0115 * REDM00022
PRHR0116 * Add IBULK(10) to COMMON /HYPER/. MGK 9-29-88 REDM00023
PRHR0117 * REDM00024
PRHR0118 COMMON /HYPER/ C(10),PHIO(10),DPHI(10),ZK(10),ZN(10),RF(10), REDM00025
PRHR0119 $ VT(10),BK(10),BM(10),RATIO(10),IBULK(10) REDM00026
PRIN0001 * REDM00027
PRIN0002 DIMENSION MATNAM(5) REDM00028
PRIN0003 * REDM00029
PRIN0004 * Add dimension statement for Selig parameters. SCM 3-3-89 REDM00030
PRIN0005 * REDM00031
PRIN0006 DIMENSION ISELIG(15),SXC(15),SXPHIO(15),SXDPHI(15),SXZK(15), REDM00032
PRIN0007 * SXZN(15),SXRF(15),SXBI(15),SXEU(15) REDM00033
PRIN0008 * REDM00034
PRIN0009 DIMENSION IDUNC(12),XC(12),XPHIO(12),XDPHI(12),XZK(12), REDM00035
PRIN0010 * XZN(12),XRF(12),XKB(12),XBM(12) REDM00036
PRIN0011 * DIMENSION IOVER(6),XE(6,8),XH(8) REDM00037
PRIN0012 * DATA ITRIA, IGRAN, ICOHE, IMIX /4HTRIA,4HGRAN,4HCOHE,4HMIXE / REDM00038
PRIN0013 * DATA XNUMIN,XNUMAX,XQ ,VOIDRG,VOIDRM, VOIDRC,SATM,SATC,PIM,PIC REDM00039
PRIN0014 * X / 0.1, 0.49, .26, 0.6, 0.5, 1.0, 0.5, .90,.05,0.20 / REDM00040
PRIN0015 * DATA IBLANK, L1H /1H , 1HL/ REDM00041
PRIN0016 * REDM00042
PRIN0017 * Add data for Selig model parameters. SCM 3-3-89 REDM00043
PRIN0018 * REDM00044
PRIN0019 * DATA PI /3.14159265/ REDM00045
PRIN0020 * DATA ISELIG/4HSW95,4HSW90,4HSW85,4HSW80,4HSW61, REDM00046
PRIN0021 * 4HML95,4HML90,4HML85,4HML80,4HML49, REDM00047
PRIN0022 * 4HCL95,4HCL90,4HCL85,4HCL80,4HCL45/ REDM00048
PRIN0023 * DATA SXC/0.0,0.0,0.0,0.0,0.0,0.0,4.0,3.5,3.0,2.5,0.0, REDM00049
PRIN0024 * 9.0,7.0,6.0,5.0,0.0/ REDM00050
PRIN0025 * DATA SXPHIO/48.0,42.0,38.0,36.0,29.0,34.0,32.0,30.0, REDM00051
PRIN0026 * 28.0,23.0,15.0,17.0,18.0,19.0,23.0/ REDM00052
PRIN0027 * DATA SXDPHI/8.0,4.0,2.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0, REDM00053
PRIN0028 * 4.0,7.0,8.0,8.5,11.0/ REDM00054
PRIN0029 * DATA SXZK/950.0,640.0,450.0,320.0,54.0,440.0,200.0,110.0, REDM00055
PRIN0030 * 75.0,16.0,120.0,75.0,50.0,35.0,16.0/ REDM00056
PRIN0031 * DATA SXZN/0.60,0.43,0.35,0.35,0.85,0.40,0.26,0.25, REDM00057
PRIN0032 * 0.23,0.95,0.45,0.54,0.60,0.66,0.95/ REDM00058
PRIN0033 * DATA SXRF/0.70,0.75,0.80,0.83,0.90,0.95,0.89,0.85, REDM00059
PRIN0034 * 0.80,0.55,1.00,0.94,0.90,0.87,0.75/ REDM00060
PRIN0035 * DATA SXBI/187.0,102.0,31.8,15.3,4.3,120.8,46.0,23.8, REDM00061
PRIN0036 * 12.8,3.3,53.0,25.5,13.0,8.8,1.8/ REDM00062
PRIN0037 * DATA SXEU/0.014,0.036,0.057,0.078,0.163,0.043,0.071,0.100, REDM00063
REDM0001 * 0.134,0.305,0.092,0.121,0.149,0.178,0.391/ REDM00064
REDM0002 * REDM00065
REDM0003 * DATA IDUNC/4HCA10,4HCA95,4HCA90,4HSM10,4HSM90, REDM00066
REDM0004 * 4HSM85,4HSC10,4HSC90,4HSC85,4HCL10, REDM00067

```

```

*      4HCL90,4HCL85/
DATA XC/0.0,0.0,0.0,0.0,0.0,0.0,3.472,2.083,
*      1.388,2.777,1.388,0.694/
*
*      Change XPHIO and XDPHI parameters from radians to degrees.
*      SCM 3-03-89
*
DATA XPHIO/42.0,36.0,33.0,36.0,32.0,30.0,
*      33.0,33.0,33.0,30.0,30.0,30.0/
DATA XDPHI/9.0,5.0,3.0,8.0,4.0,2.0,
*      0.0,0.0,0.0,0.0,0.0,0.0/
*
DATA XZK/600.0,300.0,200.0,600.0,300.0,150.0,
*      400.0,150.0,100.0,150.0,90.0,60.0/
DATA XZN/0.4,0.4,0.4,0.25,0.25,0.25,0.6,0.6,0.6,
*      0.45,0.45,0.45/
DATA XRF/0.7,0.7,0.7,0.7,0.7,0.7,0.7,0.7,0.7,0.7,
*      0.7,0.7/
DATA XBK/175.0,75.0,50.0,450.0,250.0,150.0,200.0,
*      75.0,50.0,140.0,80.0,50.0/
DATA XBM/0.2,0.2,0.2,0.0,0.0,0.0,0.5,0.5,0.5,0.2,
*      0.2,0.2/
DATA IOVER/4HGGOO,4HGFAL,4HMGOO,4HMFAL,4HCGOO,4HCFAL/
DATA XE/1100.0,550.0,600.0,400.0,250.0,150.0,
*      1300.0,750.0,850.0,550.0,325.0,200.0,
*      1500.0,850.0,1000.0,600.0,375.0,225.0,
*      1650.0,1000.0,1100.0,700.0,375.0,250.0,
*      1800.0,1100.0,1200.0,750.0,400.0,250.0,
*      1900.0,1150.0,1250.0,800.0,400.0,250.0,
*      2100.0,1300.0,1350.0,900.0,400.0,250.0,
*      2250.0,1400.0,1450.0,900.0,400.0,250.0/
DATA XH/3.4, 6.9,10.3,13.8,17.2,20.7,27.6,34.5/

MATERIAL PROPERTIES FOR SOIL-SYSTEM (CONTINUUM ELEMENTS)

WRITE(6,2000)
KPUTCK = 0
MAT = 0
MATX = 0
LIMIT = IBLANK
2 IF (LIMIT.EQ.LIH) GO TO 500
READ(5,5005) LIMIT,I,ITYP,DENX,(MATNAM(K),K=1,5)
IF(ITYP.EQ.6) GO TO 200
ITYP(I) = ITP
DEN(I) = DENX
MAT = MAT + 1
NON(I) = 0
IF(I.LT.1 .OR. I.GT.MAXMAT) GO TO 30
IF(ITYP.LT.1 .OR. ITP.GT.5) GO TO 30
GO TO 40
30 WRITE(6,6070) ITP,I
KPUTCK = KPUTCK + 1
GO TO 2
C
40 WRITE(6,907) I,(MATNAM(K),K=1,5),DEN(I)
DEN(I) = DEN(I)/1728.
GO TO (50,60,80,90,100),ITYP

      LINEAR ISOTROPIC PROPERTIES
50 READ(5,5010) E(I,1), GNU(I,1)
SMH = E(I,1)*(1.0-GNU(I,1))/(1.0+GNU(I,1))*(1.0-2.0*GNU(I,1))
SKH = GNU(I,1)/(1.0-GNU(I,1))

```

```

REDM0068      WRITE(6,904) E(I,1),GNU(I,1),SMH,SKH
REDM0069      CALL CONV(T(E(I,1),GNU(I,1))
REDM0070      GO TO 65
REDM0071      C
REDM0072      C      ANISOTROPIC PROPERTIES
REDM0073      C
REDM0074      60 READ(5,5010) D2,D3,D4,D5,D6
REDM0075      WRITE(6,905) D2,D3,D4,D5,D6
REDM0076      CALL ANISP(D2,D3,D4,D5,D6)
REDM0077      C
REDM0078      65 L=0
REDM0079      DO 70 J=1,3
REDM0080      DO 70 K=J,3
REDM0081      L=L+1
REDM0082      70 H(I,L)=CP(J,K)
REDM0083      GO TO 2
REDM0084      C
REDM0085      C      DUNCAN'S HYPERBOLIC SOIL MODEL
REDM0086      C
REDM0087      *
REDM0088      *      Add Selig IBULK(I) variable to read statement.
REDM0089      *      MGK 9-29-1988
REDM0090      *
REDM0091      80 READ(5,300) NON(I),RATIO(I),IBULK(I)
REDM0092      *
REDM0093      *      Revise soil model iteration limit. SCM
REDM0094      *
REDM0095      IF(NON(I).EQ.0.0)NON(I) = 20
REDM0096      *
REDM0097      IF(RATIO(I).EQ.0.0) RATIO(I) = 0.5
REDM0098      *
REDM0099      *      Add control statements to print out heading for
REDM0100      *      appropriate model (Duncan or Selig).
REDM0101      *      SCM 10-28-88
REDM0102      *
REDM0103      IF (IBULK(I).EQ.0) THEN
REDM0104      WRITE(6,305) NON(I),RATIO(I)
REDM0105      ELSE
REDM0106      WRITE(6,306) NON(I),RATIO(I)
REDM0107      ENDIF
REDM0108      *
REDM0109      C
REDM0110      C      DETERMINE IF MATNAM IS REQUESTING STANDARD PROPERTIES,
REDM0111      C      OTHERWISE ASSUME MATNAM=USER
REDM0112      C
REDM0113      J=0
REDM0114      *
REDM0115      *      Add logic to select Selig standard parameters. SCM 3-3-89
REDM0116      *
REDM0117      IF (IBULK(I).EQ.0) THEN
REDM0118      *
REDM0119      DO 81 N=1,12
REDM0120      IF(MATNAM(1).EQ.IDUNC(N)) J=N
REDM0121      81 CONTINUE
REDM0122      *
REDM0123      ELSE
REDM0124      DO 82 N=1,15
REDM0125      IF(MATNAM(1).EQ.ISELIG(N)) JS=N
REDM0126      82 CONTINUE
REDM0127      END IF
REDM0128      IF(J.EQ.0 .AND. JS.EQ.0) GO TO 85
REDM0129      IF(IBULK(I).NE.0) GO TO 83
REDM0130      *

```

```

REDM0131
REDM0132
REDM0133
REDM0134
REDM0135
REDM0136
REDM0137
REDM0138
REDM0139
REDM0140
REDM0141
REDM0142
REDM0143
REDM0144
REDM0145
REDM0146
REDM0147
REDM0148
REDM0149
REDM0150
REDM0151
REDM0152
REDM0153
REDM0154
REDM0155
REDM0156
REDM0157
REDM0158
REDM0159
REDM0160
REDM0161
REDM0162
REDM0163
REDM0164
REDM0165
REDM0166
REDM0167
REDM0168
REDM0169
REDM0170
REDM0171
REDM0172
REDM0173
REDM0174
REDM0175
REDM0176
REDM0177
REDM0178
REDM0179
REDM0180
REDM0181
REDM0182
REDM0183
REDM0184
REDM0185
REDM0186
REDM0187
REDM0188
REDM0189
REDM0190
REDM0191
REDM0192
REDM0193

```

```

C
C ESTABLISH STANDARD PARAMETER VALUES
C
C C(I)=XC(J)
*
* Convert from degrees to radians. SCM 3-3-89
*
PHIO(I)=XPHIO(J)*PI/180
DPHI(I)=XDPHI(J)*PI/180
*
ZK(I)=XZK(J)
ZN(I)=XZN(J)
RF(I)=XRF(J)
BK(I)=XBK(J)
BM(I)=XBM(J)
VT(I)=0.0
GO TO 86
*
* Establish standard Selig parameter values. SCM 3-3-89
*
83 C(I)=SXC(JS)
PHIO(I)=SXPHIO(JS)*PI/180
DPHI(I)=SXDPHI(JS)*PI/180
ZK(I)=SXZK(JS)
ZN(I)=SXZN(JS)
RF(I)=SXRF(JS)
BK(I)=SXBK(JS)
BM(I)=SXBK(JS)
VT(I)=0.0
GO TO 86
*
C
C USER DEFINED DUNCAN MODEL
C
85 READ(5,330)H1,H2A,H2B,H3,H4,H5
C(I) = H1
*
* Convert from degrees to radians. SCM 3-3-89
*
PHIO(I) = H2A*PI/180
DPHI(I) = H2B*PI/180
*
ZK(I) = H3
ZN(I) = H4
RF(I) = H5
READ(5,340) H7,H8,H9
BK(I)=H7
BM(I)=H8
VT(I)=H9
C
C WRITE OUT INPUT/DEFAULT DATA FOR SOIL CLASSES
C
C
*
* Convert PHIO and DPHI on output from radians to degrees.
* SCM 3-3-89
*
86 WRITE(6,310) (MATNAM(K),K=1,5),C(I),PHIO(I)*180/PI,
DPHI(I)*180/PI,ZK(I),ZN(I),RF(I)
*
* Revise write statements for hyperbolic bulk modulus
* option. MGK 9-29-88
*
IF (VT(I) .EQ. 0.0 .AND. IBULK(I) .EQ. 0) WRITE(6,320) BK(I),BM(I)
REDM0194 IF (VT(I) .EQ. 0.0 .AND. IBULK(I) .NE. 0) WRITE(6,323) BK(I),BM(I)
REDM0195 *
REDM0196 IF (VT(I) .GT. 0.0) WRITE(6,325) VT(I)
REDM0197 GO TO 2
REDM0198 C
REDM0199 OVERBURDEN PRESSURE DEPENDENT PROPERTIES
REDM0200 (FIRST DETERMINE IF STANDARD PROPERTIES ARE
REDM0201 REQUESTED)
REDM0202 C
REDM0203 90 J=0
REDM0204 DO 91 N=1,6
REDM0205 IF (MATNAM(1) .EQ. IOVER(N)) J=N
REDM0206 91 CONTINUE
REDM0207 IF (J .EQ. 0) GO TO 93
REDM0208 C
REDM0209 ESTABLISH STANDARD PROPERTY TABLES
REDM0210 C
REDM0211 NOMI=8
REDM0212 DO 92 N=1,NOMI
REDM0213 H(I,N)=XH(N)
REDM0214 E(I,N)=XE(J,N)
REDM0215 92 GNU(I,N)=0.333
REDM0216 GO TO 98
REDM0217 C
REDM0218 USER DEFINED OVERBURDEN TABLE
REDM0219 C
REDM0220 93 N = 0
REDM0221 94 N = N+1
REDM0222 READ(5,5010) D1,D2,D3
REDM0223 IF (D1+D2 .EQ. 0.0) GO TO 95
REDM0224 H(I,N) = D1
REDM0225 E(I,N) = D2
REDM0226 GNU(I,N) = D3
REDM0227 GO TO 94
REDM0228 95 NOMI = N-1
REDM0229 IF (NOMI .GE. 2 .AND. NOMI .LE. 10) GO TO 98
REDM0230 WRITE(6,6060) NOMI
REDM0231 KPUTCK = KPUTCK + 1
REDM0232 98 WRITE(6,923)
REDM0233 WRITE(6,916) (H(I,N),E(I,N),GNU(I,N),N=1,NOMI)
REDM0234 NON(I) = NOMI
REDM0235 GO TO 2
REDM0236 C
REDM0237 C *** PROCESS EXTENDED-HARDIN MODEL, DEPENDENT ON SOIL CLASS...
REDM0238 C GRANULAR, MIXED, COHESIVE, OR USER DEFINED FROM TRIAXIAL TEST.
REDM0239 C
REDM0240 100 READ(5,5010) D0, D1, D2, D3, D4, D5, NON(I)
REDM0241 C
REDM0242 C CHECK ON PERMISSIBLE VALUES OF MAX AND MIN POISSON RATIOS FOR
REDM0243 C EXTENDED HARDIN MODEL. THIS APPLYS TO ALL SOIL CLASSES AND USER.
REDM0244 C
REDM0245 IF (NON(I) .LE. 0) NON(I) = 5
REDM0246 IF (D0 .LE. 0.0 .OR. D0 .GE. 0.5) D0 = XNUMIN
REDM0247 IF (D1 .LE. 0.0 .OR. D1 .GE. 0.5) D1 = XNUMAX
REDM0248 IF (D2 .LE. 0.0) D2 = XQ
REDM0249 H(I,4) = D0
REDM0250 H(I,5) = D1
REDM0251 H(I,6) = D2
REDM0252 C
REDM0253 IF (MATNAM(1) .EQ. ITRIA) GO TO 150
REDM0254 C
REDM0255 SET DEFAULT VALUES FOR GRANULAR SOIL CLASS
REDM0256 C

```

REDM0257
 REDM0258
 REDM0259
 REDM0260
 REDM0261
 REDM0262
 REDM0263
 REDM0264
 REDM0265
 REDM0266
 REDM0267
 REDM0268
 REDM0269
 REDM0270
 REDM0271
 REDM0272
 REDM0273
 REDM0274
 REDM0275
 REDM0276
 REDM0277
 REDM0278
 REDM0279
 REDM0280
 REDM0281
 REDM0282
 REDM0283
 REDM0284
 REDM0285
 REDM0286
 REDM0287
 REDM0288
 REDM0289
 REDM0290
 REDM0291
 REDM0292
 REDM0293
 REDM0294
 REDM0295
 REDM0296
 REDM0297
 REDM0298
 REDM0299
 REDM0300
 REDM0301
 REDM0302
 REDM0303
 REDM0304
 REDM0305
 REDM0306
 REDM0307
 REDM0308
 REDM0309
 REDM0310
 REDM0311
 REDM0312
 REDM0313
 REDM0314
 REDM0315
 REDM0316
 REDM0317
 REDM0318
 REDM0319

```
C
C  SET DEFAULT VALUES FOR COHESIVE SOIL CLASS
```

```

110 IF(MATNAM(1).NE. ICORE) GO TO 120
    IF(D3.LT.0.3 .OR. D3 .GT. 3.0) D3 = VOIDRC
    IF(D4.LT.0.0 .OR. D4 .GT. 1.0) D4 = SATC
    IF(D5.LE.0.0 .OR.D5 .GT. 1.0) D5 = PIC
    A = 1.12*(1.+2.*D4)
    GO TO 130

```

```

C
C      SET DEFAULT VALUES FOR MIXED SOIL CLASS. (THIS IS DEFAULT CLASS)

```

```

120 MATNAM(1) = IMIX
      IF(D3.LE.0.1 .OR. D3.GT.1.0) D3 = VOIDRM
      IF(D4.LE.0.0 .OR. D4.GT.1.0) D4 = SATM
      IF(D5.LE.0.0 .OR. D5.GT.0.5) D5 = PIM
      A=2.54*(1.+2.*D4)

```

```
C WRITE OUT INPUT/DEFAULT DATA FOR SOIL CLASSES.
```

```
130 WRITE(6,6040) MATNAM(1),MATNAM(2),D0,D1,D2,D3,D4,D5,NON(I)
```

C COMPUTE MATERIAL CONSTANTS IN HARDINS EQUATIONS. (NOTE THE VARIABLE REDM0348
C A WAS BASED ON HARDIN EQUATIONS WITH T = 10 AND N=1. REDM0349

```
F = (2.973-D3)**2/(1.+D3)
R = (1100.-600.*D4)
H(I,1) = 1230.*F
H(I,2) = F*F*R*R/(0.6 - 0.25*D5**0.6)
H(I,3) = A
```

GO TO 2

C USER DEFINED PROPERTIES FOR HARDINS RELATIONS FROM TRIAXIAL TESTS

```
150  WRITE(6,6050) D0,D1,D2,D3,D4,D5,NON(I)
      IF (D3*D4 .GT. 0.0) GO TO 160
      WRITE(6,6055)
```

KPUTCK = KPUTCK + 1

160 $H(I, 1) = D3$

$$H(1,2) = 04$$

H(I,3) = D5

GO TO 2

C INPUT PROPERTIES FOR INTERFACE ELEMENTS

```
200 IF(MATX.EQ.0) WRITE(6,950)
    IF(I.LE.0) I=1
```

- * Revise maximum interface materials from 30 to 99.
- * SCM 5-22-89

```
IF(I.GT.99) GO TO 30
```

```

MATX = MATX+1
READ(5,5010) ANGLE(I),FCOEF(I),TENSIL(I)

```

```

REDM0320 *      Revise default value for TENSIL.  SCM 4-25-89
REDM0321 *
REDM0322 *      IF(TENSIL(I).EQ.0.0) TENSIL(I) = 1.0
REDM0323 *
REDM0324 *      WRITE(6,955) I,ANGLE(I),FCOEF(I),TENSIL(I)
REDM0325 *

```



```

6060 FORMAT(/ 79H FATAL DATA ERROR. NUMBER OF NONLINEAR DATA POINTS NOT REDM0446
      X BETWEEN 2 AND 10, NOMI = ,I5) REDM0447
6070 FORMAT(/73H FATAL DATA ERROR. MATERIAL NUMBER AN/OR IYPE ARE OUT-OF-REDM0448
      XF-BOUNDS. I,IYPE=,2I5) REDM0449
6080 FORMAT(/46H WARNING MORE MATERIALS WERE INPUT THAN DEFINED,2I5) REDM0450
6090 FORMAT(/49H FATAL DATA ERROR NOT ENOUGH MATERIALS INPUT,NEED,I5) REDM0451
6095 FORMAT(/40H WARNING** DENSITY OF FILL ABOVE MESH, ,F10.5, REDM0452
      X /44H IS NOT THE SAME AS FILL DENSITY WITHIN MESH, ,F10.5 ) REDM0453
7000 FORMAT(/48H ***FATAL DATA ERRORS IN MATERIAL INPUT, ERRORS=,I3) REDM0454
907 FORMAT(1H0//5X,23H PROPERTIES FOR MATERIAL,I3,3X,8H*****5A4, REDM0455
      1 //,10X, 9HDENSITY = ,E15.5) REDM0456
905 FORMAT(1H0, 9X,5HC-11=, E11.4/, 10X, 5HC-12=, E11.4, /, REDM0457
      1 10X,5HC-22=, E11.4/, 10X, 5HC-33=, E11.4, /, REDM0458
      2 10X,3HTH=, E11.4) REDM0459
904 FORMAT(1H0,10X,15HYOUNGS MODULUS=, E11.4, /, 11X, REDM0460
      1 15HPOISSONS RATIO=, E11.4, /, 11X, REDM0461
      2 15HCONFINED MOD.=, E11.4, /, 11X, REDM0462
      3 15HLATERAL COEFF.=, E11.4) REDM0463
916 FORMAT(1H ,9X,1E3E20.2) REDM0464
923 FORMAT(1H0,9X,31HELASTIC PROPERTIES - OVERBURDEN//20X, REDM0465
      1 10H0VERBURDEN,10X,14HYOUNGS MODULUS,6X,14HPOISSONS RATIO) REDM0466
950 FORMAT(1H0//10X,44H INTERFACE ELEMENT MATERIAL-GROUP PROPERTIES //REDM0467
      X 12X,8HMAT. NO.,3X,12HNORMAL-ANGLE,2X,13HCOEF-FRICTION,2X, REDM0468
      X 15HTENSILE-RUPTURE //) REDM0469
955 FORMAT(15X,I5,3(3X,F12.5)) REDM0470
* REDM0471
* Revise format statement to read IBULK REDM0472
* MGK 9-29-88 REDM0473
* REDM0474
300 FORMAT(15,F10.5,I5) REDM0475
* REDM0476
305 FORMAT(/5X,31HCONTROLS FOR DUNCAN SOIL MODEL /// REDM0477
      1 10X,30H MAXIMUM ITERATIONS.....,I8// REDM0478
      2 10X,30H ENTERING ELEMENT RATIO.....,F12.4//) REDM0479
* REDM0480
* Add format statement for Selig version of model. REDM0481
* MGK 9-29-88 REDM0482
* REDM0483
306 FORMAT(/5X,30HCONTROLS FOR DUNCAN SUBROUTINE,/, REDM0484
      1 7X,48HWITH SELIG HYPERBOLIC BULK MODULUS FORMULATION /// REDM0485
      2 10X,30H MAXIMUM ITERATIONS.....,I8// REDM0486
      3 10X,30H ENTERING ELEMENT RATIO.....,F12.4//) REDM0487
* REDM0488
* Revise PHIO label to add (DEG). SCM 3-3-89 REDM0489
* REDM0490
310 FORMAT(/ 5X,'HYPERBOLIC STRESS-STRAIN PARAMETERS'/// REDM0491
      1 10X,36H SOIL CLASSIFICATION ..... ,5A4// REDM0492
      2 10X,30H COHESION INTERCEPT C.....,F12.4// REDM0493
      3 10X,30H FRICTION ANGLE PHIO (DEG)....,F12.4// REDM0494
      4 10X,30H 10-FOLD REDUCTION IN PHIO....,F12.4// REDM0495
      5 10X,30H SCALED MODULUS NUMBER ZK ..,F12.4// REDM0496
      6 10X,30H MODULUS EXPONENT ZN ..... ,F12.4// REDM0497
      7 10X,30H FAILURE RATIO RF ..... ,F12.4//) REDM0498
320 FORMAT(10X,30H BULK MODULUS NUMBER BK ....,F12.4// REDM0499
      1 10X,30H BULK MODULUS EXPONENT BM ..,F12.4) REDM0500
* REDM0501
* Insert format statements to print out hyperbolic REDM0502
* bulk modulus. MGK 9-29-88 REDM0503
* REDM0504
323 FORMAT(10X,30H INIT BULK MODULUS NUMBER BI.,F12.4// REDM0505
      1 10X,30H ULT VOLUMETRIC STRAIN EU ... ,F12.4) REDM0506
* REDM0507
325 FORMAT(10X,29H CONSTANT POISSON'S RATIO VT ,F12.4//) REDM0508

330 FORMAT(6F10.5) REDM0509
340 FORMAT(3F10.5) REDM0510
END REDM0511
* * * * * REDU0001
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM REDU0002
* VERSION 1.0, MARCH 1990 REDU0003
* PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY REDU0004
* THE FEDERAL HIGHWAY ADMINISTRATION REDU0005
* * * * * REDU0006
SUBROUTINE REDUCI(NN,NCOL) REDU0007
C REDU0008
C THIS SUBROUTINE REDUCES A BLOCK OF EQUATIONS REDU0009
C REDU0010
* Uncomment this statement in all program modules for double REDU0011
* precision version. SCM 3-10-90 REDU0012
* REDU0013
* IMPLICIT DOUBLE PRECISION (A-H,O-Z) REDU0014
* REDU0015
COMMON/SYSTEM/8(240),A(240,120) REDU0016
DO 300 N=1,NN REDU0017
IF(A(N,1) .EQ. 0.0)GO TO 300 REDU0018
D =B(N)/A(N,1) REDU0019
B(N)=D REDU0020
I=N REDU0021
DO 275 L=2,NCOL REDU0022
C=A(N,L)/A(N,1) REDU0023
I=I+1 REDU0024
IF (C .EQ. 0.0) GO TO 275 REDU0025
J=0 REDU0026
DO 250 K=L,NCOL REDU0027
J=J+1 REDU0028
250 A(I,J)=A(I,J)-C*A(N,K) REDU0029
B(I)=B(I)-A(N,L)*D REDU0030
275 A(N,L)=C REDU0031
300 CONTINUE REDU0032
RETURN REDU0033
END REDU0034
* * * * * RESO0001
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM RESO0002
* VERSION 1.0, MARCH 1990 RESO0003
* PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY RESO0004
* THE FEDERAL HIGHWAY ADMINISTRATION RESO0005
* * * * * RESO0006
* Add RESULT and NPPT parameters to subroutine statement. RESO0007
* SCM 3-01-89 RESO0008
* RESO0009
SUBROUTINE RESOUT(MNO,NOD,ST,U,V,X,Y,IA,NELEM,NPT,RESULT,NPPT) RESO0010
* RESO0011
* Uncomment this statement in all program modules for double RESO0012
* precision version. SCM 3-10-90 RESO0013
* RESO0014
* IMPLICIT DOUBLE PRECISION (A-H,O-Z) RESO0015
* RESO0016
* Add character variable. SCM 3-3-89 RESO0017
* RESO0018
CHARACTER RECORD*133 RESO0019
* RESO0020
* Add RESULT variable to dimension statement. SCM 3-4-89 RESO0021
* RESO0022
DIMENSION MNO(NELEM),NOD(4,NELEM),ST(6,NELEM),U(NPT),V(NPT), RESO0023
1 X(NPT),Y(NPT),STATE(4),RESULT(20,NPPT) RESO0024
RESO0025
RESO0026

```

```

COMMON /ELEMNT/ AS(3),BS(3),C1(17,17),EST(4,3),RQ(4),RT(3),
1 STORE(17),Z4(12),ZQ(4),ZT(3),ZY(17),J1,RQ
COMMON /SIZEPR/ BIVD(3),IX,NINC
*
* Expand array dimension statements from 30 to 99.
*   SCM 5-22-89
*
COMMON/XFACE/XXMAT,ANGLE(99),PCOEF(99),TENSIL(99),ITMAX
*
DATA CON/0.0174533/
DATA STATE/4HFIX,4HSLIP,4HSLIP,4HFREE /
C
C ESTABLISH PRINTING AND PLOTTING CONTROLS.
C
*
* Expand IWRT printing control parameters. MGK 2-16-89
*
* IWRT = 0: (No finite element stresses printed out.)
* IWRT = 1: (Standard finite element stresses printed out.)
* IWRT = 2: (Above, plus Duncan data printed out at Convergence.)
*           (Also writes trace of Duncan data to Unit 16.)
* IWRT = 3: (Above, plus writes trace of XFACES data to Unit 17.)
*
* Expand IPLOT output control parameters. SCM 3-1-89
*
* IPLOT = 0: (No plot data written to Units 10 or 30.)
* IPLOT = 1: (Mesh plot data written to Unit 10.)
* IPLOT = 2: (Pipe data written to Unit 30.)
* IPLOT = 3: (Both mesh and pipe plot data written to
*            Units 10 and 30 respectively.)
*
IF(IA.GE.0) GO TO 50
IA = IABS(IA)
LUPLOT = IA/100
IPLOT = (IA-LUPLOT*100)/10
IWRT = IA-LUPLOT*100-IPLOT*10
RETURN
C
C** WRITE DISPLACEMENTS AND STRESSES TO PLOT TAPE IF REQUESTED.
C
50 IF(IPLOT.EQ.0) GO TO 100
*
* Add control logic for writing plot output. SCM 3-1-89
*
IF(IPLOT.EQ.2) GO TO 80
*
ZERO = 0.0
C
C*****WRITE STATEMENTS ELIMINATED, JUNE 78*****
C
C WRITE(LUPLOT) K,ZERO,(U(N),V(N),N=1,NPT)
C WRITE(LUPLOT) K,ZERO,(ST(4,N),ST(5,N),ZERO,ST(6,N),N=1,NELEM)
C WRITE(LUPLOT) K,ZERO,(ST(1,N),ST(2,N),ZERO,ST(3,N),N=1,NELEM)
C
C*****
*
* Revise LUPLOT from unformatted to formatted.
*   SCM 11-24-88
*
*****WRITE STATEMENTS INCORPORATED, JUNE 78*****
RES00027 * WRITE(LUPLOT) IA,(U(N),V(N),N=1,NPT)
RES00028 * WRITE(LUPLOT) IA,((ST(J,N),J=1,6),N=1,NELEM)
RES00029 *
RES00030 WRITE(LUPLOT,1000) IA
RES00031 DO 60 N=1,NPT
RES00032 WRITE(LUPLOT,1010) U(N),V(N)
RES00033 60 CONTINUE
RES00034 DO 70 N=1,NELEM
RES00035 WRITE(LUPLOT,1020) (ST(J,N),J=1,6)
RES00036 70 CONTINUE
RES00037 IF(IPLOT.LT.2) GO TO 100
RES00038 WRITE(30,1000) IA
RES00039 DO 90, N=1,NPPT
RES00040 WRITE(30,1110) (RESULT(J,N),J=1,9)
RES00041 90 CONTINUE
RES00042 *
RES00043 C
RES00044 C*****
RES00045 C
RES00046 C
RES00047 IF(IA.EQ.NINC) END FILE LUPLOT
RES00048 IF(IWRT.LE.1) GO TO 120
RES00049 *
RES00050 * Revise control logic to add printing of Duncan data.
RES00051 *   SCM 3-1-89
RES00052 *
RES00053 IF(IWRT.LE.1) GO TO 120
RES00054 REWIND 16
RES00055 NREC = 0
RES00056 MSUMR = 0
RES00057 105 READ(16,'(11)',ERR=500,END=107) IPAGE
RES00058 NREC = NREC + 1
RES00059 IF(IPAGE.EQ.1) THEN
RES00060 READ(16,6000)
RES00061 READ(16,6000) IAX,ITERX
RES00062 IF(IAX.EQ.1A) THEN
RES00063 MSUMR = MSUMR + NREC + 2
RES00064 NREC = 0
RES00065 ELSE
RES00066 NREC = NREC + 2
RES00067 END IF
RES00068 END IF
RES00069 GO TO 105
RES00070 107 REWIND 16
RES00071 IF(MSUMR.EQ.0) GO TO 120
RES00072 DO 110, NR=1,MSUMR+9
RES00073 READ(16,*,ERR=500,END=110)
RES00074 CONTINUE
RES00075 MCT = 40
RES00076 DO 118, NN=1,NREC-9
RES00077 IF(MCT.LT.40) GO TO 115
RES00078 MCT = 0
RES00079 WRITE(6,6010) IAX,ITERX
RES00080 WRITE(6,6020)
RES00081 MCT = MCT + 1
RES00082 115 READ(16,'(A133)',ERR=500,END=118) RECORD
RES00083 WRITE(6,'(A133)') RECORD
RES00084 CONTINUE
RES00085 *
RES00086 120 CONTINUE
RES00087 C
RES00088 C** PRINT OUT DISPLACEMENTS
RES00089 C
RES00090 RES00090
RES00091 RES00091
RES00092 RES00092
RES00093 RES00093
RES00094 RES00094
RES00095 RES00095
RES00096 RES00096
RES00097 RES00097
RES00098 RES00098
RES00099 RES00099
RES00100 RES00100
RES00101 RES00101
RES00102 RES00102
RES00103 RES00103
RES00104 RES00104
RES00105 RES00105
RES00106 RES00106
RES00107 RES00107
RES00108 RES00108
RES00109 RES00109
RES00110 RES00110
RES00111 RES00111
RES00112 RES00112
RES00113 RES00113
RES00114 RES00114
RES00115 RES00115
RES00116 RES00116
RES00117 RES00117
RES00118 RES00118
RES00119 RES00119
RES00120 RES00120
RES00121 RES00121
RES00122 RES00122
RES00123 RES00123
RES00124 RES00124
RES00125 RES00125
RES00126 RES00126
RES00127 RES00127
RES00128 RES00128
RES00129 RES00129
RES00130 RES00130
RES00131 RES00131
RES00132 RES00132
RES00133 RES00133
RES00134 RES00134
RES00135 RES00135
RES00136 RES00136
RES00137 RES00137
RES00138 RES00138
RES00139 RES00139
RES00140 RES00140
RES00141 RES00141
RES00142 RES00142
RES00143 RES00143
RES00144 RES00144
RES00145 RES00145
RES00146 RES00146
RES00147 RES00147
RES00148 RES00148
RES00149 RES00149
RES00150 RES00150
RES00151 RES00151
RES00152 RES00152

```



```

MCT = 40
DO 150 N=1,NPT
  IF(MCT.LT.40) GO TO 125
  MCT = 0
  WRITE(6,2000) IA
  WRITE(6,2100)
125 MCT = MCT + 1
150 WRITE(6,2200) N,X(N),Y(N),U(N),V(N)
C
C** PRINT OUT BENDING ELEMENT RESULTANTS
C

```

```

MCT = 40
DO 300 I=1,NELEM
  N = MNO(I)
  IAC = MOD(N,1000)/10
  IF(IAC.GT.1A) GO TO 300
  IF (MOD(2,I) .NE. MOD(3,I)) GO TO 300
  IF(MCT.LT.40) GO TO 240
  MCT = 0
  WRITE(6,2000) IA
  WRITE(6,3100)
240 MCT = MCT + 1
  N1 = NOD(1,I)
  N2 = NOD(2,I)
  XC = 0.5*(X(N1) + X(N2))
  YC = 0.5*(Y(N1) + Y(N2))
  DO 250 K=1,3
250 RQ(K) = 0.5*(ST(K,I)+ST(K+3,I))
  WRITE(6,3200) I,XC,YC,(RQ(K),K=1,3)
300 CONTINUE
C
C PRINT OUT INTERFACE ELEMENT FORCES
C

```

```

MCT = 40
DO 315 I=1,NELEM
  IAC = MOD(MNO(I),1000)/10
  IF(IAC.GT.1A) GO TO 315
  MN = MNO(I)/1000
  MNX = MNO(I)-MN*1000-10*IAC
  IF(MNX.LE.0) GO TO 315
  IF(MCT.LT.40) GO TO 310
  MCT = 0
  WRITE(6,2000) IA
  WRITE(6,3500)
310 MCT = MCT + 1
  N1 = NOD(1,I)
  N2 = NOD(2,I)
  N3 = NOD(3,I)
  RADA = CON*ANGLE(MN)
  C = COS(RADA)
  S = SIN(RADA)
  UNI = C*U(N1)+S*V(N1)
  USI = -S*U(N1)+C*V(N1)
  UNJ = C*U(N2)+S*V(N2)
  USJ = -S*U(N2)+C*V(N2)
  WRITE(6,3600) I,MN,STATE(MNX),U(N3),V(N3),UNI,USI,UNJ,USJ
315 CONTINUE
C
C** PRINT OUT STRESSES AND STRAINS FOR QUAD ELEMENTS
C
MCT = 40
DO 400 I=1,NELEM
  N = MNO(I)

```

```

RES00153 IAC = MOD(N,1000)/10
RES00154 IF(IAC.GT.1A) GO TO 400
RES00155 IF (MOD(2,I) .EQ. MOD(3,I)) GO TO 400
RES00156 N = N/1000
RES00157 MNX = MNO(I)-N*1000-10*IAC
RES00158 IF (MNX .NE. 0) GO TO 400
RES00159 IF(MCT.LT.40) GO TO 325
RES00160 MCT = 0
RES00161 WRITE(6,2000) IA
RES00162 WRITE(6,4100)
325 MCT = MCT + 1
RES00163 N1 = NOD(1,I)
RES00164 N2 = NOD(2,I)
RES00165 N3 = NOD(3,I)
RES00166 N4 = NOD(4,I)
RES00167 XC = 0.25*(X(N1)+X(N2)+X(N3)+X(N4))
RES00168 YC = 0.25*(Y(N1)+Y(N2)+Y(N3)+Y(N4))
RES00169 DO 350 K=1,3
RES00170 RQ(K) = ST(K+3,I)
RES00171
RES00172 C
RES00173 C PRINCIPAL STRESSES CALCULATED
RES00174 C
RES00175 CALL PRINC(RQ,ZQ)
RES00176 WRITE(6,4200) I,XC,YC,(ST(K,I),K=1,6),(ZQ(K),K=1,3)
RES00177 400 CONTINUE
RES00178 RETURN
RES00179 *
RES00180 * Add ERROR STOP for I/O errors on Unit 16. SCM 3-1-89
RES00181 *
RES00182 500 WRITE(6,6030)
RES00183 STOP 15
RES00184 *
RES00185 C
RES00186 C** FORMATS
RES00187 C
RES00188 *
RES00189 * Revise formats for plot and print output. SCM 11-24-88
RES00190 *
RES00191 1000 FORMAT(I5)
RES00192 1010 FORMAT(E12.4,1H,,E12.4)
RES00193 1020 FORMAT(E12.4,5(1H,,E12.4))
RES00194 1110 FORMAT(E12.4,8(1H,,E12.4))
RES00195 *
RES00196 2000 FORMAT(1H1,/// 5X,23H CONSTRUCTION INCREMENT, I4 ///)
RES00197 2100 FORMAT(//5X,5H NODE,7X,8HX-COORD.,7X,8HY-COORD.,8X,7RX-DISP.,
RES00198 X 8X,7HY-DISP. )
RES00199 2200 FORMAT(5X,I3,2X,4(2X,E13.4))
RES00200 3100 FORMAT(5X, //36H BENDING ELEMENT RESULTANT FORCES .. //5X,5HELMNT,
RES00201 X 6X,8HX-CENTER,6X,8HY-CENTER,8X,6HTRUST,9X,5HSHEAR,8X,6HMOMENT/)
RES00202 3200 FORMAT(I10,5(1X,E13.4))
RES00203 3500 FORMAT(// 5X,43H INTERFACE ELEMENT FORCES AND DISPLACEMENTS //
RES00204 X 5X,5HELMNT,5H MAT.,3X,5HSTATE,7X,7HN-FORCE,7X,7HS-FORCE,2X,
RES00205 X 3X, 9HN-DISP(I),5X,9HS-DISP(I),5X,9HN-DISP(J),5X,9HS-DISP(J))
RES00206 3600 FORMAT(5X,2I5,4X,A4,6(1X,E13.4))
RES00207 4100 FORMAT(5X, 7HELEMENT, 45X,
RES00208 1 28HELEMENT STRAINS AND STRESSES / ,6X,3HNO.,8XRES00271
RES00209 1 ,1HX,10X,1HY,6X,9HEPSILON-X, 2X, 9HEPSILON-Y, 3X, 7HGAMA-XY, 4X,RES00272
RES00210 2 ,7HSIGMA-X, 4X, 7HSIGMA-Y,4X,6HTAU-XY,5X,7HSIGMA-1, 4X, 7HSIGMA-2,RES00273
RES00211 3 6X, 3HANG, /)
RES00212 4200 FORMAT(I7 ,4X,1P11E11.2)
RES00213 *
RES00214 * Add formats for Duncan output. SCM 3-1-89
RES00215 *

```

```

RES00216
RES00217
RES00218
RES00219
RES00220
RES00221
RES00222
RES00223
RES00224
RES00225
RES00226
RES00227
RES00228
RES00229
RES00230
RES00231
RES00232
RES00233
RES00234
RES00235
RES00236
RES00237
RES00238
RES00239
RES00240
RES00241
RES00242
RES00243
RES00244
RES00245
RES00246
RES00247
RES00248
RES00249
RES00250
RES00251
RES00252
RES00253
RES00254
RES00255
RES00256
RES00257
RES00258
RES00259
RES00260
RES00261
RES00262
RES00263
RES00264
RES00265
RES00266
RES00267
RES00268
RES00269
RES00270
RES00271
RES00272
RES00273
RES00274
RES00275
RES00276
RES00277
RES00278

```

```

6000 FORMAT(29X,I2,15X,I2) RES00279 * SCM 4-30-89 RESP0044
6010 FORMAT(1H1,/,6X,23HCONSTRUCTION INCREMENT ,I2,15X, RES00280 * RESP0045
X31H(CONVERGENCE ACHIEVED AFTER ***,I4,29H *** ITERATIONS IF ICON RES00281 * DIMENSION NELPIP(99),NELIST(99) RESP0046
X= 1)) RES00282 * RESP0047
6020 FORMAT(1H0,15X,47H* * * ELASTIC MODULI AND CONVERGENCE PARAMETERS,RES00283 C RESP0048
X28H FOR DUNCAN SOIL MODEL * * *,/ RES00284 * GO TO (99,299) , ICOME RESP0049
X22X,69H(CONVERGENCE ERRORS (ERROR-EMOD) ARE RATIOS GIVEN BY, (NEW-RES00285 C RESP0050
XOLD)/NEW ,/,/ RES00286 C FIRST ENTRY RESP0051
X22X,42HK-TYPE = 1 ; DUNCAN POWER LAW BULK MODULUS, RES00287 C COLLECT CULVERT ELEMENTS AND FIND TOP ELEMENT RESP0052
X10X,45HSTRESS-LEV <= -1.0 ; COMPLETE TENSION FAILURE, / RES00288 C RESP0053
X22X,42HK-TYPE = 2 ; SELIG HYPERBOLIC BULK MODULUS, RES00289 99 CONTINUE RESP0054
X10X,34HSTRESS-LEV >= +1.0 ; SHEAR FAILURE, / RES00290 YMAX = -1.E+10 RESP0055
X22X,37HK-TYPE = 3 ; CONSTANT POISSON'S RATIO, // RES00291 NEP = 0 RESP0056
X3X,5HELE-N,4X,5HMAT-N,4X,6HK-TYPE,4X,4HICON,4X, RES00292 DO 100 I=1,NELEM RESP0057
X10HERROR-EMOD,4X,10HSTRESS-LEV,4X,10HYOUNGS-MOD,6X,8HBULK-MOD, RES00293 IF (MOD(2,I) .NE. MOD(3,I)) GO TO 100 RESP0058
X4X,10HPOIS-RATIO,4X,12HCONFINED-MOD, // RES00294 NEP = NEP + 1 RESP0059
6030 FORMAT(1H0,21H***FATAL I/O ERROR***/1H ,55HDUNCAN SUBROUTINE OUTPURES00295 NELPIP(NEP) = I RESP0060
XT FILE NOT PRESENT OR UNREADABLE//) RES00296 ND1 = MOD(1,I) RESP0061
+ RES00297 ND2 = MOD(2,I) RESP0062
END RES00298 * RESP0063
* * * * * RES00001 * Revise selection procedure to choose Material Number 1 as RESP0064
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM RES00002 * first beam element in top, right position from centerline, RESP0065
* VERSION 1.0, MARCH 1990 RES00003 * SCM 2-25-90 RESP0066
* PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY RES00004 * RESP0067
* THE FEDERAL HIGHWAY ADMINISTRATION RES00005 * IF(Y(ND1)+Y(ND2) .LE. YMAX) GO TO 100 RESP0068
* * * * * RES00006 * YMAX = Y(ND1) + Y(ND2) RESP0069
SUBROUTINE RESPIP (MNO,NOD,RESULT,ST,STPIPE,U,V,X,Y, RES00007 * RESP0070
1 NELEM,NPMAT,NPPT,NPT,IEXIT,ICOME) RES00008 IF(Y(ND2) .LE. YMAX) GO TO 100 RESP0071
RES00009 YMAX = Y(ND2) RESP0072
C THIS SUBROUTINE ORDERS PIPE ELEMENTS AND PUTS PIPE RESPONSES RES00010 * RESP0073
C INTO MATRIX RESULTS. RES00011 NETOP = I RESP0074
C RES00012 100 CONTINUE RESP0075
C RESULT(1,N) = X-COORDINATE ON PIPE (X=Y=0, AT PIPE CENTER, FOR RES00013 IF(NEP.EQ.0) RETURN RESP0076
C RESULT(2,N) = Y-COORDINATE ON PIPE LEVEL = 2 ) RES00014 C RESP0077
C RESULT(2,N) = Y-COORDINATE ON PIPE LEVEL = 2) RES00015 C FIND BEAM ELEMENTS CONNECTED CLOCKWISE FROM TOP ELEMENT. RESP0078
C RESULT(3,N) = X-DISPLACEMENT RES00016 C RESP0079
C RESULT(4,N) = Y-DISPLACEMENT RES00017 KK = 1 RESP0080
C RESULT(5,N) = BENDING MOMENT (POSITIVE = INNER FIBER TENSION) RES00018 NEL = NETOP RESP0081
C RESULT(6,N) = THRUST FORCE (COMPRESSION = NEGATIVE) RES00019 150 NELIST(KK) = NEL RESP0082
C RESULT(7,N) = SHEAR FORCE RES00020 IF(KK.EQ. NEP) GO TO 250 RESP0083
C RESULT(8,N) = NORMAL PRESSURE ON PIPE (POSITIVE OUTWARD NORMAL) RES00021 NA1 = MOD(1,NEL) RESP0084
C RESULT(9,N) = TANGENTIAL PRESSURE ON PIPE (POSITIVE CLOCKWISE) RES00022 DO 200 I=1,NEP RESP0085
C RES00023 NEL = NELPIP(I) RESP0086
C THE REMAINING RESULT ARRAY IS DEPENDENT ON PIPE TYPE .. RES00024 NB2 = MOD(2,NEL) RESP0087
C RES00025 IF(NA1.NE.NB2) GO TO 200 RESP0088
* Uncomment this statement in all program modules for double RES00026 KK = KK +1 RESP0089
* precision version. SCM 3-10-90 RES00027 GO TO 150 RESP0090
* RES00028 200 CONTINUE RESP0091
* IMPLICIT DOUBLE PRECISION (A-H,O-Z) RES00029 C RESP0092
* RES00030 C FIND BEAM ELEMENTS CONNECTED COUNTER-CLOCKWISE FROM TOP ELEMENT. RESP0093
* DIMENSION MNO(NELEM),NOD(4,NELEM),RESULT(20,NPPT) RES00031 C RESP0094
* 1 ,ST(6,NELEM),STPIPE(2,NPPT),U(NPT),V(NPT),X(NPT),Y(NPT) RES00032 DO 220 I=1,KK RESP0095
* RES00033 NELIST(NEP-I+1) = NELIST(KK-I+1) RESP0096
* Revise dimensions of XMONI and XTHRU arrays from 30 to 99. RES00034 C RESP0097
* SCM 4-30-89 RES00035 NEL = NETOP RESP0098
* RES00036 NELIST(NEP-KK+1) = NEL RESP0099
* DIMENSION XMONI(99),XTHRU(99) RES00037 IF(KK.EQ.NEP) GO TO 250 RESP0100
* RES00038 NA2 = MOD(2,NEL) RESP0101
* COMMON /SIZEPR/ BLVD(3),IX,NINC RES00039 DO 230 I=1,NEP RESP0102
* COMMON /MATERL/ CP(3,3),DEN(10),E(10,10),GNY(10,10),H(10,10), RES00040 NEL = NELPIP(I) RESP0103
* 1 ITYPE(10),NON(10) RES00041 NB1 = MOD(1,NEL) RESP0104
* RES00042 IF(NA2.NE.NB1) GO TO 230 RESP0105
* Revise dimensions of NELPIP and NELIST arrays from 30 to 99. RES00043 KK = KK + 1 RESP0106

```

```

      GO TO 225
230  CONTINUE
      C
      WRITE MESSAGE DENOTING UNCONNECTED STRUCTURES.
      C
      WRITE(6,3000) NEP,KK
      DO 240 I=1,KK
240  NELIST(I) = NELIST(NEP-KK+I)
      C
      C** NOTE, KK = NUMBER OF CONNECTED ELEMENTS, NEP = ALL BEAM ELMTS.
      C** CHECK NUMBER OF CONNECTED ELEMENTS WITH PRESCRIBED VALUE
      *
      *   Revise limitation that continuous elements be connected.
      *   SCM 4-30-89
      *
      *250 IF (NPMAT .NE. KK) GO TO 650
      *
250  CONTINUE
      C
      C** CHECK THE BEAM SEQUENCE NUMBER OF EACH BEAM ELEMENT IN THE
      C** CLOCKWISE ORDER OF THE CONNECTED PIPE ELEMENTS
      C
      DO 285 I=1,NPMAT
      NEL = NELIST(I)
      MAT = MNO(NEL)/1000
      IF (MAT .NE. 1) GO TO 600
      C
      DO 275 J=1,NEP
275  IF (NEL.EQ.NELPIP(J)) NELPIP(J) = -NELPIP(J)
      C
285  CONTINUE
      C
      C** CONSIDER BEAM ELEMENTS NOT PART OF PIPE SEQUENCE. MAKE SURE
      C** THEY HAVE MATERIAL NUMBERS GREATER THAN NPMAT.
      C
      IF (NEP.EQ.NPMAT) GO TO 295
      DO 290 J=1,NEP
      NEL = NELPIP(J)
      IF (NEL.LE.0) GO TO 290
      MAT = MNO(NEL)/1000
      IF (MAT .LT. NPMAT) GO TO 675
290  CONTINUE
295  CONTINUE
      C
      C DETERMINE IF PIPE IS OPEN OR CLOSED.
      C
      ICURVE = 1
      NELEFT = NELIST(1)
      NERITE = NELIST(NPMAT)
      NDLEFT = NOD(2,NELEFT)
      NDRITE = NOD(1,NERITE)
      IF (NDLEFT.EQ.NDRITE) ICURVE = 0
      NPPT = NPMAT + ICURVE
      C
      C PRINT OUT THE ELEMENTS COMPRISING PIPE.
      C
      C
      WRITE(6,2000) (NELIST(I),I=1,NPMAT)
      C
      RETURN
      C
      C SECOND ENTRY
      C
      C PUT STRUCTURAL RESPONSES INTO RESULT VECTOR.

```

```

RESP0107  C
RESP0108 299 CONTINUE
RESP0109 IF (NEP.EQ.0) RETURN
RESP0110 DO 300 I=1,NPMAT
RESP0111 NE = NELIST(I)
RESP0112 ND1 = NOD(1,NE)
RESP0113 ND2 = NOD(2,NE)
RESP0114 RESULT(1,I) = X(ND2)
RESP0115 RESULT(2,I) = Y(ND2)
RESP0116 RESULT(3,I) = U(ND2)
RESP0117 300 RESULT(4,I) = V(ND2)
RESP0118 C
RESP0119 IF (ICURVE.EQ.0) GO TO 310
RESP0120 RESULT(1,NPPT) = X(ND1)
RESP0121 RESULT(2,NPPT) = Y(ND1)
RESP0122 RESULT(3,NPPT) = U(ND1)
RESP0123 RESULT(4,NPPT) = V(ND1)
RESP0124 310 CONTINUE
RESP0125 C
RESP0126 C** SAVE THE PREVIOUS CONVERGED SOLUTION FOR MOMENT AND THRUST
RESP0127 C
RESP0128 IF (IEXIT.EQ.-1) GO TO 360
RESP0129 DO 350 NUD = 1, NPPT
RESP0130 XMONT(NUD) = RESULT(5,NUD)
RESP0131 350 XTHRU(NUD) = RESULT(6,NUD)
RESP0132 360 CONTINUE
RESP0133 C
RESP0134 C DISTRIBUTE ELEMENT RESPONSES TO NODES
RESP0135 C
      I = 1
      I1 = I+1
      400 NEL = NELIST(I)
      NE2 = NELIST(I1)
      ND1 = NOD(2,NEL)
      ND2 = NOD(2,NE2)
      ND3 = NOD(1,NE2)
      S1 = SQRT((X(ND1)-X(ND2))**2+(Y(ND1)-Y(ND2))**2)
      S2 = SQRT((X(ND3)-X(ND2))**2+(Y(ND3)-Y(ND2))**2)
      S3 = SQRT((X(ND3)-X(ND1))**2+(Y(ND3)-Y(ND1))**2)
      IF (I.EQ.1) SD1 = S1/2.
      IF (I1.EQ.NPMAT) SD2 = S2/2.0
      FX = STPIPE(1,I) -STPIPE(1,I1)
      FY = STPIPE(2,I) -STPIPE(2,I1)
      COSA = (X(ND3)-X(ND1))/S3
      SINA = (Y(ND1)-Y(ND3))/S3
      FN = FY*COSA + FX*SINA
      FT = -FX*COSA + FY*SINA
      C
      RESULT(5,I1) = (ST(3,NEL) + ST(6,NE2))/2.0
      RESULT(6,I1) = (ST(1,NEL) + ST(4,NE2))/2.0
      RESULT(7,I1) = (ST(2,NEL) + ST(5,NE2)) / 2.0
      RESULT(8,I1) = FN/(0.5*(S1+S2))
      RESULT(9,I1) = FT/(0.5*(S1+S2))
      RESULT(19,I1) = RESULT(5,I1) -XMONT(I1)
      RESULT(20,I1) = RESULT(6,I1) -XTHRU(I1)
      IF (I1.LT. 1) GO TO 500
      IF (ICURVE.EQ.1.AND.I1.EQ.NPMAT) GO TO 450
      I = I + 1
      I1 = I + 1
      IF (ICURVE.EQ.0 .AND. I.EQ.NPMAT) I1 = 1
      GO TO 400
      C
      C GET END POINTS FOR OPEN SECTIONS (OR SYMMETRIC HALVES)

```

```

RESP0170
RESP0171
RESP0172
RESP0173
RESP0174
RESP0175
RESP0176
RESP0177
RESP0178
RESP0179
RESP0180
RESP0181
RESP0182
RESP0183
RESP0184
RESP0185
RESP0186
RESP0187
RESP0188
RESP0189
RESP0190
RESP0191
RESP0192
RESP0193
RESP0194
RESP0195
RESP0196
RESP0197
RESP0198
RESP0199
RESP0200
RESP0201
RESP0202
RESP0203
RESP0204
RESP0205
RESP0206
RESP0207
RESP0208
RESP0209
RESP0210
RESP0211
RESP0212
RESP0213
RESP0214
RESP0215
RESP0216
RESP0217
RESP0218
RESP0219
RESP0220
RESP0221
RESP0222
RESP0223
RESP0224
RESP0225
RESP0226
RESP0227
RESP0228
RESP0229
RESP0230
RESP0231
RESP0232

```

```

C
450 CONTINUE
    RESULT(5,1) = ST(6,NELEFT)
    RESULT(6,1) = ST(4,NELEFT)
    RESULT(7,1) = 0.0
    RESULT(8,1) = -STPIPE(2,1) / SD1
    RESULT(9,1) = 0.0
    RESULT(19,1) = RESULT(5,1)-XMONT(1)
    RESULT(20,1) = RESULT(6,1) - XTHRU(1)
    RESULT(5,NPPT) = ST(3,NERITE)
    RESULT(6,NPPT) = ST(1,NERITE)
    RESULT(7,NPPT) = 0.0
    RESULT(8,NPPT) = -STPIPE(2,NPMT) /SD2
    RESULT(9,NPPT) = 0.0
    RESULT(19,NPPT) = RESULT(5,NPPT) - XMONT(NPPT)
    RESULT(20,NPPT) = RESULT(6,NPPT) - XTHRU(NPPT)

C
C** INITILIZE RESULT MATRIX
C
500 RETURN
C
600 WRITE(6,4000) NEL,MAT,I
    STOP 10
650 WRITE(6,4100) NPMAT,KK
    STOP 10
675 WRITE(6,4200) NEL
    STOP 10
C
2000 FORMAT(///10X,46H AN ORDERED LIST OF BEAM ELEMENTS AROUND PIPE. /
X 10X,56H ORDERING IS CLOCKWISE FROM TOP TO BOTTOM OR LEFT TO RIGHTRESP0262
X/(10X,10I5))
3000 FORMAT(///37H ALL BEAM ELEMENTS ARE NOT CONNECTED. /
X 37H THE TOTAL NUMBER OF BEAM ELEMENTS IS ,I5 /
X 41H THE NUMBER OF CONNECTED BEAM ELEMENTS IS ,I5/
X 54H THE LIST OF BEAM ELEMENTS BELOW WILL BE CONSIDERED TO /
X 54H TO COMPRISE THE PIPE. OTHER BEAM ELEMENTS WILL NOT BE /
X 57H PROCESSED BY THE PIPE SUBROUTINES. (NOTE IF YOU DID NOT /
X 56H INTEND DISCONNECTED STRUCTURES CHECK BEAM CONNECTIVITY. )
4000 FORMAT(///64H FATAL DATA ERROR. BEAM MATERIAL NUMBER IS NOT PROPERRESP0271
1LY DEFINED ,/
X /15H ELEMENT NUMBER,I5,/,16H MATERIAL NUMBER,I5,/
X 26H MATERIAL NUMBER SHOULD BE , I5)
4100 FORMAT (/// 40H * * * * * FATAL DATA ERROR * * * * * ,/
1 44H NUMBER OF SPECIFIED CONNECTED BEAM ELEMENTS ,I5,RESP0276
2 41H NUMBER OF ACTUAL CONNECTED BEAM ELEMENTS, I5)
4200 FORMAT (/// 40H * * * * * FATAL DATA ERROR * * * * * ,/
1 67H UNCONNECTED BEAM MATERIAL NUMBER SMALLER THAN NPMARESP0279
2T ELEMENT NUMBER, I5)
END
* * * * *
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
* VERSION 1.0, MARCH 1990
* PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
* THE FEDERAL HIGHWAY ADMINISTRATION
* * * * *
SUBROUTINE SAVED (X, Y, NOD,NODB,BIV,KODE,COOE,TITLE,NPT,NELEM,
1 IPLOT,NBPTC,NPUTCK,KPUTCK,LSTOP,NINC,LUDATA,
2 LUPLLOT,NSMAT,NPMAT,NPPT)
*
* Uncomment this statement in all program modules for double
* precision version. SCM 3-10-90
*
* IMPLICIT DOUBLE PRECISION (A-H,O-Z)
RESP0233
RESP0234
RESP0235
RESP0236
RESP0237
RESP0238
RESP0239
RESP0240
RESP0241
RESP0242
RESP0243
RESP0244
RESP0245
RESP0246
RESP0247
RESP0248
RESP0249
RESP0250
RESP0251
RESP0252
RESP0253
RESP0254
RESP0255
RESP0256
RESP0257
RESP0258
RESP0259
RESP0260
RESP0261
RESP0262
RESP0263
RESP0264
RESP0265
RESP0266
RESP0267
RESP0268
RESP0269
RESP0270
RESP0271
RESP0272
RESP0273
RESP0274
RESP0275
RESP0276
RESP0277
RESP0278
RESP0279
RESP0280
RESP0281
SAVD0001
SAVD0002
SAVD0003
SAVD0004
SAVD0005
SAVD0006
SAVD0007
SAVD0008
SAVD0009
SAVD0010
SAVD0011
SAVD0012
SAVD0013
SAVD0014
*
* DIMENSION X(1), Y(1), NOD(LSTOP,1), NODB(1), BIV(3,1), KODE(1),
1 CODE(1), TITLE(17)
* DATA TPSCAN, FZERO, IZERO /1.0, 0.0, 0/
C
C** INITIALIZE PARAMETERS
C
NUMMAT = NSMAT + 1
IF (KPUTCK .NE. 0) NPUTCK = 1
IF(NPUTCK.NE.0 .AND. IPLOT.NE.0) IPLOT =-1
DO 20 N = 1, NPT
20 CODE(N) = 0.0
DO 30 N = 1, NBPTC
30 CONTINUE
DO 40 N = 1, NELEM
40 CONTINUE
KODE(N) = NOD(5,N)
IF(NOD(2,N).EQ.NOD(3,N)) KODE(N) = 11
IF(NOD(7,N).EQ.1) KODE(N) = 11
*
* SAVE MESH DATA FOR PLOTTING AND AS INPUT DATA FOR HEROIC
C
*
* Revise I/O Unit from LUDATA to Unit 14.
* SCM 11-24-88
*
* REWIND 14
* IF(IPLOT.EQ.0) GO TO 50
* IF(IPLOT.EQ.2) GO TO 120
*
* Revise LUPLLOT from unformatted to formatted.
* SCM 11-24-88
*****WRITE STATEMENTS PERMANENTLY INCORPORATED (JUNE 78)*****
* WRITE (LUPLLOT) (TITLE(K),K=1,17),NPT,NELEM,NUMMAT,NPUTCK,NINC
* WRITE (LUPLLOT) (CODE(N),X(N),Y(N),N=1,NPT)
* WRITE (LUPLLOT) ((NOD(K,N),K=1,4),KODE(N),NOD(6,N),N=1,NELEM)
*
* WRITE (LUPLLOT,6010) (TITLE(K),K=1,17)
* WRITE (LUPLLOT,6015) NPT,NELEM,NUMMAT,NPUTCK,NBPTC,NINC
* DO 100 N=1,NPT
100 CONTINUE
* DO 110 N=1,NELEM
110 CONTINUE
* IF(IPLOT.LT.2) GO TO 50
120 WRITE(30,6010) (TITLE(K),K=1,17)
WRITE(30,6030) NINC,NPMAT,NPPT
C
*****WRITE STATEMENTS PERMANENTLY ELIMINATED (JUNE 78)*****
C WRITE (LUPLLOT) (TITLE(K), K = 1,8), NPT, NELEM, NUMMAT, NPUTCK,
1 FZERO, NINC, IZERO, FZERO, FZERO, TPSCAN, TPSCAN,
2 (IZERO,K=1,12)
C WRITE (LUPLLOT) (CODE(N), X(N), Y(N), N = 1,NPT),
1 ((NOD(K,N),K=1,4),KODE(N),N=1,NELEM), (IZERO,K=1,4)
C

```

```

C*****SAVD0078
C      SAVD0079
50 CONTINUE      SAVD0080
      DO 45 N = 1, NELEM      SAVD0081
      NOD(6,N) = 1000*NOD(5,N) + 10*NOD(6,N) + NOD(7,N)      SAVD0082
45 CONTINUE      SAVD0083
      IF(IPLLOT.LT.0) END FILE LUPLLOT      SAVD0084
C      SAVD0085
*      SAVD0086
*      Revise I/O unit from LUDATA to Unit 14.      SAVD0087
*      SCM 11-14-88      SAVD0088
*      SAVD0089
      WRITE (14) NPT,NELEM,NBPTC,NINC,NPUTCK      SAVD0090
      WRITE (14) (X(N),Y(N), N=1,NPT),      SAVD0091
1      ((NOD(K,N), K=1,4), NOD(6,N), N=1,NELEM),      SAVD0092
2      (NODB(N), (BIV(K,N), K=1,3), N=1,NBPTC)      SAVD0093
*      SAVD0094
C      SAVD0095
      WRITE (6,6005)      SAVD0096
      RETURN      SAVD0097
6005 FORMAT ( 53H0* * * * * MESH DATA HAS BEEN SAVED * * * * *)      SAVD0098
6010 FORMAT(1H',17A4,1H')      SAVD0099
6015 FORMAT(15,5(1H,,15))      SAVD0100
6020 FORMAT(15,2(1H,,E12.4))      SAVD0101
6025 FORMAT(15,6(1H,,15))      SAVD0102
6030 FORMAT(15,2(1H,,15))      SAVD0103
      END      SAVD0104
*      SAVD0001
*      CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM      SAVD0002
*      VERSION 1.0, MARCH 1990      SAVD0003
*      PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY      SAVD0004
*      THE FEDERAL HIGHWAY ADMINISTRATION      SAVD0005
*      SAVD0006
*      SAVD0007
      SUBROUTINE SAVED (X, Y, NOD,NODB,BIV,KODE,CODE,TITLE,NPT,NELEM,      SAVD0008
1      IPLLOT,NBPTC,NPUTCK,KPUTCK,LSTOP,NINC,LUDATA,      SAVD0009
2      LUPLLOT,NSMAT,NPMAT,NPPT)      SAVD0010
*      SAVD0011
*      Uncomment this statement in all program modules for double      SAVD0012
*      precision version. SCM 3-10-90      SAVD0013
*      SAVD0014
*      IMPLICIT DOUBLE PRECISION (A-H,O-Z)      SAVD0015
*      SAVD0016
      DIMENSION X(1), Y(1), NOD(LSTOP,1), NODB(1), BIV(3,1), KODE(1),      SAVD0017
1      CODE(1), TITLE(17)      SAVD0018
C      DATA TPSCAN, FZERO, IZERO /1.0, 0.0, 0/      SAVD0019
C      SAVD0020
C** INITIALIZE PARAMETERS      SAVD0021
C      SAVD0022
      NUMMAT = NSMAT + 1      SAVD0023
      IF (KPUTCK.NE.0) NPUTCK = 1      SAVD0024
      IF(NPUTCK.NE.0 .AND. IPLLOT.NE.0) IPLLOT = -1      SAVD0025
      DO 20 N = 1, NPT      SAVD0026
      CODE(N) = 0.0      SAVD0027
      DO 30 N = 1, NBPTC      SAVD0028
      NODE = NODB(N)/10000      SAVD0029
      IF (MOD(NODB(N),100) .GT. 1) CODE(NODE) = 1.0      SAVD0030
      IF (MOD(NODB(N),10) .EQ. 1) CODE(NODE) = 2.0      SAVD0031
      IF (MOD(NODB(N),100) .EQ. 11) CODE(NODE) = 3.0      SAVD0032
30 CONTINUE      SAVD0033
      DO 40 N = 1, NELEM      SAVD0034
      KODE(N) = NOD(5,N)      SAVD0035
      IF(NOD(2,N).EQ.NOD(3,N)) KODE(N) = 11      SAVD0036
      IF(NOD(7,N).EQ.1) KODE(N) = 11

```

40 CONTINUE

C** SAVE MESH DATA FOR PLOTTING AND AS INPUT DATA FOR HEROIC

* Revise I/O Unit from LUDATA to Unit 14.
 * SCM 11-24-88

* REWIND 14
 * IF(IPLLOT.EQ.0) GO TO 50
 * IF(IPLLOT.EQ.2) GO TO 120

* Revise LUPLLOT from unformatted to formatted.
 * SCM 11-24-88

*****WRITE STATEMENTS PERMANENTLY INCORPORATED (JUNE 78)*****

* WRITE (LUPLLOT) (TITLE(K), K=1,17), NPT, NELEM, NUMMAT, NPUTCK, NINC
 * WRITE (LUPLLOT) (CODE(N), X(N), Y(N), N=1, NPT)
 * WRITE (LUPLLOT) ((NOD(K,N), K=1,4), KODE(N), NOD(6,N), N=1, NELEM)

* WRITE (LUPLLOT, 6010) (TITLE(K), K=1,17)
 * WRITE (LUPLLOT, 6015) NPT, NELEM, NUMMAT, NPUTCK, NBPTC, NINC
 * DO 100 N=1, NPT
 * WRITE (LUPLLOT, 6020) N, X(N), Y(N)
 100 CONTINUE
 * DO 110 N=1, NELEM
 * WRITE (LUPLLOT, 6025) N, (NOD(K,N), K=1,4), KODE(N), NOD(6,N)
 110 CONTINUE
 * IF(IPLLOT.LT.2) GO TO 50
 120 WRITE (30, 6010) (TITLE(K), K=1,17)
 * WRITE (30, 6030) NINC, NPMAT, NPPT

*****WRITE STATEMENTS PERMANENTLY ELIMINATED (JUNE 78)*****

C WRITE (LUPLLOT) (TITLE(K), K = 1,8), NPT, NELEM, NUMMAT, NPUTCK, SAVD0072
 C 1 FZERO, NINC, IZERO, FZERO, FZERO, TPSCAN, TPSCAN, SAVD0073
 C 2 (IZERO, K=1,12) SAVD0074
 C WRITE (LUPLLOT) (CODE(N), X(N), Y(N), N = 1,NPT), SAVD0075
 C 1 ((NOD(K,N), K=1,4), KODE(N), N=1, NELEM), (IZERO, K=1,4) SAVD0076
 C SAVD0077

50 CONTINUE SAVD0079
 DO 45 N = 1, NELEM SAVD0080
 NOD(6,N) = 1000*NOD(5,N) + 10*NOD(6,N) + NOD(7,N) SAVD0081
45 CONTINUE SAVD0082
 IF(IPLLOT.LT.0) END FILE LUPLLOT SAVD0083

* Revise I/O unit from LUDATA to Unit 14.
 * SCM 11-14-88

* WRITE (14) NPT,NELEM,NBPTC,NINC,NPUTCK
 * WRITE (14) (X(N),Y(N), N=1,NPT),
 1 ((NOD(K,N), K=1,4), NOD(6,N), N=1,NELEM),
 2 (NODB(N), (BIV(K,N), K=1,3), N=1,NBPTC)

* WRITE (6,6005)
 * RETURN

6005 FORMAT (53H0* * * * * MESH DATA HAS BEEN SAVED * * * * *) SAVD0098
 6010 FORMAT(1H',17A4,1H') SAVD0099


```

6015 FORMAT(15,5(1H,,15))
6020 FORMAT(15,2(1H,,E12.4))
6025 FORMAT(15,6(1H,,15))
6030 FORMAT(15,2(1H,,15))
END
* * * * *
CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
*
*   VERSION 1.0, MARCH 1990
*
*   PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
*   THE FEDERAL HIGHWAY ADMINISTRATION
* * * * *
SUBROUTINE SAVG (NODGEN, XYGEN, NUMGNP, NNP, MODEG, XCOORD, YCOORD)
*
*   Uncomment this statement in all program modules for double
*   precision version. SCM 3-10-90
*
*   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
*   DIMENSION NODGEN(3,1), XYGEN(3,1)
*   IF (NNP.EQ.0) GO TO 50
*   NUMGNP = NUMGNP + 1
*   NODGEN(2,NUMGNP) = NNP
*   NODGEN(3,NUMGNP) = MODEG
*   XYGEN(1,NUMGNP) = XCOORD
*   XYGEN(2,NUMGNP) = YCOORD
*   RETURN
50 WRITE (6,6000)
DO 60 NGP = 1,NUMGNP
IF (NODGEN(3,NGP).EQ.-1) WRITE (6,6002) NODGEN(2,NGP),
1 XYGEN(1,NGP), XYGEN(2,NGP)
60 IF (NODGEN(3,NGP).NE.-1) WRITE (6,6004) NODGEN(2,NGP),
1 NODGEN(3,NGP), XYGEN(1,NGP), XYGEN(2,NGP)
RETURN
6000 FORMAT ( 74H11LISTING OF NODES (EXCEPT LAPLACE GENERATED NODES) AS
1GENERATED FROM INPUT/ 65H0NODE GENERATION
2 X-COORDINATE Y-COORDINATE/31H NUMBER MODE
6002 FORMAT ( 16, 24X, E19.3, E16.3)
6004 FORMAT ( 2H I,14, I11, 13X, E19.3, E16.3)
END
* * * * *
CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
*
*   VERSION 1.0, MARCH 1990
*
*   PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
*   THE FEDERAL HIGHWAY ADMINISTRATION
* * * * *
SUBROUTINE SETU(THETA,C1,C2,C3,C4,SQRAL,XP,XQ,U)
C
C   SUBROUTINE FOR EVALUATING CORRECTIONAL SOLUTION MATRIX.
C
*
*   Uncomment this statement in all program modules for double
*   precision version. SCM 3-10-90
*
*   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
*   DIMENSION U(6,6)
*   DO 100 I=1,6
*   DO 100 J=1,6
100 U(I,J) = 0.0
SQ = SIN(XQ*THETA)
CQ = COS(XQ*THETA)
EXPP = EXP(XP*THETA)
EXPN = EXP(-XP*THETA)
SAVD0100
SAVD0101
SAVD0102
SAVD0103
SAVD0104
SAVG0001
SAVG0002
SAVG0003
SAVG0004
SAVG0005
SAVG0006
SAVG0007
SAVG0008
SAVG0009
SAVG0010
SAVG0011
SAVG0012
SAVG0013
SAVG0014
SAVG0015
SAVG0016
SAVG0017
SAVG0018
SAVG0019
SAVG0020
SAVG0021
SAVG0022
SAVG0023
SAVG0024
SAVG0025
SAVG0026
SAVG0027
SAVG0028
SAVG0029
SAVG0030
SAVG0031
SAVG0032
SAVG0033
SAVG0034
SETU0001
SETU0002
SETU0003
SETU0004
SETU0005
SETU0006
SETU0007
SETU0008
SETU0009
SETU0010
SETU0011
SETU0012
SETU0013
SETU0014
SETU0015
SETU0016
SETU0017
SETU0018
SETU0019
SETU0020
SETU0021
SETU0022
SETU0023
SETU0024
U(1,1) = EXPP/SQRAL*(XP*CQ+XQ*SQ)
U(1,2) = EXPP/SQRAL*(-XQ*CQ+XP*SQ)
U(1,3) = EXPN/SQRAL*(-XP*CQ+XQ*SQ)
U(1,4) = EXPN/SQRAL*(-XQ*CQ-XP*SQ)
U(2,1) = EXPP*CQ
U(2,2) = EXPP*SQ
U(2,3) = EXPN*CQ
U(2,4) = EXPN*SQ
U(3,1) = EXPP*(XP*CQ-XQ*SQ)
U(3,2) = EXPP*(XQ*CQ+XP*SQ)
U(3,3) = EXPN*(-XP*CQ-XQ*SQ)
U(3,4) = EXPN*(XQ*CQ-XP*SQ)
U(4,1) = EXPP*(-0.5*CQ-C3*SQ)
U(4,2) = EXPP*(C3*CQ-0.5*SQ)
U(4,3) = EXPN*(-0.5*CQ-C3*SQ)
U(4,4) = EXPN*(-C3*CQ-0.5*SQ)
U(5,1) = EXPP*(-XP*C1*CQ-XQ*C2*SQ)
U(5,2) = EXPP*(XQ*C2*CQ-XP*C1*SQ)
U(5,3) = EXPN*(XP*C1*CQ-XQ*C2*SQ)
U(5,4) = EXPN*(XQ*C2*CQ+XP*C1*SQ)
U(6,1) = EXPP*(C4*CQ+C3*SQ)
U(6,2) = EXPP*(-C3*CQ+C4*SQ)
U(6,3) = EXPN*(C4*CQ-C3*SQ)
U(6,4) = EXPN*(C3*CQ+C4*SQ)
U(5,5) = THETA
U(5,6) = -1.0
U(6,5) = THETA
U(6,6) = -1.0
RETURN
END
* * * * *
CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
*
*   VERSION 1.0, MARCH 1990
*
*   PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
*   THE FEDERAL HIGHWAY ADMINISTRATION
* * * * *
SUBROUTINE SLPJNT
C
C   NEW ADDITION SUBROUTINE IN 1983 FOR JOINT SLIPPING BEHAVIOR
C
C   SUBROUTINE SLPJNT(ICOME,LEVEL,NPMAT,NPPT,PDIA,PIPMAT,PE,
*PNU,PA,IA,JCON,ICON)
C
C   THIS ROUTINE CONTROLS THE PROCESSING OF SLIPPING JOINTS
C   FOR BOTH LEVEL 1 AND LEVEL 2,3 SOLUTIONS.
C   THERE ARE THREE ENTRY POINTS INTO THIS ROUTINE
C   CALLED FROM THREE PLACES IN SUBROUTINE STEEL.
C   ENTRY POINTS ARE CONTROLLED BY THE VARIABLE ICOME AS FOLLOWS:
C   ICOME=1 , READ INPUT DATA FOR JOINTS
C   ICOME=2 , COMPUTE CURRENT AXIAL YOUNGS MODULUS 'PENEN'
C   AND STORE IN PIPMAT AS EFFECTIVE AREA
C   (ITERATE TO CONVERGENCE).
C   ICOME=3 , UPDATE CONVERGED SOLUTIONS AND PRINT RESULTS.
C   THE ONLY OTHER COMMUNICATION TO THIS SUBROUTINE IS THROUGH
C   COMMON/XJOINT/ WHICH PROVIDES CURRENT AXIAL STRAIN INCREMENTS FROM
C   SUBROUTINE BEAMST(LEVEL 2.3) OR SUBROUTINE BURNS (LEVEL 1).
C
*
*   Uncomment this statement in all program modules for double
*   precision version. SCM 3-10-90
*
*   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
*   Expand pipe dimension statements from 30 to 99.
SLPJ0001
SLPJ0002
SLPJ0003
SLPJ0004
SLPJ0005
SLPJ0006
SLPJ0007
SLPJ0008
SLPJ0009
SLPJ0010
SLPJ0011
SLPJ0012
SLPJ0013
SLPJ0014
SLPJ0015
SLPJ0016
SLPJ0017
SLPJ0018
SLPJ0019
SLPJ0020
SLPJ0021
SLPJ0022
SLPJ0023
SLPJ0024
SLPJ0025
SLPJ0026
SLPJ0027
SLPJ0028
SLPJ0029
SLPJ0030
SLPJ0031
SLPJ0032
SLPJ0033

```



```

      SCM 5-22-89
      DIMENSION JLOC(15), PIPMAT(5,99), PRSTRN(15), JFLAG(15), PSTRES(15),
      * STRES1(15), XJC(15), PEOLDX(15), STRN1(15), SLOTLR(15), SLOTL1(15)
      Expand array dimension statements from 30 to 99.
      SCM 5-22-89
      COMMON/XJOINT/DSTRN(99), XLONG(99), JOINT
      COMMON/LIMITS/PSLIP, SSTRN, PFAIL
      DATA XPSLIP, XPFail, XRPES, XRPEF, XRPEF, XSLOTL/4950., 33000., .000001,
      * 0.5, .000001, 1.0/
      DATA TOLER, ITMAX, NJMAX/.01, 40, 15/
C
C STATEMENT FUNCTION FOR RELATING JOINT RESPONSES
C TO EFFECTIVE RESPONSES
C
      FACTOR(PEJ)=PEJ/PE*(1.-XJR)+XJR
C
C GO TO PROPER ENTRY POSITION
C
      GO TO (100,200,400), ICOME
C
C FIRST ENTRY, READ JOINT DATA, SET DEFAULTS AND CONSTANTS
C
100 READ(5,1000) PSLIP, PFAIL, RPES, RPEP, RPEF, SLOTL, NUMJ, JSLOTLR
      IF(LEVEL.NE.1) READ(5,1100) (JLOC(N), N=1, NUMJ)
      IF(LEVEL.NE.1 .AND. JSLOTLR.NE.0) READ(5,1200) (SLOTLR(N), N=1, NUMJ)
C
C SET DEFAULT VALUES FOR INPUT VARIABLES, CHECK DATA
C
      IF(PSLIP.EQ.0.0) PSLIP=XPSLIP
      IF(PFAIL.EQ.0.0) PFAIL=XPFail
      IF(RPES.LE.0.0) RPES=XRPEF
      IF(RPEP.LE.0.0) RPEP=XRPEF
      IF(RPEF.LE.0.0) RPEF=XRPEF
      IF(SLOTL.EQ.0.0) SLOTL=XSLOTL
      IF(NUMJ.EQ.0) NUMJ=1
      IF(NUMJ.GT.NJMAX) WRITE(6,2500)
      DO 105 NJ=1, NUMJ
      IF(LEVEL.EQ.1) GO TO 105
      IF(JLOC(NJ).GT.0 .OR. JLOC(NJ).LE.NPMAT) GO TO 105
      WRITE(6,2000) NJ, JLOC(NJ)
105 CONTINUE
C
C COMPUTE CONSTANTS FOR MODULI VALUES AND STRAIN AT END OF INITIAL
C ELASTIC RANGE
C
      PES=RPES*PE
      PEP=RPEP*PE
      PEF=RPEF*PE
      ESTRN=PSLIP/(PE/(1.-PNU**2))
      SSTRN=1.0
      PCSLIP=PSLIP+(1.-ESTRN)*PES/(1.-PNU**2)
      IF(LEVEL.EQ.1) XJR=SLOTL*FLOAT(NUMJ)/(PDIA*3.1415926535898)
C
C INITIALIZE PREVIOUS VALUES
C
      PENEW=PE
      DO 110 NJ=1, NUMJ
      PEOLDX(NJ)=PE
      SLPJ0034
      SLPJ0035
      SLPJ0036
      SLPJ0037
      SLPJ0038
      SLPJ0039
      SLPJ0040
      SLPJ0041
      SLPJ0042
      SLPJ0043
      SLPJ0044
      SLPJ0045
      SLPJ0046
      SLPJ0047
      SLPJ0048
      SLPJ0049
      SLPJ0050
      SLPJ0051
      SLPJ0052
      SLPJ0053
      SLPJ0054
      SLPJ0055
      SLPJ0056
      SLPJ0057
      SLPJ0058
      SLPJ0059
      SLPJ0060
      SLPJ0061
      SLPJ0062
      SLPJ0063
      SLPJ0064
      SLPJ0065
      SLPJ0066
      SLPJ0067
      SLPJ0068
      SLPJ0069
      SLPJ0070
      SLPJ0071
      SLPJ0072
      SLPJ0073
      SLPJ0074
      SLPJ0075
      SLPJ0076
      SLPJ0077
      SLPJ0078
      SLPJ0079
      SLPJ0080
      SLPJ0081
      SLPJ0082
      SLPJ0083
      SLPJ0084
      SLPJ0085
      SLPJ0086
      SLPJ0087
      SLPJ0088
      SLPJ0089
      SLPJ0090
      SLPJ0091
      SLPJ0092
      SLPJ0093
      SLPJ0094
      SLPJ0095
      SLPJ0096
      PSTRES(NJ)=0.0
      PRSTRN(NJ)=0.0
110
C WRITE OUT JOINT DATA TO BE USED
C
      WRITE(6,2100) PSLIP, PFAIL, PES, PEP, PEF, SLOTL, NUMJ
      DO 150 NJ=1, NUMJ
      IF(JSLOTLR.EQ.0) SLOTLR(NJ)=1.0
      SLOTL1(NJ)=SLOTL*SLOTLR(NJ)
150 CONTINUE
      IF(LEVEL.GT.1) WRITE(6,2200) (NJ, JLOC(NJ), SLOTL1(NJ), NJ=1, NUMJ)
      IF(PFAIL.LE.PCSLIP) WRITE(6,2050)
      IF(PFAIL.LT.PCSLIP) STOP
      RETURN
C
C SECOND ENTRY, COMPUTE CURRENT AXIAL MODULUS AND STORE IN PIPMAT
C AS EFFECTIVE AREA. THIS ENTRY WILL BE REPEATED UNTIL CONVERGENCE
C OCCURS.
C
200 JCON=JCON+1
C
C COMPUTATIONS FOR LEVEL 1 SOLUTION (ENTRY 2)
C
      IF(LEVEL.NE.1) GO TO 300
      PEOLD=PENEW
      STRNPR=PRSTRN(1)
      DSTRNJ=DSTRN(1)/FACTOR(PEOLD)
      STRN=STRNPR+DSTRNJ
      STRESP=PSTRES(1)
      STRES=STRESP+DSTRNJ*PEOLD/(1.-PNU**2)
      CALL JMOD(STRN, STRNPR, PE, PES, PEP, PEF, STRES, STRESP, PEOLD, PENEW, PNU
      *, IFLAG, JCON, 1)
      STRN1(1)=STRN
      STRES1(1)=STRES
      IF(JOINT.EQ.2 .AND. JCON.EQ.1) WRITE(6,2600) IA, PSLIP, SSTRN, PFAIL
C
C CHECK ON CONVERGENCE, NOTE: CONVERGENCE WITH JCON=-1
C
      ERR=ABS((PENEW-PEOLD)/PEOLD)
      IF(ERR.LT.TOLER) JCON=-1
      IF(JCON.GT.ITMAX) JCON=-2
      IF(JOINT.EQ.2) WRITE(6,2650) STRN, STRES, STRNPR, STRESP, PEOLD,
      * PENEW, IFLAG, JCON, ICON
C
C UPDATE CURRENT EFFECTIVE AREA IN PIPMAT MATRIX AND RETURN
C
      DO 250 N=1, NPMAT
      PESTAR=PENEW/FACTOR(PENEW)
250 PIPMAT(3,N)=PESTAR*PA/PE
      RETURN
C
C COMPUTATIONS FOR LEVEL 2 OR 3 SOLUTIONS (ENTRY 2)
C
300 ERRSUM=0.0
      IF(JOINT.EQ.2 .AND. JCON.EQ.1) WRITE(6,2700) IA, PSLIP, SSTRN, PFAIL
      DO 350 NJ=1, NUMJ
      PEOLD=PEOLDX(NJ)
      STRNPR=PRSTRN(NJ)
      MNJ=JLOC(NJ)
      XJR=SLOTL1(NJ)/XLONG(MNJ)
      DSTRNJ=DSTRN(MNJ)/FACTOR(PEOLD)
      STRN=STRNPR+DSTRNJ
      STRESP=PSTRES(NJ)
      SLPJ0097
      SLPJ0098
      SLPJ0099
      SLPJ0100
      SLPJ0101
      SLPJ0102
      SLPJ0103
      SLPJ0104
      SLPJ0105
      SLPJ0106
      SLPJ0107
      SLPJ0108
      SLPJ0109
      SLPJ0110
      SLPJ0111
      SLPJ0112
      SLPJ0113
      SLPJ0114
      SLPJ0115
      SLPJ0116
      SLPJ0117
      SLPJ0118
      SLPJ0119
      SLPJ0120
      SLPJ0121
      SLPJ0122
      SLPJ0123
      SLPJ0124
      SLPJ0125
      SLPJ0126
      SLPJ0127
      SLPJ0128
      SLPJ0129
      SLPJ0130
      SLPJ0131
      SLPJ0132
      SLPJ0133
      SLPJ0134
      SLPJ0135
      SLPJ0136
      SLPJ0137
      SLPJ0138
      SLPJ0139
      SLPJ0140
      SLPJ0141
      SLPJ0142
      SLPJ0143
      SLPJ0144
      SLPJ0145
      SLPJ0146
      SLPJ0147
      SLPJ0148
      SLPJ0149
      SLPJ0150
      SLPJ0151
      SLPJ0152
      SLPJ0153
      SLPJ0154
      SLPJ0155
      SLPJ0156
      SLPJ0157
      SLPJ0158
      SLPJ0159

```

```

      STRES=STRESF+DSTRNJ*PEOLD/(1.-PNU**2)
      CALL JMOD(STRN,STRNPR,PE,PES,PEP,PEF,STRES,STRESF,PEOLD,PENEW,PNU
      *,IFLAG,JCON,NJ)
      STRN1(NJ)=STRN
      STRES1(NJ)=STRES
      IF(JOINT.EQ.2) WRITE(6,2750)NJ,STRN,STRES,STRNPR,STRESF,PEOLD
      *,PENEW,IFLAG,JCON,ICON
      IF(JOINT.EQ.2 .AND. NJ.EQ.NUMJ) WRITE(6,2800)
C
C      ACCUMULATE ERROR SUM OF ALL JOINTS
C
      ERR=ABS((PENEW-PEOLD)/PEOLD)
      ERRSUM=ERRSUM+ERR
C
C      UPDATE CURRENT EFFECTIVE AREA IN PIPMAT MATRIX
C      AND SAVE IFLAG NUMBER
C
      PEOLDX(NJ)=PENEW
      PESTAR=PENEW/FACTOR(PENEW)
      PIPMAT(3,MNJ)=PESTAR*PA/PE
      JFLAG(NJ)=IFLAG
350  CONTINUE
C
C      CHECK FOR CONVERGENCE AND RETURN TO STEEL SUBROUTINE
C      JCON.GT.0 , NO CONVERGENCE YET, ENTRY 2 WILL BE REPEATED
C      JCON=-1 , CONVERGED, ENTRY 3 WILL BE CALLED
C      JCON=-2 , PROGRAM WILL STOP , NO CONVERGENCE
C
      AVER=ERRSUM/FLOAT(NUMJ)
      IF(AVER.LT.TOLER)JCON=-1
      IF(JCON.GT.ITMAX)JCON=-2
      RETURN
C
C      THIRD ENTRY, UPDATE JOINT AXIAL STRAIN VECTOR
C      AND PRINT RESULTS ABOUT JOINT BEHAVIOR
C
400  IF(LEVEL.GT.1)GO TO 500
      PRSTRN(1)=STRN1(1)
      APRSTR=ABS(PRSTRN(1))
      PSTRES(1)=STRES1(1)
      XJC(1)=(APRSTR-ESTRN)/(SSTRN-ESTRN)
      IF(XJC(1).LT.0.0) XJC(1)=0.0
      IF(XJC(1).GT.1.0) XJC(1)=1.0
      WRITE(6,2300) IA,PRSTRN(1),PSTRES(1),XJC(1),IFLAG
      RETURN
C
C      UPDATE AND OUTPUT FOR LEVEL 2,3
C
500  DO 550 NJ=1,NUMJ
      MNJ=JLOC(NJ)
      PENEW=PEOLDX(NJ)
      XJR=SLOTL1(NJ)/XLONG(MNJ)
      DSTRNJ=DSTRN(MNJ)/FACTOR(PENEW)
      PRSTRN(NJ)=STRN1(NJ)
      APRSTR=ABS(PRSTRN(NJ))
      PSTRES(NJ)=STRES1(NJ)
      XJC(NJ)=(APRSTR-ESTRN)/(SSTRN-ESTRN)
      IF(XJC(NJ).LT.0.0) XJC(NJ)=0.0
      IF(XJC(NJ).GT.1.0) XJC(NJ)=1.0
550  CONTINUE
      WRITE(6,2400) IA,(NJ,PRSTRN(NJ),PSTRES(NJ),XJC(NJ),JFLAG(NJ),
      *,NJ=1,NUMJ)
      RETURN

```

```

SLPJ0160
SLPJ0161
SLPJ0162
SLPJ0163
SLPJ0164
SLPJ0165
SLPJ0166
SLPJ0167
SLPJ0168
SLPJ0169
SLPJ0170
SLPJ0171
SLPJ0172
SLPJ0173
SLPJ0174
SLPJ0175
SLPJ0176
SLPJ0177
SLPJ0178
SLPJ0179
SLPJ0180
SLPJ0181
SLPJ0182
SLPJ0183
SLPJ0184
SLPJ0185
SLPJ0186
SLPJ0187
SLPJ0188
SLPJ0189
SLPJ0190
SLPJ0191
SLPJ0192
SLPJ0193
SLPJ0194
SLPJ0195
SLPJ0196
SLPJ0197
SLPJ0198
SLPJ0199
SLPJ0200
SLPJ0201
SLPJ0202
SLPJ0203
SLPJ0204
SLPJ0205
SLPJ0206
SLPJ0207
SLPJ0208
SLPJ0209
SLPJ0210
SLPJ0211
SLPJ0212
SLPJ0213
SLPJ0214
SLPJ0215
SLPJ0216
SLPJ0217
SLPJ0218
SLPJ0219
SLPJ0220
SLPJ0221
SLPJ0222
C
C      READ FORMAT
C
1000  FORMAT(6F10.0,2I5)
1100  FORMAT(15I4)
1200  FORMAT(15F4.0)
C
C      WRITE FORMAT
C
2000  FORMAT(///28H ***FATAL ERROR IN DATA *** ,/,16H JOINT NUMBER,15,
      * 18H JOINT LOCATION IS ,15,/)
2050  FORMAT(///28H ***FATAL ERROR IN DATA *** ,/,
      * 39H PFAIL IS TOO SMALL OR PES IS TOO LARGE)
2100  FORMAT(1H1,///,10X,27H SLOTTED JOINTS PROPERTIES,///
      * 10X,27H SLIPPING STRESS.....,E15.5,/,
      * 10X,27H FAILURE STRESS.....,E15.5,/,
      * 10X,27H SLIPPING-JOINT MODULUS.....,E15.5,/,
      * 10X,27H POSTSLIPPING-JOINT MODULUS,E15.5,/,
      * 10X,27H JOINT FAILURE MODULUS ....,E15.5,/,
      * 10X,27H JOINT SLOT LENGTH.....,E15.5,/,
      * 10X,27H NUMBER OF JOINTS.....,15,/)
2200  FORMAT(///,13H JOINT NUMBER,10X,14HJOINT LOCATION,10X,
      * 13H JOINT LENGTH,/,
      * (6X,15,18X,15,13X,E13.4,/)
2300  FORMAT(1H1,///,41H SLOTTED JOINTS RESPONSES AT LOAD STEP =,15,///
      * 5X,7H STRAIN,5X,7H STRESS,5X,20H JOINT CLOSURE RATIO,
      * 5X,6H IFLAG,/,2E13.3,10X,E13.3,5X,15)
2400  FORMAT(1H1,///,41H SLIPPING JOINTS RESPONSES AT LOAD STEP =,15,///
      *5X,8HJOINT NO,5X,7H STRAIN,5X,7H STRESS,5X,19HJOINT CLOSURE RATIO,
      *5X,6H IFLAG,/, (8X,15,2E13.3,9X,E13.3,5X,15))
2500  FORMAT(///,18H ***FATAL ERROR***,/,
      * 43H NUMBER OF JOINTS .GT. MAX.NUMBER OF JOINTS)
2600  FORMAT(1H1,///,6X,41H SLOTTED JOINT ITERATIONS AT LOAD STEP =,15,
      * //,6X,7H PSLIP=,E13.3,/,6X,7H SSTRN=,E13.3,/,6X,
      * 7H PFAIL=,E13.3,///,6X,7H STRN .6X,7H STRESS,6X,7H STRNPR,6X,
      * 7H STRESF,6X,7H PEOLD .6X,7H PENEW .4X,6H IFLAG,4X,5H JCON,4X,
      * 5H ICON,/)
2650  FORMAT(6E13.5,3(4X,15))
2700  FORMAT(1H1,///,6X,41H SLOTTED JOINT ITERATIONS AT LOAD STEP =,15,
      * //,6X,7H PSLIP=,E13.3,/,6X,7H SSTRN=,E13.3,/,6X,
      * 7H PFAIL=,E13.3,///,2X,9H JOINT NO,6X,7H STRN .6X,7H STRESS,6X,
      * 7H STRNPR,6X,7H STRESF,6X,7H PEOLD .6X,7H PENEW .4X,6H IFLAG,4X,
      * 5H JCON,4X,5H ICON,/)
2750  FORMAT(6X,15,6E13.5,3(4X,15))
2800  FORMAT(//)
C
C      * * * * *
C      END
C      * * * * *
C      CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
C      *
C      VERSION 1.0, MARCH 1990
C      *
C      PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
C      *
C      THE FEDERAL HIGHWAY ADMINISTRATION
C      * * * * *
C      SUBROUTINE STEEL(IA,ICOME,IEXIT,LEVEL,NINC,NPMAT,NPPT,PDIA,PIPMAT,
C      1 RESULT,SK,SM,XMODE)
C
C      SUBROUTINE FOR CORRUGATED STEEL CULVERTS.
C
C      THE FOLLOWING ADDITIONS TO THE * RESULT * ARRAY ARE CALCULATED
C      IN THIS SUBROUTINE.
C
C      RESULT(10,N) = BENDING STRESSES IN OUTER FIBERS.
C      RESULT(11,N) = THRUST STRESS (AVERAGE OVER SECTION)

```

```

SLPJ0223
SLPJ0224
SLPJ0225
SLPJ0226
SLPJ0227
SLPJ0228
SLPJ0229
SLPJ0230
SLPJ0231
SLPJ0232
SLPJ0233
SLPJ0234
SLPJ0235
SLPJ0236
SLPJ0237
SLPJ0238
SLPJ0239
SLPJ0240
SLPJ0241
SLPJ0242
SLPJ0243
SLPJ0244
SLPJ0245
SLPJ0246
SLPJ0247
SLPJ0248
SLPJ0249
SLPJ0250
SLPJ0251
SLPJ0252
SLPJ0253
SLPJ0254
SLPJ0255
SLPJ0256
SLPJ0257
SLPJ0258
SLPJ0259
SLPJ0260
SLPJ0261
SLPJ0262
SLPJ0263
SLPJ0264
SLPJ0265
SLPJ0266
SLPJ0267
SLPJ0268
SLPJ0269
STEL0001
STEL0002
STEL0003
STEL0004
STEL0005
STEL0006
STEL0007
STEL0008
STEL0009
STEL0010
STEL0011
STEL0012
STEL0013
STEL0014
STEL0015
STEL0016

```

```

C      RESULT(12,N) = SHEAR STRESS (AVERAGE OVER SECTION)
C
C      Uncomment this statement in all program modules for double
C      precision version. SCM 3-10-90
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C      Revise dimensions of PIPMAT and RESULT arrays from 30 to 99.
C      SCM 4-30-89
C
C      DIMENSION CPLIST(10),FF(2),FS(3),PALIST(35),PFS(3),PHINGE(5),
C      1 PILIST(35),PIPMAT(5,99),PSLIST(35),RESULT(20,99)
C
C      NEW ADDITION COMMON STATEMENT IN 1983 FOR JOINT SLIPPING BEHAVIOR
C
C      Expand array dimension statements from 30 to 99.
C      SCM 5-22-89
C
C      COMMON/XJOINT/DSTRN(99),XLONG(99),JOINT
C
C      DATA E,XNU,YIELD/30000000.,0.3,33000./
C      DATA NFS,FS/3,3.,4.,2./
C      DATA FE,FIMIN,GAGE20/0.0433,0.02,0.05,0.038/
C      DATA ANALYS,DESIGN/4HANAL,4HDESI/
C      DATA CPLIST,BXB,NUMCP,NUMGA/ 4H 3/2,4H 2,4H 8/3,4H 3,4H 6,
C      1 4H1/4, 2*4H1/2,4H1,4H2,3H X,5,7/
C      DATA PALIST/
C      X 0.03800,0.05066,0.06341,0.07916,0.11091,0.14266,0.17441,
C      X 0.04075,0.05433,0.06791,0.08491,0.11900,0.15316,0.18741,
C      X 0.03875,0.05158,0.06458,0.08066,0.11300,0.14533,0.17775,
C      X 0.04450,0.05925,0.07416,0.09275,0.13000,0.16733,0.20483,
C      X 0.12966,0.16691,0.20408,0.22825,0.26658,0.30483,0.34325/
C      DATA PILIST/
C      X 0.00025,0.00034,0.00044,0.00056,0.00085,0.00120,0.00163,
C      X 0.00144,0.00153,0.00194,0.00245,0.00354,0.00471,0.00599,
C      X 0.00112,0.00150,0.00189,0.00239,0.00342,0.00453,0.0057,
C      X 0.00515,0.00689,0.00865,0.01088,0.01545,0.02017,0.02508,
C      X 0.06041,0.07816,0.09616,0.10800,0.12691,0.14616,0.16583/
C      DATA PSLIST/
C      X 0.00177,0.00230,0.00283,0.00349,0.00483,0.00627,0.00787,
C      X 0.00427,0.00560,0.00693,0.00854,0.01171,0.01485,0.01805,
C      X 0.00419,0.00549,0.00676,0.00831,0.01133,0.01428,0.01724,
C      X 0.00995,0.01315,0.01634,0.02025,0.02798,0.03557,0.04308,
C      X 0.05741,0.07325,0.08883,0.09891,0.11466,0.13016,0.14575/
C
C      GO TO (100,200),ICOME
C
C      FIRST ENTRY ESTABLISHES PIPE PROPERTIES, AND SAFETY FACTORS
C
C      'JOINT' IS A NEW ADDITION TO THE NEXT READ STATEMENT IN 1983
C      AS AN OPTION TO USE THE SLIPPING JOINT BEHAVIOR OR NOT.
C
C      100 READ(5,1000) NONLIN,PDIA,PE,PNU,PYIELD,PDEN,PE2,JOINT
C      IF(PE.LE.0.0) PE = E
C      IF(PNU.LE.0.0) PNU = XNU
C      IF(PYIELD.LE.0.0) PYIELD = YIELD
C      IF(NONLIN.NE.1.AND.NONLIN.NE.2) NONLIN = 0
C      FIMIN = PDIA**2/(PE*PF(1))
C      IF(PIMIN.GT.FIMIN) PIMIN = PIMIN*FF(1)/FF(2)
C
C      STEL0017
C      STEL0018
C      STEL0019
C      STEL0020
C      STEL0021
C      STEL0022
C      STEL0023
C      STEL0024
C      STEL0025
C      STEL0026
C      STEL0027
C      STEL0028
C      STEL0029
C      STEL0030
C      STEL0031
C      STEL0032
C      STEL0033
C      STEL0034
C      STEL0035
C      STEL0036
C      STEL0037
C      STEL0038
C      STEL0039
C      STEL0040
C      STEL0041
C      STEL0042
C      STEL0043
C      STEL0044
C      STEL0045
C      STEL0046
C      STEL0047
C      STEL0048
C      STEL0049
C      STEL0050
C      STEL0051
C      STEL0052
C      STEL0053
C      STEL0054
C      STEL0055
C      STEL0056
C      STEL0057
C      STEL0058
C      STEL0059
C      STEL0060
C      STEL0061
C      STEL0062
C      STEL0063
C      STEL0064
C      STEL0065
C      STEL0066
C      STEL0067
C      STEL0068
C      STEL0069
C      STEL0070
C      STEL0071
C      STEL0072
C      STEL0073
C      STEL0074
C      STEL0075
C      STEL0076
C      STEL0077
C      STEL0078
C      STEL0079
C
C      WRITE(6,2000) PDIA,PE,PNU,PYIELD,PDEN,NONLIN
C      IF(NONLIN.EQ.2) WRITE(6,2001) PE2
C      PHINGE(1) = 0.0
C      IF(XMODE.EQ.DESIGN) GO TO 110
C      READ(5,1010) PA,PI,PS
C      WRITE(6,2010) PA,PI,PS
C      GO TO 130
C
C      110 READ(5,1010) (PFS(I),I=1,NFS)
C      DO 120 I=1,NFS
C      IF(PFS(I).GT.0.0) GO TO 120
C      PFS(I) = FS(I)
C      120 CONTINUE
C      WRITE(6,2100) (PFS(I),I=1,NFS)
C
C      PAMIN = GAGE20
C      PA = PAMIN
C      PI = PIMIN
C      NOTE...SECTION MODULUS FOR STANDARD CORRUGATIONS HAS THE RELATION,
C      PS = PI/(1.465*SQRT(PI/PA) + 0.375*PA)
C      ITER = 0
C
C      INITIALIZE PIPE PROPERTIES ALL AROUND PIPE TO BE THE SAME
C
C      NEW ADDITION STATEMENTS IN 1983 FOR JOINT SLIPPING BEHAVIOR.
C      INPUT JOINT PROPERTIES IF JOINT .GT. 0
C
C      130 IF(JOINT.EQ.0) GO TO 135
C      CALL SLIPJNT(ICOME,LEVEL,NPMAT,NPPT,PDIA,PIPMAT,PE,PNU,
C      *PA,IA,JCON,ICON)
C      DO 140 N=1,NPMAT
C      PIPMAT(1,N) = PE
C      PIPMAT(2,N) = PNU
C      PIPMAT(3,N) = PA
C      PIPMAT(4,N) = PI
C      PIPMAT(5,N) = PDEN*PA
C      140
C      DO 145 N=1,NPPT
C      RESULT(14,N) = PA
C      RESULT(15,N) = PI
C      RESULT(16,N) = PI/PS
C      ICON = 0
C      JCON = 0
C      PESUM = PE
C      RETURN
C      145
C
C      SECOND ENTRY DETERMINES MAXIMUM STRUCTURAL RESPONSES FOR
C      EVALUATING THE PIPE IN TERMS OF FAILURE CRITERIA. IN THE DESIGN
C      MODE, THE PROGRAM ITERATES TO FIND ADEQUATE SECTION PROPERTIES.
C
C      IEXIT = 0
C
C      IF(NONLIN.EQ.1.AND. IA.EQ.NINC) CALL HINGE(PIPMAT,NPMAT,PS,PDIA,
C      X PYIELD,SM,SK,RESULT,NPPT,PHINGE)
C
C      IF NONLIN = 2, UPDATE NONLINEAR BEAM PROPERTIES (EFFECTIVE BENDING
C      AND AXIAL STIFFNESS). CONVERGENCE IS SIGNALLED BY SUBROUTINE EMOD
C      RETURNING ICON=-1. ITERATION LIMIT SETS ICON=-2.
C
C      IF(NONLIN.NE.2) GO TO 201
C
C      STEL0080
C      STEL0081
C      STEL0082
C      STEL0083
C      STEL0084
C      STEL0085
C      STEL0086
C      STEL0087
C      STEL0088
C      STEL0089
C      STEL0090
C      STEL0091
C      STEL0092
C      STEL0093
C      STEL0094
C      STEL0095
C      STEL0096
C      STEL0097
C      STEL0098
C      STEL0099
C      STEL0100
C      STEL0101
C      STEL0102
C      STEL0103
C      STEL0104
C      STEL0105
C      STEL0106
C      STEL0107
C      STEL0108
C      STEL0109
C      STEL0110
C      STEL0111
C      STEL0112
C      STEL0113
C      STEL0114
C      STEL0115
C      STEL0116
C      STEL0117
C      STEL0118
C      STEL0119
C      STEL0120
C      STEL0121
C      STEL0122
C      STEL0123
C      STEL0124
C      STEL0125
C      STEL0126
C      STEL0127
C      STEL0128
C      STEL0129
C      STEL0130
C      STEL0131
C      STEL0132
C      STEL0133
C      STEL0134
C      STEL0135
C      STEL0136
C      STEL0137
C      STEL0138
C      STEL0139
C      STEL0140
C      STEL0141
C      STEL0142

```

```

      ICON = ICON + 1
      CALL EMOD(IA,ICON,NPMAT,NPPT,PA,PE,PE2,PESUM,PI,PIPMAT,
1      PNU,PS,PYIELD,RESULT,ICOME)
C
C      NEW ADDITION STATEMENT IN 1983 FOR JOINT SLIPPING BEHAVIOR.
C      MODIFY AXIAL STIFFNESS TO SIMULATE SLIPPING JOINTS,THIS OVER-RIDES
C      MODIFICATIONS BY EMOD.CONVERGENCE IS SIGNED BY SUBROUTINE SLPJNT
C      RETURNING JCON=-1 . ITERATION LIMIT SETS JCON=-2
C
201 IF(JOINT.EQ.0) GO TO 208
      CALL SLPJNT(ICOME,LEVEL,NPMAT,NPPT,PDIA,PIPMAT,PE,PNU,
      *PA,IA,JCON,ICON)
      IF(JCON.EQ.-2) WRITE (6,2999)JCON,ICON
C
C      CHECK FOR CONVERGENCE OF BOTH ICON AND JCON. CONVERGENCE DOES
C      NOT OCCUR IF ICON OR JCON .GT. 0
C
208 IF(ICON.LE.0 .AND. JCON.LE.0) GO TO 203
      IF(ICON.EQ.-1) ICON=0
      IEXIT = -1
      RETURN
C
C      CHECK IF NONLINEAR SOIL MODEL HAS CONVERGED ...
C
203 IF(NINC.GT.0) GO TO 204
      IEXIT = -1
      RETURN
C
C      EVALUATE CURRENT MAXIMUM STRESSES, DISPLACEMENTS, AND PRESSURES.
C
204 TMAX = 0.0
      BMAX = 0.0
      PAVG = 0.0
      UYMAX = RESULT(4,1)
      UYMIN = RESULT(4,1)
C
      DO 210 N=1,NPPT
        RESULT(10,N) = -ABS(RESULT(5,N)/PS)
        RESULT(11,N) = RESULT(6,N)/PA
        RESULT(12,N) = RESULT(7,N)/PA
        IF(NONLIN.EQ.0) GO TO 205
        IF(NONLIN.EQ.2) GO TO 202
        IF(ABS(RESULT(10,N)).GT.PYIELD) RESULT(10,N) = -PYIELD
        GO TO 205
202 IF(RESULT(18,N).LT.1.0) GO TO 205
        SIGMAX=PYIELD*(1.0+PE2*(RESULT(18,N)-1.0)/PE)
        RESULT(10,N) = -(SIGMAX-ABS(RESULT(11,N)))
205 IF(ABS(RESULT(10,N)).GT.BMAX) BMAX = ABS(RESULT(10,N))
        IF(NONLIN.EQ.2) GO TO 206
        IF(SIGMAX.GT.BMAX .AND. RESULT(18,N).GT.1.0) BMAX=SIGMAX
206 CONTINUE
        IF(ABS(RESULT(11,N)).GT.TMAX) TMAX = ABS(RESULT(11,N))
        IF(RESULT(4,N).GT.UYMAX) UYMAX = RESULT(4,N)
        IF(RESULT(4,N).LT.UYMIN) UYMIN = RESULT(4,N)
        PAVG = PAVG + RESULT(8,N)
210 CONTINUE
        PAVG = ABS(PAVG/NPPT)
        DISP = UYMAX-UYMIN
        IF (SM.EQ. 0.0) SM = 1.0
        PBUCK = PDIA**3/(36.*PESUM*SM*(1.-SK))*(1.-PNU**2)
        PCRIT = SQRT(PI/PBUCK)
        IF(NONLIN.EQ.1 .AND. PHINGE(1).LT.0.0) GO TO 300
        IF(XMODE.EQ.ANALYS) GO TO 300
      STEL0143 C
      STEL0144 C
      STEL0145 C
      STEL0146 C
      STEL0147 C
      STEL0148 C
      STEL0149 C
      STEL0150 C
      STEL0151 C
      STEL0152 C
      STEL0153 C
      STEL0154 C
      STEL0155 C
      STEL0156 C
      STEL0157 C
      STEL0158 C
      STEL0159 C
      STEL0160 C
      STEL0161 C
      STEL0162 C
      STEL0163 C
      STEL0164 C
      STEL0165 *
      STEL0166 *
      STEL0167 *
      STEL0168 *
      STEL0169 *
      STEL0170 C
      STEL0171 C
      STEL0172 C
      STEL0173 C
      STEL0174 C
      STEL0175 C
      STEL0176 C
      STEL0177 C
      STEL0178 C
      STEL0179 C
      STEL0180 C
      STEL0181 C
      STEL0182 C
      STEL0183 C
      STEL0184 C
      STEL0185 C
      STEL0186 C
      STEL0187 C
      STEL0188 C
      STEL0189 C
      STEL0190 C
      STEL0191 C
      STEL0192 C
      STEL0193 C
      STEL0194 C
      STEL0195 C
      STEL0196 C
      STEL0197 C
      STEL0198 C
      STEL0199 C
      STEL0200 C
      STEL0201 C
      STEL0202 C
      STEL0203 C
      STEL0204 C
      STEL0205 C
      C
      C      DESIGN... REVISE SECTION PROPERTIES AS REQUIRED
      C
      C      ICON = 0
      C      IF(IA.LT.NINC) RETURN
      C      ITER = ITER + 1
      C      ICHKA = 0
      C      ICHKI = 0
      C
      C      UPDATE THRUST AREA AS REQUIRED BY MAX THRUST.
      C
      C      PANEW = PFS(1)*TMAX*PA/PYIELD
      C      IF(PANEW.LT.PAMIN) PANEW = PAMIN
      C      IF(ABS((PANEW-PA)/PANEW).LT.0.05) ICHKA = 1
      C      PA = PANEW
      C
      C      UPDATE MOMENT OF INERTIA AS REQUIRED BY ... (1) DEFLECTION LIMIT
      C      AS A PERCENT OF DIAMETER, OR (2) ELASTIC BUCKLING AS GIVEN BY
      C      CHELAPATI, HIGHWAY RESEARCH RECORD, NO. 413, 1972.
      C
      C      PINewD = PFS(2)*DISP*PI/(0.2*PDIA)
      C      PINewB = (PFS(3)*PAVG)**2*PBUCK
      C
      C      *
      C      *      Revise intrinsic funtion to generic form. SCM 3-10-90
      C      *
      C      PINew = MAX(PINewD,PINewB)
      C
      C      IF(PINew.LT.PIMIN) PINew = PIMIN
      C      IF(ABS((PINew-PI)/PINew).LT.0.05) ICHKI = 1
      C      PI = PINew
      C      PS = PI/(1.465*SQRT(PI/PA)+0.375*PA)
      C
      C      UPDATE SECTION PROPERTIES AND CHECK FOR CONVERGENCE.
      C
      C      IF(ICHKA*ICHKI.EQ.1) GO TO 250
      C      IF(ITER.GT.5) WRITE(6,2200) ITER,PA,PI,PS
      C      IF(ITER.GT.5) GO TO 300
      C      IEXIT = -2
      C      GO TO 130
      C
      C      SEARCH STANDARD CORRUGATION TABLES FOR SUITABLE SECTIONS WHICH
      C      SATISFY THE DESIGN SOLUTION, LIST THEM.
      C
250 WRITE(6,2300) ITER,PA,PI,PS
      SUMMAX = 0.0
      IJBEST = 0
      DO 280 I=1,NUMCF
        NGAGE = 0
        DO 260 J=1,NUMGA
          IJ = J+NUMGA*(I-1)
          CPA = PALIST(IJ)
          CPI = PILIST(IJ)
          IF(CPA.LT.PA .OR. CPI.LT.PI) GO TO 260
          NGAGE = 22-2*J
          IF(I.EQ.5) NGAGE= 14-2*J+J/4
          GO TO 270
260 CONTINUE
          WRITE(6,2400) CPLIST(I),BXB,CPLIST(I+5)
          GO TO 280
270 WRITE(6,2500) CPLIST(I),BXB,CPLIST(I+5),NGAGE,CPA,CPI
          SUMS = PA/CPA + PI/CPI
          IF(SUMS.LT.SUMMAX) GO TO 280
          SUMMAX = SUMS
      STEL0206 C
      STEL0207 C
      STEL0208 C
      STEL0209 C
      STEL0210 C
      STEL0211 C
      STEL0212 C
      STEL0213 C
      STEL0214 C
      STEL0215 C
      STEL0216 C
      STEL0217 C
      STEL0218 C
      STEL0219 C
      STEL0220 C
      STEL0221 C
      STEL0222 C
      STEL0223 C
      STEL0224 C
      STEL0225 C
      STEL0226 C
      STEL0227 C
      STEL0228 C
      STEL0229 C
      STEL0230 C
      STEL0231 C
      STEL0232 C
      STEL0233 C
      STEL0234 C
      STEL0235 C
      STEL0236 C
      STEL0237 C
      STEL0238 C
      STEL0239 C
      STEL0240 C
      STEL0241 C
      STEL0242 C
      STEL0243 C
      STEL0244 C
      STEL0245 C
      STEL0246 C
      STEL0247 C
      STEL0248 C
      STEL0249 C
      STEL0250 C
      STEL0251 C
      STEL0252 C
      STEL0253 C
      STEL0254 C
      STEL0255 C
      STEL0256 C
      STEL0257 C
      STEL0258 C
      STEL0259 C
      STEL0260 C
      STEL0261 C
      STEL0262 C
      STEL0263 C
      STEL0264 C
      STEL0265 C
      STEL0266 C
      STEL0267 C
      STEL0268 C

```



```

      IJBEST = IJ
      NGBEST = NGAGE
      CONTINUE
C
C      CHOOSE MOST EFFICIENT SECTION FROM LIST AND ANALYSE.
C
      IF(IJBEST.EQ.0) WRITE(6,2600)
      IF(IJBEST.EQ.0) GO TO 300
      PA = PALIST(IJBEST)
      PI = PILIST(IJBEST)
      PS = PSLIST(IJBEST)
      IBEST = (IJBEST-1)/NUMGA+1
      WRITE(6,2700) CPLIST(IBEST),BXH,CPLIST(IBEST+5),NGBEST
      XMODE = ANALYS
      IEXIT = -2
      GO TO 130
C
C      PRINTOUT STRUCTURAL RESULTS AND EVALUATION.
C
C      CONTINUE
C
C      PRINT FINITE ELEMENT FIELD RESPONSES.
C
      IF (LEVEL .EQ. 1) GO TO 330
      ICOME = 3
      CALL PRHERO(IA, ICOME, IEXIT, LEVEL, NINC, NPMAT, NPPT, PDIA, PIPMAT,
1      RESULT, SK, SM)
      ICOME = 2
      CONTINUE
330 ICOME=3
      IF(NONLIN.EQ.2) CALL EMOD(IA, ICON, NPMAT, NPPT, PA, PE, PE2, PESUM, PI,
1      PIPMAT, PNU, PS, PYIELD, RESULT, ICOME)
C
C      NEW ADDITION STATEMENTS IN 1983 FOR JOINT SLIPPING BEHAVIOR
C      CALL TO SLPJNT TO PRINTOUT RESPONSES.
C
      IF(JOINT.NE.0)CALL SLPJNT(ICOME,LEVEL,NPMAT,NPPT,PDIA,
*PIPMAT,PE,PNU,PA,IA,JCON,ICON)
      ICOME=2
      WRITE(6,2800) IA,(RESULT(1,J),RESULT(3,J),RESULT(8,J),RESULT(5,J),
X RESULT(6,J),RESULT(7,J),RESULT(2,J),RESULT(4,J),RESULT(9,J),
X RESULT(10,J),RESULT(11,J),RESULT(12,J),J=1,NPPT)
C
      IF(NONLIN.EQ.2) WRITE(6,2990) IA,(J,RESULT(19,J),RESULT(20,J),
X RESULT(18,J),RESULT(13,J),J=1,NPPT)
C
      IF (TMAX*DISP*PAVG*BMAX .EQ. 0.0) GO TO 350
      CIRCR = PYIELD/TMAX
      DISPR = (0.2*PDIA)/DISP
      BUCKR = PCRT/PAVG
      BENDR = PYIELD/BMAX
      HANDR = PI/PIMIN
      WRITE(6,2900) IA,CIRCR,DISPR,BUCKR,BENDR,HANDR
350 CONTINUE
      IF(NONLIN.NE.1) GO TO 400
      IF(PHINGE(1).LT.0.0 .OR. PHINGE(1).GT.1.0) GO TO 400
      WRITE(6,2950) (PHINGE(I),I=1,5)
400 IEXIT = 1
      IF(IA.LT.NINC) IEXIT= 0
      IF(JCON.EQ.-2) ICON=-2
      IF(ICON.EQ.-3) WRITE(6,2995)
      IF(ICON.EQ.-2) STOP
      ICON = 0

```

```

STEL0269      JCON = 0
STEL0270      RETURN
C
STEL0271      C
STEL0272      C      FORMATS
STEL0273      C
STEL0274      1000 FORMAT(15,6F10.0,15)
STEL0275      1010 FORMAT(3F10.0)
STEL0276      2000 FORMAT(/// 10X,35H PIPE PROPERTIES ARE AS FOLLOWS ... //
STEL0277      X 15X,36H AVERAGE PIPE DIAMETER (IN.) ..... , E15.5 //
STEL0278      X 15X,36H YOUNGS MODULUS OF PIPE (PSI)..... , E15.5 //
STEL0279      X 15X,36H POISSONS RATIO OF PIPE (-) ..... , E15.5 //
STEL0280      X 15X,36H YIELD STRESS OF PIPE (PSI)..... , E15.5 //
STEL0281      X 15X,36H DENSITY OF PIPE (PCI)..... , E15.5 //
STEL0282      X 15X,36H MATERIAL CHARACTER, NONLIN ..... , 15 /
STEL0283      X 17X,25H NONLIN=0, IMPLIES LINEAR /
STEL0284      X 17X,30H NONLIN=1, IMPLIES YIELD-HINGE /
STEL0285      X 17X,34H NONLIN=2, IMPLIES BILINEAR CURVE //
STEL0286      2001 FORMAT( /15X,36H MODULUS OF UPPER BI-CURVE .....E15.5 //)
STEL0287      2010 FORMAT(// 10X, 31H SECTION PROPERTIES OF PIPE ... //
STEL0288      X 15X, 36H THRUST AREA (IN**2/IN) ..... , F10.5 //
STEL0289      X 15X, 36H MOM. OF INERTIA (IN**4/IN) ..... , F10.5 //
STEL0290      X 15X, 36H SECTION MODULUS (IN**3/IN) ..... , F10.5 //
STEL0291      2100 FORMAT(/// 10X,43H DESIRED SAFETY FACTORS FOR PIPE DESIGN ... ///
STEL0292      X 15X,36H THRUST STRESS YIELDING ..... , F10.5 //
STEL0293      X 15X,36H DEFLECTION COLLAPSE,(0.2*D) ..... , F10.5 //
STEL0294      X 15X,36H ELASTIC BUCKLING ..... , F10.5 / )
STEL0295      2200 FORMAT(1H1,/// 10X,50H DESIGN ITERATIVE PROCEDURE DID NOT CONVERGESTEL0358
STEL0296      X AFTER ,13/ 10X,55H ITERATIONS. THE LAST SOLUTION IS PRINTED AND ESTEL0359
STEL0297      XVALUATED, / 10X,37H WHERE THE SECTION PROPERTIES ARE ... //
STEL0298      X 15X,36H THRUST AREA PER UNIT LENGTH ..... , E15.5 //
STEL0299      X 15X,36H MOM. OF INERTIA PER UNIT LENGTH ..... , E15.5 //
STEL0300      X 15X,36H SECTION MODULUS PER UNIT LENGTH ... , E15.5 //)
STEL0301      2300 FORMAT(1H1,///10X,19H DESIGN RESULTS AT ,12,11H ITERATIONS //
STEL0302      X 15X, 35H *** REQUIRED THRUST AREA ..... , F10.5 //
STEL0303      X 15X, 35H *** REQUIRED MOM. OR INERTIA ..... , F10.5 //
STEL0304      X 15X, 35H *** REQUIRED SECTION MODULUS ..... , F10.5 ///
STEL0305      X 10X, 49H THE FOLLOWING STEEL CORRUGATED SECTIONS MEET THE /
STEL0306      X 10X, 37H ABOVE REQUIREMENTS WITH MINIMUM AREA //
STEL0307      X15X,12H CORRUGATION,3X,5H GAGE,3X,12H THRUST AREA,3X,13H MOM. INERSTEL0370
STEL0308      XTIA /)
STEL0309      2400 FORMAT(17X,3A4,3X,22H (NO ADEQUATE SECTION) )
STEL0310      2500 FORMAT(17X,3A4,6X,12,5X,F10.5,5X,F10.5)
STEL0311      2600 FORMAT(/// 10X,37H NO STANDARD CORRUGATION IS ADEQUATE. /
STEL0312      X 10X, 45H AN ANALYSIS OF THE REQUIRED SECTION FOLLOWS. )
STEL0313      2700 FORMAT(/// 10X, 38H THE CORRUGATION WITH MINIMUM AREA AND /
STEL0314      X10X,21H MOMENT OF INERTIA IS,3A4,5H GAGE,13,/
STEL0315      X 10X, 41H AN ANALYSIS OF THIS CORRUGATION FOLLOWS. )
STEL0316      2800 FORMAT(1H1,///10X,32H STRUCTURAL RESPONSES OF CULVERT ,
STEL0317      X 3X, 12H LOAD STEP = ,12 //
STEL0318      X 4X,9H X-COORD.,5X,8H X-DISP.,4X,9H N-PRES.,6X,7H MOMENT,6X,
STEL0319      X 7H THRUST,7X,6H SHEAR /4X,9H Y-COORD.,5X,8H Y-DISP.,5X,8H S-PRES.STEL0382
STEL0320      X ,4X,9H M-STRESS,4X,9H T-STRESS,4X,9H S-STRESS ,
STEL0321      X (/F13.2,5E13.3/F13.2,5E13.3)
STEL0322      2900 FORMAT(1H1,///10X,34H CALCULATED SAFETY FACTORS AT STEP, 13 ///
STEL0323      X15X,50H THRUST STRESS ..... (YIELD-STRESS/MAX-STRESS) =,F10.2//
STEL0324      X15X,50H DISPLACEMENT ..... (DIAMETER*0.2/MAX-DISP.) =,F10.2//
STEL0325      X15X,50H ELASTIC BUCKLING ..... (P-CRITICAL/P-AVERAGE) =,F10.2//STEL0388
STEL0326      X 10X, 20H PERFORMANCE FACTORS ///
STEL0327      X15X,50H MAXIMUM STRESS ..... (YIELD-STRESS/MAX-STRESS) =,F10.2 //STEL0390
STEL0328      X15X,50H HANDLING REQUIREMENT ..... (F*EI/D**2) =,F10.2 //STEL0391
STEL0329      2950 FORMAT(/// 10X, 27H PLASTIC HINGING EVALUATION ///
STEL0330      X 15X,42H RATIO OF LOAD AT INITIAL YIELDING ..... , F10.5 //
STEL0331      X 15X,42H RATIO OF MAXIMUM HINGE PENETRATION .... , F10.5 //

```

```

STEL0332
STEL0333
STEL0334
STEL0335
STEL0336
STEL0337
STEL0338
STEL0339
STEL0340
STEL0341
STEL0342
STEL0343
STEL0344
STEL0345
STEL0346
STEL0347
STEL0348
STEL0349
STEL0350
STEL0351
STEL0352
STEL0353
STEL0354
STEL0355
STEL0356
STEL0357
STEL0358
STEL0359
STEL0360
STEL0361
STEL0362
STEL0363
STEL0364
STEL0365
STEL0366
STEL0367
STEL0368
STEL0369
STEL0370
STEL0371
STEL0372
STEL0373
STEL0374
STEL0375
STEL0376
STEL0377
STEL0378
STEL0379
STEL0380
STEL0381
STEL0382
STEL0383
STEL0384
STEL0385
STEL0386
STEL0387
STEL0388
STEL0389
STEL0390
STEL0391
STEL0392
STEL0393
STEL0394

```

```

X 15X,42H RATIO OF CIRCUMFERENCE IN YIELDING .... , F10.5 // STEL0395
X 15X,42H RATIO OF MAX-DISP. DUE TO YIELDING .... , F10.5 // STEL0396
X 15X,42H RATIO OF MAX-STRAIN/YIELD-STRAIN ..... , F10.5 // STEL0397
2990 FORMAT(1H1,///10X,40H NONLINEAR CHARACTERISTICS, LOAD STEP =,I3//STEL0398
X5X,73H NODE INNER-FIBER OUTER-FIBER STRAIN RATIO STEL0399
X FRACTION OF /21X,57HSTRAIN STRAIN MAX-TO-YIELD STEL0400
X WALL YIELDED // (5X,15,2E17.5,7X,F10.5,7X,F10.5)) STEL0401
2995 FORMAT(1H0,///37(2H *)//73H THE SOLUTION WILL NOT CONTINUE DUE TSTEL0402
XO THE RATIO OF ERROR IN THE MODULUS ///47H IT IS RECOMMENDABLE TSTEL0403
XO USE SMALLER LOAD STEPS ///37(2H *)// STEL0404
2999 FORMAT(///,5X,6H JCON=,I5,/,5X,6H ICON=,I5,/) STEL0405
END STEL0406
* * * * * STFB0001
* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM STFB0002
* VERSION 1.0, MARCH 1990 STFB0003
* PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY STFB0004
* THE FEDERAL HIGHWAY ADMINISTRATION STFB0005
* * * * * STFB0006
SUBROUTINE STFSUB STFB0007
C STEFB0008
C THIS SUBROUTINE FORMS THE ELEMENT MATRIX FOR A TRIANGULAR STEFB0009
C ELEMENT STEFB0010
C STEFB0011
+ Uncomment this statement in all program modules for double STEFB0012
+ precision version. SCM 3-10-90 STEFB0013
+ STEFB0014
+ IMPLICIT DOUBLE PRECISION (A-H,O-Z) STEFB0015
+ STEFB0016
DIMENSION AL(3,3),BT(3,3),CT(12,12),CTP(12,17),GA(3,3),ZT(12) STEFB0017
COMMON /ELEMNT/ AS(3),BS(3),C1(17,17),EST(4,3),RQ(4),RT(3), STEFB0018
1 STORE(17),Z4(12),ZQ(4),ZDUM(3),ZY(17),J1,RO STEFB0019
COMMON /SIZEPR/ BIVD(3),IX,NINC STEFB0020
COMMON /MATERL/ CP(3,3),DEN(10),E(10,10),GNU(10,10),H(10,10), STEFB0021
1 IYPE(10),NON(10) STEFB0022
COMMON /ESTABL/ ITRW(28,2),NETCL(17,2),T(28,2) STEFB0023
CALL GEOM(AA,D1,D2,D3) STEFB0024
DO 155 N=1,3 STEFB0025
BN=BS(N)/12.0 STEFB0026
AN=AS(N)/12.0 STEFB0027
DO 150 I=1,3 STEFB0028
AL(N,I)=2.0*BN *BS(I) STEFB0029
BT(N,I)=AN *BS(I)+AS(I)*BN STEFB0030
150 GA(N,I)=2.0*AN *AS(I) STEFB0031
AL(N,N)=D1/12.0 STEFB0032
BT(N,N)=D2/12.0 STEFB0033
155 GA(N,N)=D3/12.0 STEFB0034
DU=0.5/AA STEFB0035
C11=CP(1,1)*DU STEFB0036
C12=CP(1,2)*DU STEFB0037
C13=CP(1,3)*DU STEFB0038
C22=CP(2,2)*DU STEFB0039
C23=CP(2,3)*DU STEFB0040
C33=CP(3,3)*DU STEFB0041
FY = -RO*AA/8. STEFB0042
C STEFB0043
C CHECK FOR DATA ERROR STEFB0044
C STEFB0045
IF(AA.LE.0.0) WRITE(6,911) IX STEFB0046
911 FORMAT (// 7X,21HDATA ERROR IN ELEMENT , 14//) STEFB0047
C STEFB0048
C FORM ELEMENT STIFFNESS MATRIX STEFB0049
C STEFB0050
DO 250 N=1,3 STEFB0051

```

```

DUI = 1.0 STFB0052
IF(N*J1.LT.2 .OR. N*J1.GT.4) DUI = 2.0 STFB0053
BN=BS(N) STFB0054
AN=AS(N) STFB0055
ZT(N )=0.0 STFB0056
ZT(N+3) = FY*DUI STFB0057
ZT(N+6)=0.0 STFB0058
ZT(N+9) = 0.0 STFB0059
DO 250 I=1,3 STFB0060
D2=-1.0/3.0 STFB0061
D4=AL(I,N) STFB0062
D5=BT(I,N) STFB0063
D6=GA(I,N) STFB0064
BI=BS(I) STFB0065
AI=AS(I) STFB0066
X1=BI*BN STFB0067
X2=BI*AN STFB0068
X3=AI*BN STFB0069
X4=AI*AN STFB0070
X5=X2+X3 STFB0071
CT(N,I)= C11*X1+C13* X5 +C33*X4 STFB0072
CT(N,I+3)= C12*X3+C13*X1+C23*X4+C33*X2 STFB0073
CT(N,I+6)=D2*CT(N,I) STFB0074
CT(N,I+9)=D2*CT(N,I+3) STFB0075
CT(N+3,I+3)= C22*X4+C23* X5 +C33*X1 STFB0076
CT(N+3,I+6)=D2*(C13*X1+C12*X2+ C33*X3+C23*X4) STFB0077
CT(N+3,I+9)=D2*CT(N+3,I+3) STFB0078
CT(N+6,I+6)=C11*D4+2.0*C13*D5+C33*D6 STFB0079
CT(N+6,I+9)=C13*D4 +(C12+C33)*D5+C23*D6 STFB0080
250 CT(N+9,I+9)=C22*D6+2.0*C23*D5+C33*D4 STFB0081
DO 300 N=1,12 STFB0082
DO 300 I=N,12 STFB0083
300 CT(I,N)=CT(N,I) STFB0084
C STFB0085
C SUB-ELEMENT UNKNOWNNS TRANSFORMED TO ELEMENT UNKNOWNNS STFB0086
C STFB0087
KB=1 STFB0088
MC=1 STFB0089
DO 400 K=1,17 STFB0090
KA=NETCL(K,J1) STFB0091
DO 320 MM=1,KA STFB0092
M=ITRW(KB,J1) STFB0093
ZY(K)=ZY(K)+2T(M)*T(KB,J1) STFB0094
320 KB=KB+1 STFB0095
DO 380 L=1,12 STFB0096
CTP(L,K)=0.0 STFB0097
MP=MC STFB0098
DO 380 MM=1,KA STFB0099
M=ITRW(MP,J1) STFB0100
CTP(L,K)=CTP(L,K)+CT(L,M)*T(MP,J1) STFB0101
380 MP=MP+1 STFB0102
400 MC=MC+KA STFB0103
MC=1 STFB0104
DO 410 I=1,17 STFB0105
KA=NETCL(I,J1) STFB0106
DO 405 K=1,17 STFB0107
MP=MC STFB0108
DO 405 MM=1,KA STFB0109
M=ITRW(MP,J1) STFB0110
C1(I,K)=C1(I,K)+CTP(M,K)*T(MP,J1) STFB0111
405 MP=MP+1 STFB0112
410 MC=MC+KA STFB0113
RETURN STFB0114

```


STFN00115
STFN00001
STFN00002
STFN00003
STFN00004
STFN00005
STFN00006
STFN00007
STFN00008
STFN00009
STFN00010
STFN00011
STFN00012
STFN00013
STFN00014
STFN00015
STFN00016
STFN00017
STFN00018
STFN00019
STFN00020
STFN00021
STFN00022
STFN00023
STFN00024
STFN00025
STFN00026
STFN00027
STFN00028
STFN00029
STFN00030
STFN00031
STFN00032
STFN00033
STFN00034
STFN00035
STFN00036
STFN00037
STFN00038
STFN00039
STFN00040
STFN00041
STFN00042
STFN00043
STFN00044
STFN00045
STFN00046
STFN00047
STFN00048
STFN00049
STFN00050
STFN00051
STFN00052
STFN00053
STFN00054
STFN00055
STFN00056
STFN00057
STFN00058
STFN00059
STFN00060
STFN00061
STFN00062

```

J1=J
C
C      NODE COORDINATES OF SUB-ELEMENT ARE FOUND
C
DO 100 K=1,3
RT(K) = RQ(K)
100 ZT(K) = ZQ(K)
IF(JTRY,EQ.1) GO TO 105
IF(J1 ,EQ. 2) GO TO 103
RT(3) = RQ(4)
ZT(3) = ZQ(4)
GO TO 105
103 RT(1) = RQ(4)
ZT(1) = ZQ(4)
105 M=6*J1-7
DO 220 K=1,3
M=M+2
STORE(M) = RT(K)
220 STORE(M+1) = ZT(K)
CALL STFSUB
300 CONTINUE
C
C      THE ELEMENT UNKNOWNNS ARE ELIMINATED
C
IF(JTRY,EQ.1) GO TO 313
DO 312 K=1,9
LL=17-K
KK=18-K
DU3=C1(KK,KK)
DU2=ZY(KK)
DO 312 L=1,LL
DU1=C1(L,KK)/DU3
DO 310 M=1,LL
310 C1(L,M)=C1(L,M)-C1(KK,M)*DU1
312 ZY(L)=ZY(L)-DU1*DU2
C
C      STIFFNESS MATRIX STORED
C
313 STORE(17) = IX
WRITE(12)(ZY(M),STORE(M),(C1(M,N),N=1,17),M=1,17)
1 ((CP(J,K),K=1,3),J=1,3)
C
C      THE BOUNDARY CONDITIONS ARE CONSIDERED
C
DO 334 J=1,4
NRQ = NOD(J,IX)
DO 331 K=1,NBPTC
KK=IABS(NODB(K))
KL=KK/10000
IF (NRQ .NE. KL) GO TO 331
XX=1.0
IF (NODB(K) .LT. 0) XX=0.0
NODB(K)=-KK
NR=NU*(J-1)
IIFLG(1)=MOD(KK,100)/10
IIFLG(2)=MOD(KK,10)
IIFLG(3)=IIFLG(1)+IIFLG(2)
*
* Add extended boundary condition definition. MKG
*
IF (IIFLG(1) .EQ. 2) IIFLG(3) = 0
IF (IIFLG(2) .EQ. 2) IIFLG(3) = 0

```

STFN0063
STFN0064
STFN0065
STFN0066
STFN0067
STFN0068
STFN0069
STFN0070
STFN0071
STFN0072
STFN0073
STFN0074
STFN0075
STFN0076
STFN0077
STFN0078
STFN0079
STFN0080
STFN0081
STFN0082
STFN0083
STFN0084
STFN0085
STFN0086
STFN0087
STFN0088
STFN0089
STFN0090
STFN0091
STFN0092
STFN0093
STFN0094
STFN0095
STFN0096
STFN0097
STFN0098
STFN0099
STFN0100
STFN0101
STFN0102
STFN0103
STFN0104
STFN0105
STFN0106
STFN0107
STFN0108
STFN0109
STFN0110
STFN0111
STFN0112
STFN0113
STFN0114
STFN0115
STFN0116
STFN0117
STFN0118
STFN0119
STFN0120
STFN0121
STFN0122
STFN0123
STFN0124
STFN0125

```

      BIVD(1) = BIV(1,K)
      BIVD(2) = BIV(2,K)
      BIVD(3)=0.0
      IAN=MOD(KK,10000)/100
      ANG = BIV(3,K)
      IF(ANG .EQ. 0.0) GO TO 320
C
C      TRANSFORMATION TO N-S AXES
C
      CA=COS(ANG)
      SA=SIN(ANG)
      D1=ZY(NR+1)
      D2=ZY(NR+2)
      ZY(NR+1)=D1*CA+D2*SA
      ZY(NR+2)=-D1*SA+D2*CA
      DO 315 JJ=1,8
      D1=C1(NR+1,JJ)
      D2=C1(NR+2,JJ)
      C1(NR+1,JJ)=D1*CA+D2*SA
315  C1(NR+2,JJ)=-D1*SA+D2*CA
      DO 318 II=1,8
      D1=C1(II,NR+1)
      D2=C1(II,NR+2)
      C1(II,NR+1)=D1*CA+D2*SA
318  C1(II,NR+2)=-D1*SA+D2*CA
      DO 320 N=1,NU
      DU1=0.0
      IF(IAN .EQ. IA) DU1=BIVD( N)
      NR=NR+1
C
C      LOADS ADDED IN
C
      ZY(NR)=ZY(NR)+XX*DU1
      IF(IIFLG(N) .EQ. 0) GO TO 330
C
C      DISPLACEMENTS SPECIFIED
C
      ZY(NR)=DU1*XX
      C1(NR,NR)=0.0
      DO 327 M=1,8
      ZY(M)=ZY(M)-C1(M,NR)*DU1
      C1(M,NR)=0.0
327  C1(NR,M)=0.0
330  CONTINUE
331  CONTINUE
334  CONTINUE
C
C      THE ELEMENT MATRIX IS NOW ADDED INTO THE SYSTEM MATRIX
C
      NRCC=0
      IJU=LB1*NEQ
      DO 350 K=1,4
      KK = NOD(K,IX)
      NR=NQ(KK)/100-1
      NRM=NR-IJU
      DO 350 M=1,NU
      NRCC=NRCC+1
      NRM=NRM+1
      NRCC=0
      DO 345 L=1,4
      JJ = MOD(L,IX)
      NCN=NQ(JJ)/100-NR-M
      DO 345 N=1,NU
      STFN0126      NRCC=NRCC+1
      STFN0127      NCN=NCN+1
      STFN0128      IF (NCN.LT. 1) GO TO 344
      STFN0129      A(NRM,NCN) = A(NRM,NCN) +C1(NRCC,NRCC)
      STFN0130      344 CONTINUE
      STFN0131      345 CONTINUE
      STFN0132      350 B(NRM) = B(NRM) + ZY(NRCC)
      STFN0133      RETURN
      STFN0134      END
      STFN0135      * * * * *
      STFN0136      * CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
      STFN0137      * VERSION 1.0, MARCH 1990
      STFN0138      * PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
      STFN0139      * THE FEDERAL HIGHWAY ADMINISTRATION
      STFN0140      * * * * *
      STFN0141      SUBROUTINE STRESS (BIV,MNO,NOD,NODB,NQ,ST,STPIPE,U,V,IA,NBPTC,
      STFN0142      1 NELEM,NELEMT,NPMAT,NPPT,NPT,SKAVG,SHAVG)
      STFN0143      *
      STFN0144      * Uncomment this statement in all program modules for double
      STFN0145      * precision version. SCM 3-10-90
      STFN0146      *
      STFN0147      * IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      STFN0148      *
      STFN0149      DIMENSION BIV(3,NBPTC),MNO(NELEM),NOD(4,NELEM),NODB(NBPTC),NQ(NPT)
      STFN0150      1 ,ST(6,NELEM),STPIPE(2,NPPT),U(NPT),V(NPT)
      STFN0151      COMMON /ELEMNT/ AS(3),BS(3),C1(17,17),EST(4,3),RQ(4),RT(3),
      STFN0152      1 STORE(17),Z4(12),ZQ(4),ZT(3),ZY(17),JI,RO
      STFN0153      COMMON /MATERL/ CP(3,3),DEN(10),E(10,10),GNU(10,10),H(10,10),
      STFN0154      1 ITYPE(10),NON(10)
      STFN0155      COMMON/SYSTEM/B(240),A(240,120)
      STFN0156      COMMON /ESTABL/ ITRW(28,2),NETCL(17,2),T(28,2)
      STFN0157      C
      STFN0158      C
      STFN0159      C
      STFN0160      DO 600 K=1,NBPTC
      STFN0161      KK=IABS(NODB(K))
      STFN0162      NODB(K)=KK
      STFN0163      ANG = BIV(3,K)
      STFN0164      IF(ANG .EQ. 0.0) GO TO 600
      STFN0165      K1=KK/10000
      STFN0166      JJ=NQ(K1)/100
      STFN0167      IF(JJ.EQ.NQ(K1+1)/100) GO TO 600
      STFN0168      CT=COS(ANG)
      STFN0169      SA=SIN(ANG)
      STFN0170      D1 = B(JJ)
      STFN0171      D2 = B(JJ+1)
      STFN0172      B(JJ) = D1 *CT -D2*SA
      STFN0173      B(JJ+1) = D1*SA +D2*CT
      STFN0174      600 CONTINUE
      STFN0175      C
      STFN0176      C
      STFN0177      C
      STFN0178      SHAVG = 0.0
      STFN0179      SKAVG = 0.0
      STFN0180      SMKNEI = 0.0
      STFN0181      REWIND 12
      STFN0182      DO 726 III=1,NELEMT
      STFN0183      C
      STFN0184      C
      STFN0185      C
      STFN0186      RECALL ELEMENT STIFFNESS MATRIX
      STFN0187      READ(12) (ZY(M),STORE(M),(C1(M,N),N=1,17),M=1,17),
      STFN0188      1 ((CP(J,K),K=1,3),J=1,3)
      I=STORE(17)+0.1
      STFN0189
      STFN0190
      STFN0191
      STFN0192
      STFN0193
      STFN0194
      STFN0195
      STFN0196
      STFN0197
      STRS0001
      STRS0002
      STRS0003
      STRS0004
      STRS0005
      STRS0006
      STRS0007
      STRS0008
      STRS0009
      STRS0010
      STRS0011
      STRS0012
      STRS0013
      STRS0014
      STRS0015
      STRS0016
      STRS0017
      STRS0018
      STRS0019
      STRS0020
      STRS0021
      STRS0022
      STRS0023
      STRS0024
      STRS0025
      STRS0026
      STRS0027
      STRS0028
      STRS0029
      STRS0030
      STRS0031
      STRS0032
      STRS0033
      STRS0034
      STRS0035
      STRS0036
      STRS0037
      STRS0038
      STRS0039
      STRS0040
      STRS0041
      STRS0042
      STRS0043
      STRS0044
      STRS0045
      STRS0046
      STRS0047
      STRS0048
      STRS0049
      STRS0050
      STRS0051
      STRS0052
      STRS0053
      STRS0054

```

```

MN=MNO(I)/1000
IAC = MOD(MNO(I),1000)/10
MNX = MNO(I)-1000*MN-10*IAC
NOD(1,I) = IABS(NOD(1,I))
C
C CALCULATE ELEMENT UNKNOWNNS
C
L=1
DO 610 J=1,4
K = NOD(J,I)
ND=NQ(K)/100
ZY(L) = B(ND)
ZY(L+1) = B(ND+1)
L= L+2
610
C
C BEAM ELEMENT COMPUTATIONS
C
IF(MNX.GT.0) GO TO 615
IF (NOD(2,I) .NE. NOD(3,I)) GO TO 621
ZY(6) = B(ND+2)
CALL BEAMST(NOD,NQ,I,NELEM,NPT,MN)
RT(1) = RQ(4)
RT(2) = ZQ(4)
RT(3) = 0.0
GO TO 718
C
C INTERFACE ELEMENT RECOVERY OF FORCES AND DISPLACEMENTS
C
615 ND1 = NOD(1,I)
ND2= NOD(2,I)
ND3 = NOD(3,I)
NDF1 = NQ(ND1)/100
NDF2 = NQ(ND2)/100
NDF3 = NQ(ND3)/100
RQ(1) = B(NDF3)
RQ(2) = B(NDF3+1)
RQ(3) = B(NDF3) - ST(3,I)
ZQ(1) = B(NDF3+1) - ST(4,I)
ZQ(2) = B(NDF2)-B(NDF1) - ST(5,I)
ZQ(3) = B(NDF2+1)-B(NDF1+1) - ST(6,I)
RT(1) = 0.0
RT(2) = 0.0
RT(3) = 0.0
GO TO 718
C
621 JTRY=2
IF (NOD(3,I) .NE. NOD(4,I)) GO TO 625
JTRY=1
DO 622 J=9,17
ZY(J)=0.0
GO TO 658
625 DO 630 J=1,8
DO 630 K=9,17
630 ZY(K)=ZY(K)-C1(K,J)*ZY(J)
ZY(9)=ZY(9)/C1(9,9)
DO 650 J=10,17
KA=J-1
DO 645 K=9,KA
645 ZY(J)=ZY(J)-ZY(K)*C1(J,K)
650 ZY(J)=ZY(J)/C1(J,J)
658 DO 660 K=1,4
660 RQ(K) =0.0
C
STRS0055 C CALCULATE SUB-ELEMENT EXPANSION COEFFICIENTS
STRS0056 C
STRS0057 DO 715 J1=1,JTRY
STRS0058 DO 670 K=1,12
STRS0059 670 Z4(K)=0.0
STRS0060 M=6*J1-7
STRS0061 DO 675 K=1,3
STRS0062 M=M+2
STRS0063 RT(K)=STORE(M)
STRS0064 675 ZT(K)=STORE(M+1)
STRS0065 J1P=J1
STRS0066 DU=0.5
STRS0067 IF (JTRY .EQ. 2)GO TO 677
STRS0068 DU=1.0
STRS0069 J1P=2
STRS0070 677 CALL GEOM(AA,D1,D2,D3)
STRS0071 DU=DU/AA
STRS0072 MP=1
STRS0073 DO 680 K=1,17
STRS0074 KA=NETCL(K,J1)
STRS0075 DO 680 MM=1,KA
STRS0076 M=ITRW(MP,J1)
STRS0077 Z4(M)=Z4(M)+ZY(K)*T(MP,J1)
STRS0078 680 MP=MP+1
STRS0079 C
STRS0080 C STRAINS CALCULATED FOR SUBELEMENTS
STRS0081 C
STRS0082 EX=0.
STRS0083 EY=0.0
STRS0084 GAMXY=0.0
STRS0085 DO 701 K=1,3
STRS0086 D1=AS(K)
STRS0087 D2=BS(K)
STRS0088 EX=EX+Z4(K)*D2
STRS0089 EY=EY+Z4(K+3)*D1
STRS0090 701 GAMXY=GAMXY+Z4(K)*D1+Z4(K+3)*D2
STRS0091 RQ(1)=EX*DU+RQ(1)
STRS0092 RQ(2)=EY*DU+RQ(2)
STRS0093 RQ(3)=GAMXY*DU+RQ(3)
STRS0094 DU=(1.0-DU*AA)*0.5/AA
STRS0095 IF(J1 .EQ. 2)GO TO 705
STRS0096 RQ(1)=RQ(1)-DU*BS(1)*(Z4(7)-Z4(8)-Z4(9))
STRS0097 RQ(2)=RQ(2)-DU*AS(1)*(Z4(10)-Z4(11)-Z4(12))
STRS0098 RQ(3)=RQ(3)-DU*(AS(1)*(Z4(7)-Z4(8)-Z4(9)) +BS(1)*(Z4(10)-Z4(11)
STRS0099 1 -Z4(12)))
STRS0100 GO TO 715
STRS0101 705 RQ(1)=RQ(1)-DU *BS(3)*(Z4(9)-Z4(8)-Z4(7))
STRS0102 RQ(2)=RQ(2)-DU *AS(3)*(Z4(12)-Z4(11)-Z4(10))
STRS0103 RQ(3)=RQ(3)-DU *(AS(3)*(Z4(9)-Z4(8)-Z4(7)) +BS(3)*(Z4(12)-Z4(11)
STRS0104 1 -Z4(10)))
STRS0105 715 CONTINUE
STRS0106 C
STRS0107 C INCREMENTAL STRESSES CALCULATED
STRS0108 C
STRS0109 DO 717 L=1,3
STRS0110 DU=0.0
STRS0111 DO 716 J=1,3
STRS0112 716 DU=DU+RQ(J)*CP(L,J)
STRS0113 717 ZQ(L)=DU
STRS0114 C
STRS0115 C COMPUTE AVERAGE SOIL PROPERTIES FOR BUCKLING AND HINGING
STRS0116 C
STRS0117 RT(1) = 0.0
STRS0118 STRS0118
STRS0119 STRS0119
STRS0120 STRS0120
STRS0121 STRS0121
STRS0122 STRS0122
STRS0123 STRS0123
STRS0124 STRS0124
STRS0125 STRS0125
STRS0126 STRS0126
STRS0127 STRS0127
STRS0128 STRS0128
STRS0129 STRS0129
STRS0130 STRS0130
STRS0131 STRS0131
STRS0132 STRS0132
STRS0133 STRS0133
STRS0134 STRS0134
STRS0135 STRS0135
STRS0136 STRS0136
STRS0137 STRS0137
STRS0138 STRS0138
STRS0139 STRS0139
STRS0140 STRS0140
STRS0141 STRS0141
STRS0142 STRS0142
STRS0143 STRS0143
STRS0144 STRS0144
STRS0145 STRS0145
STRS0146 STRS0146
STRS0147 STRS0147
STRS0148 STRS0148
STRS0149 STRS0149
STRS0150 STRS0150
STRS0151 STRS0151
STRS0152 STRS0152
STRS0153 STRS0153
STRS0154 STRS0154
STRS0155 STRS0155
STRS0156 STRS0156
STRS0157 STRS0157
STRS0158 STRS0158
STRS0159 STRS0159
STRS0160 STRS0160
STRS0161 STRS0161
STRS0162 STRS0162
STRS0163 STRS0163
STRS0164 STRS0164
STRS0165 STRS0165
STRS0166 STRS0166
STRS0167 STRS0167
STRS0168 STRS0168
STRS0169 STRS0169
STRS0170 STRS0170
STRS0171 STRS0171
STRS0172 STRS0172
STRS0173 STRS0173
STRS0174 STRS0174
STRS0175 STRS0175
STRS0176 STRS0176
STRS0177 STRS0177
STRS0178 STRS0178
STRS0179 STRS0179
STRS0180 STRS0180

```

```

      RT(2) = 0.0
      RT(3) = 0.0
C
      SMAVG = SMAVG + CP(1,1)
      SKAVG = SKAVG + CP(1,2)/CP(1,1)
      SMKNEI = SMKNEI + 1.
C
      INCREMENTAL STRESSES AND STRAINS ADDED TO TOTAL VALUES
C
718 CONTINUE
      DO 719 J=1,3
        ST(J+3,I) = ST(J+3,I) + ZQ(J)
719 ST(J,I) = ST(J,I) + RQ(J)
C
C** PIPE RESPONSES
C
      IF (MOD(2,I) .NE. MOD(3,I)) GO TO 726
      IBEAM = MNO(I)/1000
      STPIPE(1,IBEAM) = STPIPE(1,IBEAM) + RT(1)
      STPIPE(2,IBEAM) = STPIPE(2,IBEAM) + RT(2)
726 CONTINUE
C
      IF (SMKNEI .EQ. 0.0) GO TO 750
      SMAVG = SMAVG/SMKNEI
      SKAVG = SKAVG/SMKNEI
750 CONTINUE
C
C** INCREMENTAL DISPLACEMENTS ARE ADDED
C
      DO 785 I=1,NPT
        IF (MOD(MQ(I),100) .EQ. 0) GO TO 785
        JJ=MQ(I)/100
        DU = B(JJ)
        DV = B(JJ+1)
        U(I)=U(I)+DU
        V(I)=V(I)+DV
785 CONTINUE
      REWIND 12
      RETURN
      END
+ + + + +
+ CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
+   VERSION 1.0, MARCH 1990
+   PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
+   THE FEDERAL HIGHWAY ADMINISTRATION
+ + + + +
      SUBROUTINE XCAN(LUREAD,IGO)
C
C      MODIFY CAN LUDATA TAPE FOR INTERFACE ELEMENTS AROUND PIPE.
C
C      Uncomment this statement in all program modules for double
C      precision version. SCM 3-10-90
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DIMENSION ISWAS(110,2),TITLE(17),NFX(11),MODX(11),XCX(11),YCX(11)
      ,NBADD(4),INCC(11)
      DATA INCC /4*1,2,2,3,3,3,4/
      DATA NADDB,NBADD / 4, 40, 70, 39, 69 /
      DATA IMORE,ILAST/1H ,1HL/
C
      LUREAD = 10
      IF(IGO.EQ.0) RETURN
      STRS0181
      STRS0182
      STRS0183
      STRS0184
      STRS0185
      STRS0186
      STRS0187
      STRS0188
      STRS0189
      STRS0190
      STRS0191
      STRS0192
      STRS0193
      STRS0194
      STRS0195
      STRS0196
      STRS0197
      STRS0198
      STRS0199
      STRS0200
      STRS0201
      STRS0202
      STRS0203
      STRS0204
      STRS0205
      STRS0206
      STRS0207
      STRS0208
      STRS0209
      STRS0210
      STRS0211
      STRS0212
      STRS0213
      STRS0214
      STRS0215
      STRS0216
      STRS0217
      STRS0218
      STRS0219
      STRS0220
      XCAN0001
      XCAN0002
      XCAN0003
      XCAN0004
      XCAN0005
      XCAN0006
      XCAN0007
      XCAN0008
      XCAN0009
      XCAN0010
      XCAN0011
      XCAN0012
      XCAN0013
      XCAN0014
      XCAN0015
      XCAN0016
      XCAN0017
      XCAN0018
      XCAN0019
      XCAN0020
      XCAN0021
      XCAN0022
      XCAN0023
      LURITE = 11
      REWIND LURITE
      REWIND LUREAD
C
      ESTABLISH IS-WAS LIST FOR NODES TO INCLUDE INTERFACE NODES.
C
      ISWAS(1,1) = 1
      ISWAS(1,2) = 0
      DO 50 N=2,110
        IZONE = 0
        IF(N.GE.38.AND.N.LE.42) IZONE = 1
        IF(N.EQ.49) IZONE = 1
        IF(N.GE.56.AND.N.LE.60) IZONE = 1
        ISWAS(N,1) = ISWAS(N-1,1) + 1 + 2*ISWAS(N-1,2)
        ISWAS(N,2) = IZONE
50
C
      READ, MODIFY AND SAVE CONTROL INPUT
C
      READ (LUREAD,6105) PREP,TITLE,WORD2,NINC,MGENPR,NPUTCK,IPLOT,IWRT,XCAN0042
      X NUMNP,NUMEL,NBPTC
      NUMNP = NUMNP + 22
      NUMEL = NUMEL + 11
      NBPTC = NBPTC + NADDB
      WRITE(LURITE,6105) PREP,TITLE,WORD2,NINC,MGENPR,NPUTCK,IPLOT,IWRT,XCAN0047
      X NUMNP,NUMEL,NBPTC
C
      READ, MODIFY AND SAVE NODAL GENERATION INPUT
C
      N = 1
      READ(LUREAD,6110) LIMIT,NNP,MODE,XCOORD,YCOORD,IZERO,SPACNG,FZERO,XCAN0053
      NX = ISWAS(NNP,1)
C
      CHANGE RELATIVELY ADDRESSED NODES
C
      IF (MODE .LT. 100) GO TO 130
      KRELAD = MODE/100
      IF (KRELAD .NE. 1 .AND. KRELAD .NE. 3) GO TO 120
      *
      * Revise intrinsic function to generic form. SCM 3-10-90
      *
      NPP = INT(XCOORD)
      XCOORD = FLOAT(ISWAS(NPP,1))
      120 IF (KRELAD .NE. 2 .AND. KRELAD .NE. 3) GO TO 130
      *
      * NPP = INT(YCOORD)
      *
      YCOORD = FLOAT(ISWAS(NPP,1))
      130 CONTINUE
      IF(ISWAS(NNP,2).EQ.0) GO TO 150
      NFX(N) = NX
      MODX(N) = MODE
      XCX(N) = XCOORD
      YCX(N) = YCOORD
      N = N+1
      150 WRITE(LURITE,6110) IMORE,NX,MODE,XCOORD,YCOORD,IZERO,SPACNG,FZERO,XCAN0079
      IF(LIMIT.NE.ILAST) GO TO 100
C
      IX = IMORE
      DO 200 K=1,2
        DO 200 N=1,11
          IF(N.EQ.11 .AND. K.EQ.2) IX = ILAST
          NX = NFX(N) + K
          XCAN0024
          XCAN0025
          XCAN0026
          XCAN0027
          XCAN0028
          XCAN0029
          XCAN0030
          XCAN0031
          XCAN0032
          XCAN0033
          XCAN0034
          XCAN0035
          XCAN0036
          XCAN0037
          XCAN0038
          XCAN0039
          XCAN0040
          XCAN0041
          XCAN0042
          XCAN0043
          XCAN0044
          XCAN0045
          XCAN0046
          XCAN0047
          XCAN0048
          XCAN0049
          XCAN0050
          XCAN0051
          XCAN0052
          XCAN0053
          XCAN0054
          XCAN0055
          XCAN0056
          XCAN0057
          XCAN0058
          XCAN0059
          XCAN0060
          XCAN0061
          XCAN0062
          XCAN0063
          XCAN0064
          XCAN0065
          XCAN0066
          XCAN0067
          XCAN0068
          XCAN0069
          XCAN0070
          XCAN0071
          XCAN0072
          XCAN0073
          XCAN0074
          XCAN0075
          XCAN0076
          XCAN0077
          XCAN0078
          XCAN0079
          XCAN0080
          XCAN0081
          XCAN0082
          XCAN0083
          XCAN0084
          XCAN0085
          XCAN0086

```

200	WRITE(LURITE,6110) IX,NX,MODX(N),XCX(N),YCX(N),IZERO,FZERO,FZERO	XCAN0087	REWIND LURITE	XCAN0150
C	CONTINUE	XCAN0088	LUREAD = 11	XCAN0151
C		XCAN0089	RETURN	XCAN0152
C	READ, MODIFY AND SAVE ELEMENT GENERATION INPUT	XCAN0090		XCAN0153
C		XCAN0091	C FORMATS	XCAN0154
	INCRT = 0	XCAN0092		XCAN0155
300	READ(LUREAD,6115) LIMIT,NEL,N1,N2,N3,N4,NUMMAT,INCR,IZERO,IZERO,	XCAN0093	6105 FORMAT (A4,1X,17A4,3X,A4/8I5)	XCAN0156
X	IZERO,IZERO	XCAN0094	6110 FORMAT (A1, I4, I5, 2F10.3, I5, 5X,2F10.3)	XCAN0157
	IF (NEL.EQ. 90 .AND. INCR.EQ. 1) INCRT = 1	XCAN0095	6115 FORMAT (A1, I4, 10I5)	XCAN0158
	IQUAD = 2	XCAN0096	6120 FORMAT (A1, I4, I5, F10.2, I5, 2F10.2, 3I5, 2F10.2)	XCAN0159
	IF(NEL.LE.10) IQUAD = 0	XCAN0097	END	XCAN0160
	NX1 = ISWAS(N1,1) + IQUAD*ISWAS(N1,2)	XCAN0098	* * * * *	XCN20001
	IF(N1.EQ.0) NX1 = 0	XCAN0099	* CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM	XCN20002
	NX2 = ISWAS(N2,1) + IQUAD*ISWAS(N2,2)	XCAN0100	* VERSION 1.0, MARCH 1990	XCN20003
	IF(N2.EQ.0) NX2 = 0	XCAN0101	* PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY	XCN20004
	NX3 = ISWAS(N3,1) + IQUAD*ISWAS(N3,2)	XCAN0102	* THE FEDERAL HIGHWAY ADMINISTRATION	XCN20005
	IF(N3.EQ.0) NX3 = 0	XCAN0103	* * * * *	XCN20006
	NX4 = ISWAS(N4,1) + IQUAD*ISWAS(N4,2)	XCAN0104	SUBROUTINE XCAN2(LUREAD,IGO)	XCN20007
	IF(N4.EQ.0) NX4 = 0	XCAN0105		XCN20008
	WRITE(LURITE,6115) IMORE,NEL,NX1,NX2,NX3,NX4,NUMMAT,INCR,IZERO,	XCAN0106	C MODIFY CAN LUDATA TAPE FOR INTERFACE ELEMENTS ALONG TRENCH FACE	XCN20009
X	IZERO,IZERO,IZERO	XCAN0107	C	XCN20010
	IF(LIMIT.NE.ILAST) GO TO 300	XCAN0108	* Uncomment this statement in all program modules for double	XCN20011
C		XCAN0109	* precision version. SCM 3-10-90	XCN20012
	NEL = NUMEL-11	XCAN0110		XCN20013
	IX = IMORE	XCAN0111	* IMPLICIT DOUBLE PRECISION (A-H,O-Z)	XCN20014
	IONE = 1	XCAN0112		XCN20015
	DO 350 N=1,11	XCAN0113	DIMENSION ISWAS(110,2),TITLE(17),NFX(11),MODX(11),XCX(11),YCX(11)	XCN20016
	IF(N.EQ.11) IX = ILAST	XCAN0114	DIMENSION IADDN(7),INCC(7)	XCN20017
	NEL = NEL + 1	XCAN0115	DIMENSION NELNX(8)	XCN20018
	NX1 = NFX(N)	XCAN0116	DATA NELNX /23,24,25,26,71,78,85,92/	XCN20019
	NX2 = NX1 + 2	XCAN0117	DATA IADDN /51,62,71,82,90,98,106/	XCN20020
	NX3 = NX1 + 1	XCAN0118	DATA INCC /2,3,3,4,3*5/	XCN20021
	INCF = INCC(N)	XCAN0119	DATA IMORE,ILAST/1H ,1HL/	XCN20022
	IF (INCRT.EQ. 1) INCF = 1	XCAN0120		XCN20023
	WRITE(LURITE,6115) IX,NEL,NX1,NX2,NX3,IZERO,N,INCF,IONE,IZERO,	XCAN0121	C LUREAD = 10	XCN20024
X	IZERO,IZERO	XCAN0122	IF(IGO.EQ.0) RETURN	XCN20025
350	CONTINUE	XCAN0123	LURITE = 11	XCN20026
C		XCAN0124	REWIND LURITE	XCN20027
C	READ, MODIFY AND SAVE BOUNDARY CONDION INPUT	XCAN0125	REWIND LUREAD	XCN20028
C		XCAN0126		XCN20029
400	READ(LUREAD,6120) LIMIT,ND1,ICX,FZERO,ICY,FZERO,FZERO,INC,ND2,	XCAN0127	C ESTABLISH IS-WAS LIST FOR NODES TO INCLUDE INTERFACE NODES.	XCN20030
X	NNPINC,PLOAD,PLOAD	XCAN0128	C	XCN20031
	NX1 = ISWAS(ND1,1)	XCAN0129	ISWAS(1,1) = 1	XCN20032
	NX2 = 0	XCAN0130	ISWAS(1,2) = 0	XCN20033
	IF(ND2.GT.0) NX2= ISWAS(ND2,1)	XCAN0131	DO 50 N=2,110	XCN20034
	WRITE(LURITE,6120) IMORE,NX1,ICX,FZERO,ICY,FZERO,FZERO,INC,NX2,	XCAN0132	IZONE = 0	XCN20035
X	NNPINC,PLOAD,PLOAD	XCAN0133	DO 20 K = 1, 7	XCN20036
	IF(LIMIT.NE.ILAST) GO TO 400	XCAN0134	20 IF (IADDN(K).EQ. N) IZONE = 1	XCN20037
C		XCAN0135	ISWAS(N,1) = ISWAS(N-1,1) + 1 + 2*ISWAS(N-1,2)	XCN20038
	IL = IMORE	XCAN0136	ISWAS(N,2) = IZONE	XCN20039
	DO 450 N = 1, NADDB	XCAN0137	WRITE(6,6081) N,ISWAS(N,1),ISWAS(N,2),IZONE	XCN20040
	IF (N.EQ. NADDB) IL = ILAST	XCAN0138	6081 FORMAT (10X,6110)	XCN20041
	NX1 = NBADD(N)	XCAN0139	50 CONTINUE	XCN20042
	ICX = IONE	XCAN0140		XCN20043
	ICY = IZERO	XCAN0141	C READ, MODIFY AND SAVE CONTROL INPUT	XCN20044
	IF (N.GT. 2) ICX = IZERO	XCAN0142	C	XCN20045
	IF (N.GT. 2) ICY = IONE	XCAN0143	READ (LUREAD,6105) PREP,TITLE,WORD2,NINC,MGENPR,NPUTCK,IPL0T,IWRT,	XCN20046
450	WRITE(LURITE,6120) IL,NX1,ICX,FZERO,ICY,FZERO,FZERO,IONE,	XCAN0144	X NUMNP,NUMEL,NBPTC	XCN20047
1	IZERO,IZERO,FZERO,FZERO	XCAN0145	NUMNP = NUMNP + 14	XCN20048
C		XCAN0146	NUMEL = NUMEL + 7	XCN20049
C	REDEFINE INPUT TAPE UNIT	XCAN0147	NBPTC = NBPTC + 2	XCN20050
C		XCAN0148	WRITE(LURITE,6105) PREP,TITLE,WORD2,NINC,MGENPR,NPUTCK,IPL0T,IWRT,	XCN20051
	REWIND LUREAD	XCAN0149	X NUMNP,NUMEL,NBPTC	XCN20052


```

C      READ, MODIFY AND SAVE NODAL GENERATION INPUT
C
C      N = 1
100  READ(LUREAD,6110) LIMIT,NNP,MODE,XCOORD,YCOORD,IZERO,SPACNG,FZERO
      WRITE(6,6110) LIMIT,NNP,MODE,XCOORD,YCOORD,IZERO,SPACNG,FZERO
      WRITE(6,6081) ISWAS(NNP,1),ISWAS(NNP,2)
      NX = ISWAS(NNP,1)
C
C      CHANGE RELATIVELY ADDRESSED NODES
C
C      IF (MODE .LT. 100) GO TO 130
      KRELAD = MODE/100
      IF (KRELAD .NE. 1 .AND. KRELAD .NE. 3) GO TO 120
+
+      Revise intrinsic function to generic form.
+
+      NPP = INT(XCOORD)
+
+      XCOORD = FLOAT(ISWAS(NPP,1))
120  IF (KRELAD .NE. 2 .AND. KRELAD .NE. 3) GO TO 130
+
+      NPP = INT(YCOORD)
+
+      YCOORD = FLOAT(ISWAS(NPP,1))
130  CONTINUE
      IF (ISWAS(NNP,2) .EQ. 0) GO TO 150
      NPX(N) = NX
      MODX(N) = MODE
      XCX(N) = XCOORD
      YCX(N) = YCOORD
      N = N+1
150  WRITE(LURITE,6110) IMORE,NX,MODE,XCOORD,YCOORD,IZERO,SPACNG,FZERO
      IF (LIMIT .NE. ILAST) GO TO 100
C
C      IX = IMORE
      DO 200 K=1,2
      DO 200 N = 1, 7
      IF (N.EQ. 7 .AND. K.EQ.2) IX = ILAST
      NN = IADDN(N)
      NPX(N) = ISWAS(NN,1)
      NX = NPX(N) + K
      MODX(N) = 300
      XCX(N) = FLOAT(NX-K)
      YCX(N) = FLOAT(NX-K)
      WRITE(6,6110) IX,NX,MODX(N),XCX(N),YCX(N)
      WRITE(LURITE,6110) IX,NX,MODX(N),XCX(N),YCX(N),IZERO,FZERO,FZERO
200  CONTINUE
C
C      READ, MODIFY AND SAVE ELEMENT GENERATION INPUT
C
C      INCRT = 0
300  READ(LUREAD,6115) LIMIT,NEL,N1,N2,N3,N4,NUMMAT,INCR,IZERO,IZERO,
      X IZERO,IZERO
      IF (NEL .EQ. 90 .AND. INCR .EQ. 1) INCRT = 1
      IQUAD = 2
      IF (NEL .LE. 10) IQUAD = 0
      DO 250 NK = 1, 8
      IF (NEL .EQ. NELNK(NK)) IQUAD=0
250  CONTINUE
      NX1 = ISWAS(N1,1) + IQUAD*ISWAS(N1,2)
      IF (N1.EQ.0) NX1 = 0
      NX2 = ISWAS(N2,1) + IQUAD*ISWAS(N2,2)
      IF (N2.EQ.0) NX2 = 0
      NX3 = ISWAS(N3,1) + IQUAD*ISWAS(N3,2)
      IF (N3.EQ.0) NX3 = 0
      NX4 = ISWAS(N4,1) + IQUAD*ISWAS(N4,2)
      IF (N4.EQ.0) NX4 = 0
      WRITE(LURITE,6115) IMORE,NEL,NX1,NX2,NX3,NX4,NUMMAT,INCR,IZERO,
      X IZERO,IZERO,IZERO
      IF (LIMIT .NE. ILAST) GO TO 300
C
C      NEL = NUMEL - 7
      IX = IMORE
      IONE = 1
      DO 350 N = 1, 7
      IF (N.EQ. 7) IX = ILAST
      NEL = NEL + 1
      NX1 = NPX(N)
      NX2 = NX1 + 2
      NX3 = NX1 + 1
      INCF = INCC(N)
      IF (INCR .EQ. 1) INCF = 1
      WRITE(LURITE,6115) IX,NEL,NX1,NX2,NX3,IZERO,N,INCF,IONE,IZERO,
      X IZERO,IZERO
350  CONTINUE
C
C      READ(LUREAD,6120) LIMIT,ND1,ICX,FZERO,ICY,FZERO,FZERO,INC,ND2,
      X NNPINC,FLOAD,FLOAD
      NX1 = ISWAS(ND1,1)
      NX2 = 0
      IF (ND2.GT.0) NX2 = ISWAS(ND2,1)
      IF (LIMIT.EQ.ILAST) IMORE = ILAST
      IF (NX1 .EQ. 121 .AND. NX2 .EQ. 118) NX2 = 120
      WRITE(LURITE,6120) IMORE,NX1,ICX,FZERO,ICY,FZERO,FZERO,INC,
      X NNPINC,FLOAD,FLOAD
      WRITE(6,6120) IMORE,NX1,ICX,FZERO,ICY,FZERO,FZERO,INC,
      1 NX2,NNPINC,FLOAD,FLOAD
      IF (LIMIT.NE.ILAST) GO TO 400
C
C      REDEFINE INPUT TAPE UNIT
C
C      REWIND LUREAD
      REWIND LURITE
      LUREAD = 11
      RETURN
C
C      FORMATS
C
6105 FORMAT (A4,I4,17A4,3X,A4/BI5)
6110 FORMAT (A1,I4,I5,2F10.3,I5,5X,2F10.3)
6115 FORMAT (A1,I4,10I5)
6120 FORMAT (A1,I4,I5,F10.2,I5,2F10.2,3I5,2F10.2)
END
+ * * * * *
+ CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM
+ VERSION 1.0, MARCH 1990
+ PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY
+ THE FEDERAL HIGHWAY ADMINISTRATION
+ * * * * *
+ SUBROUTINE XFACES(MN,IAC,MNX,NEL,ICON,ITER,NELEM,NPT,MNO,NOD,ST,
+ 1 U,V,IA)
+
+ Uncomment this statement in all program modules for double
+ precision version. SCM 3-10-90

```


XFAC0012
XFAC0013
XFAC0014
XFAC0015
XFAC0016
XFAC0017
XFAC0018
XFAC0019
XFAC0020
XFAC0021
XFAC0022
XFAC0023
XFAC0024
XFAC0025
XFAC0026
XFAC0027
XFAC0028
XFAC0029
XFAC0030
XFAC0031
XFAC0032
XFAC0033
XFAC0034
XFAC0035
XFAC0036
XFAC0037
XFAC0038
XFAC0039
XFAC0040
XFAC0041
XFAC0042
XFAC0043
XFAC0044
XFAC0045
XFAC0046
XFAC0047
XFAC0048
XFAC0049
XFAC0050
XFAC0051
XFAC0052
XFAC0053
XFAC0055
XFAC0055
XFAC0056
XFAC0057
XFAC0058
XFAC0059
XFAC0060
XFAC0061
XFAC0062
XFAC0063
XFAC0064
XFAC0065
XFAC0066
XFAC0067
XFAC0068
XFAC0069
XFAC0070
XFAC0071
XFAC0072
XFAC0073
XFAC0074

```

C      EQUILIBRATE DISPLACEMENTS WHEN INTERFACE FIRST ENTERS SYSTEM
C
      IF(ITER.NE.1) GO TO 2
      IF(IAC.NE.1A) GO TO 2
      ND1 = NOD(1,NEL)
      ND2 = NOD(2,NEL)
      IF (U(ND1).EQ.0.0) U(ND1)=U(ND2)
      IF (U(ND2).EQ.0.0) U(ND2)=U(ND1)
      IF (V(ND1).EQ.0.0) V(ND1)=V(ND2)
      IF (V(ND2).EQ.0.0) V(ND2)=V(ND1)
2     CONTINUE
C
      COMPUTE TESTING PARAMETERS
C
      MNXP=MNX
C
      LKCHNG = 0
      RADA = ANGLE(MN)*CON
      C = COS(RADA)
      S = SIN(RADA)
      ND1 = NOD(1,NEL)
      ND2 = NOD(2,NEL)
      PN = ST(1,NEL)
      PS = ST(2,NEL)
      PNPRES = ST(1,NEL)-ST(3,NEL)
      PSPRES = ST(2,NEL)-ST(4,NEL)
      RDUS = -S*ST(5,NEL) + C*ST(6,NEL)
      UN = C*(U(ND2)-U(ND1)) + S*(V(ND2)-V(ND1))
      BREAK = PN - TENSIL(MN)
      RESIST = FCOEF(MN)*ABS(PN)
      APS = ABS(PS)
C
      IF FREE-FREE IN PREVIOUS LOAD STEP, SET ABARN AND ABARS NONZERO.
C
      ABARN = 0.0
      ABARS = 0.0
      RDUN = C*ST(5,NEL) + S*ST(6,NEL)
      UNPRE = UN-RDUN
      IF(UNPRE.LT.1.0E10-9) GO TO 5
      IF(RDUN.GT.-1.0E10-9) GO TO 5
      ABARN = -UNPRE
      ABARS = RDUS*ABARN/RDUN
5     CONTINUE
*
*     Lines added to force a FREE-FREE state.  SCM 10-18-89
*
      IF(XFLAG(NEL).EQ.1) THEN
          MNX=4
          RN=0.0
          RS=0.0
          PN2=-PNPRE
          PS2=-PSPRE
          CHECK=0.0
          GO TO 100
          END IF
C
C      DETERMINE CURRENT STATE OF ELEMENT
C
      GO TO (10,20,20,30),MNX
C
      MNX =1, FIX-FIX ON ENTRY. ANY STATE POSSIBLE ON EXIT

```

XFAC0075
XFAC0076
XFAC0077
XFAC0078
XFAC0079
XFAC0080
XFAC0081
XFAC0082
XFAC0083
XFAC0084
XFAC0085
XFAC0086
XFAC0087
XFAC0088
XFAC0089
XFAC0090
XFAC0091
XFAC0092
XFAC0093
XFAC0094
XFAC0095
XFAC0096
XFAC0097
XFAC0098
XFAC0099
XFAC0100
XFAC0101
XFAC0102
XFAC0103
XFAC0104
XFAC0105
XFAC0106
XFAC0107
XFAC0108
XFAC0109
XFAC0110
XFAC0111
XFAC0112
XFAC0113
XFAC0114
XFAC0115
XFAC0116
XFAC0117
XFAC0118
XFAC0119
XFAC0120
XFAC0121
XFAC0122
XFAC0123
XFAC0124
XFAC0125
XFAC0126
XFAC0127
XFAC0128
XFAC0129
XFAC0130
XFAC0131
XFAC0132
XFAC0133
XFAC0134
XFAC0135
XFAC0136
XFAC0137

```

C
10 IF(BREAK.LE.0.0) GO TO 11
   LKCHNG = 1
   RN = 0.0
   RS = 0.0
   PN2 = -PNPRE
   PS2 = -PSPRE
   MNX = 4
*
* Lines added to force a FREE-FREE state. SCM 10-18-89
*
XFFLAG(NEL) = 1
CHECK = 0.0
GO TO 100
C
11 IF(APS.LE.RESIST) GO TO 12
   LKCHNG = 1
   RN = 1.0
   RS = 0.0
   SIGN = PS/APS
   PN2 = 0.0
   PS2 = SIGN*RESIST - PSPRE
*
* Add relaxation function to enhance convergence. SCM 3-20-90
*
IF(ITER.GT.2) PS2 = 0.5*PS2 + 0.5*ST(4,NEL)
MNX = 2
IF(SIGN.GT.0.0) MNX = 3
GO TO 100
C
12 RN = 1.0
   RS = 1.0
   PN2 = 0.0
   PS2 = 0.0
   GO TO 100
C
C MNX = 2 OR 3, FIX-FREE ON ENTRY, ANY STATE POSSIBLE ON EXIT
C
20 IF(BREAK.LE.0.0) GO TO 21
   LKCHNG = 1
   RN = 0.0
   RS = 0.0
   PN2 = -PNPRE
   PS2 = -PSPRE
   MNX = 4
*
* Lines added to force a FREE-FREE state. SCM 10-18-89
*
XFFLAG(NEL) = 1
CHECK = 0.0
GO TO 100
C
21 SIGN = 1.0
   IF(MNX.EQ.2) SIGN = -1.0
   IF(PCOEF(MN).EQ.0.0) SIGN=0.0
   IF(RDUS*SIGN.LT.0.0) GO TO 22
   RN = 1.0
   RS = 0.0
   PN2 = 0.0
   PS2 = SIGN*RESIST - PSPRE

```

```

XFAC0138 *
XFAC0139 * Add relaxation function to enhance convergence. SCM 3-20-90
XFAC0140 *
XFAC0141 IF(ITER.GT.2) PS2 = 0.5*PS2 + 0.5*ST(4,NEL)
XFAC0142 *
XFAC0143 IF(PS2.EQ.0.0) GO TO 100
XFAC0144 CHECK = ABS((PS2-ST(4,NEL))/PS2)
XFAC0145 IF(CHECK.GT.0.05) LKCHNG = 1
XFAC0146 GO TO 100
C
22 LKCHNG = 1
   RN = 1.0
   RS = 1.0
   PN2 = 0.0
   PS2 = 0.0
   MNX = 1.0
   GO TO 100
C
C MNX = 4, FREE-FREE ON ENTRY, FREE-FREE OR FIX-FIX ON EXIT
C
30 DIS = SQRT(ST(5,NEL)**2 + ST(6,NEL)**2)
   IF(UN/DIS.LT.-0.000001) GO TO 31
   RN = 0.0
   RS = 0.0
   PN2 = -PNPRE
   PS2 = -PSPRE
   GO TO 100
C
31 LKCHNG = 1
   RN = 1.0
   RS = 1.0
   PN2 = 0.0
   PS2 = 0.0
   MNX = 1
C
C CONSTRUCT ELEMENT STIFFNESS (CONSTRAINT) MATRIX
C
100 CONTINUE
   C1(1,5) = -C*RN
   C1(1,6) = S*RS
   C1(2,5) = -S*RN
   C1(2,6) = -C*RS
   C1(3,5) = C*RN
   C1(3,6) = -S*RS
   C1(4,5) = S*RN
   C1(4,6) = C*RS
   DO 150 I=1,6
   DO 150 J=1,6
   C1(J,I) = C1(I,J)
150
C
C COMPUTE INTERFACE LOAD INCREMENTS
C
IF(ITER.NE.1) GO TO 175
   PN2 = 0.0
   PS2 = 0.0
175 PX2 = C*PN2 - S*PS2
   PY2 = S*PN2 + C*PS2
   ZY(1) = PX2
   ZY(2) = PY2
   ZY(3) = -PX2
   ZY(4) = -PY2
   ZY(5) = PN2*(1.-RN) + ABARN*RN
   ZY(6) = PS2*(1.-RS) + ABARS*RS

```

```

XFAC0201
XFAC0202
XFAC0203
XFAC0204
XFAC0205
XFAC0206
XFAC0207
XFAC0208
XFAC0209
XFAC0210
XFAC0211
XFAC0212
XFAC0213
XFAC0214
XFAC0215
XFAC0216
XFAC0217
XFAC0218
XFAC0219
XFAC0220
XFAC0221
XFAC0222
XFAC0223
XFAC0224
XFAC0225
XFAC0226
XFAC0227
XFAC0228
XFAC0229
XFAC0230
XFAC0231
XFAC0232
XFAC0233
XFAC0234
XFAC0235
XFAC0236
XFAC0237
XFAC0238
XFAC0239
XFAC0240
XFAC0241
XFAC0242
XFAC0243
XFAC0244
XFAC0245
XFAC0246
XFAC0247
XFAC0248
XFAC0249
XFAC0250
XFAC0251
XFAC0252
XFAC0253
XFAC0254
XFAC0255
XFAC0256
XFAC0257
XFAC0258
XFAC0259
XFAC0260
XFAC0261
XFAC0262
XFAC0263

```