Auburn University

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

## 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

## LIST OF TABLES

## LISTS OF FIGURES

# 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 vaied 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 $lb/ft^3$) 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 moluli 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, $\nu$. 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, $\sigma_H$, to the vertical stress, $\sigma_V$, using the relationship:

$$\sigma_H = K_0 \sigma_V \tag{2.1}$$

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

$$K_0 = 1 - \sin\phi \tag{2.2}$$

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

$$\nu = \frac{K_0}{1 + K_0} \tag{2.3}$$

For example, for a $\phi$ of 44 degrees, typical values of $K_0$ and $\nu$ 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, $\nu$, 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

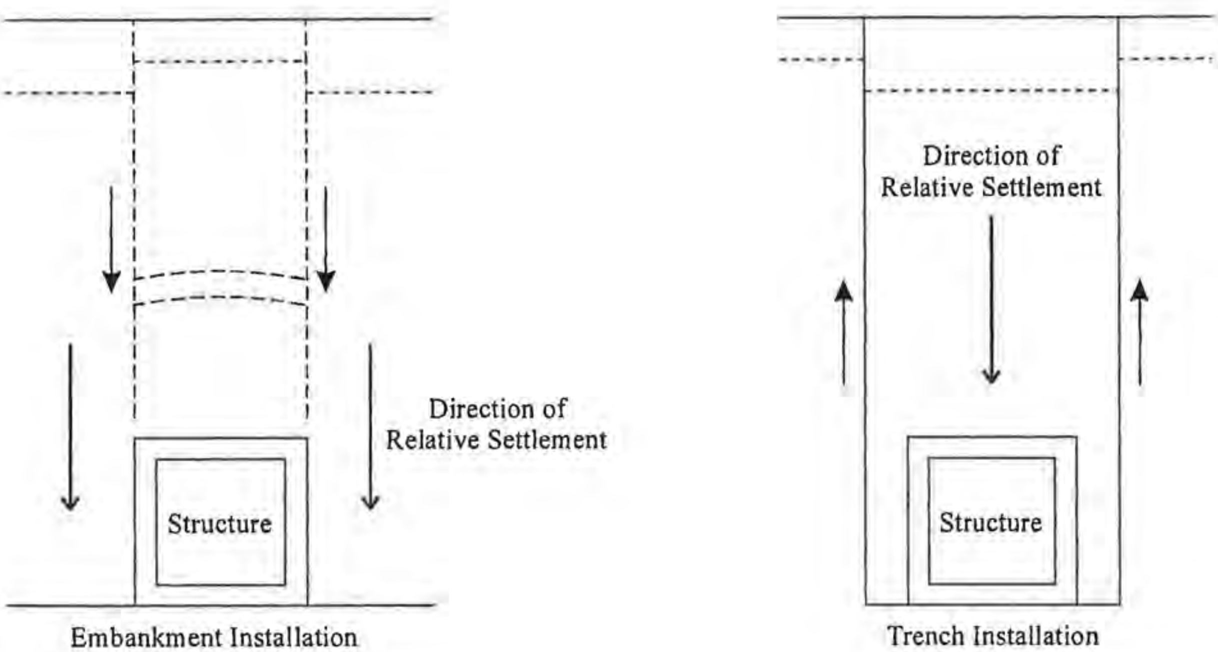



Figure 3.1: Pressure Transfer within a Soil-Structure System

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

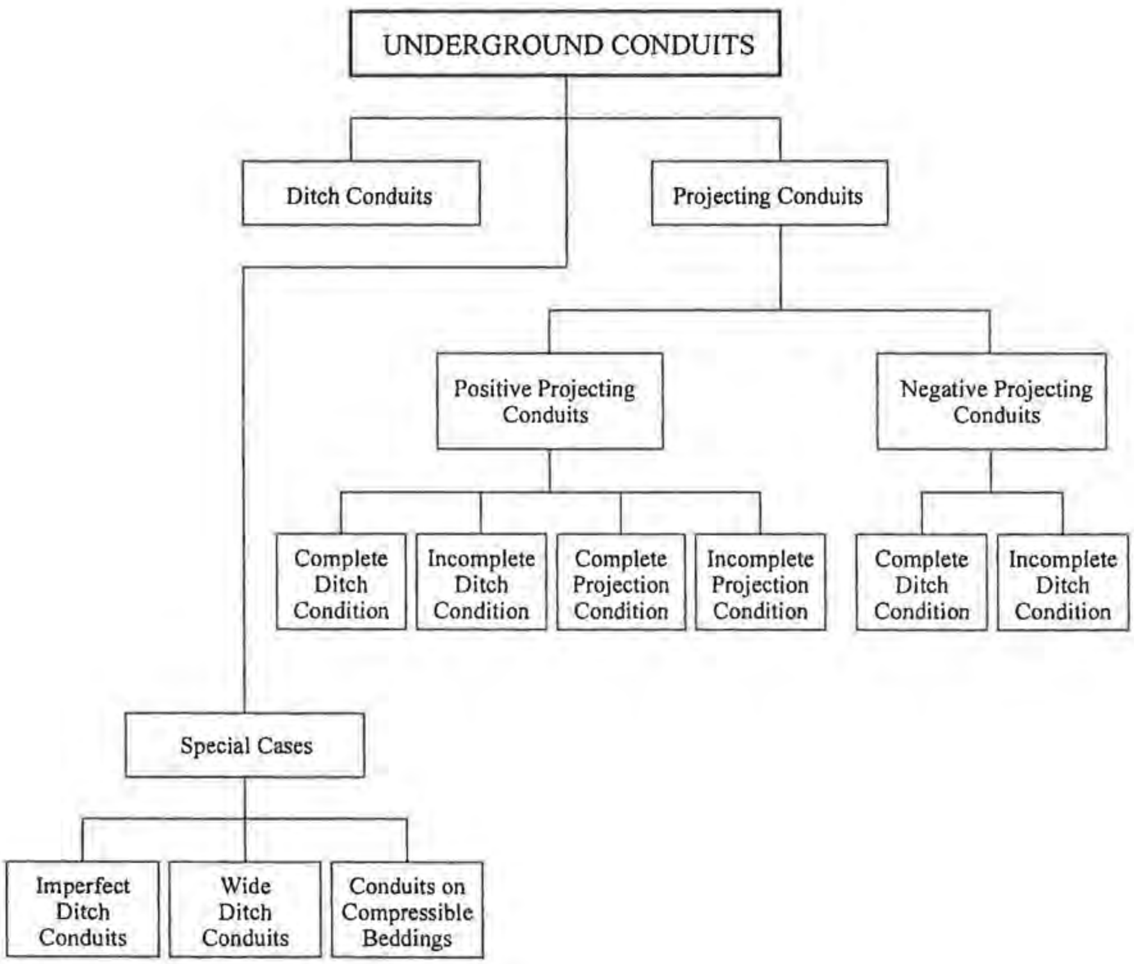



Figure 3.2: Construction Conditions of Underground Conduits[31]

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.

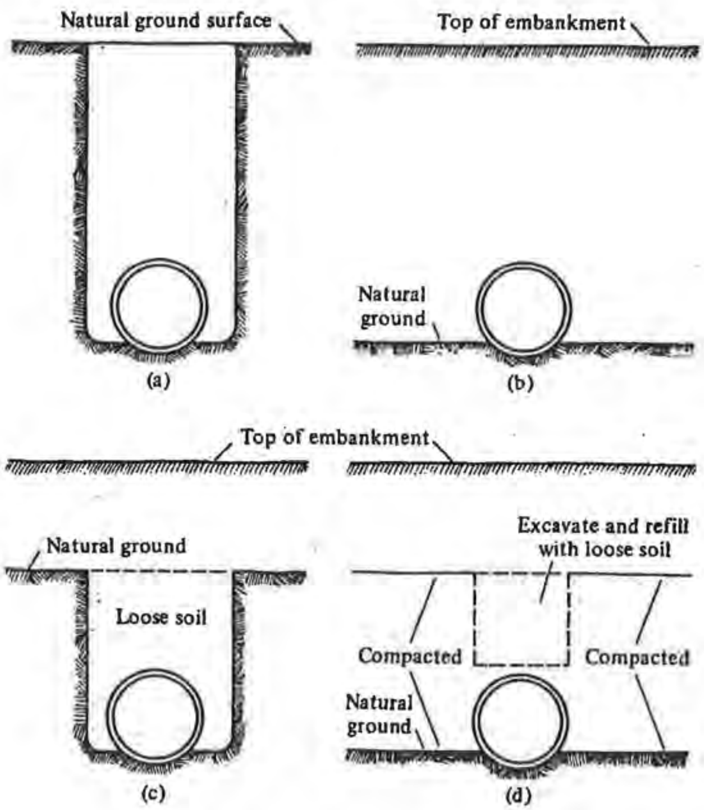



Figure 3.3: Various Classes of Conduit Installations

(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:

$$W = C_d \gamma B_d^2 \quad \textit{for ditch conduits} \tag{3.1}$$

$$W = C_c \gamma B_c^2 \quad \textit{for positive projecting conduits} \tag{3.2}$$

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

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,

$\mu$ = $\tan\phi$ = coefficient of internal friction of fill material,

$\mu'$ = $\tan\phi'$ = 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 | $C_c = 1.93\ H/B_c - 0.17$ | | |

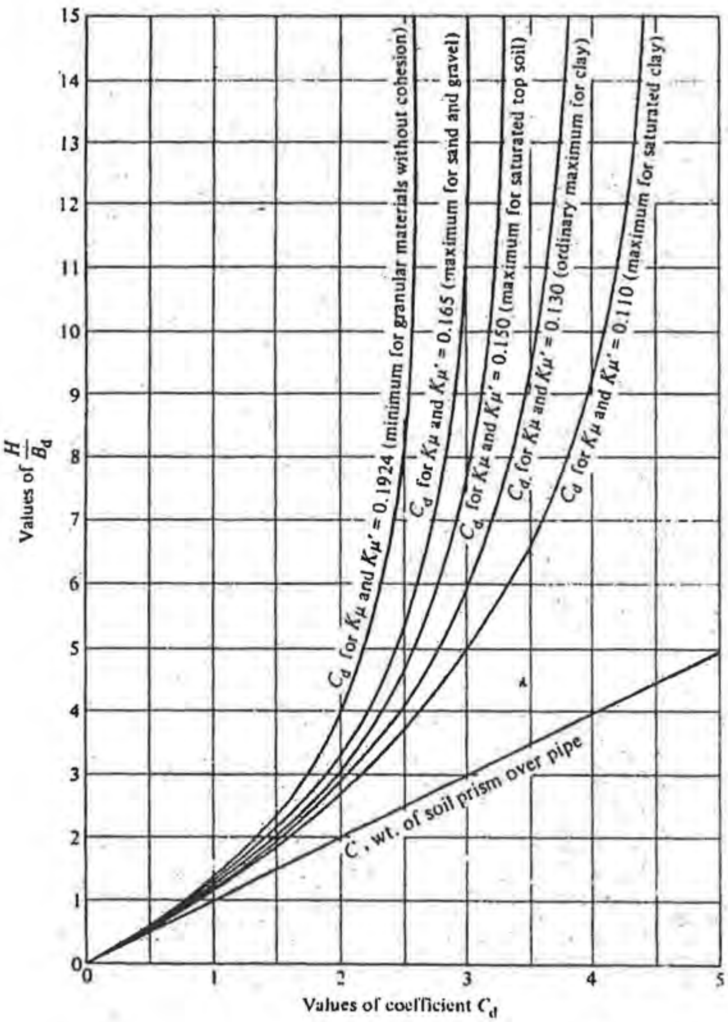



Figure 3.4: Diagram for Coefficient $C_d$ for Ditch Conduits

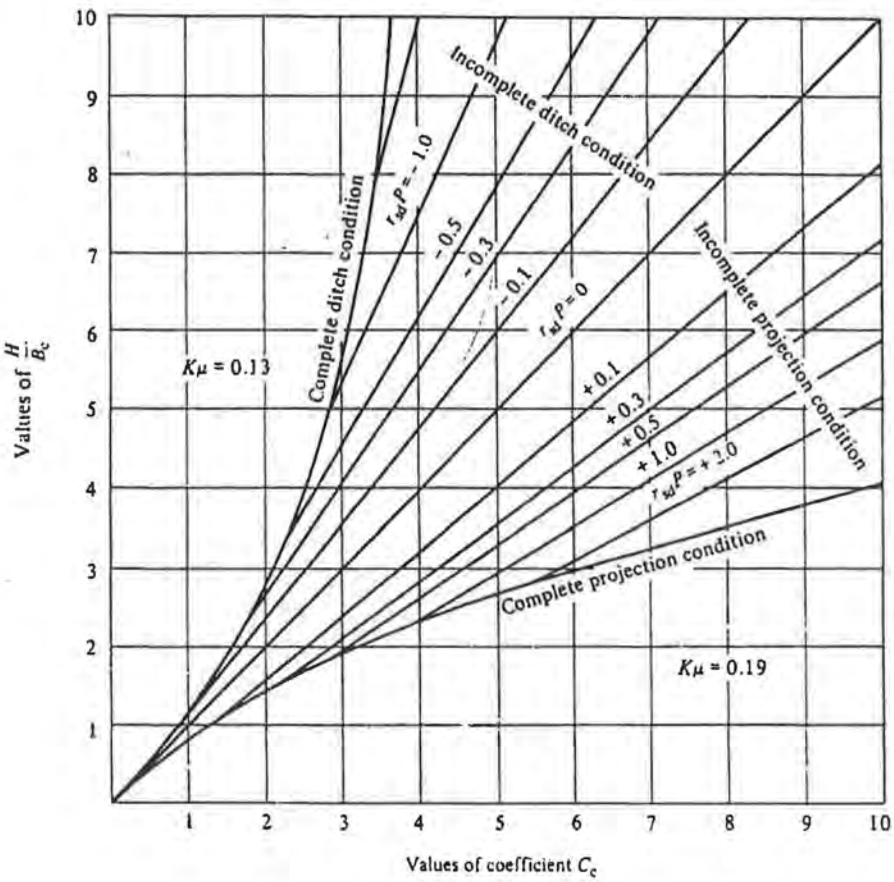



Figure 3.5: Diagram for Coefficient $C_c$ for Positive Projecting Conduits

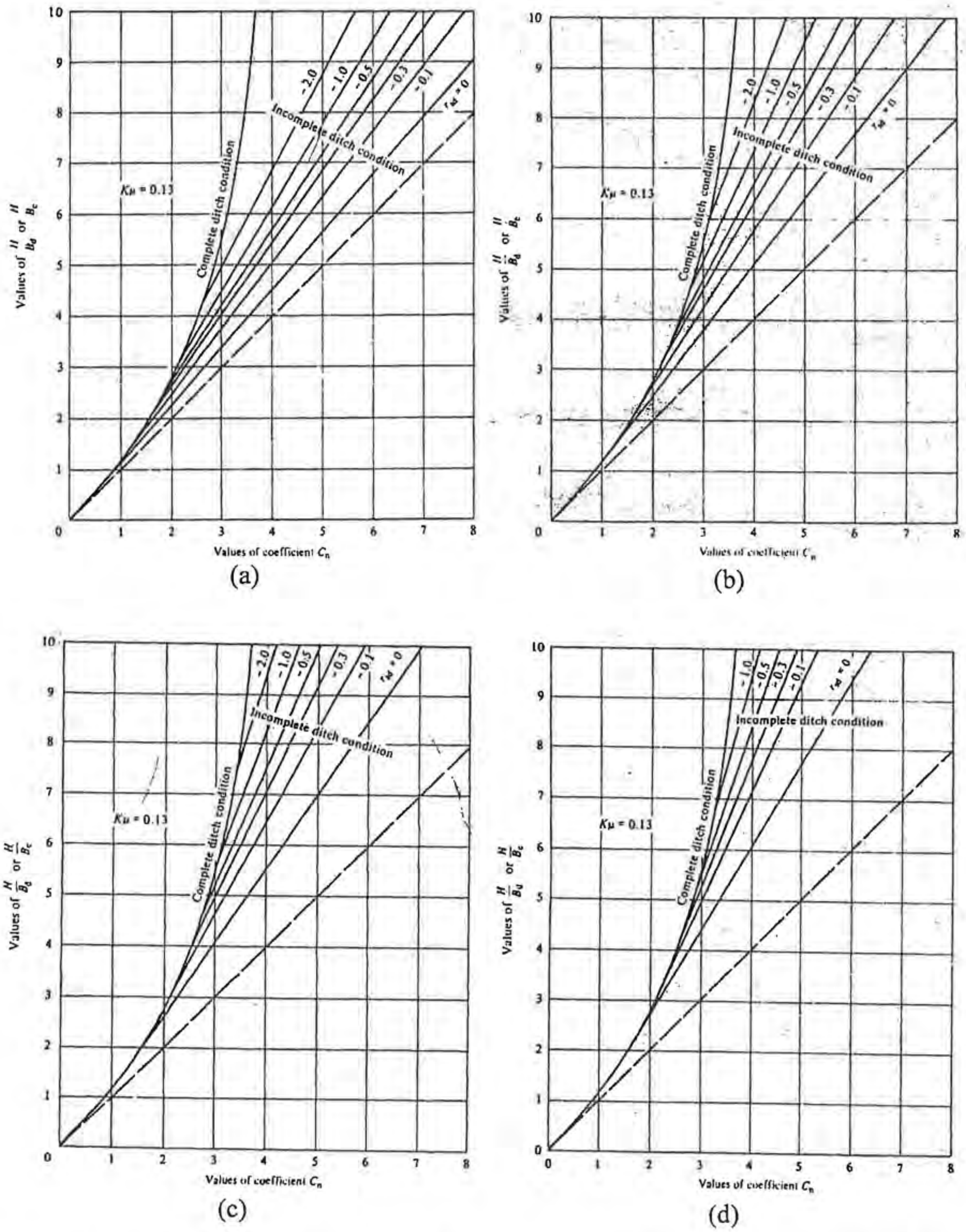



Figure 3.6: Diagrams for Coefficient $C_n$ for Negative Projection Conduits and Imperfect Ditch Conditions (a) $p' = 0.5$ (b) $p' = 1.0$ (c) $p' = 1.5$ (d) $p' = 2.0$

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:

$$W_e = F_e w B_c H \tag{3.4}$$

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

*Embankment Installation*

$$F_{e1} = 1 + 0.20\frac{H}{B_c} \tag{3.5}$$

*Trench Installation*

$$F_{e2} = \frac{C_d B_d^2}{H B_c} \tag{3.6}$$

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:

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

where $\sigma_1$, $\sigma_3$ = the maximum and minimum principal stresses, respectively; $\varepsilon$ = 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, $E_i$, and $b$ is the reciprocal of the asymptotic value of the stress difference which the stress-strain curve approaches at infinite strain $(\sigma_1\text{-}\sigma_3)_{ult}$. 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:

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

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 $(\sigma_1\text{-}\sigma_3)$ 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 $R_f$:

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

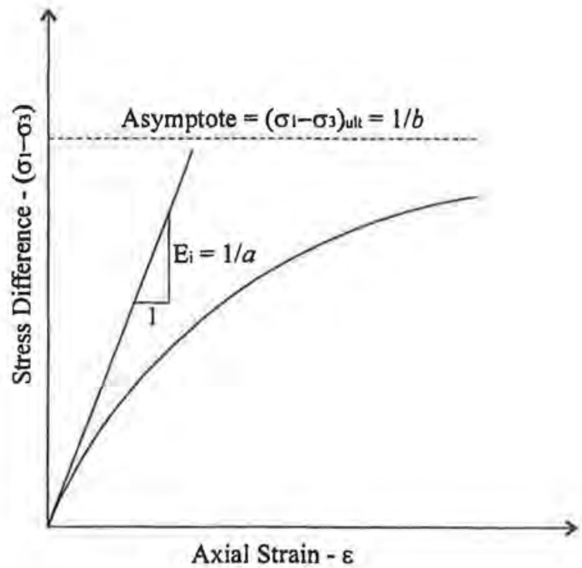



Figure 3.7: Hyperbolic Stress-Strain Curve

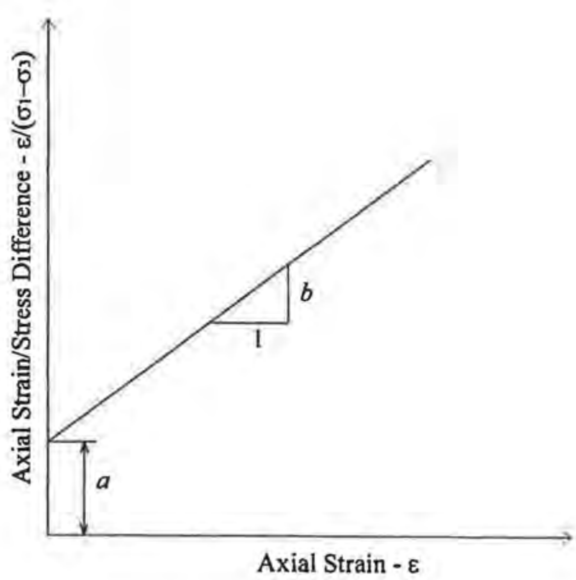



Figure 3.8: Transformed Hyperbolic Stress-Strain Curve

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

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

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

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

where $E_i$ = the initial tangent modulus; $\sigma_3$ = the minimum principal stress; $p_a$ = atmospheric pressure expressed in the same pressure units as $E_i$ and $\sigma_3$; $K$ = a modulus number; and $n$ = the exponent determining the rate of variation of $E_i$ with $\sigma_3$.

If it is assumed that failure will occur with no change in the value of $\sigma_3$, the relationship between compressive strength and confining pressure may be expressed conveniently in terms of the Mohr-Coulomb failure criterion as

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

where $c$, $\phi$ = 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 $K$, $n$, $c$, $\phi$, 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:

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

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

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

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:

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

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

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

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

$$\varepsilon_a = \frac{-\varepsilon_r}{v_i - d\varepsilon_r} \tag{3.17}$$

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

$$-\frac{\varepsilon_r}{\varepsilon_a} = v_i - d\varepsilon_r \tag{3.18}$$

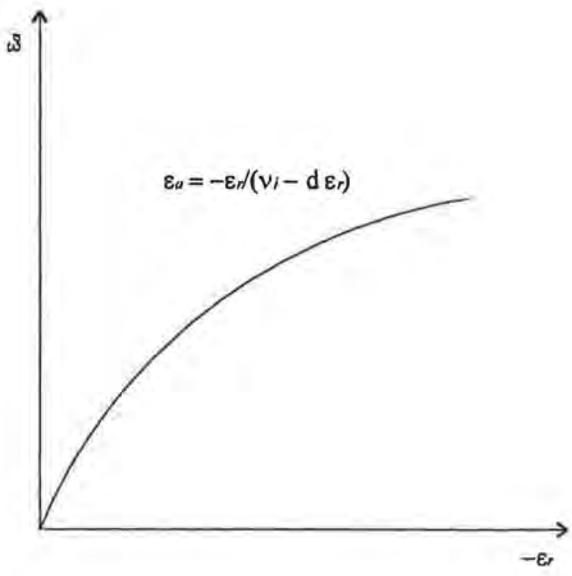



Figure 3.9: Hyperbolic Axial Strain-Radial Strain Curve

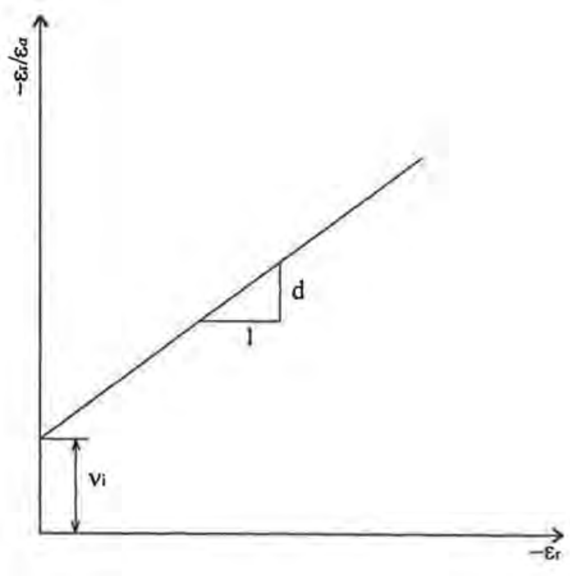



Figure 3.10: Transformed Hyperbolic Axial Strain-Radial Strain Curve

In Equation 3.18, $\nu_i$ 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 $\nu_i$ is equal to one half for any value of confining pressure. This variation of $\nu_i$ with respect to $\sigma_3$ may be expressed by the equation

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

where $G$ is the value of $\nu_i$ at a confining pressure of one atmosphere, and $F$ is the reduction in $\nu_i$ for a ten-fold increase in $\sigma_3$. After differentiating Equation 3.17 with respect to $\varepsilon_r$, 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:

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

where $\sigma_1$, $\sigma_3$ = maximum and minimum principal stresses, respectively; $K$ = modulus number; $n$ = modulus exponent; $c$ = cohesion intercept; $\phi$ = 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 $(\sigma_1$-$\sigma_3)$ 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 $\sigma_3$ |
| $n$ | Modulus exponent | |
| $c$ | Cohesion intercept | Relate $(\sigma_1\text{-}\sigma_3)$ to $\sigma_3$ |
| $\phi$ | Friction angle | |
| $R_f$ | Failure ratio | Relate $(\sigma_1\text{-}\sigma_3)_{ult}$ to $(\sigma_1\text{-}\sigma_3)_f$ |
| $G$ | Poisson's ratio parameter | Value of $\nu_i$ at $\sigma_3=p_a$ |
| $F$ | Poisson's ratio parameter | Decrease in $\nu_i$ for ten-fold increase in $\sigma_3$ |
| $d$ | Poisson's ratio parameter | Rate of increase of $\nu_t$ with strain |

elasticity, the value of the bulk modulus is defined by

$$B = \frac{\Delta\sigma_1 + \Delta\sigma_2 + \Delta\sigma_3}{3\varepsilon_v} \tag{3.21}$$

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 $(\sigma_1\text{-}\sigma_3)$ increases while the confining pressure is held constant, Equation 3.21 may be expressed as:

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

The value of bulk modulus for a conventional triaxial compression test may be calculated using the value of $(\sigma_1\text{-}\sigma_3)$ 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:

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

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:

$$\nu_t = \frac{1}{2} - \frac{E_t}{6B} \tag{3.24}$$

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:

$$\tau = \frac{\Delta_s}{a + b\Delta_s} \tag{3.25}$$

where $\tau$ = interface shear stress; $\Delta_s$ = 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) | $\phi_o(\Delta\phi)$ | K | n | $R_f$ | G | F | d | $K_b$ | 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 | .35 | .10 | 2.4 | 360 | .11 |
| 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 | 13.0 | 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) | | | | 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: Type | Compaction: Max. Dry Unit Wt. (PCF) | Compaction: Opt. W/C | Compaction: Dry U. Wt. (PCF) | Compaction: W/C | C (TSF) | $\phi_o(\Delta\phi)$ | K | n | $R_f$ | G | F | d | $K_b$ | m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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) | $\phi_o(\Delta\phi)$ | K | n | $R_f$ | G | F | d | $K_b$ | 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 Silly Sand (Rodman Dam) | Mod.AASHO | 101.1 | 13.6 | 101.3 | 13.6 | .00 | 44( 6) | 420 | .67 | .76 | | | | | |
| SP | SP-9E | Poorly Graded Silly Sand (Rodman Dam) | Mod.AASHO | 101.1 | 13.6 | 96.2 | 13.3 | .00 | 44(11) | 850 | .79 | .92 | | | | | |
| SP | SP-9F | Poorly Graded Silly 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 | $R_f$ | G | F | d | $K_b$ | m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Type | Max. Dry Unit Wt. (PCF) | Opt. W/C | Dry U. Wt. (PCF) | W/C | | | | | | | | | | |
| CL | CL-10B | Sandy Clay (Somerville Dam) | Std.AASHO | 110.7 | 15.0 | 106.5 | 15.0 | .55 | 22 | 290 | .27 | .91 | | | | | |
| CL | CL-11A | Sandy Clay (Somerville Dam) | Std.AASHO | 107.5 | 16.8 | 100.3 | 13.5 | .78 | 28 | 680 | - .36 | .84 | | | | | |
| CL | CL-11B | Sandy Clay (Somerville Dam) | Std.AASHO | 107.5 | 16.8 | 106.5 | 13.3 | 1.50 | 25 | 600 | .18 | .68 | | | | | |
| CL | CL-11C | Sandy Clay (Somerville Dam) | Std.AASHO | 107.5 | 16.8 | 102.6 | 19.3 | .74 | 6 | 23 | .32 | .61 | | | | | |
| CL | CL-11D | Sandy Clay (Somerville Dam) | Std.AASHO | 107.5 | 16.8 | 106.7 | 16.7 | .91 | 18 | 280 | .60 | .93 | | | | | |
| CL | CL-11E | Sandy Clay (Somerville Dam) | Std.AASHO | 107.5 | 16.8 | 101.5 | 16.3 | .66 | 20 | 220 | .23 | .90 | | | | | |
| CL | CL-12A | Sandy Clay (Somerville Dam) | Std.AASHO | 106.1 | 17.2 | 105.0 | 18.6 | 1.30 | 8 | 140 | .20 | .84 | | | | | |
| CL | CL-12C | Sandy Clay (Somerville Dam) | Std.AASHO | 106.1 | 17.2 | 101.9 | 17.1 | 1.00 | 13 | 120 | .09 | .83 | | | | | |
| CL | CL-12D | Sandy Clay (Somerville Dam) | Std.AASHO | 106.1 | 17.2 | 103.0 | 19.7 | .80 | 2 | 47 | .33 | .82 | | | | | |
| CL | CL-12E | Sandy Clay (Somerville Dam) | Std.AASHO | 106.1 | 17.2 | 106.5 | 13.9 | 1.50 | 24 | 950 | - .15 | .90 | | | | | |
| CL | CL-12F | Sandy Clay (Somerville Dam) | Std.AASHO | 106.1 | 17.2 | 108.3 | 16.9 | 1.50 | 8 | 470 | .00 | .95 | | | | | |
| CL | CL-13A | Sandy Clay (Somerville Dam) | Std.AASHO | 104.9 | 17.6 | 98.7 | 20.8 | .67 | 4 | 75 | .44 | .88 | | | | | |
| CL | CL-13B | Sandy Clay (Somerville Dam) | Std.AASHO | 104.9 | 17.6 | 104.9 | 14.8 | 1.80 | 23 | 840 | - .19 | .84 | | | | | |
| CL | CL-13C | Sandy Clay (Somerville Dam) | Std.AASHO | 104.9 | 17.6 | 101.2 | 17.4 | 1.20 | 12 | 270 | .06 | .87 | | | | | |
| CL | CL-13D | Sandy Clay (Somerville Dam) | Std.AASHO | 104.9 | 17.6 | 100.5 | 14.2 | 1.40 | 29 | 1100 | - .36 | .83 | | | | | |
| CL | CL-13E | Sandy Clay (Somerville Dam) | Std.AASHO | 104.9 | 17.6 | 104.4 | 17.5 | 1.40 | 13 | 410 | .15 | .87 | | | | | |
| CL | CL-14 | Lean Clay (Clinton Dam) | Std.AASHO | 103.0 | 21.2 | 98.0 | 24.0 | .77 | 2 | 57 | .43 | .86 | | | | | |
| CL | CL-16C | Lean Clay (Clinton Dam) | Std.AASHO | 105.0 | 20.2 | 99.7 | 22.9 | .97 | 1 | 110 | .43 | .90 | | | | | |
| CL | CL-17A | Lean Clay (Clinton Dam) | Std.AASHO | 101.0 | 20.1 | 99.1 | 22.7 | 1.10 | 2 | 100 | .27 | .89 | | | | | |
| CL | CL-17B | Lean Clay (Clinton Dam) | Std.AASHO | 101.0 | 20.1 | 98.1 | 23.9 | .99 | 1 | 160 | .54 | .97 | | | | | |
| CL | CL-17C | Lean Clay (Clinton Dam) | Std.AASHO | 101.0 | 20.1 | 98.9 | 22.7 | 1.10 | 3 | 130 | .46 | .91 | | | | | |
| CL | CL-19A | Lean Clay (Clinton Dam) | Std.AASHO | 102.0 | 19.9 | 96.8 | 22.7 | .78 | 2 | 53 | .41 | .85 | | | | | |
| CL | CL-24A | Sandy Clay (Chatfield Dam) | Std.AASHO | 104.0 | 19.3 | 97.6 | 23.4 | 1.20 | 0 | 240 | .00 | .95 | | | | | |
| CL | CL-25A | Sandy Clay (Chatfield Dam) | Std.AASHO | 113.0 | 15.1 | 107.4 | 18.1 | .95 | 0 | 160 | .00 | .93 | | | | | |
| CL | CL-28 | Sandy Clay (Proctor Dam) | Std.AASHO | 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.AASHO | 94.0 | 26.5 | 90.0 | 28.8 | .61 | 4 | 92 | .21 | .89 | | | | | |
| CH | CH-3A | Fat Clay (Montoe Dam) | Std.AASHO | 95.5 | 26.5 | 89.3 | 31.1 | .37 | 0 | 21 | .00 | .65 | | | | | |
| CH | CH-3B | Fat Clay (Montoe Dam) | Std.AASHO | 95.5 | 26.5 | 92.6 | 28.6 | .51 | 1 | 67 | .02 | .79 | | | | | |
| CH | CH-4 | Fat Clay (Montoe Dam) | Std.AASHO | 100.0 | 22.7 | 96.4 | 26.5 | .63 | 1 | 65 | .14 | .77 | | | | | |
| CH | CH-5A | Fat Clay (Chatfield Dam) | Std.AASHO | 95.0 | 24.4 | 90.3 | 27.4 | 1.20 | 0 | 36 | .72 | .91 | | | | | |
| CH | CH-5B | Fat Clay (Chatfield Dam) | Std.AASHO | 95.0 | 24.4 | 90.7 | 24.4 | 1.50 | 2 | 52 | .66 | .89 | | | | | |

$$\frac{\Delta_s}{\tau} = a + b\Delta_s \tag{3.26}$$

If values of $\Delta_s/\tau$ are plotted against values of $\Delta_s$, 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, $\tau_{ult}$, is larger than the peak value of shear stress or strength of the interface, $\tau_f$. It has been found to be convenient to express $\tau_{ult}$ in terms of $\tau_f$ by means of a failure ratio, $R_f$, as follows:

$$\tau_f = R_f \tau_{ult} \tag{3.27}$$

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]

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

where $K_I$ = dimensionless stiffness number; $n$ = stiffness exponent; $\gamma_w$ = 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 $\sigma_n$. 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:

$$\tau_f = \sigma_n \tan\delta \tag{3.29}$$

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:

$$\tau = \frac{\Delta_s}{\dfrac{1}{K_I \gamma_w \left(\dfrac{\sigma_n}{p_a}\right)^n} + \dfrac{R_f \Delta_s}{\sigma_n \tan\delta}} \tag{3.30}$$

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 $\Delta_s$, and eliminating $\Delta_s$ from the resulting equation, the tangent stiffness value may be expressed as:

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

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:

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

where $H_i$, $\gamma_i$ = 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, $\nu$, 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

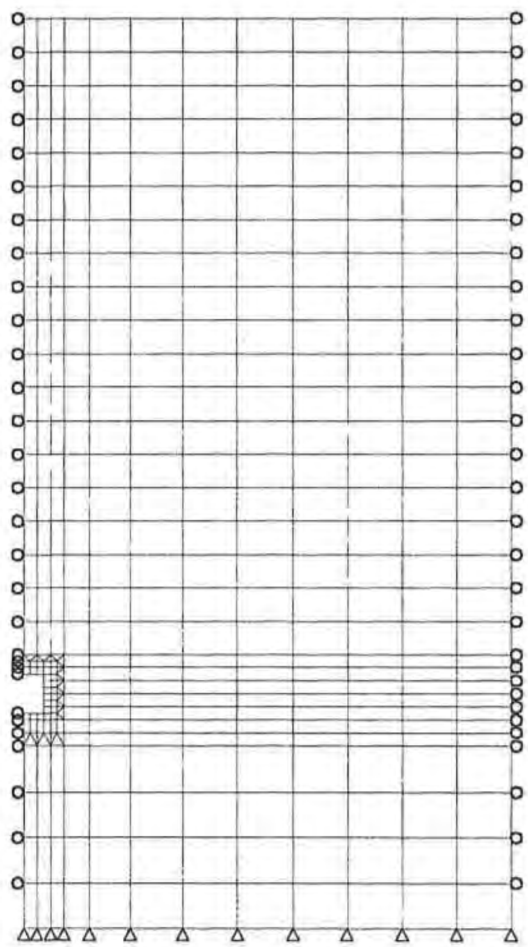


Figure 4.1: Typical Soil-Structure Finite Element Model

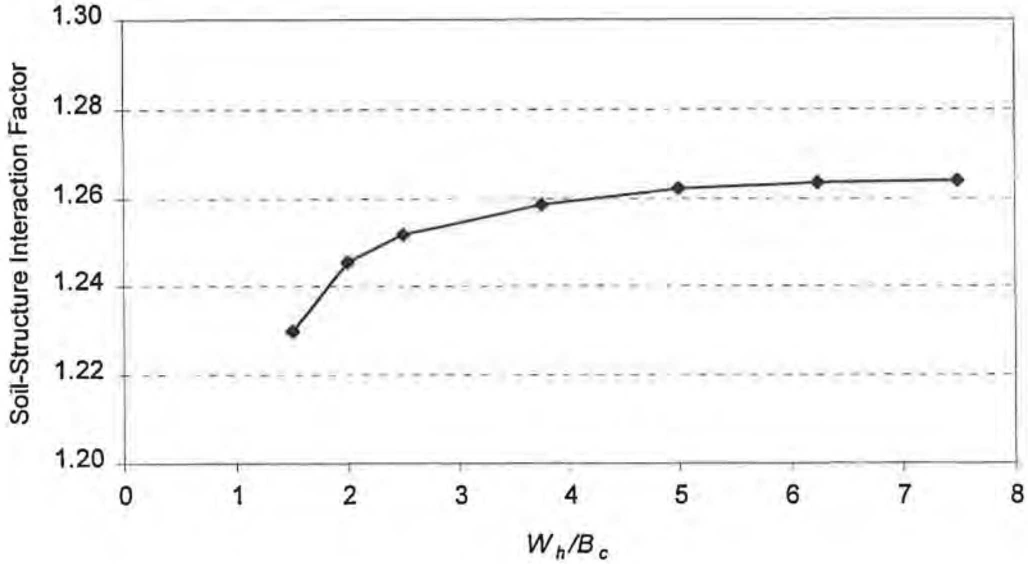



Figure 4.2: Soil-Structure Interaction Factor vs. $W_h/B_c$

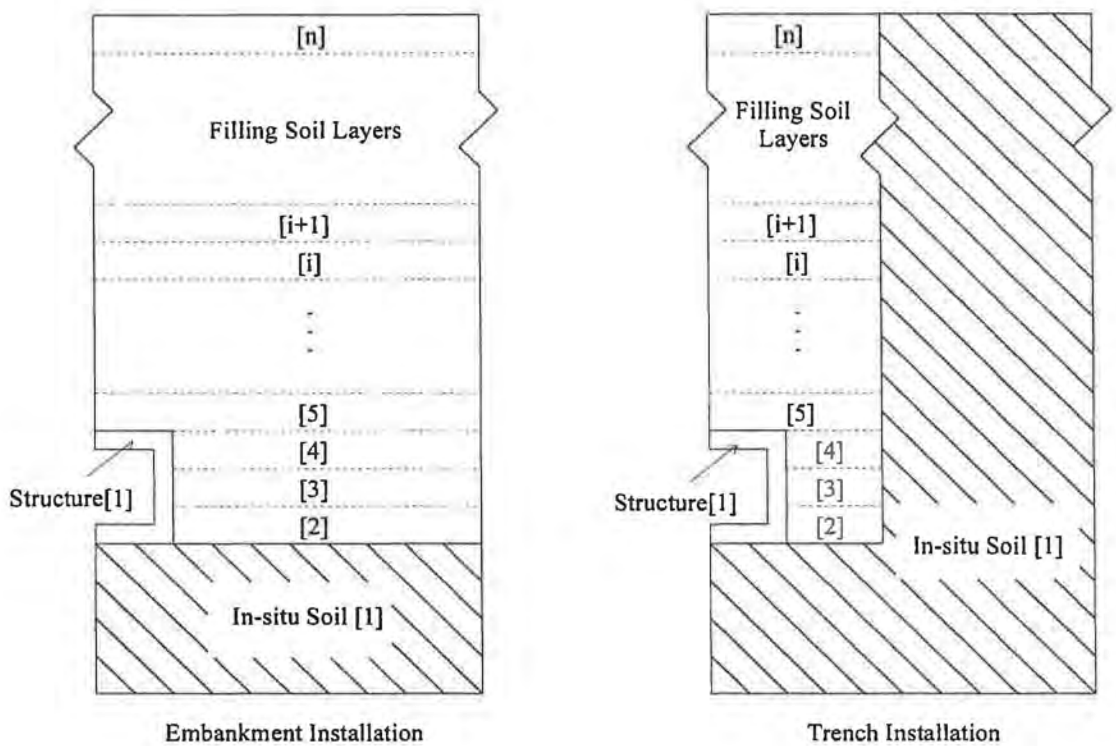



Figure 4.3: Incremental Sequences for Embankment and Trench Installation

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) | $E_{interface\ element}/E_{soil}$ | | | | | |
|---|---|---|---|---|---|---|
| | 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:

$$k = \frac{d}{2} K_{st} \tag{4.1}$$

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

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

Note that $K_{st}$ is a function of normal stress, $\sigma_n$ and shear stress, $\tau$, 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 |
|---|---|
| $\delta$ | 24° |
| $K_I$ | 40,000 |
| $N$ | 1.0 |
| $R_f$ | 0.9 |

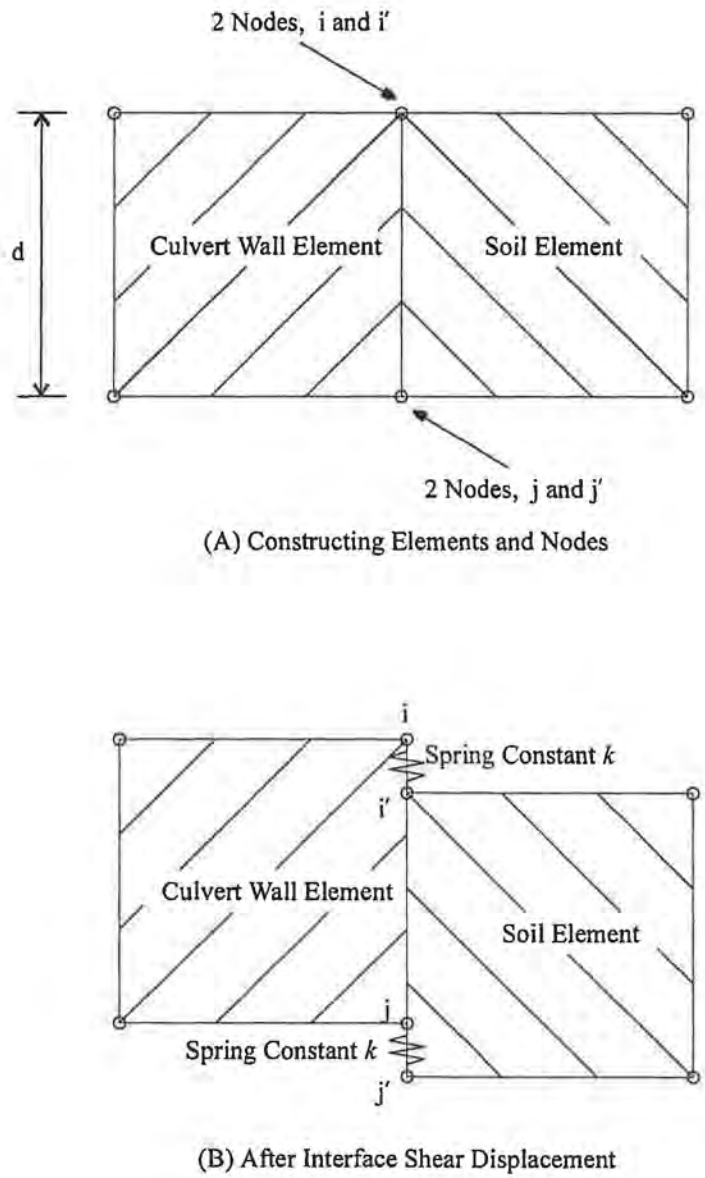



Figure 4.4: Spring Interface Modeling

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

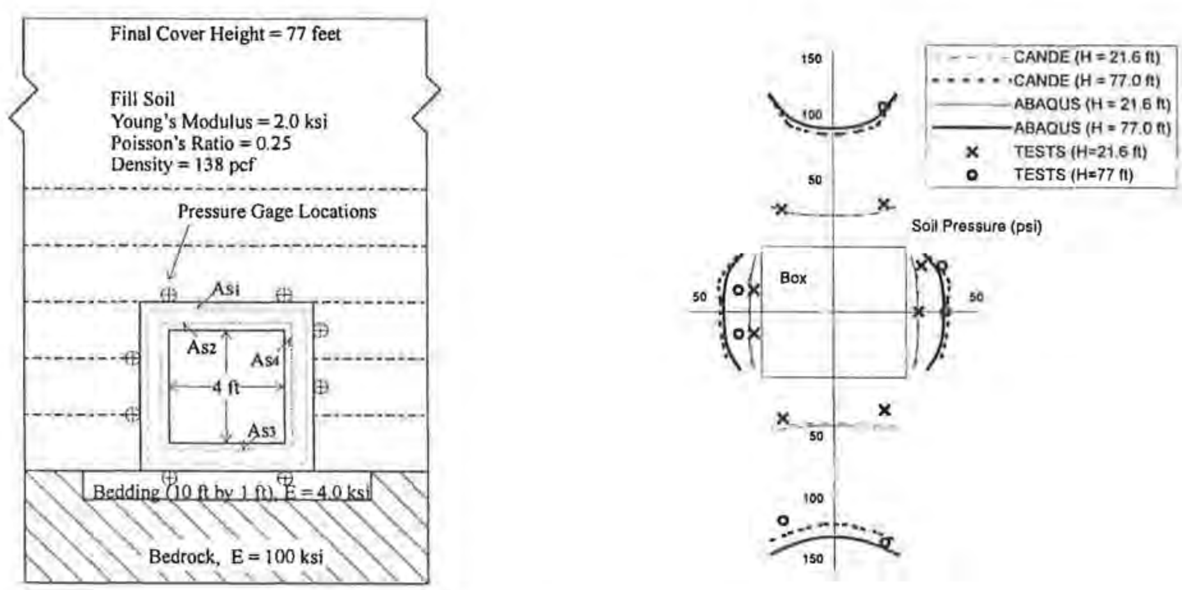



Figure 4.5: Schematics of Test Setup and Data Comparison

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

Static Stresses and Movements in Embankments).

## 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 |
| $\phi$ (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.

$$D_E = 1.009H^{0.048} \text{ on Yielding Foundation} \tag{5.1}$$

$$D_E = 1.016H^{0.068} \text{ on Unyielding Foundation} \tag{5.2}$$

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

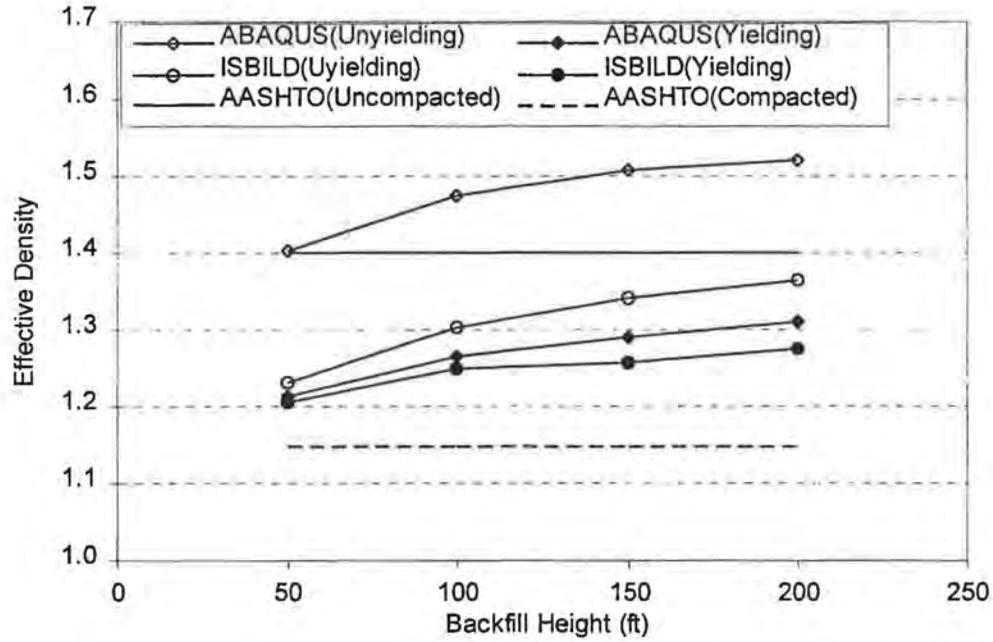



Figure 5.1: Effective Density for Embankment Installations

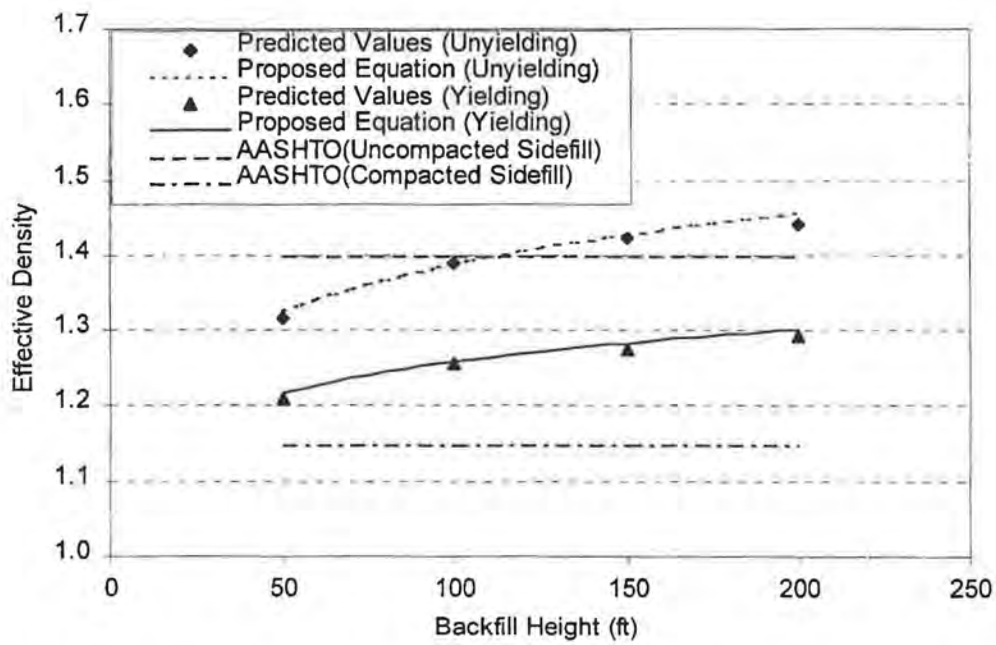
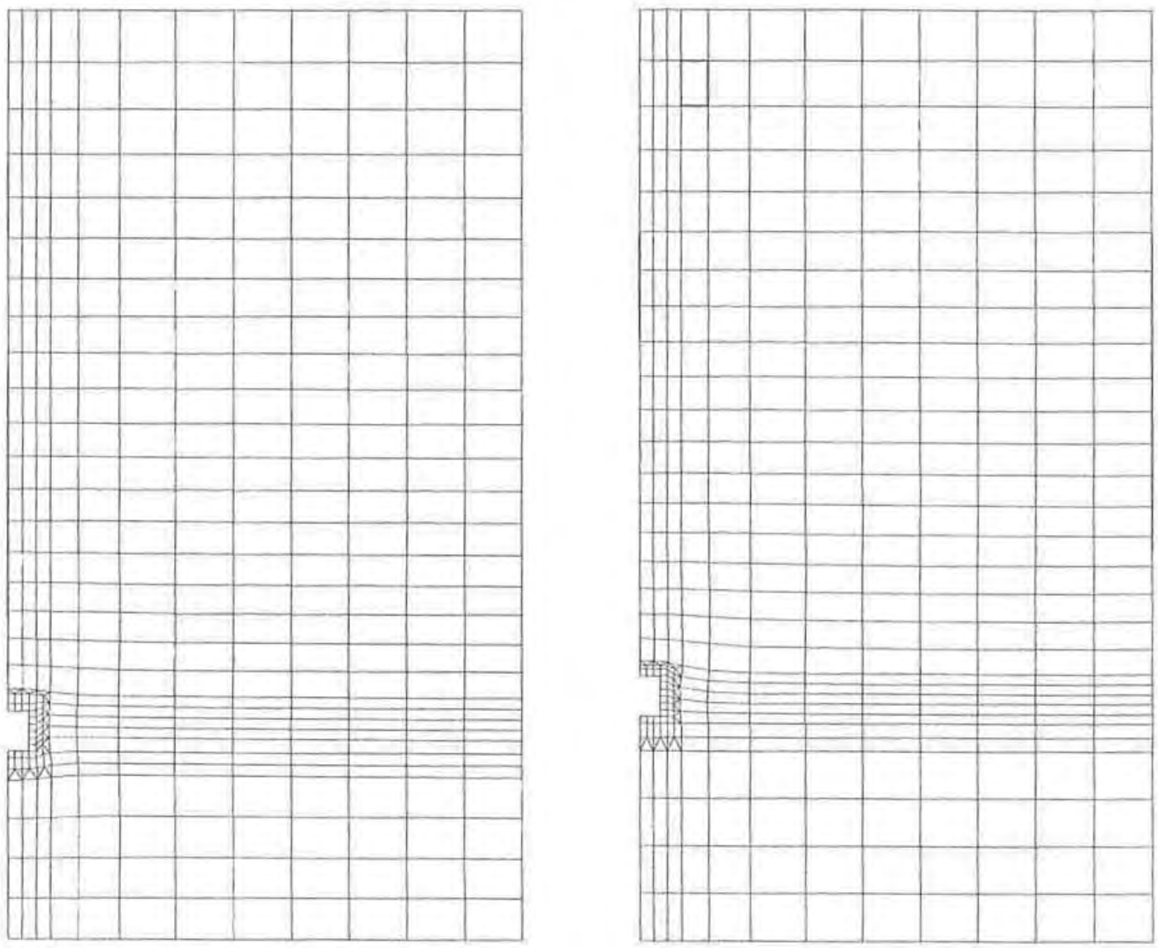



Figure 5.2: Effective Density Curves from Proposed Equations for Embankment Installations



(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 ASSHTO 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 weigh 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 |

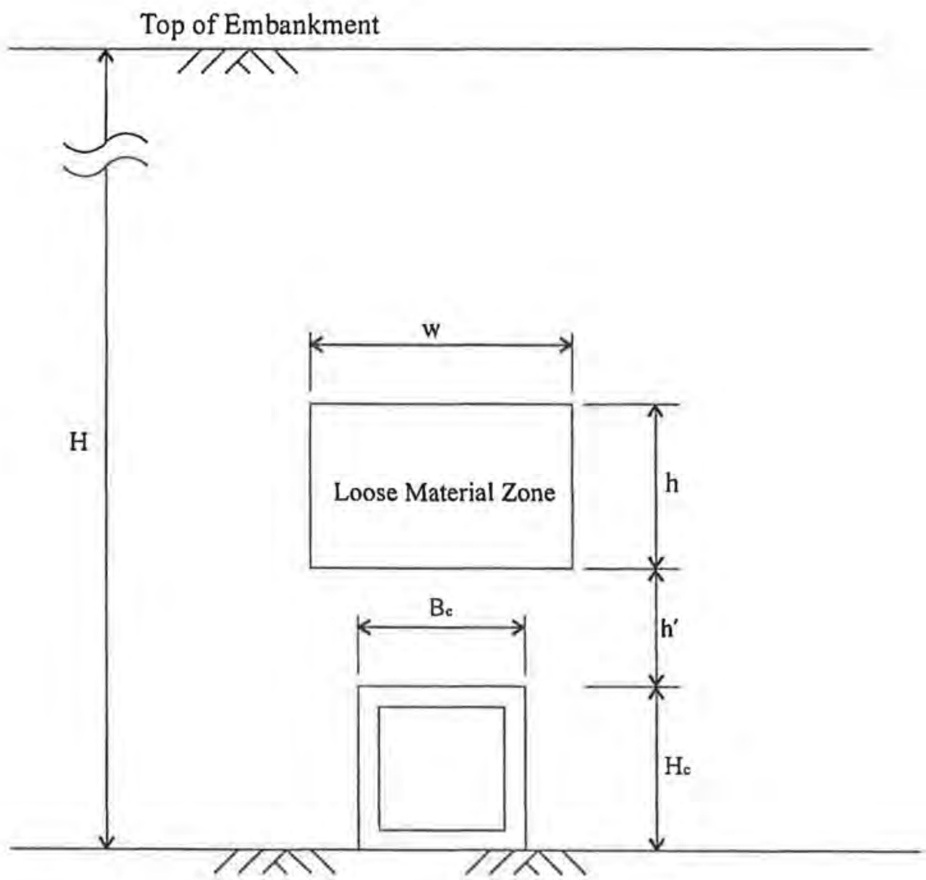



Figure 5.3: Imperfect Trench Installation

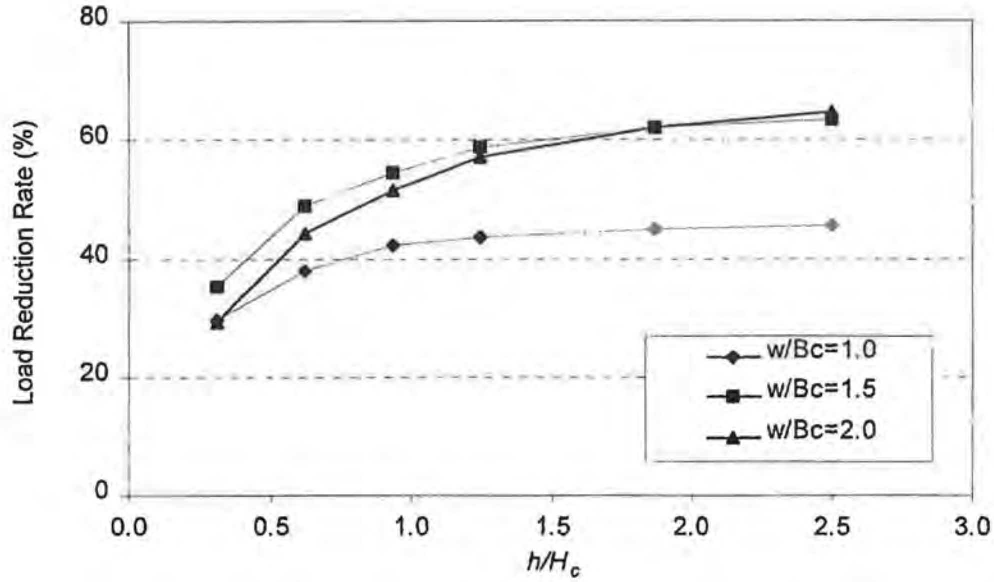



Figure 5.4: Load Reduction Rate vs. $h/H_c$ (Different $w/B_c$)

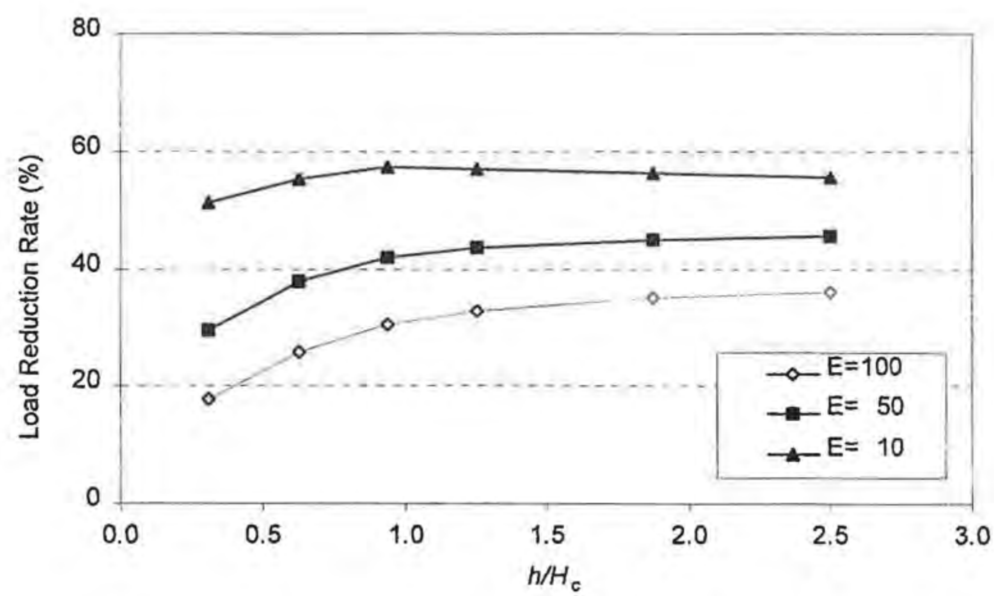



Figure 5.5: Load Reduction Rate vs. $h/H_c$ (Different Young's Modulus)

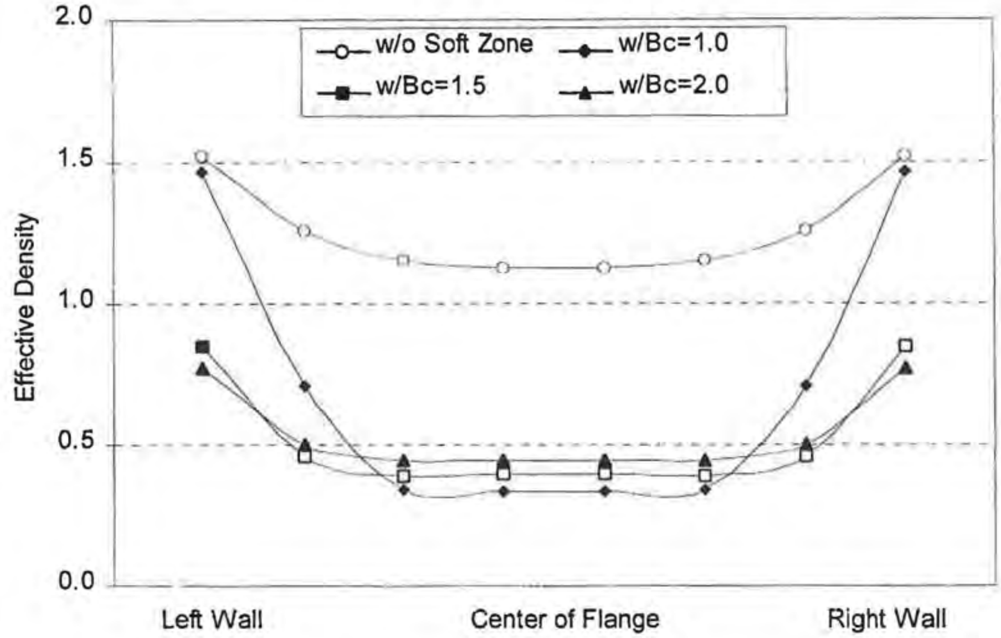



Figure 5.6: Effective Density Distribution on the Box (Different $w/B_c$)

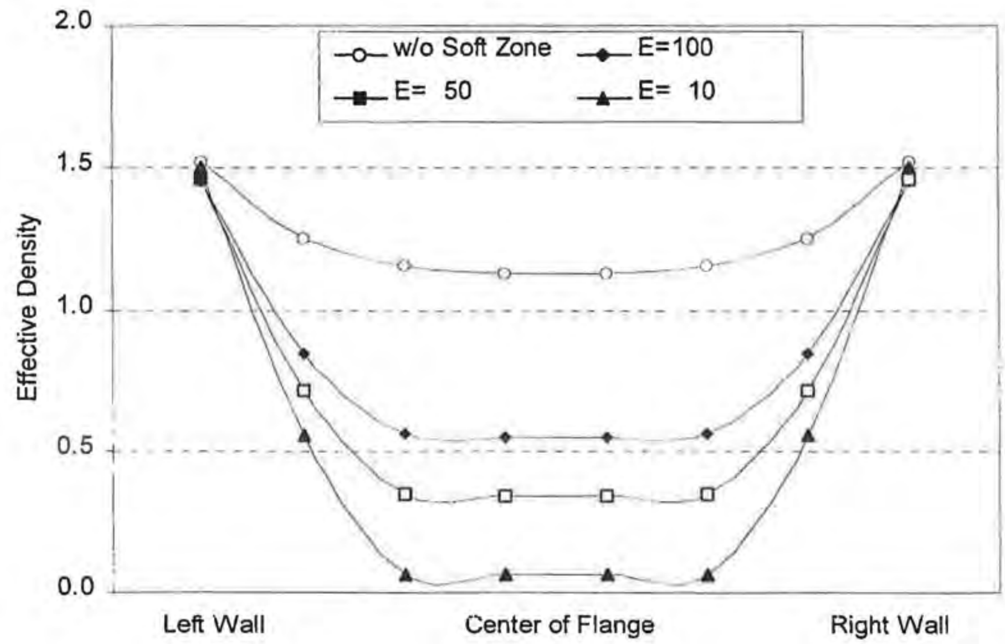



Figure 5.7: Effective Density Distribution on the Box (Different Young's Modulus)

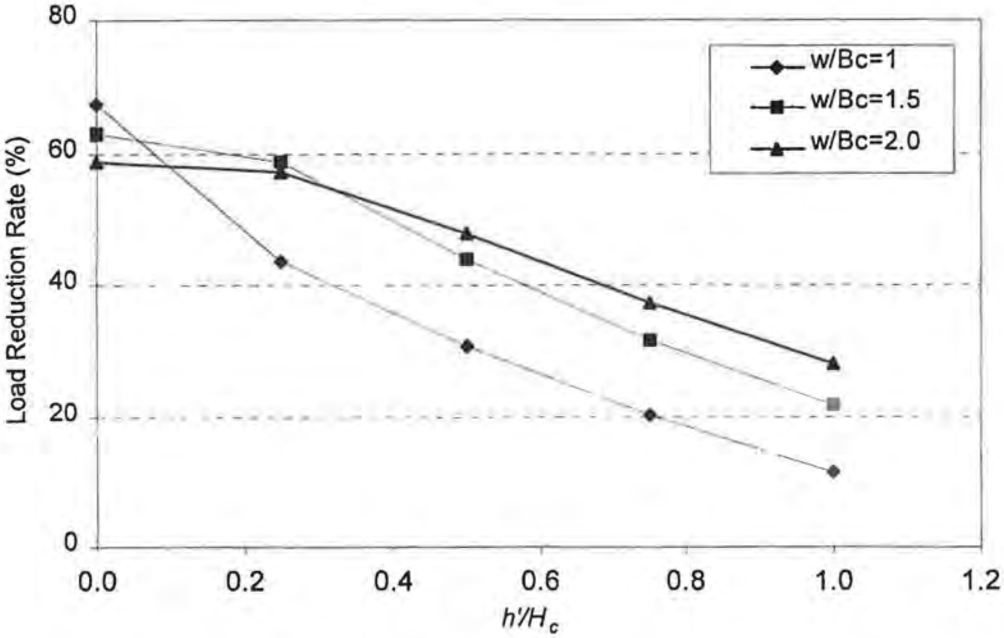



Figure 5.8: Load Reduction Rate vs. $h'/H_c$

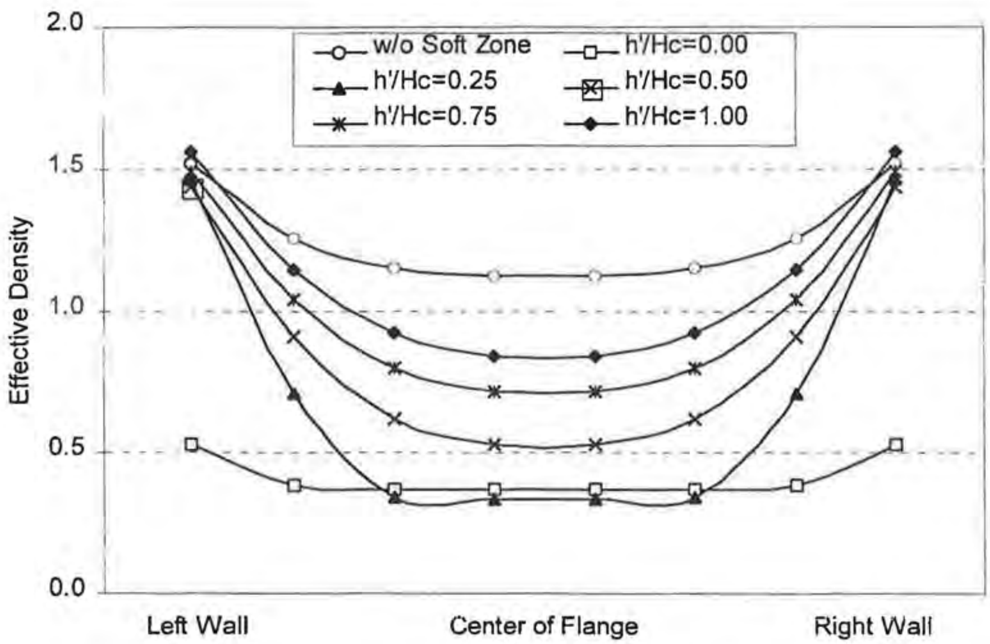



Figure 5.9: Effective Density Distribution ($w/B_c$=1, $h/H_c$=1.25)

Table 5.3 Vertical Displacements at Soft Zone (ft)

| | E=100 (ksf) | | | E=50 (ksf) | | | E=10 (ksf) | | |
|---|---|---|---|---|---|---|---|---|---|
| | ν=0.4 | ν=0.1 | Diff. (%) | ν=0.4 | ν=0.1 | Diff. (%) | ν=0.4 | ν=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 |

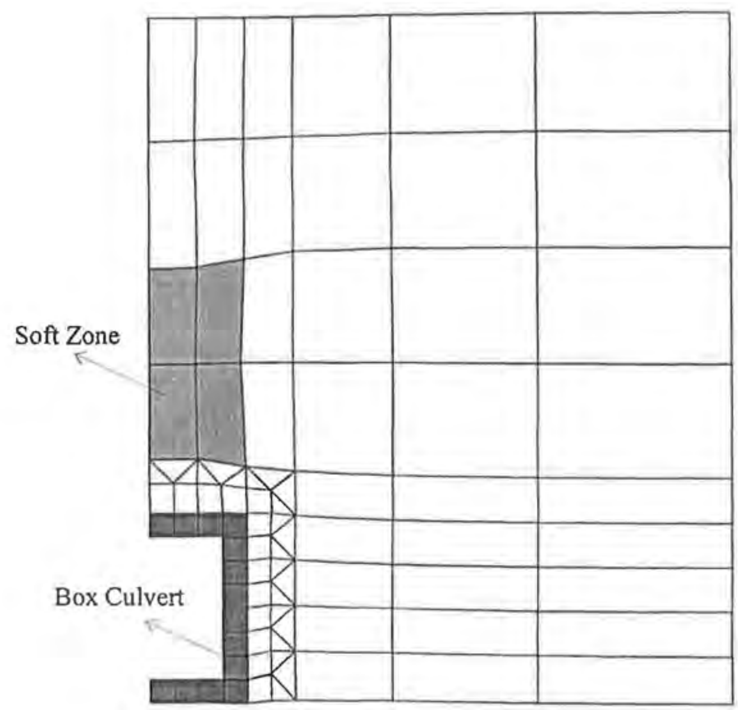



Figure 5.10: Close-up View of Deformed Shape near the Soft Zone

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).

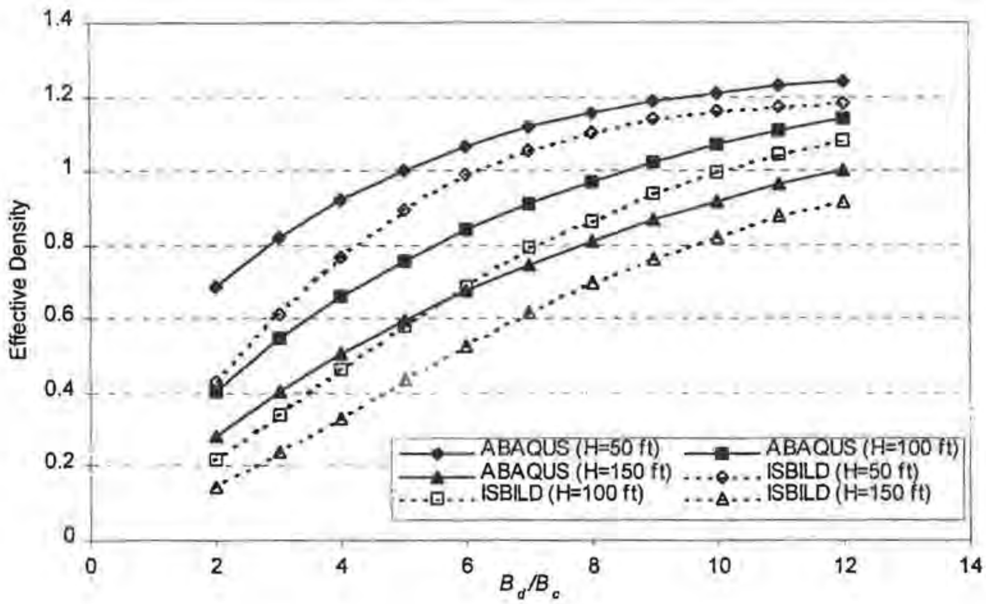



Figure 5.11: Effective Density vs. $B_d/B_c$.

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*

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

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

*Unyielding Foundation*

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

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

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.

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

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

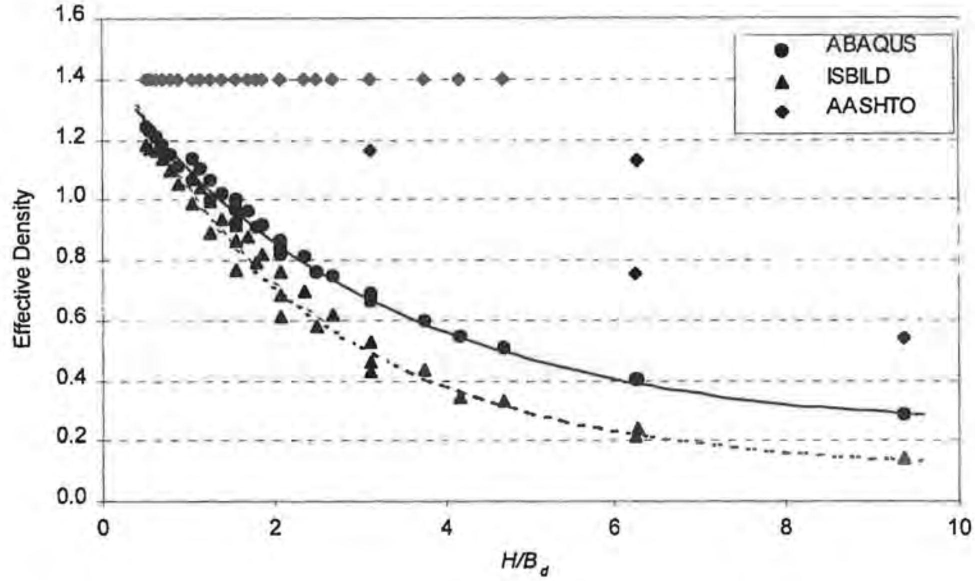



Figure 5.12: Effective Density vs. *H/B*$_d$ on Yielding Foundation.

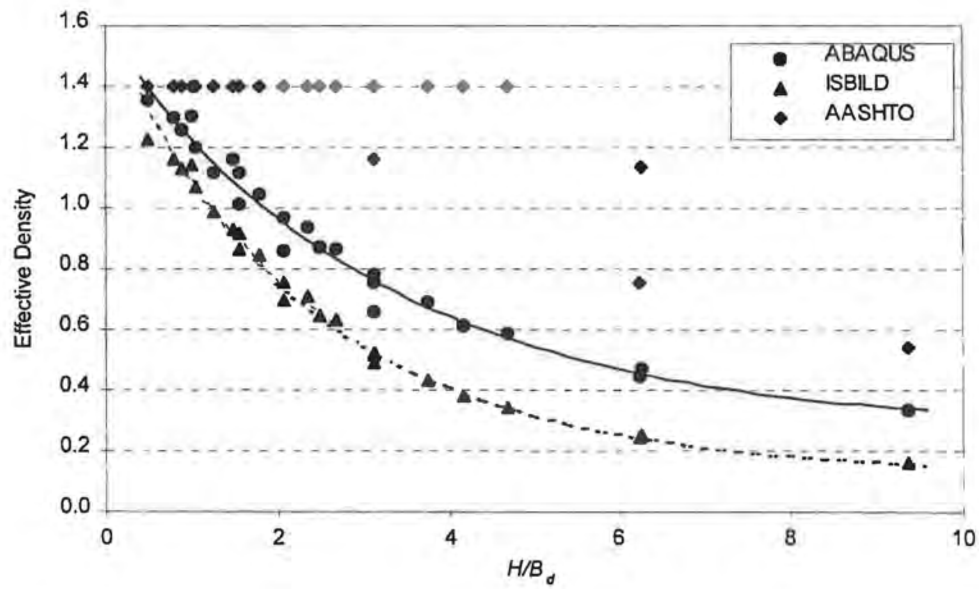



Figure 5.13: Effective Density vs. *H/B*$_d$ on Unyielding Foundation

## 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. $\theta$ 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 $\theta$ 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

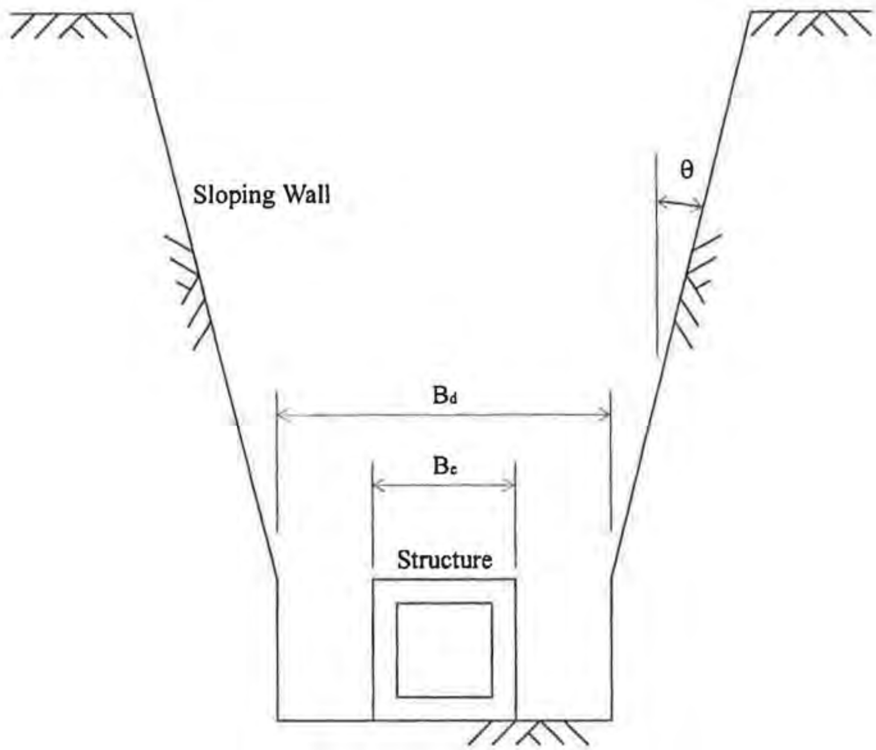



Figure 5.14: Trench Installation with Sloping Sidewalls

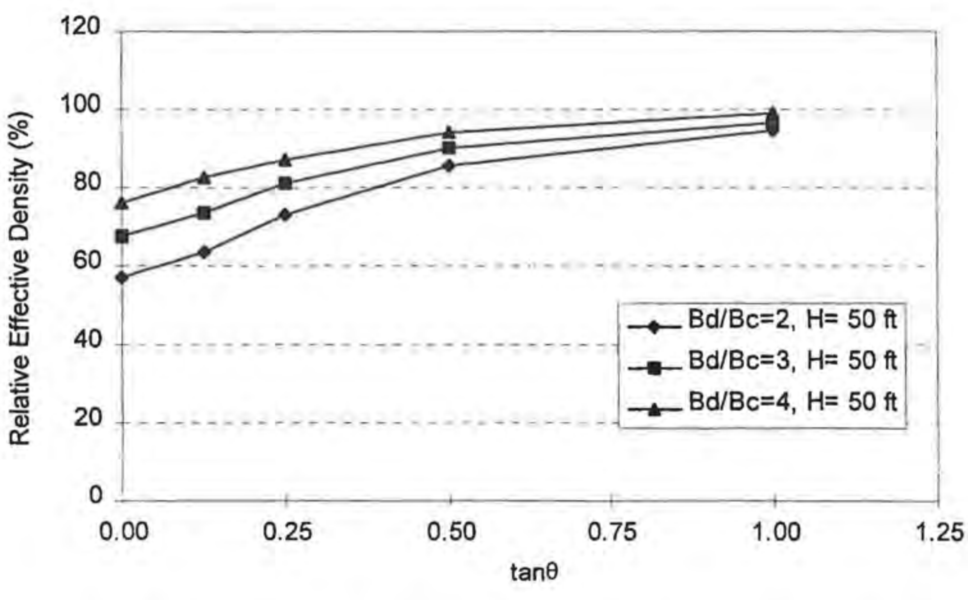



Figure 5.15: Effect of Sloping Sidewall in Trench Installations (*H*=50 feet)

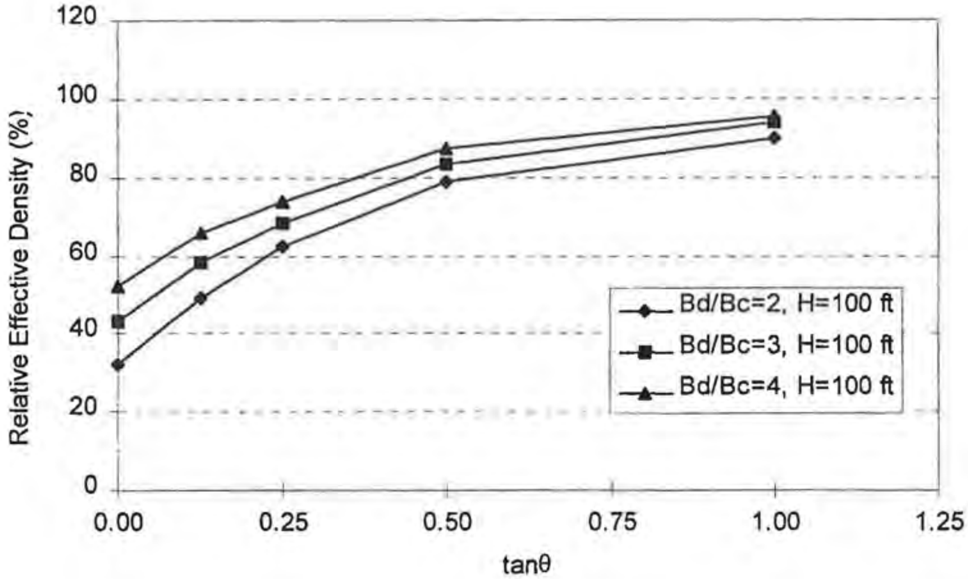



Figure 5.16: Effect of Sloping Sidewall in Trench Installations ($H$=100 feet)

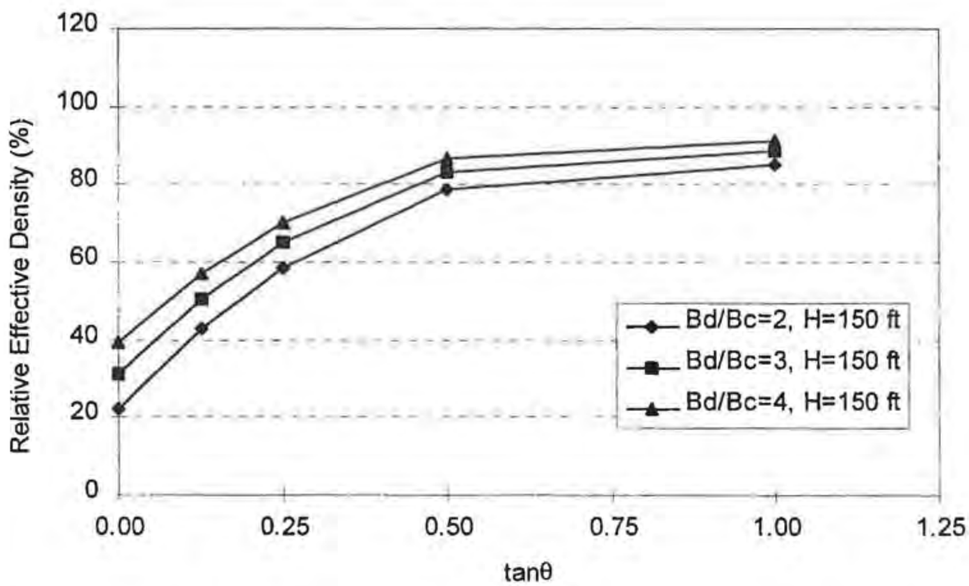



Figure 5.17: Effect of Sloping Sidewall in Trench Installations ($H$=150 feet)

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 × 8 ft |
| Fill height, $H$: | 160 ft |
| Weight of filling soil, $\gamma$: | 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

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

From AASHTO provisions, Equations 3.4 and 3.5,

$$F_{e1} = 1 + 0.20\frac{H}{B_c} = 1 + 0.20\frac{160}{8} = 5.00 \ > 1.40$$

$$W_{AASHTO} = F_{e1} \gamma B_c H = (1.40)(0.12)(8)(160) = 215.04 \ \ (k/ft)$$

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,

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

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 × 8 ft |
| Fill height, $H$: | 175 ft |
| Weight of filling soil, $\gamma$: | 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:

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

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 ration, $r_{sd}$ 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:

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

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 × 8 ft |
| Fill height, $H$: | 120 ft |
| Weight of filling soil, $\gamma$: | 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 $\alpha$, corresponding to $H$ = 120 ft and $B_d/B_c$ = 2. Using a linear interpolation, the relative effective density, $\alpha$, is computed to be 78%. Therefore, the total earth load is calculated as follows:

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

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:

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

$W_{AASHTO} = F_e w B_c H = (0.733)(0.12)(8)(120) = 84.44 \ (k / ft)$

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 $\tan\theta$ = ½, 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:

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

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

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 Divi., 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 CULVERVERT
*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
82,83,84,85,86,98,99,100,101,102,103,104,105,106,107,
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
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 HEAD,T,TIME,AA
      DIMENSION HEAD(20),T(10),L(1),A(1)
      COMMON /ISOP/ E1,E2,RR(4),ZZ(4),LM(8),P(8),S(12,12),STR(3,12),
     1 STS(3,8),VJAC,MPRINT
      COMMON AA(8000000)
      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
      MTOTAL=8000000
      MTOTD2=MTOTAL/2
      IN=5
      IO=6
      OPEN ( 4,FILE='f04',FORM='UNFORMATTED')
      OPEN ( 5,FILE=' ')
      OPEN ( 6,FILE=' ')
      OPEN ( 7,FILE='f07',FORM='UNFORMATTED')
      OPEN ( 8,FILE='f08',FORM='UNFORMATTED')
      OPEN ( 9,FILE='f09',FORM='UNFORMATTED')
      OPEN (10,FILE='f10',FORM='UNFORMATTED')
      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 NMXEQB,NPUNCH,IN,IO,INEQ)
      CALL SECOND (T(2))
      N23=N22+NEQ
```

```
      N24=N23+NEQB
      IF(N24.LT.MTOTD2) 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 UNKNOWNS
C
      CALL SYMBAN (A(N17),A(NNN1),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(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
  250 T(I)=T(I+1)-T(I)
      T(9)=T(5)+T(6)+T(7)+T(8)
      WRITE (IO,2110) T(5),T(6),T(7),T(8),T(8)
      IF(IT.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),ZZ(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
      REWIND 10
      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
      REWIND 13
      REWIND 14
      NA=14
      NUME=NUM3
      IF (MM .NE. 1) GO TO 150
      NA=13
      NUME=NUMEL
      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.MB) MM=0
      MM=MM+1
      NSHIFT=NSHIFT+NE2B
  500 CONTINUE
      RETURN
      END
      SUBROUTINE CALBLK (ID,NCEL,NCNP,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),LM(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/
C
C     MATRIX OF DERIVATIVES
C
      SM=1.0D0-SX
      SP=1.0D0+SX
      TM=1.0D0-TY
      TP=1.0D0+TY
C
      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)
C
      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
C
      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
C
C     JACOBIAN
C
      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     FORM STRAIN DISPLACEMENT MATRIX
C
      DO 200 K=1,6
      I=II(K)
      J=JJ(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     CALCULATE SHEAR MODULUS, BULK MODULUS AND POISSON'S RATIO VALUES
C
      DEVSTR=PRS(1)-PRS(2)
C     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     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     PI=355./113.
C     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)
CC      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 /ISOP/ E1,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/
C
C     FORM STRAIN-DISPLACEMENT MATRIX
C
      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,NUMNP,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,NUMNP
      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 (IO,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 (IO,2010)
      DO 600 L=1,NUMPC
      READ (IN,*) IBC,JBC,PRES1,PRES2
      WRITE (IO,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/3.0D0
      FX(JBC)=FX(JBC)+(PRES1+TEMP2)*DY/3.0D0
      FY(IBC)=FY(IBC)+(TEMP1+PRES2)*DX/3.0D0
      FY(JBC)=FY(JBC)+(PRES1+TEMP2)*DX/3.0D0
  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 (IO,2030)
      GO TO 770
  760 WRITE (IO,2035)
  770 CONTINUE
      DO 800 N=1,NUMNP
  800 WRITE (IO,2040) N,FX(N),FY(N)
C1000 FORMAT (2I5)
C1010 FORMAT (I5,2F10.0)
C1020 FORMAT (2I5,2F10.0)
 2000 FORMAT (/,
     1 35H0NUMBER OF N.P. FORCE CARDS********,I3,/,
     2 35H0NUMBER 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 /ISOP/ E1,E2,RR(4),ZZ(4),LM(8),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 STIFNESS 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-DISPLACEMEMT 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/ E1,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 (IO,2000) LN,IT
      GO TO 16
   15 LNMLAY=LN-NLAY
      WRITE (IO,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 (IO,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 (IO,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(I5,2F10.2,3F15.4)
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 (EMPR,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*DATAN2(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 (15H1LAYER NUMBER =,I3,15H    ITERATION =,I3,/,
     1 59H ***********************************************************,/)
 2005 FORMAT (12H1LOAD CASE =,I3,15H    ITERATION =,I3,/,
     1 59H ***********************************************************,/)
 2010 FORMAT ('0  NP        DELTA-X        DELTA-Y         X-DISP',
     1 '       Y-DISP          TOTAL  NP',/)
 2050 FORMAT (I5,5F14.4,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 (I5,10F11.3,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       SLMAX',
     4 '    SLPRES  ELE',/)
 2250 FORMAT (I5,10F11.3,I5)
CXXXXXXXXXXXXX
C2500 FORMAT (I5,3F10.4)
C2550 FORMAT (I5,2F10.4)
      END
      SUBROUTINE LAYOUT  (MAXB,ID,INP,NCEL,NCNP,NOMEL,NOMNP,NPHUMP,NLDP,
     1 EMPR,X,Y,BULK,SHEAR,POIS,SLMAX,DISP,STRESS,STRAIN,XCP,YCP,NUMELT,
     2 NUMNPT,NUMCEL,NUMCNP,NFEL,NUMMAT,NUMLD,NLAY,NEQ,NEQB,MBAND,PATM,
     3 MTMN17,NMXEQB,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),SLMAX(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
      NMXEQB=MTMN17/(MBAND+2)/2
      NEQB=NMXEQB
      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
      SINITX(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)
      II1=NPHUMP(LN,1)
      II2=NPHUMP(LN,2)
      II3=NPHUMP(LN,3)
      II4=NPHUMP(LN,4)
      IF ((X(II2)-X(II1)) .NE. 0.0)TEMP1=(Y(II2)-Y(II1))/(X(II2)-X(II1))
      IF ((X(II3)-X(II2)) .NE. 0.0)TEMP2=(Y(II3)-Y(II2))/(X(II3)-X(II2))
      IF ((X(II4)-X(II3)) .NE. 0.0)TEMP3=(Y(II4)-Y(II3))/(X(II4)-X(II3))
      DO 770 N=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(II1)) GO TO 731
      IF (XCP(N) .LE. X(II2)) GO TO 732
      IF (XCP(N) .LE. X(II3)) GO TO 733
      IF (XCP(N) .LE. X(II4)) GO TO 734
      SLOPE=0.0
      HT=Y(II4)-YCP(N)
C     WRITE(6,44)Y(II4),YCP(N)
C  44 FORMAT(3X,'Y(II4)',2X,F10.2,3X,'YCP(N)',2X,F10.2)
      GO TO 740
  731 SLOPE=0.0
      HT=Y(II1)-YCP(N)
      GO TO 740
  732 SLOPE=TEMP1
      HT=Y(II1)+(XCP(N)-X(II1))*SLOPE-YCP(N)
      GO TO 740
  733 SLOPE=TEMP2
      HT=Y(II2)+(XCP(N)-X(II2))*SLOPE-YCP(N)
      GO TO 740
  734 SLOPE=TEMP3
      HT=Y(II3)+(XCP(N)-X(II3))*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 23HNODAL 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 35H0BAND WIDTH************************,I4,/,
     2 35H0NUMBER OF EQUATIONS***************,I4,/,
     3 35H0NUMBER 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,I6,10X,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 (35H0NUMBER OF LAYER IN FOUNDATION*****,I4,/,
     1 35H0ELEVATION OF RIGID BOUNDARY*******,F8.3,/)
 2220 FORMAT (47H0LAYER  MAT. NO.  INCLUSIVE ELEMENTS ELEVATION,/)
 2230 FORMAT (I5,3I10,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)
*
C
C     NEW ADDITION COMMON STATEMENT IN 1983 FOR JOINT SLIPPING BEHAVIOR
C
*     Expand array dimension statements from 30 to 99.
*          SCM  5-22-89
*
      COMMON/XJOINT/DSTRN(99),XLONG(99),JOINT
C
*     Add common statement for output control.  SCM 3-18-89
*
      COMMON /OUTPUT/ IWRT,IPLOT,MAXIT,JCON
*
      DATA NLIST,PLIST /5,4HSTEE,4HALUM,4HCONC,4HPLAS,4HBASI,4HL
     1                  4HINUM,4HRETE,4HTIC ,4HC    /
      DATA DESIGN, ANALYS, STOP/4HDESI,4HANAL,4HSTOP/
      DATA SOLVE/4HBURN,4HF.E.,4HF.E.,4HS    ,4HAUTO,4HUSER/
      DATA XPLOT /4HPLTD/
C
*       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')
*
*       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
        ELSEIF(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
C
        NPROB = NPROB + 1
        WRITE(6,2100) NPROB,HED,XMODE,SOLVE(LEVEL),SOLVE(LEVEL+3),
       1              PLIST(NPIPE),PLIST(NPIPE+5)
C
C       READ OR ESTABLISH PIPE PROPERTIES FROM PIPE LIBRARY
C
        ICOME = 1
        GO TO 300
120     CONTINUE
C
C       READ SOIL AND SYSTEM PROPERTIES FROM SOLUTION LIBRARY
C
        GO TO 400
130     CONTINUE
C
C       SOLVE SYSTEM FOR PIPE RESPONSES, ITERATE AS REQUIRED.
C
        IEXIT = 1
140     ICOME = 2
```

```
      GO TO 400                                                         ACAN0118
150   CONTINUE                                                          ACAN0119
      GO TO 300                                                         ACAN0120
160   CONTINUE                                                          ACAN0121
*                                                                       ACAN0122
*     Delete Unit 30 output from ACANDE and add to RESOUT.              ACAN0123
*          SCM  3-03-89                                                 ACAN0124
*                                                                       ACAN0125
*     IF (IEXIT.GE.0 .AND. HED(15).EQ.XPLOT) WRITE(30) NPPT,NINC,IA,((REACAN0126
*    1SULT(N,J),N=1,12),J=1,NPPT)                                       ACAN0127
*                                                                       ACAN0128
      IF(IEXIT.NE.1) GO TO 140                                          ACAN0129
C                                                                       ACAN0130
C     RETURN TO READ A NEW PROBLEM                                      ACAN0131
C                                                                       ACAN0132
      GO TO 100                                                         ACAN0133
C                                                                       ACAN0134
C     ...                                                               ACAN0135
C     PIPE LIBRARY                                                      ACAN0136
C                                                                       ACAN0137
300   GO TO (301,302,303,304,305) , NPIPE                               ACAN0138
C                                                                       ACAN0139
 301  CALL STEEL (IA,ICOME,IEXIT,LEVEL,NINC,NPMAT,NPPT,PDIA,PIPMAT,     ACAN0140
     1            RESULT,SK,SM,XMODE)                                   ACAN0141
      GO TO 310                                                         ACAN0142
302   CALL ALUMIN(IA,ICOME,IEXIT,LEVEL,NINC,NPMAT,NPPT,PDIA,PIPMAT,     ACAN0143
     1            RESULT,SK,SM,XMODE)                                   ACAN0144
      GO TO 310                                                         ACAN0145
303   CALL CONCRE(IA,ICOME,IEXIT,LEVEL,NINC,NPMAT,NPPT,PDIA,PIPMAT,     ACAN0146
     1            RESULT,XMODE)                                         ACAN0147
      GO TO 310                                                         ACAN0148
304   CALL PLASTI(IA,ICOME,IEXIT,LEVEL,NINC,NPMAT,NPPT,PDIA,PIPMAT,     ACAN0149
     1            RESULT,SK,SM,XMODE)                                   ACAN0150
      GO TO 310                                                         ACAN0151
305   CALL BASIC (IA,ICOME,IEXIT,LEVEL,NINC,NPMAT,NPPT,PDIA,PIPMAT,     ACAN0152
     1            RESULT,XMODE)                                         ACAN0153
C                                                                       ACAN0154
 310  CONTINUE                                                          ACAN0155
      GO TO (120,160) , ICOME                                           ACAN0156
C                                                                       ACAN0157
C     ...                                                               ACAN0158
C     SOLUTION LEVEL LIBRARY                                            ACAN0159
C                                                                       ACAN0160
 400 GO TO (401,402,402), LEVEL                                         ACAN0161
C                                                                       ACAN0162
401   CALL BURNS (IA,ICOME,IEXIT,NINC,NPMAT,NPPT,PDIA,PIPMAT,RESULT,    ACAN0163
     1            SK,SM)                                                ACAN0164
      GO TO 410                                                         ACAN0165
*                                                                       ACAN0166
*     Change subroutine call parameter LEVEL to LCODE for canned        ACAN0167
*          mesh additions.   SCM  3-18-89                               ACAN0168
*                                                                       ACAN0169
 402  CALL PRHERO(IA,ICOME,IEXIT,LCODE,NINC,NPMAT,NPPT,PDIA,PIPMAT,     ACAN0170
     1            RESULT,SK,SM)                                         ACAN0171
C                                                                       ACAN0172
410   GO TO (130,150),ICOME                                             ACAN0173
C                                                                       ACAN0174
C     ERROR EXIT                                                        ACAN0175
C                                                                       ACAN0176
500   WRITE(6,3000) XMODE,LEVEL,PTYPE                                   ACAN0177
      STOP 10                                                           ACAN0178
C                                                                       ACAN0179
C     NORMAL EXIT                                                       ACAN0180
C                                                                       ACAN0181
  900 WRITE(6,6000)                                                     ACAN0182
      STOP                                                              ACAN0183
C                                                                       ACAN0184
C     FORMATS                                                           ACAN0185
C                                                                       ACAN0186
 1000 FORMAT (A4,3X,I1,1X,A4,3X,15A4,2I2)                               ACAN0187
C                                                                       ACAN0188
*     Revise title page of Unit 6 output file for 1989 version.         ACAN0189
*          SCM 3-10-90                                                  ACAN0190
*                                                                       ACAN0191
 2000 FORMAT(1H1,//                                                     ACAN0192
     X,/12X,57H     CCCC  AA     N    N DDDD   EEEEE      8888   9999ACAN0193
     X,/12X,57H    CC     AAA    NN   N DD  D  EE          88  8 99  9ACAN0194
     X,/12X,56H   CC     AA A   NNN  N  DD  D EEEE DDD  88888   9999 ACAN0195
     X,/12X,55H  CC     AAAAA  NN N N  DD  D  EE          88  8     9 ACAN0196
     X,/12X,54H  CCCC  AA   A NN  NN  DDDDD  EEEEE       8888      9  ACAN0197
     X,/                                                                ACAN0198
     X,/12X,52HIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII    ACAN0199
     X,/12X,51H           IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII    ACAN0200
     X,/12X,50H                                               III    ACAN0201
     X,/12X,49H             (C)ULVERT (AN)ALYSIS AND (DE)SIGN  II    ACAN0202
     X,/12X,48H                                               I    ACAN0203
     X,/                                                                ACAN0204
     X,/12X,35H                      VERSION  1.0                   ACAN0205
     X,/12X,34H                       MARCH 1990                    ACAN0206
     X,/                                                                ACAN0207
     X,/12X,47H          PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY        ACAN0208
     X,/12X,46H            THE FEDERAL HIGHWAY ADMINISTRATION         ACAN0209
     X)                                                                 ACAN0210
 2010 FORMAT(1H ,//                                                     ACAN0211
     X,/12X,27H        PIPE TYPE LIBRARIES:                             ACAN0212
     X,/12X,33H             ALUMINUM (CORRUGATED)                       ACAN0213
     X,/12X,33H             BASIC (ANALYSIS ONLY)                       ACAN0214
     X,/12X,46H                   (UNCONNECTED ELEMENT OPTION)          ACAN0215
     X,/12X,33H             CONCRETE (REINFORCED)                       ACAN0216
     X,/12X,32H             PLASTIC (SOLID WALL)                        ACAN0217
     X,/12X,30H             STEEL (CORRUGATED)                          ACAN0218
     X,/12X,40H                   (SLOTTED JOINT OPTION)                ACAN0219
     X,/                                                                ACAN0220
     X,/12X,23H        SOLUTION LEVELS:                                 ACAN0221
     X,/12X,43H             LEVEL 1: ELASTIC THEORY - BURNS             ACAN0222
     X,/12X,42H             LEVEL 2: FINITE ELEMENT - AUTO              ACAN0223
     X,/12X,51H                      CIRCULAR AND ELLIPTICAL (CAN1)     ACAN0224
     X,/12X,51H                      CONCRETE BOX CULVERT  (CANBOX)     ACAN0225
     X,/12X,51H                      2 OR 3 SEGMENT ARCHES (CANARI)     ACAN0226
     X,/12X,42H             LEVEL 3: FINITE ELEMENT - USER              ACAN0227
     X)                                                                 ACAN0228
*                                                                       ACAN0229
2100  FORMAT(1H1,/// 10X,19H *** PROBLEM NUMBER,I3,4H ***///10X,15A4/// ACAN0230
     X 10X, 21H EXECUTION MODE ..... ,A4 //                             ACAN0231
     X 10X, 21H SOLUTION LEVEL ..... ,2A4 //                            ACAN0232
     X 10X, 21H CULVERT TYPE ....... ,2A4)                              ACAN0233
3000  FORMAT(////28H *** FATAL ERROR IN DATA *** //  9H MODE IS ,A4,4X, ACAN0234
     X10H LEVEL IS ,I2,4X,14H PIPE-TYPE IS ,A4)                         ACAN0235
 6000 FORMAT (////// 10X,40H * * * * NORMAL EXIT FROM CANDE * * * *    )ACAN0236
      END                                                               ACAN0237
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ACAN0001
*     CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM                AFUN0002
*                     VERSION 1.0, MARCH 1990                           AFUN0003
*              PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY                    AFUN0004
*                THE FEDERAL HIGHWAY ADMINISTRATION                     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
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
*     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),PILIST(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 PILIST/
     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/
C
      GO TO (100,200),ICOME
C
C     FIRST ENTRY ESTABLISHES PIPE PROPERTIES, AND SAFETY FACTORS
  100 READ(5,1000) NONLIN,PDIA,PE,PNU,PYIELD,PELONG,PDEN,PE2
      IF(PE.LE.0.0) PE = E
      IF(PNU.LE. 0.0) PNU = XNU
      IF(PYIELD.LE.0.0) PYIELD = YIELD
      IF(PELONG.LT.PYIELD*(1.-PNU**2)/PE) PELONG = ELONG
      IF(NONLIN.NE.1 .AND. NONLIN.NE.2) NONLIN = 0
      IF(NONLIN.EQ.2 .AND. PE2.LE.0.0) PE2 = E2
C
      WRITE(6,2000) PDIA,PE,PNU,PYIELD,PELONG,PDEN,NONLIN
      IF(NONLIN.EQ.2) WRITE(6,2001) PE2
      PIMIN = PDIA**2/(PE*FF)
      PHINGE(1) = 0.0
      IF(XMODE.EQ.DESIGN) GO TO 110
      READ(5,1010) PA,PI,PS
      WRITE(6,2010) PA,PI,PS
      GO TO 130
C
  110 READ(5,1010) (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
      PAMIN = GAGE20
      PA = PAMIN
      PI = PIMIN
C     NOTE...SECTION MODULUS FOR ALUMINUM CORRUGATIONS HAS THE RELATION,
      PS = PI/(1.417*SQRT(PI/PA) + 0.450*PA)
      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
C
      DO 145 N=1,NPPT
      RESULT(14,N) = PA
      RESULT(15,N) = PI
  145 RESULT(16,N) = PI/PS
      ICON = 0
      PESUM = PE
      RETURN
C
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
  200 IEXIT = 0
C
      IF(NONLIN.EQ.1 .AND. IA.EQ.NINC) CALL HINGE(PIPMAT,NPMAT,PS,PDIA,
     X PYIELD,SM,SK,RESULT,NPPT,PHINGE)
```

```
      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)
      PCRIT = 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
      IF(ABS((PANEW-PA)/PANEW).LT.0.05) ICHKA = 1
      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., OR (3) STRAIN
C     RUPTURE IN THE OUTER FIBERS.
C
      PINEWD = PFS(2)*DISP*PI/(0.2*PDIA)
      PINEWB = (PFS(3)*PAVG)**2*PBUCK
      PINEWS = PFS(4)*EPSMAX*PI/PELONG
*
*     Revise intrinsic function to generic form.
*         SCM 3-10-90
*
      PINEW = MAX(PINEWD,PINEWB,PINEWS)
*
      IF(PINEW.LT.PIMIN) PINEW = PIMIN
      IF(ABS((PINEW-PI)/PINEW).LT.0.05) ICHKI = 1
      PI = PINEW
      PS = PI/(1.417*SQRT(PI/PA) + 0.450*PA)
C
C     UPDATE SECTION PROPERTIES AND CHECK FOR CONVERGENCE.
C
      IF(ICHKA*ICHKI.EQ.1) GO TO 250
      IF(ITER.GT.5) WRITE(6,2200) ITER,PA,PI,PS
      IF(ITER.GT.5) GO TO 300
      IEXIT = -2
      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,NUMCP
      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.4) NGAGE= 14-2*J+J/4
      IF(I.EQ.5) NGAGE= 0
      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
      IJBEST = IJ
      NGBEST = NGAGE
280   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),BXB,CPLIST(IBEST+5),NGBEST
      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 = PCRIT/PAVG
      STRNR = PELONG/EPSMAX
      BENDR = PYIELD/BMAX
      HANDR = PI/PIMIN
      WRITE(6,2900) IA,CIRCR,DISPR,BUCKR,STRNR,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
      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 /
     X 17X,34H NONLIN=2, IMPLIES BILINEAR CURVE  /)
2001  FORMAT( /15X,36H MODULUS OF UPPER BI-CURVE .........,E15.5 /)
2010  FORMAT(// 10X, 31H SECTION PROPERTIES OF PIPE ...  //
     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 THRUST STRESS YIELDING ............ , F10.5 //
     X 15X,36H DEFLECTION COLLAPSE,(0.2*D) ....... , F10.5 //
     X 15X,36H ELASTIC BUCKLING .................. , F10.5 //
     X 15X,36H OUTER FIBER RUPTURE ............... , 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 E
     XVALUATED, / 10X,37H WHERE THE 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 //)
2300  FORMAT(1H1,//10X,19H DESIGN RESULTS AT ,I2,11H ITERATIONS //
     X 15X, 35H *** REQUIRED THRUST AREA ......... , F10.5 //
     X 15X, 35H *** REQUIRED MOM. OR INERTIA ..... , F10.5 //
     X 15X, 35H *** REQUIRED SECTION MODULUS ..... , F10.5 ////
     X 10X, 52H THE FOLLOWING ALUMINUM CORRUGATED SECTIONS MEET THE  /
     X 10X, 37H ABOVE REQUIREMENTS WITH MINIMUM AREA  //
     X15X,12H CORRUGATION,3X,5H GAGE,3X,12H THRUST AREA,3X,13H MOM. INER
     XTIA /)
2400  FORMAT(17X,3A4,3X,22H (NO ADEQUATE SECTION) )
2500  FORMAT(17X,3A4,6X,I2,5X,F10.5,5X,F10.5)
2600  FORMAT(//// 10X,37H NO STANDARD CORRUGATION IS ADEQUATE. /
     X 10X, 45H AN ANALYSIS OF THE REQUIRED SECTION FOLLOWS. )
2700  FORMAT(//// 10X, 38H THE CORRUGATION WITH MINIMUM AREA AND /
     X10X,21H MOMENT OF INERTIA IS,3A4,5H GAGE,I3, /
     X 10X, 41H AN ANALYSIS OF THIS CORRUGATION FOLLOWS. )
2800  FORMAT(1H1,//10X,32H STRUCTURAL RESPONSES OF CULVERT ,
     X 3X, 12H LOAD STEP =  ,I2 //
     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 ,4X,9H M-STRESS,4X,9H T-STRESS,4X,9H S-STRESS ,
     X (//F13.2,5E13.3/F13.2,5E13.3))
2900  FORMAT(1H1,////10X,34H CALCULATED SAFETY FACTORS AT STEP, I3 ///
     X15X,50H THRUST STRESS ....... (YIELD-STRESS/MAX-STRESS) =,F10.2//
     X15X,50H DISPLACEMENT ......... (DIAMETER*0.2/MAX-DISP.) =,F10.2//
     X15X,50H ELASTIC BUCKLING ....... (P-CRITICAL/P-AVERAGE) =,F10.2//
     X15X,50H STRAIN RUPTURE .... (RUPTURE-STRAIN/MAX-STRAIN) =,F10.2///
     X 10X, 20H PERFORMANCE FACTORS  ///
     X15X,50H BENDING STRESS ...... (YIELD-STRESS/MAX-STRESS) =,F10.2 //
     X15X,50H HANDLING REQUIREMENT ............. (FF*EI/D**2) =,F10.2 /)
2950  FORMAT(/// 10X, 27H PLASTIC HINGING EVALUATION ///
     X 15X,42H   RATIO OF LOAD AT INITIAL YIELDING ..... , F10.5 //
     X 15X,42H   RATIO OF MAXIMUM HINGE PENETRATION .... , F10.5 //
     X 15X,42H   RATIO OF CIRCUMFERENCE IN YIELDING .... , F10.5 //
     X 15X,42H   RATIO OF MAX-DISP. DUE TO YIELDING .... , F10.5 //
     X 15X,42H   RATIO OF MAX-STRAIN/YIELD-STRAIN ...... , F10.5 //)
2990  FORMAT(1H1,///10X,40H NONLINEAR CHARACTERISTICS,  LOAD STEP =,I3//
     X5X,73H NODE        INNER-FIBER        OUTER-FIBER       STRAIN RATIO
     X  FRACTION OF /21X,57HSTRAIN             STRAIN        MAX-TO-YIELD
     X WALL YIELDED // (5X,I5,2E17.5,7X,F10.5,7X,F10.5))
      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 ANISP(C11, C12, C22, C33, PI)
C
C     THIS SUBROUTINE TRANSFORMS ORTHOTROPIC PROPERTIES FROM THEIR
C     PREFERED DIRECTIONS TO GLOBAL COORDINATES
C
*     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),NON(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)
C
C     GENERAL SUBROUTINE FOR ANALYSING ARBITRARY PIPES. NO EVALUATION
C     OF STRUCTURAL RESPONSES ARE MADE.
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 PIPMAT(5,99),RESULT(20,99)
*
      GO TO (100,200),ICOME
C
C     FIRST ENTRY READS PIPE PROPERIES.
C
  100 IF(NPMAT.EQ.0) RETURN
*
*     Add NEXTRA parameter to read statement for input of
*          unconnected elements.     MGK  1-17-89
*
      READ(5,1000) PDIA,NEXTRA
      NTOTAL = NPMAT + NEXTRA
*
  150 READ(5,1500) ISEQ1,ISEQ2,(PIPMAT(J,ISEQ1),J=1,5)
      IF(ISEQ1.LE.0) GO TO 400
*
*     Revise statements to substitute NTOTAL for NPMAT to
*          implement unconnected element option.  MGK 1-17-89
*
      IF(ISEQ2.GT.NTOTAL) GO TO 400
*
      IF(ISEQ2.LE.ISEQ1) ISEQ2=ISEQ1
      DO 175 I=ISEQ1,ISEQ2,1
      DO 175 J=1,5
  175 PIPMAT(J,I) = PIPMAT(J,ISEQ1)
*
      IF(ISEQ2.EQ.NTOTAL) GO TO 190
*
      GO TO 150
*
*     Revise output statement to use NTOTAL instead of NPMAT for
*          unconnected element option. MGK  1-17-89
*
  190 WRITE(6,2000) PDIA, (J, (PIPMAT(I,J),I=1,5),J=1,NTOTAL)
      IF(NTOTAL.GT.99) GO TO 400
*
      RETURN
C
C     SECOND ENTRY PRINTS RESULTS
C
  200 CONTINUE
C
C     CHECK TO SEE IF NONLINEAR SOIL MODEL HAS CONVERGED ...
C
      IF(NINC .GT. 0) GO TO 210
      IEXIT = -1
      RETURN
C
  210 CONTINUE
C
C     OUTPUT FINITE ELEMENT FIELD DATA (IF REQUESTED)
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
      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,
     X7H MOMENT ,4X,8H DENSITY / 3X,7H NUMBER ,4X,8H MODULUS ,6X,
     X 6H RATIO ,7X,5H AREA ,4X,8H INERTIA /(5X,I5,5E12.4))
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 ,  (//F13.2,5E13.3/F13.2,2E13.3))
3000  FORMAT( ///27H FATAL DATA ERROR IN BASIC /
     X44H 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
*     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
      STORE(2)=-SA
C
C     NEW ADDITION STATEMENT IN 1983 FOR JOINT SLIPPING BEHAVIOR.
C
      STORE(3)=XL
C
      EB = CP(1,1)
      GNUB = CP(1,2)
      AB = CP(1,3)
      XIB = CP(2,1)
      EB = EB/(1.0-GNUB*GNUB)
      DU=EB*AB/XL
      C1(1,1)=DU
      C1(4,1)=-DU
      C1(1,4)=-DU
      C1(4,4)=DU
      XL=0.5*XL
      RQ(1)=0.0
      RQ(3)=0.0
      D=.25/XL
      RQ(2)=-D
      RQ(4)=D
      ZQ(2)=D
      ZQ(4)=D
      D=D/XL
      ZQ(1)=D
      ZQ(3)=-D
      D=8.0*XL*EB*XIB
      DO 120 L=1,4
      LS=L+1+L/3
      DO 120 J=1,4
      JS=J+1+J/3
C
  120 C1(LS,JS)=D*(RQ(J)*RQ(L)+3.       *ZQ(J)*ZQ(L))
C
C     LOCAL UNKNOWNS TRANSFORMED TO GLOBAL UNKNOWNS
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,NQ,I,NELEM,NPT,MN)
```

```
C
C     THIS SUBROUTINE CALCULATES BEAM FORCES, SHEARS AND MOMENTS
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),NQ(NPT)
      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/SYSTEM/B(240),A(240,120)
C
C     NEW ADDITION COMMON STATEMENT IN 1983 FOR JOINT SLIPPING BEHAVIOR
C
*     Expand array dimension statements from 30 to 99.
*         SCM  5-22-89
*
      COMMON/XJOINT/DSTRN(99),XLONG(99),JOINT
C
C
C     STIFFNESS MATRIX TIMES DISPLACEMENT
C
      COST=STORE(1)
      SINT=STORE(2)
C
C     NEW ADDITION STATEMENT IN 1983 FOR JOINT SLIPPING BEHAVIOR.
C
      XL=STORE(3)
C
      ZY(5)=ZY(4)
      ZY(4)=ZY(3)
      N1 = NOD(1,I)
      ND=NQ(N1)/100+2
      ZY(3) = B(ND)
      DO 100 K=1,6
      DU=0.0
      DO 90 L=1,6
   90 DU=DU+C1(K,L)*ZY(L)
  100 STORE(K)=DU
      RQ(1)=-STORE(1)*COST-STORE(2)*SINT
      RQ(2)=STORE(1)*SINT-STORE(2)*COST
      RQ(3) = STORE(3)
      RQ(4) = STORE(1)
      ZQ(1)=STORE(4)*COST+STORE(5)*SINT
      ZQ(2)=-STORE(4)*SINT+STORE(5)*COST
      ZQ(3) = -STORE(6)
      ZQ(4) = STORE(2)
C
C     NEW ADDITION STATEMENTS IN 1983 FOR JOINT SLIPPING BEHAVIOR.
C
      IF(JOINT.EQ.0)RETURN
      UL1=COST*ZY(1)+SINT*ZY(2)
      UL2=COST*ZY(4)+SINT*ZY(5)
      DSTRN(MN)=(UL2-UL1)/XL
      XLONG(MN)=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 CANAR1 (LUDATA,KPUTCK,DENSTY)
C
C     ADDED TO CANDE IN 1989.
C
C     GENERATES TWO OR THREE SEGMENT ARCH MESH WITH CURVED OR
C         STRAIGHT SEGMENTS. DEVELOPED BY SYRO STEEL COMPANY.
*
*     Uncomment this statement in all program modules for double
*         precision version. SCM 3-10-90
*
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
      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),TPMESH(4)
      DIMENSION R(3),THETA(4),CHORD(3),ARCLEN(3),XCEN(3),YCEN(3),
     X    ANG(20),XPN(20),YPN(20),DELTA(3),NDEV(3)
      DATA PI/3.14159265/
      DATA L1H, IBLANK, PREP, STOP, XMOD/1HL,1H ,4HPREP,4HSTOP,4HMOD /
      DATA IZERO, IONE, FZERO ,XBLANK /0,1,0.0,4H    /
      DATA TPMESH /4HEMBA,4HTREN,4HHOMO,4H    /
      DATA INCLEV, INCMAX, NUMNP, NUMEL, NBPTC / 8,20,244,209,35 /
      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 /
      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 /
      DATA NODMDA / 245,0,    258,202,  265,302,  266,0,    279,202,
     X    286,302,  287,0,    300,202,  307,302 /
      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,
X           1.0,75.0,0.0,          4.0,75.0,0.79,         5.0,75.0,0.0,
X           6.0,75.0,0.0,          8.0,75.0,1.0,          2*0.0,1.26,
X           1.0,90.0,0.0,          4.0,90.0,0.79,         5.0,90.0,0.0,
X           6.0,90.0,0.0,          8.0,90.0,0.0,          2*0.0,1.26,
X           0.0,127.0,0.0,         2*0.0,1.26,            0.0,157.0,0.0,
X           2*0.0,1.26,            0.0,189.0,0.0,         2*0.0,1.26,
X           0.0,223.0,0.0,         2*0.0,1.26,            1.0,244.0,0.0,
X           0.0,244.0,1.0,         2*0.0,1.26,            3*0.0,
X           2*198.0,0.0,           2*0.0,1.00,            3*0.0,
X           2*194.0,0.0,           2*0.0,1.00,            3*0.0,
X           2*190.0,0.0,           2*0.0,1.00,            3*0.0 /
 DATA XYAND2/ 2*167.0,0.0,         2*0.0,1.00,            3*0.0,
X           2*164.0,0.0,           2*0.0,1.00,            3*0.0,
X           2*161.0,0.0,           2*0.0,1.00,            3*0.0,
X           2*158.0,0.0,           2*0.0,1.00,            3*0.0,
X           2*138.0,0.0,           2*0.0,1.00,            3*0.0,
X           2*136.0,0.0,           2*0.0,1.00,            3*0.0,
X           2*134.0,0.0,           2*0.0,1.00,            3*0.0,
X           2*0.0,1.0,             3*0.0,                 236.0,216.0,1.0,
X           3*0.0,                 2*132.0,0.0,           2*0.0,1.00,
X           3*0.0,                 2*130.0,0.0,           2*0.0,1.00,
X           3*0.0,                 2*128.0,0.0,           2*0.0,1.00,
X           3*0.0,                 2*109.0,0.0,           2*0.0,1.00,
X           3*0.0,                 2*115.0,0.0,           2*0.0,1.00,
X           3*0.0,                 2*118.0,0.0,           2*0.0,1.00,
X           3*0.0,                 2*103.0,0.0,           2*0.0,1.00,
X           3*0.0,                 2*97.0,0.0,            2*0.0,1.00,
X           3*0.0,                 2*91.0,0.0,            2*0.0,1.00 /
 DATA XYANDA / 3*0.0,              0.0,245.0,1.0,         15.0,245.0,1.26,
X           3*0.0,                 0.0,266.0,1.0,         15.0,266.0,1.26,
X           3*0.0,                 0.0,287.0,1.0,         15.0,287.0,1.26/
 DATA LNOD1 /
X 1,194,198,0,0,1,1,0,0,0,0,        2,190,194,0,0,2,1,0,0,0,0,
X 3,167,190,0,0,3,1,0,0,0,0,        4,164,167,0,0,4,1,0,0,0,0,
X 5,161,164,0,0,5,1,0,0,0,0,        6,158,161,0,0,6,1,0,0,0,0,
X 7,138,158,0,0,7,1,0,0,0,0,        8,136,138,0,0,8,1,0,0,0,0,
X 9,134,136,0,0,9,1,0,0,0,0,        10,132,134,0,0,10,1,0,0,0,0,
X 11,130,132,0,0,11,1,0,0,0,0.      12,128,130,0,0,12,1,0,0,0,0,
X 13,118,128,0,0,13,1,0,0,0,0,      14,115,118,0,0,14,1,0,0,0,0,
X 15,109,115,0,0,15,1,0,0,0,0,      16,103,109,0,0,16,1,0,0,0,0,
X 17,97,103,0,0,17,1,0,0,0,0,       18,91,97,0,0,18,1,0,0,0,0,
X 19,80,91,0,0,19,1,0,0,0,0 /
 DATA LNOD2 /
X 20,1,2,17,16,1,1,0,1,1,0,         34,16,17,32,31,1,1,0,1,1,0,
X 48,31,32,47,46,1,1,0,1,1,0,       62,46,47,62,61,1,1,0,1,1,0,
X 76,61,62,77,76,1,1,0,1,1,0,
X 79,64,65,80,79,2,1,0,1,1,0,       81,66,67,82,81,1,1,0,1,1,0,
X 90,83,84,96,0,3,2,0,0,0,0,        91,96,84,121,102,3,2,0,0,0,0,
X 92,84,85,122,121,3,2,0,1,1,0,     98,102,121,108,0,3,3,0,0,0,0,
X 99,108,121,151,114,3,3,0,0,0,0,  100,121,122.152,151,3,3,0,1,1,0,
X 106,114,151,150,0,3,4,0,0,0,0,   107,150,151,183,0,3,4,0,0,0,0,
X 108,150,183,182,0,3,4,0,0,0,0,   109,151,152,184,183,3,4,0,1,1,0,
X 115,182,183,217,216,3,7,0,1,1,0,122,216,217,238,237,3,8,0,1,1,0,
X 129,80,81,94,93,3,2,0,1,1,0,     132,93,94,100,99,3,2,0,1,1,0,
X 135,99,100,106,105,3,3,0,1,1,0, 138,105,106,112,111,3,3,0,1.1.0,
X 141,111,112,148,117,3,4,0,0,0,0,142,112,113,149,148,3,4,0,1,1,0,
X 144,117,148,147,120,3,4,0,0,0,0,145,148,149,181,147,3,4,0,0,0,0,
X 146,149,150,182,181,3,4,0,0,0,0,147,200,196,197,201,3,5,0,0,0,0,
X 148,196,192,193,197,3,5,0,0,0,0,149,192,169,170,193,3,5,0,0,0,0,
X 150,169,166,171,170,3,5,0,0,0,0,151,166,163,172,171,3,5,0,0,0,0,
X 152,163,160,173,172,3,5,0,0,0,0,153,160,140,174,173,3,5,0,0,0,0,
X 154,140,141,175,174,3,5,0,1,1,0,158,144,145,146,178,3,5,0,0,0,0 /
 DATA LNOD3 /
X 159,145,120,147,146,3,5,0,0,0,0,160,201,197,203,202,3,6,0,0,0,0,
X 161,197,193,204,203.3,6,0,0,0,0,162,193,170,205,204,3,6,0,0,0,0,
X 163,170,171,206,205,3,6,0,1,1,0,170,177,178,179,212,3,6,0,0,0,0,
X 171,178,146,180,179,3,6,0,0,0,0,172,146,147,181,180,3,6,0,0,0,0,
X 173,202,203,225,224,3,7,0,1,1,0,182,211,212,213,233,3,7,0,0,0,0,
X 183,212,179,214,213,3,7,0,0,0,0,184,179,180,215,214,3,7,0,0,0,0,
X 185,180,181,182,215,3,7,0,0,0,0,186,182,216,215,0,3,7,0,0,0,0,
X 187,213,234,233,0,3,8,0,0,0,0,  188,213,214,235,234,3,8,0,1,1,0 /
 DATA LNOD4 /
X 191,198,200,199,0,1,5,1,0,0,0,  192,194,196,195,0,2,5,1,0,0,0,
X 193,190,192,191,0,3,5,1,0,0,0,  194,167,169,168,0,4,5,1,0,0,0,
X 195,164,166,165,0,5,5,1,0,0,0,  196,161,163,162,0,6,5,1,0,0,0,
X 197,158,160,159,0,7,5,1,0,0,0,  198,138,140,139,0,8,5,1,0,0,0,
X 199,136,141,137,0,9,5,1,0,0,0,  200,134,142,135,0,10,5,1,0,0,0,
X 201,132,143,133,0,11.5,1,0,0,0, 202,130,144,131,0,12,5,1,0,0,0,
X 203,128,145,129,0,13,5,1,0,0,0, 204,118,120,119,0,14,4,1,0,0,0,
X 205,115,117,116,0,15,4,1,0,0,0, 206,109,111,110,0,16,3,1,0,0,0,
X 207,103,105,104,0,17,3,1,0,0,0, 208,97,99,98,0,18,2,1,0,0,0,
X 209,91,93,92,0,19,2,1,0,0,0 /
 DATA LNODA /
X 191,224,225,246,245,3,9,0,1,1,0,
X 211,245,246,267,266,3;10,0,1,1,0,
X 231,266,267,288,287,3,11,0,1,1,0 /
 DATA NODBC1 / 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,30,45,60,75,90,
X          127,157,189,223,224,202,201,200,199,198,76,61,46,31,16 /
 DATA NODBCA / 245,265,266,286,287,307 /
C
C     INITIALIZE VARIABLES
C
      DO 1,J=1,3
           R(J) = 0.0
           THETA(J) = 0.0
           CHORD(J) = 0.0
           ARCLEN(J) = 0.0
           XCEN(J) = 0.0
           YCEN(J) = 0.0
    1      DELTA(J) = 0.0
      DO 2,J=1,20
           ANG(J) = 0.0
           XPN(J) = 0.0
    2      YPN(J) = 0.0
      DO 3,J=1,307
           X(J) = 0.0
    3      Y(J) = 0.0
      HPRESS = 0
      INCADD = 0
      PINC = 0
      XDIV = 0
      YDIV = 0
C
C     READ DATA CARDS THAT SPECIFY THE CANNED MESH
C
      READ(5,1000) WORD,TITLE,WORD1
      READ(5,1005) IPLOT,IWRT,MGENPR,NINC,HTCOVR,DENSTY,TRNDEP,TRNWID,
     X             TRNSLP
      READ(5,1010) RISE,HFSPAN,SDRISE,FTDEP,FTWIDO,FTWIDI,SPCFAC
      READ(5,1020) (R(J),THETA(J),J=1,3),THETA(4),NTN,NCN
C
C     CONVERT INPUT TO PROPER UNITS
C
      HTCOVR = HTCOVR*12
      TRNDEP = TRNDEP*12
```

```
      TRNWID = TRNWID*12
C
C     CHECK INPUT PARAMETERS
C
      DO 4,MESHTP=1,3
           IF(WORD.EQ.TPMESH(MESHTP)) GO TO 5
    4      CONTINUE
      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(HTCOVR.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.HTCOVR+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)
                KPUTCK = KPUTCK + 1
                END IF
           END IF
      IF(FTWIDI.LT.0.1*HFSPAN .OR. FTWIDI.GT.0.5*HFSPAN) THEN
           WRITE(6,2035)
           KPUTCK = KPUTCK + 1
           END IF
      IF(FTDEP.LT.0.05*RISE .OR. FTDEP.GT.0.3*RISE) THEN
           WRITE(6,2040)
           KPUTCK = KPUTCK + 1
           END IF
      IF(SPCFAC.LT.0.0 .OR. SPCFAC.GT.1.3) THEN
           WRITE(6,2042)
           KPUTCK = KPUTCK + 1
           END IF
      IF(SDRISE.GT.0.0) THEN
           IF(NTN.GT.17 .OR. NTN.LT.0) THEN
                WRITE(6,2045)
                KPUTCK = KPUTCK + 1
                END IF
           IF(NCN.GT.18 .OR. NCN.LT.0) THEN
                WRITE(6,2050)
                KPUTCK = KPUTCK + 1
                END IF
           ELSE
           IF(NTN.GT.18 .OR. NTN.LT.0) THEN
                WRITE(6,2045)
                KPUTCK = KPUTCK + 1
                END IF
           IF(NCN.GT.19 .OR. NCN.LT.0) THEN
                WRITE(6,2050)
                KPUTCK = KPUTCK + 1
                END IF
           END IF
C
C     DETERMINE MESH SIZE BASED ON HTCOVR
C
      IF(HTCOVR.LE.0.4*RISE .AND. HTCOVR.GE.0.1*RISE) THEN
           JNMODA = 0
           JXYA = 0
           JLNODA = 0
           JBCA = 0
           KCOVER = 0
           END IF
      IF(HTCOVR.LE.0.7*RISE .AND. HTCOVR.GT.0.4*RISE) THEN
           JNMODA = 3
           JXYA = 3
           JLNODA = 1
           JBCA = 2
           KCOVER = 1
           INCLEV = 9
           NUMNP = 265
           NUMEL = 229
           NBPTC = 37
           END IF
      IF(HTCOVR.LE.1.0*RISE .AND. HTCOVR.GT.0.7*RISE) THEN
           JNMODA = 6
           JXYA = 6
           JLNODA = 2
           JBCA = 4
           KCOVER = 2
           INCLEV = 10
           NUMNP = 286
```

```
            NUMEL = 249
            NBPTC = 39
            END IF
      IF(HTCOVR.GT.1.0*RISE) THEN
            JNMODA = 9
            JXYA = 9
            JLNODA = 3
            JBCA = 6
            KCOVER = 3
            INCLEV = 11
            NUMNP = 307
            NUMEL = 269
            NBPTC = 41
            END IF
C
C     CONTATENATE DATA ARRAYS
C
      DO 10,NUM=1,107
            DO 10,NUM1=1,2
   10       NODMOD(NUM1,NUM) = NODMD1(NUM1,NUM)
      DO 20,NUM=1,55
            DO 20,NUM1=1,2
   20       NODMOD(NUM1,NUM+107) = NODMD2(NUM1,NUM)
      IF(JNMODA.GT.0) THEN
            DO 25,NUM=1,JNMODA
                  DO 25,NUM1=1,2
   25             NODMOD(NUM1,NUM+162) = NODMDA(NUM1,NUM)
            END IF
      DO 30,NUM=1,60
            DO 30,NUM1=1,3
   30       XYANDS(NUM1,NUM) = XYAND1(NUM1,NUM)
      DO 40,NUM=1,51
            DO 40,NUM1=1,3
   40       XYANDS(NUM1,NUM+60) = XYAND2(NUM1,NUM)
      IF(JXYA.GT.0) THEN
            DO 45,NUM=1,JXYA
                  DO 45,NUM1=1,3
   45             XYANDS(NUM1,NUM+111) = XYANDA(NUM1,NUM)
            END IF
      DO 50,NUM=1,19
            DO 50,NUM1=1,11
   50       LNOD(NUM1,NUM) = LNOD1(NUM1,NUM)
      DO 60,NUM=1,37
            DO 60,NUM1=1,11
   60       LNOD(NUM1,NUM+19) = LNOD2(NUM1,NUM)
      DO 70,NUM=1,16
            DO 70,NUM1=1,11
   70       LNOD(NUM1,NUM+56) = LNOD3(NUM1,NUM)
      IF(JLNODA.GT.0) THEN
            DO 75,NUM=1,JLNODA
                  DO 75,NUM1=1,11
   75             LNOD(NUM1,NUM+72) = LNODA(NUM1,NUM)
            DO 77,NUM=1,19
                  DO 76,NUM1=1,11
   76             LNOD(NUM1,NUM+72+JLNODA) = LNOD4(NUM1,NUM)
   77             LNOD(1,NUM+72+JLNODA) = 190 + JLNODA*20 + NUM
            GO TO 85
            END IF
      DO 80,NUM=1,19
            DO 80,NUM1=1,11
   80       LNOD(NUM1,NUM+72) = LNOD4(NUM1,NUM)
   85 DO 86,NUM=1,35
   86       NODBC(NUM) = NODBC1(NUM)
      IF(JBCA.GT.0) THEN
            DO 87,NUM=1,JBCA
   87       NODBC(NUM+35) = NODBCA(NUM)
            END IF
C
C     ESTABLISH SPACING FACTORS AROUND ARCH
C
      SPF = SPCFAC
      SPC = SPCFAC - 1
      SPVERT = 1.5*SPC*(RISE**2/(RISE**2+HFSPAN**2))
      SPHORZ = 0.75*SPC*(HFSPAN**2/(HFSPAN**2+RISE**2))
      DO 90,I=53,80,3
   90       XYANDS(3,I) = 1 + SPC
      DO 92,I=87,111,3
   92       XYANDS(3,I) = 1 + SPC
      XYANDS(3,82) =  1 + SPVERT
      XYANDS(3,49) = 1 + SPHORZ
      DO 93,I=113,119,3
   93       XYANDS(3,I) = 1 + SPHORZ
C
C**** BEGIN SOLUTION OF BEAM ELEMENT GEOMETRY *************************
C
C     CALCULATE CHORD AND ARC LENGTHS FOR CURVED SEGMENTS
C
      DO 120,J=1,4
  120       THETA(J) = THETA(J)*PI/180
      DO 150,J=1,3
            IF(ABS(R(J)).GT.0.0) THEN
                  CHORD(J) = 2*ABS(R(J))*SIN(THETA(J)/2)
                  ARCLEN(J) = ABS(R(J))*THETA(J)
            ELSE
                  DELTA(J) = THETA(J)
                  THETA(J) = 0.0
            END IF
  150 CONTINUE
C
C     ASSIGN BEGINNING COORDINATES
C
      X1 = 0.0
      Y1 = RISE
C
C     ASSIGN END COORDINATES FOR TWO SEGMENT PROBLEM
C
      IF(SDRISE.EQ.0.0) THEN
            X3 = HFSPAN
            Y3 = 0.0
            GO TO 180
C
C     ASSIGN END COORDINATES FOR THREE SEGMENT PROBLEM
C
      ELSE
            X4 = HFSPAN
            Y4 = 0.0
            Y3 = SDRISE
C
C     SOLVE INTERSECTION OF CURVED SIDE SEGMENT
C
            IF(ARCLEN(3).GT.0.0) THEN
                  X3 = X4+CHORD(3)*COS(PI/2+THETA(4)+THETA(3)/2)
C
C     SOLVE INTERSECTION OF STRAIGHT SIDE SEGMENT
C
            ELSE
```

```
                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)
      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)))) -
     X                    THETA(1)*(J-1)/(NTN-1)
                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)/
     X                   (NCN-NTN+1)
                YPN(J) = Y2 - CHORD(2)*SIN(DELTA(2))*(J-NTN)/
```

```
     X                  (NCN-NTN+1)
  220       CONTINUE
C
C     FIND BEAM COORDINATES - CURVED CORNER SEGMENT
C
      ELSE IF(ARCLEN(2).GT.0.0) THEN
           DO 230,J=NTN+1,NCN
                ANG(J) = ATAN(ABS((Y2-YCEN(2))/(X2-XCEN(2)))) -
     X                   THETA(2)*(J-NTN)/(NCN-NTN)
                XPN(J) = XCEN(2) + ABS(R(2))*COS(ANG(J))
                YPN(J) = YCEN(2) + ABS(R(2))*SIN(ANG(J))
  230       CONTINUE
      END IF
C
C     FIND BEAM COORDINATES - STRAIGHT SIDE SEGMENT
C
      IF(ARCLEN(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)/
     X                   (NSN-NCN+1)
                YPN(J) = Y3 - CHORD(3)*SIN(DELTA(3))*(J-NCN)/
     X                   (NSN-NCN+1)
  240       CONTINUE
C
C     FIND BEAM COORDINATES - CURVED SIDE SEGMENT
C
      ELSE IF(ARCLEN(3).GT.0.0 .AND. SDRISE.GT.0.0) THEN
           DO 250,J=NCN+1,NSN
                ANG(J) = ATAN(ABS((Y3-YCEN(3))/(X3-XCEN(3)))) -
     X                   THETA(3)*(J-NCN)/(NSN-NCN+1)
                XPN(J) = XCEN(3) + ABS(R(3))*COS(ANG(J))
                YPN(J) = YCEN(3) + ABS(R(3))*SIN(ANG(J))
  250       CONTINUE
      END IF
C
C**** END SOLUTION OF BEAM ELEMENT GEOMETRY *****************************
C
C     ASSIGN COORDINATES TO KEY NODAL POINTS
C
  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)
      Y(130) = YPN(12)
      X(128) = XPN(13)
      Y(128) = YPN(13)
      X(118) = XPN(14)
      Y(118) = YPN(14)
      X(115) = XPN(15)
      Y(115) = YPN(15)
      X(109) = XPN(16)
      Y(109) = YPN(16)
      X(103) = XPN(17)
      Y(103) = YPN(17)
      X(97) = XPN(18)
      Y(97) = YPN(18)
      X(91) = XPN(19)
      Y(91) = YPN(19)
C
      X(15) = 6*HFSPAN
      X(8) = HFSPAN + HFSPAN/4 + RISE/4
      X(6) = HFSPAN + FTWIDO
      X(5) = HFSPAN
      X(4) = HFSPAN - FTWIDI
      X(1) = 0.0
      Y(1) = -1.5*RISE
      Y(75) = -FTDEP
      Y(90) = 0.0
      X(237) = X(8)
      X(258) = X(8)
      X(279) = X(8)
      X(300) = X(8)
      XINC = (X(15)-X(8))/(1+1.26+1.26**2+1.26**3+1.26**4+1.26**5+
     X                     1.26**6)
      X(121) = X(8) + XINC
      X(151) = X(121)
      X(183) = X(121)
      X(217) = X(121)
C
      IF(KCOVER.EQ.0) Y(244) = RISE + HTCOVR
      IF(KCOVER.EQ.1) THEN
           YINC = HTCOVR/(1 + SPF + SPF**2 + SPF**3)
           Y(244) = RISE + YINC + SPF*YINC + (SPF**2)*YINC
           Y(245) = RISE + HTCOVR
      ELSE IF(KCOVER.EQ.2) THEN
           YINC = HTCOVR/(1 + SPF + SPF**2 + SPF**3 + SPF**4)
           Y(244) = RISE + YINC + SPF*YINC + (SPF**2)*YINC
           Y(245) = Y(244) + (SPF**3)*YINC
           Y(266) = RISE + HTCOVR
      ELSE IF(KCOVER.EQ.3) THEN
           IF(HTCOVR.GT.1.5*RISE) THEN
                YINC = 1.5*RISE/(1+SPF+SPF**2+SPF**3+SPF**4+SPF**5)
                Y(244) = RISE + YINC + SPF*YINC + (SPF**2)*YINC
                Y(245) = Y(244) + (SPF**3)*YINC
                Y(266) = Y(245) + (SPF**4)*YINC
                Y(287) = RISE + 1.5*RISE
                HPRESS = HTCOVR - 1.5*RISE
                INCADD = NINC - INCLEV
                IF(INCADD.LE.0) INCADD = -1
                NBPTC = NBPTC + IABS(INCADD)*21
                PINC = -(HPRESS*DENSTY)/(1728*IABS(INCADD))
                ELSE
                YINC = HTCOVR/(1+SPF+SPF**2+SPF**3+SPF**4+SPF**5)
                Y(244) = RISE + YINC + SPF*YINC + (SPF**2)*YINC
                Y(245) = Y(244) + (SPF**3)*YINC
                Y(266) = Y(245) + (SPF**4)*YINC
```

```
                Y(287) = RISE + HTCOVR
                END IF
            END IF
C
C           REVISE MESH FOR TRENCH CONDITION
C
            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
 310                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
 320                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
 330                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
 340                LNOD(7,I) = 1
                TRNDEP = 4*YDIV
            ELSE
                DO 350,I=27,38
                    LNOD(6,I) = 1
 350                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)
                    ELSE
                        DO 360,I=92,93
                            LNOD(6,I) = 1
 360                        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
 370                    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
 380                    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
C
C           REVISE MESH FOR HOMOGENEOUS MATERIAL
C
 400        IF(WORD.NE.TPMESH(3)) GO TO 440
            DO 410,I=20,72+JLNODA
 410            LNOD(6,I) = 1
C
C           RECORD MESH DOCUMENTATION PARAMETERS
C
 440        WRITE(6,4000) WORD
            IF(WORD.EQ.TPMESH(2)) WRITE(6,4010) TRNDEP/12,TRNWID/12,TRNSLP
            WRITE(6,4020)SPCFAC,HTCOVR/12,(HTCOVR-HPRESS)/12,DENSTY,(3+KCOVER)
            WRITE(6,4100) RISE,HFSPAN,SDRISE,FTDEP,FTWIDO,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)
```

```
              WRITE(6,4300) R(N),THETA(N)*180/PI,ARCLEN(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,IPLOT,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 = L1H
  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 = L1H
  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+JBCA
      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+JBCA .AND. HPRESS.EQ.0) LIMIT  = L1H
  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
```

```
      IF(HPRESS.GT.0) THEN
          LIMIT = IBLANK
          DO 550, N=1,IABS(INCADD)
          IF(N.EQ.IABS(INCADD)) LIMIT = L1H
          IF(INCADD.GT.0) THEN
              IADD = NINC - INCADD + N
              ELSE
              IADD = NINC + INCADD + N
              END IF
  550     WRITE (LUDATA,6120) LIMIT,287,IZERO,FZERO,IZERO,FZERO,
     X                        FZERO,IADD,307,IONE,PINC,PINC
          END IF
      REWIND LUDATA
      RETURN
C
C     ERROR EXIT
C
  600 STOP '***FATAL ERROR IN ARCH GEOMETRY - CHECK INPUT***'
C
C     FORMATS
C
 1000 FORMAT (A4,17A4,A3)
 1005 FORMAT (4I5,5F10.0)
 1010 FORMAT (7F10.0)
 1020 FORMAT (7F10.0,2I5)
 2000 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,74HTHE CONTROL WORD FOR 
     XMESH TYPE MUST BE "EMBA", "TREN", OR "HOMO", BUT IS ",A4,1H"/1H , 
     X17H(SEE USER MANUAL)/)
 2003 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,34HCONTROL WORD1 MUST BE
     X MOD OR BLANK/1H ,17H(SEE USER MANUAL)/)
 2005 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,72HNUMBER OF CONSTRUCTIO
     XN INCREMENTS "NINC" MUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE US
     XER MANUAL)/)
 2010 FORMAT (1H0,34HSPECIFIED COVER HEIGHT TO SHALLOW./1H0,34HPROGRAM W
     XILL SET HTCOVR = 0.1*RISE/)
 2011 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,24HINVALID ENTRY FOR "RI
     XSE"/)
 2012 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,26HINVALID ENTRY FOR "HF
     XSPAN"/)
 2013 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,26HINVALID ENTRY FOR "SD
     XRISE"/)
 2014 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,31HINVALID ENTRY FOR RAD
     XIUS "R(I)"/)
 2015 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,52HTRENCH WIDTH "TRNWID"
     XMUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE USER MANUAL)/)
 2020 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,52HTRENCH DEPTH "TRNDEP"
     XMUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE USER MANUAL)/)
 2025 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,52HTRENCH SLOPE "TRNSLP"
     XMUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE USER MANUAL)/)
 2030 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,62HOUTSIDE FOOTING WIDTH
     X "FTWIDO" MUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE USER MANUAL)
     X/)
 2035 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,61HINSIDE FOOTING WIDTH 
     X"FTWIDI" MUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE USER MANUAL)/
     X)
 2040 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,53HFOOTING DEPTH "FTDEP"
     X MUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE USER MANUAL)/)
 2042 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,55HSPACING FACTOR "SPCFA
     XC" MUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE USER MANUAL)/)
 2045 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,72HNUMBER OF TOP SEGMENT
     X NODAL POINTS "NTN" MUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEE US
     XER MANUAL)/)
 2050 FORMAT (1H0,22H***FATAL DATA ERROR***/1H ,75HNUMBER OF CORNER SEGM
```

```
     XENT NODAL POINTS "NCN" MUST BE WITHIN ALLOWABLE LIMITS/1H ,17H(SEECANA0951
     X USER MANUAL)/)                                                  CANA0952
 4000 FORMAT (1H1,48H* * * BEGIN GENERATION OF CANNED ARCH MESH * * *///CANA0953
     X1X,23H*** MESH PARAMETERS ***/                                   CANA0954
     X6X,50HTYPE OF MESH -------------------------------------,1X,A4)  CANA0955
 4010 FORMAT (                                                         CANA0956
     X6X,50HDEPTH OF TRENCH ABOVE FOOTING LEVEL (FT) ---------,F8.2/   CANA0957
     X6X,50HWIDTH OF TRENCH AT FOOTING LEVEL (FT) ------------,F8.2/   CANA0958
     X6X,50HSLOPE OF TRENCH WALL (H/V) -----------------------,F8.2)   CANA0959
 4020 FORMAT (                                                         CANA0960
     X6X,50HMESH SPACING PARAMETER ---------------------------,F8.2/   CANA0961
     X6X,50HTOTAL SOIL HEIGHT ABOVE CROWN (FT) ---------------,F8.2/   CANA0962
     X6X,50HMESH HEIGHT ABOVE CROWN (FT) ---------------------,F8.2/   CANA0963
     X6X,50HSOIL DENSITY ABOVE MESH (PCF) --------------------,F8.2/   CANA0964
     X6X,50HNUMBER OF ELEMENT LAYERS ABOVE CROWN -------------,I5//)   CANA0965
 4100 FORMAT (1X,23H*** ARCH PARAMETERS ***/                           CANA0966
     X6X,50HCENTERLINE RISE OF ARCH (IN) ---------------------,F8.2/   CANA0967
     X6X,50HBOTTOM HALF SPAN OF ARCH (IN) --------------------,F8.2/   CANA0968
     X6X,50HSIDE RISE OF ARCH ABOVE FOOTING (IN) -------------,F8.2/   CANA0969
     X6X,50HFOOTING DEPTH (IN) -------------------------------,F8.2/   CANA0970
     X6X,50HINSIDE FOOTING WIDTH (IN) ------------------------,F8.2/   CANA0971
     X6X,50HOUTSIDE FOOTING WIDTH (IN) -----------------------,F8.2)   CANA0972
 4110 FORMAT (                                                         CANA0973
     X6X,50HNUMBER OF ARCH SEGMENTS DEFINED ------------------,5H   3) CANA0974
 4120 FORMAT (                                                         CANA0975
     X6X,50HNUMBER OF ARCH SEGMENTS DEFINED ------------------,5H   2) CANA0976
 4200 FORMAT (/1X,26H*** CURVED TOP SEGMENT ***)                       CANA0977
 4210 FORMAT (/1X,29H*** CURVED CORNER SEGMENT ***)                    CANA0978
 4220 FORMAT (/1X,27H*** CURVED SIDE SEGMENT ***)                      CANA0979
 4230 FORMAT (/1X,28H*** STRAIGHT TOP SEGMENT ***)                     CANA0980
 4240 FORMAT (/1X,31H*** STRAIGHT CORNER SEGMENT ***)                  CANA0981
 4250 FORMAT (/1X,29H*** STRAIGHT SIDE SEGMENT ***)                    CANA0982
 4300 FORMAT (                                                         CANA0983
     X6X,50HRADIUS (IN) --------------------------------------,F8.2/   CANA0984
     X6X,50HCENTRAL ANGLE (DEG) ------------------------------,F8.2/   CANA0985
     X6X,50HSEGMENT ARC LENGTH (IN) --------------------------,F8.2/   CANA0986
     X6X,50HNUMBER OF BEAM ELEMENTS IN SEGMENT ---------------,I5)     CANA0987
 4310 FORMAT (                                                         CANA0988
     X6X,50HBASE ANGLE (DEG) ---------------------------------,F8.2)   CANA0989
 4320 FORMAT (                                                         CANA0990
     X6X,50HDEFLECTION ANGLE (DEG) ---------------------------,F8.2/   CANA0991
     X6X,50HSEGMENT LENGTH (IN) ------------------------------,F8.2/   CANA0992
     X6X,50HNUMBER OF BEAM ELEMENTS IN SEGMENT ---------------,I5)     CANA0993
 6105 FORMAT (A4,1X,17A4,3X,A4/8I5)                                    CANA0994
 6110 FORMAT (A1, I4, I5, 2F10.3, I5, 5X,2F10.3)                       CANA0995
 6115 FORMAT (A1,I4,10I5)                                              CANA0996
 6120 FORMAT ( A1, I4, I5, F10.2, I5, 2F10.2, 3I5, 2F10.2)             CANA0997
      END                                                              CANA0998
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *CANB0001
*     CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM               CANB0002
*                VERSION 1.0, MARCH 1990                               CANB0003
*           PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY                      CANB0004
*             THE FEDERAL HIGHWAY ADMINISTRATION                       CANB0005
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *CANB0006
      SUBROUTINE CANBOX (LUDATA,KPUTCK,DENSTY)                         CANB0007
C                                                                      CANB0008
C     SUBROUTINE FOR R/C BOX MESH .... REQUIRES FDIA.LT.0.0            CANB0009
C                                                                      CANB0010
*     Uncomment this statement in all program modules for double       CANB0011
*         precision version. SCM 3-10-90                               CANB0012
*                                                                      CANB0013
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)                              CANB0014
*                                                                      CANB0015
      DIMENSION X(20),Y(20),TPMESH(6),LNOD1(5,4),LNOD2(7,6),LNOD3(7,2), CANB0016
     1LNOD4(7,6),NODBC(38),MATTYP(6),TITLE(17)                         CANB0017
      DATA IZERO,IONE,FZERO/0,1,0.0/                                   CANB0018
      DATA L1H,IBLANK,PREP/1HL,1H ,4HPREP/                             CANB0019
      DATA TPMESH/4HEMBA,4HTREN,4HNKME,4HCH  ,4HNT  ,4H    /           CANB0020
      DATA LNOD1/15,1,2,13,12, 55,0,0,0,0,  60,0,0,0,0, 64,54,55,66,65/ CANB0021
      DATA LNOD2/65,60,61,68,67,3,2,     71,0,0,0,0,3,2,               CANB0022
     1           77,0,0,0,0,3,3,         83,0,0,0,0,3,3,               CANB0023
     2           89,0,0,0,0,3,4,         94,93,94,101,100,1,1/         CANB0024
      DATA LNOD3/95,95,96,107,106,3,4,  100,100,101,112,111,1,1/       CANB0025
      DATA MATTYP/4HINSI,4HTU  ,4HBEDD,4HING ,4HFILL,4H    /           CANB0026
      DATA LNOD4/101,102,103,114,113,3,5,    111,0,0,0,0,3,6,          CANB0027
     1           121,0,0,0,0,3,7,     131,0,0,0,0,3,8,                 CANB0028
     2           141,0,0,0,0,3,9,     150,155,156,167,166,1,1/         CANB0029
      DATA NODBC/1,2,3,4,5,           6,7,8,9,10,     11,12,22,23,33,  CANB0030
     1           34,44,45,55,56,     66,73,80,87,94,101,102,112,113,123,CANB0031
     2           124,134,135,145,146,156,157,167/                      CANB0032
C                                                                      CANB0033
C     READ INPUT DATA TO CONTROL BOX MESH                              CANB0034
C                                                                      CANB0035
      READ(5,1000)WORD,TITLE,WORD2                                     CANB0036
      READ(5,1010)IPLOT,IWRT,MGENPR,NINC,R1,R2,HTCOVR,DENSTY,TRNWID,   CANB0037
     1 BDEPTH                                                          CANB0038
C                                                                      CANB0039
C     NX1=NODES IN X-DIRECTION ALONG BOX                               CANB0040
C     NX2=NODES IN Y-DIRECTION ALONG BOX                               CANB0041
C                                                                      CANB0042
      NX1=5                                                            CANB0043
      NY1=7                                                            CANB0044
      DO 10 MESHTP=1,2                                                 CANB0045
      IF(WORD.EQ.TPMESH(MESHTP)) GO TO 20                              CANB0046
   10 CONTINUE                                                         CANB0047
      WRITE(6,2000)WORD                                                CANB0048
      STOP                                                             CANB0049
   20 R1T=2.0*R1                                                       CANB0050
      R2T=2.0*R2                                                       CANB0051
      WRITE(6,2010)TITLE,TPMESH(MESHTP),TPMESH(MESHTP+2),TPMESH(MESHTP+4CANB0052
     1),IPLOT,IWRT,MGENPR,NINC,R1T,R2T                                 CANB0053
      HTOPB=3.0*R2/12.                                                 CANB0054
      IF(HTCOVR.LT.HTOPB) HTOPB=HTCOVR                                 CANB0055
      IF(WORD.EQ.TPMESH(2)) WRITE(6,2120)TRNWID                        CANB0056
      WRITE(6,2100) HTCOVR,HTOPB,DENSTY                                CANB0057
C                                                                      CANB0058
C     TRANSFORM UNITS TO POUNDS AND INCHES                             CANB0059
C                                                                      CANB0060
      HTCOVR=HTCOVR*12.                                                CANB0061
      DENSIT=DENSTY/12.**3                                             CANB0062
      TRNWID=TRNWID*12.0                                               CANB0063
C                                                                      CANB0064
C     SET MASTER X-COORDINATES                                         CANB0065
C                                                                      CANB0066
      NX2=11                                                           CANB0067
      X(1)=0.0                                                         CANB0068
      DX=R1/FLOAT(NX1-1)                                               CANB0069
      DO 30 I=2,NX1                                                    CANB0070
   30 X(I)=X(I-1)+DX                                                   CANB0071
      IF(WORD.EQ.TPMESH(2)) GO TO 70                                   CANB0072
      XTR=5.0*R1                                                       CANB0073
      GO TO 80                                                         CANB0074
   70 XTR=R1+TRNWID                                                    CANB0075
   80 XTRMIN=1.10*R1                                                   CANB0076
      IF(XTR.LE.XTRMIN) XTR=1.101*R1                                   CANB0077
      NTR=NX1                                                          CANB0078
```

```
      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     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     NY1 ARE SPACED EVENLY ALONG VERTICAL WALL OF BOX
C
      Y(6)=-R2
      DY=R2/3.0
      DO 60 I=7,12
60    Y(I)=Y(I-1)+DY
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
90    Y(I)=Y(I-1)+DY
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     REMAINING SOIL WITH EQUIVALENT PRESSURE LOAD IN BOUNDARY CONDITION
C
      NHC=12
      IF(HTCOVR.LT.0.0) GO TO 95
      GO TO 96
95    WRITE(6,2070)HTCOVR
      STOP
C
C     YCOVR=MAXIMUM Y COORDINATE
C     NHC=MAXIMUM NUMBER OF VERTICAL NODES IF HTCOVR.GE.4R2
C     IUP=STARTING VERTICAL NUMBER OF ROWS ON PIPE TOP,ASCENDING
C     IDN=STARTING VERTICAL NUMBER OF ROWS ON PIPE BOTTOM,DESCENDING
C
96    YCOVR=HTCOVR+R2
      NHC=17
      IUP=12
      IDN=6
      DO 100 I=1,5
      Y(IUP+I)=-Y(IDN-I)
      IF(Y(IUP+I).GT.YCOVR) NHC=IUP+I
      IF(Y(IUP+I).GT.YCOVR) GO TO 110
100   CONTINUE
      GO TO 120
C
C     RESET Y-COORDINATE TO MATCH  " YCOVR "
C
110   DYTOP=(Y(NHC)-YCOVR)*0.5
      DYBOT=YCOVR-Y(NHC-1)
      IF(NHC.LE.14) GO TO 115
      IF(DYTOP.GT.DYBOT) NHC=NHC-1
      Y(NHC)=YCOVR
      GO TO 120
115   NHC=14
      Y(NHC)=YCOVR
      Y(NHC-1)=(Y(NHC)+Y(NHC-2))*0.5
120   CONTINUE
C
C     ADJUST BOTTOM Y-COORDINATE Y(5) TO BE LOCATED AT BOTTOM OF BEDDING
C
      IF(BDEPTH.LE.0.0) GO TO 140
C
C     LIMITS FOR MAXIMUM AND MINIMUM BEDDING THICKNESS
C
      BDEMIN=0.1*R2
      BDEMAX=0.667*R2
      IF(BDEPTH.LT.BDEMIN) BDEPTH=BDEMIN
      IF(BDEPTH.GT.BDEMAX) BDEPTH=BDEMAX
      Y(5)=Y(6)-BDEPTH
      GO TO 130
140   WRITE(6,2020)
130   CONTINUE
C
C     COMPUTE THE NUMBER OF NODES,NUMBER OF ELEMENTS AND ESTIMATE
C     MAXIMUM NUMBER OF BOUNDARY CONDITIONS(FOR STORAGE SPACE ONLY)
C
      NUMNP=11*NHC-5*4
      NUMEL=10*(NHC-1)-6*4+14
      NBPTC=200
      NPUTCK=0
      NPMAT=14
C
C     WRITE MASTER CONTROL DATA TO UNIT LUDATA TO SIMULATE LEVEL 3 INPUT
C
      WRITE(LUDATA,2030)PREP,TITLE,WORD2,NINC,MGENPR,NPUTCK,IPLOT,IWRT,
     1 NUMNP,NUMEL,NBPTC
C
C     WRITE NODAL POINT DATA TO UNIT LUDATA FOR LEVEL 3 INPUT
C
      LIMIT=IBLANK
      NODE=0
C
C     VERTICAL NODE COUNTER
C
      DO 150 J=1,NHC
      ISTART=1
      IF(J.GT.6 .AND. J.LT.12) ISTART=5
      DO 150 I=ISTART,11
      NODE=NODE+1
      IF(NODE.EQ.NUMNP) LIMIT=L1H
150   WRITE(LUDATA,2040)LIMIT,NODE,IZERO,X(I),Y(J),IZERO,FZERO,FZERO
C
C     GENERATE ELEMENT DATA
C
C     GET PIPE ELEMENT CONNECTIVITY
C
      LIMIT=IBLANK
```

```
      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(6,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
C     GENERATE SOIL BLOCK # 3 (SIDE)
C
      DO 250 I=1,2
      INTRAL=0
      NUMLAY=0
      INTERL=0
      MAT=LNOD3(6,I)
      INC=LNOD3(7,I)
      IF(NINC.LT.0) INC=1
      IF(I.EQ.1) GO TO 240
      INTRAL=1
      NUMLAY=1
      IF(NMOVE.EQ.11) INC=4
      IF(NINC.LT.0) INC=1
      IF(NMOVE.EQ.11) MAT=3
      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=I
      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=I
      INC=1
      INTRAL=1
```

```
      NUMLAY=NHC-12                                                     CANB0331
      INTERL=11                                                         CANB0332
      IF(NMOVE.EQ.11) INC=NHC-8                                         CANB0333
      IF(NINC.LT.0) INC=1                                               CANB0334
      IF(NMOVE.EQ.11) MAT=3                                             CANB0335
      WRITE(LUDATA,2050)LIMIT,LASTEL,NODEI,NODEJ,NODEK,NODEL,MAT,INC,   CANB0336
     1 IZERO,INTRAL,NUMLAY,INTERL                                       CANB0337
      GO TO 400                                                         CANB0338
  310 IF(I.EQ.6) GO TO 260                                              CANB0339
      IF(NMOVE.EQ.11) GO TO 260                                         CANB0340
      NELI=LNOD4(1,I)+NMOVE-1                                           CANB0341
      MAT=1                                                             CANB0342
      INC=1                                                             CANB0343
      IF(NELI.EQ.LASTEL) GO TO 260                                      CANB0344
      WRITE(LUDATA,2050)LIMIT,NELI,IZERO,IZERO,IZERO,IZERO,MAT,INC,     CANB0345
     1 IZERO,INTRAL,NUMLAY,INTERL                                       CANB0346
  260 CONTINUE                                                          CANB0347
C                                                                       CANB0348
C     ALL CONNECTIVITY IS DONE                                          CANB0349
C                                                                       CANB0350
C     BOUNDARY CONDITION GENERATION                                     CANB0351
C     OVERBURDEN PRESSURE INTO INCREMENTS                                CANB0352
C                                                                       CANB0353
  400 LIMIT=IBLANK                                                      CANB0354
      COVLIM=3.0*R2                                                     CANB0355
      IF(HTCOVR.LE.COVLIM) GO TO 440                                    CANB0356
      OVERP=(HTCOVR-3.0*R2)*DENSIT                                      CANB0357
      WRITE(6,2110) OVERP                                               CANB0358
      IF(NINC.LE.9) GO TO 420                                           CANB0359
      OVERP=OVERP/FLOAT(NINC-9)                                         CANB0360
      IDINC=NINC-9                                                      CANB0361
      DO 430 I=1,IDINC                                                  CANB0362
      NFIRST=157                                                        CANB0363
      NLAST=167                                                         CANB0364
      NEWINC=9+I                                                        CANB0365
      PLOAD=-OVERP                                                      CANB0366
      WRITE(LUDATA,2060)LIMIT,NFIRST,IZERO,FZERO,IZERO,FZERO,FZERO,     CANB0367
     1 NEWINC,NLAST,IONE,PLOAD,PLOAD                                    CANB0368
  430 CONTINUE                                                          CANB0369
      GO TO 440                                                         CANB0370
  420 IF(NINC.GT.0) GO TO 440                                           CANB0371
      NFIRST=157                                                        CANB0372
      NLAST=167                                                         CANB0373
      NEWINC=1                                                          CANB0374
      PLOAD=-OVERP                                                      CANB0375
      WRITE(LUDATA,2060)LIMIT,NFIRST,IZERO,FZERO,IZERO,FZERO,FZERO,     CANB0376
     1 NEWINC,NLAST,IONE,PLOAD,PLOAD                                    CANB0377
C                                                                       CANB0378
C     BOUNDARY CONDITIONS (DISPLACEMENTS)                               CANB0379
C                                                                       CANB0380
  440 LASTNO=(NHC-12)*11+112                                            CANB0381
      K=IONE                                                            CANB0382
      DO 450 I=1,38                                                     CANB0383
      IF(I.GT.11) K=IZERO                                               CANB0384
      IF(NODBC(I).EQ.LASTNO) LIMIT=LIH                                  CANB0385
      WRITE(LUDATA,2060)LIMIT,NODBC(I),IONE,FZERO,K,FZERO,FZERO,IONE,   CANB0386
     1 IZERO,IZERO,FZERO,FZERO                                          CANB0387
      IF(LIMIT.EQ.LIH) GO TO 460                                        CANB0388
  450 CONTINUE                                                          CANB0389
C                                                                       CANB0390
C     PRINT MATERIAL NUMBERS                                            CANB0391
C                                                                       CANB0392
  460 WRITE(6,2080)                                                     CANB0393
      DO 470 K=1,3                                                      CANB0394
      NUMMAT=K                                                          CANB0395
      K1=2*K-1                                                          CANB0396
      K2=K1+1                                                           CANB0397
  470 WRITE(6,2090)MATTYP(K1),MATTYP(K2),NUMMAT                         CANB0398
      REWIND LUDATA                                                     CANB0399
      RETURN                                                            CANB0400
C                                                                       CANB0401
C     FORMATS                                                           CANB0402
C                                                                       CANB0403
 1000 FORMAT(A4,17A4,A4)                                                CANB0404
 1010 FORMAT (4I5,6F10.0)                                               CANB0405
 2000 FORMAT(1H0,'  FATAL DATA ERROR. THE CONTROL WORD MUST BE * EMBA ORCANB0406
     1 TREN * ,BUT IS * ',A4,' * ')                                     CANB0407
 2010 FORMAT(1H1,'  * * BEGIN GENERATION OF CANNED MESH * *'///         CANB0408
     1          '  THE DATA TO BE RUN IS ENTITLED           '//1H0,17A4//CANB0409
     2          '  TYPE OF MESH ------------------------- ',3A4//       CANB0410
     3          '  PLOTTING DATA SAVED ------------------ ',I5//        CANB0411
     4          '  PRINT SOIL RESPONSES ----------------- ',I5//        CANB0412
     5          '  PRINT CONTROL FOR PREP OUTPUT -------- ',I5//        CANB0413
     6          '  NUMBER OF CONSTRUCTION INCREMENTS ---- ',I5//        CANB0414
     7          '  SPAN OF BOX -------------------------- ',F9.2//      CANB0415
     8          '  HEIGHT OF BOX ------------------------ ',F9.2/)      CANB0416
 2120 FORMAT(1H0,'  TRENCH WIDTH (FT) -------------------- ',F9.2/)     CANB0417
 2100 FORMAT(1H0,' SOIL ABOVE TOP OF BOX (FT) ----------- ',F9.2//      CANB0418
     1          '  MESH HEIGHT ABOVE TOP OF BOX (FT) ---- ',F9.2//      CANB0419
     2          '  SOIL DENSITY ABOVE MESH (PCF)--------- ',F9.2/)      CANB0420
 2110 FORMAT(1H0,' PRESSURE OF TRUNCATED SOIL (PSI)------ ',F9.2)       CANB0421
 2020 FORMAT(///,1H ,'WE ARE GOING TO ASSUME NO BEDDING MATERIAL'//)    CANB0422
 2030 FORMAT(A4,1X,17A4,3X,A4/8I5)                                      CANB0423
 2040 FORMAT(A1,I4,I5,2F10.3,I5,5X,2F10.3)                              CANB0424
 2050 FORMAT(A1,I4,10I5)                                                CANB0425
 2060 FORMAT(A1,I4,I5,F10.2,I5,2F10.2,3I5,2F10.2)                       CANB0426
 2070 FORMAT(1H0,'   HTCOVR=',F10.2//'  FATAL DATA ERROR FOR THE HEIGHT CANB0427
     1OF COVER OF SOIL'//'  HTCOVR MUST BE GREATER OR EQUAL THAN ZERO') CANB0428
 2080 FORMAT(1H0,//' IDENTIFICATION OF MATERIAL ZONE WITH MATERIAL NUMBECANB0429
     1R'///10X,'MATERIAL-ZONE',3X,'MATERIAL NO.'/)                      CANB0430
 2090 FORMAT(1H0,16X,2A4,3X,I2)                                         CANB0431
      END                                                               CANB0432
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * CONC0001
*     CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM                CONC0002
*                    VERSION 1.0, MARCH 1990                            CONC0003
*              PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY                    CONC0004
*                THE FEDERAL HIGHWAY ADMINISTRATION                     CONC0005
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * CONC0006
      SUBROUTINE CONCRE(IA,ICOME,IEXIT,LEVEL,NINC,NPMAT,NPPT,PDIA,      CONC0007
     1                  PIPMAT,RESULT,XMODE)                            CONC0008
C                                                                       CONC0009
C     SUBROUTINE FOR REINFORCED CONCRETE PIPE CULVERTS                  CONC0010
C                                                                       CONC0011
*     Uncomment this statement in all program modules for double        CONC0012
*          precision version. SCM 3-10-90                               CONC0013
*                                                                       CONC0014
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               CONC0015
*                                                                       CONC0016
*     Revise pipe dimension statements from 30 to 99.                   CONC0017
*          SCM  5-22-89                                                 CONC0018
*                                                                       CONC0019
      DIMENSION ASI(99),ASO(99),PFS(3),PIPMAT(5,99),PPFS(3),PSTRNI(99), CONC0020
     1          PSTRNO(99),RESULT(20,99),SAFE(4),STNMAT(5),TBI(99),     CONC0021
     2          TBO(99),WALLA(33),WALLB(33),WALLC(33),PTV(99)           CONC0022
*                                                                       CONC0023
C                                                                       CONC0024
```

```
C     TABLES BELOW GIVE STANDARD WALL THICKNESSES DERIVED FROM ASTM C-76
C     USERS MAY SUBSTITUTE THEIR OWN TABLES OF WALL THICKNESSES USING
C     THE SAME FORMAT AS BELOW.   PIPE SIZES FOR WHICH NO STANDARD WALL
C     IS AVAILABLE SHOULD HAVE  0.  IN THE MATRIX AT THOSE LOCATIONS
C
C     FORMAT FOR LOCATING PIPE SIZES IS AS FOLLOWS--
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
      COMMON /PEDRO/ STSMAX(30,11),STNMAX(30,11),SM1,SM2,SM3,PCE2,FTC,
     X FYC,FCC,PFSY,SM5,PTSN(30,2),PTSS(30,2)
      DATA WALLA/
     1      1.750,  1.875,  2.000,  2.250,  2.500,  2.375,  2.750,
     2      2.875,  3.000,  3.250,  3.500,  3.750,  4.000,  4.250,
     3      4.500,  4.750,  5.000,  5.250,  5.500,  5.750,  6.000,
     4      6.250,  6.500,  6.750,  7.000,  7.250,  7.500,  7.750,
     5      8.000,  8.250,  8.500,  8.750,  9.000/
      DATA WALLB/
     1      2.000,  2.250,  2.500,  2.750,  3.000,  3.250,  3.500,
     2      3.750,  4.000,  4.250,  4.500,  4.750,  5.000,  5.250,
     3      5.500,  5.750,  6.000,  6.250,  6.500,  6.750,  7.000,
     4      7.250,  7.500,  7.750,  8.000,  8.250,  8.500,  8.750,
     5      9.000,  9.250,  9.500,  9.750, 10.000/
      DATA WALLC/
     1      2.750,  3.000,  3.250,  3.500,  3.750,  4.000,  4.250,
     2      4.500,  4.750,  5.000,  5.250,  5.500,  5.750,  6.000,
     3      6.250,  6.500,  6.750,  7.000,  7.250,  7.500,  7.750,
     4      8.000,  8.250,  8.500,  8.750,  9.000,  9.250,  9.500,
     5      9.750, 10.000, 10.250, 10.500, 10.750/
C
      DATA NFS,PPFS/ 3,1.6,2.,2./
      DATA FPC,XNU,FSY,SE,SNU,SLS/4000.,.17,40000.,29000000.,.3,2.0/
      DATA ANALYS, ARBI  , CIRC  , DESIGN, ELLI  , FIX  , OPT  , STD
     1 / 4HANAL, 4HARBI, 4HCIRC, 4HDESI, 4HELLI, 3HFIX, 3HOPT, 3HSTD /
      DATA NTMAX,CW01/20,0.01/
C
      GO TO (100,200),ICOME
C
C     FIRST ENTRY ESTABLISHES PIPE PROPERTIES AND SAFETY FACTORS
C
100   READ(5,1000) PDIA,PT,RSHAPE,NONLIN,STNMAT(1),STNMAT(2),STNMAT(3),
     X PFPC,PCE,PNU,PDEN,PFSY,PSE,PSNU,SL
      ITER=0
      IF(PDIA.EQ.0.0) GO TO 950
      IF(PFPC.EQ.0.) PFPC=FPC
      PFPT = 5.0*SQRT(PFPC)
      PFVT = 2.0*SQRT(PFPC)
      PDENT = PDEN
      IF(PDENT.LE.0.) PDENT = 150.
      IF(PCE.EQ.0.) PCE = 33.*PDENT**1.5*SQRT(PFPC)
      IF(PNU.EQ.0.) PNU=XNU
      IF(PFSY.EQ.0.) PFSY=FSY
      IF(PSE.EQ.0.) PSE=SE
      IF(PSNU.EQ.0.) PSNU=SNU
      IF(SL.EQ.0.0) SL=SLS
      IF(PDIA.LT.0.0) WRITE(6,2001)
      WRITE(6,2000) PDIA,PFPC,PCE,PNU,PDEN,PFSY,PSE,PSNU
C
      PCENU = PCE/(1.-PNU**2)
      PSENU = PSE/(1.-PSNU**2)
      PN = PSENU/PCENU
      IF(NONLIN.LE.0) NONLIN = 1
      IF(NONLIN.GE.3) NONLIN = 3
      IF(STNMAT(1).LE.0.0) STNMAT(1) = 0.0
      IF(STNMAT(2).LE.0.0) STNMAT(2) = 0.5*PFPC/PCENU
      IF(STNMAT(3).LE.0.0) STNMAT(3) = 0.002
      IF(STNMAT(3).LE.STNMAT(2)) STNMAT(3) = 1.001*STNMAT(2)
      PCE2 = (PFPC-PCENU*STNMAT(2))/(STNMAT(3)-STNMAT(2))
      STNMAT(4) = PCE2/PCENU
      STNMAT(5) = PFSY/PSENU
      SM1=STNMAT(1)
      SM2=-STNMAT(2)
      SM3=-STNMAT(3)
      SM5=STNMAT(5)
      FTC=SM1*PCENU
      FYC=-STNMAT(2)*PCENU
      FCC=-PFPC
      WRITE(6,2002) NONLIN,STNMAT(1),STNMAT(2),STNMAT(3),STNMAT(5),SL
C
C     READ SECTION PROPERTIES FOR ANALYSIS
C
      IF(PDIA.LT.0.0) GO TO 111
      IF(RSHAPE.NE.ELLI .AND. RSHAPE.NE.ARBI) RSHAPE= CIRC
      IF(XMODE.EQ.DESIGN) GO TO 120
      NREAD =1
      IF(RSHAPE.EQ.ARBI) NREAD = NPPT
      DO 110 N=1,NREAD
      READ(5,1001) ASI(N),ASO(N),TBI(N),TBO(N)
      IF(TBI(N).LE.0.0) TBI(N) = 1.25
      IF(TBO(N).LE.0.0) TBO(N) = 1.25
110   CONTINUE
C
C     WRITE SECTION PROPERTIES TO BE ANALYSED.
C
      WRITE(6,2005) PT,RSHAPE
      IF(RSHAPE.EQ.CIRC) WRITE(6,2010) ASI(1),ASO(1),TBI(1),TBO(1)
      IF(RSHAPE.EQ.ELLI) WRITE(6,2011) ASI(1),TBI(1)
      IF(RSHAPE.EQ.ARBI) WRITE(6,2012) (ASI(N),ASO(N),TBI(N),TBO(N),
     X N=1,NPPT)
      GO TO 165
C
C     READ ARBITRARY CONCRETE SECTION PROPERTIES FOR NEW OPTION OF
C     VARIABLE THICKNESS (LEVEL 2-BOX OR 3)
C
111   IF(LEVEL.EQ.3) GO TO 112
      NDTOP=5
      NDSIDE=7
      NPPT=2*(NDTOP-1)+NDSIDE
      NPMAT=NPPT-1
112   IF(RSHAPE.NE.ARBI) GO TO 114
      DO 113 N=1,NPPT
      READ(5,1001) ASI(N),ASO(N),TBI(N),TBO(N),PTV(N)
      IF(TBI(N).LE.0.0) TBI(N)=1.25
      IF(TBO(N).LE.0.0) TBO(N)=1.25
      IF(PTV(N).LE.0.0) PTV(N)=PT
113   CONTINUE
      GO TO 118
C
C     READ BOX CULVERT SECTION PROPERTIES FOR LEVEL 2-BOX AND RSHAPE=STD
C
114   IF(LEVEL.NE.2) GO TO 960
      READ(5,1001) PTT,PTS,PTB,HH,HV
```

```
      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
116   PTV(NBOT)= PTB
C
C     LOOP THROUGH FOR SIDE WALL (NEGLECTING CORNER NODES)
C
      LIM2=NDSIDE+NDTOP-2
      NTOP1=NDTOP+1
      DO 117 N=NTOP1,LIM2
      ASO(N)=AS1
      ASI(N)=AS4
117   PTV(N)=PTS
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
      IF(TBI(1).LE.0.0) TBI(1) = 1.25
      IF(SRATIO.LE.0.0) SRATIO = 0.75
      DO 130 I=1,NFS
      IF(PFS(I).GT.0.0) GO TO 130
      PFS(I)=PPFS(I)
130   CONTINUE
C
C     NGOAL = 1, IMPLIES OPTIMUM WALL DESIGN
C     NGOAL = 2, IMPLIES STANDARD WALL DESIGN
C     NGOAL = 3, IMPLIES FIXED WALL THICKNESS
C
      NGOAL = 0
      IF(GOAL.EQ.OPT) NGOAL = 1
      IF(GOAL.EQ.FIX .AND. PT.GT.0.0) NGOAL = 3
      IF(NGOAL.EQ.0) GOAL = STD
      IF(NGOAL.EQ.0) NGOAL = 2
      WRITE(6,2100) (PFS(I),I=1,NFS)
      IF(ALCW.NE.0.) WRITE(6,2101) ALCW
      IF(ALCW.EQ.0.) ALCW = 1000000.
      WRITE(6,2102) RSHAPE,TBI(1),GOAL
      IF(NGOAL.EQ.3) WRITE(6,2103)  PT
      IF(RSHAPE.EQ.CIRC) WRITE(6,2104) SRATIO
      PFSC = CW01/ALCW
C
C     ASSIGN MINIMUM WALL THICKNESS FOR DESIGN
C     IF THICKNESS IS SPECIFIED IT WILL REMAIN CONSTANT
C     IF IT IS ADEQUATE
C
      IF(NGOAL.EQ.3) GO TO 160
141   GO TO (151,152),NGOAL
151   PT = .05*PDIA
      GO TO 160
152   CONTINUE
      ISIZE = PDIA/3 -3
      IF(ISIZE.GE.1.AND.ISIZE.LE.33) GO TO 155
153   NGOAL = 1
      WRITE(6,2018)
      GO TO 141
155   PT=WALLA(ISIZE)
      IF(PT.LE.0.) PT = WALLB(ISIZE)
      IF(PT.LE.0.) PT = WALLC(ISIZE)
      IF(PT.LE.0.) GO TO 153
160   PTMIN = PT
C
C     ASSIGN MINIMUM STEEL AREA BASED ON HANDLING DEFLECTION,
C     TEMPERATURE STEEL, OR MINIMUM PRACTICAL REINFORCEMENT
C
      AS = (.13*PDIA**3 + 130.*PDIA**2) / (.49*PSE*PT**2) - .0184*PT/PN
      IF(AS.LT..0014*PT)  AS = .0014*PT
      IF(AS.LT. .00583) AS = .00583
      ASMIN = AS
C
C     ESTABLISH RELATIVE DESIGN PROPORTIONS OF STEEL FOR CIRCULAR AND
C     ELLIPTICAL CAGES
C
      IF(RSHAPE.EQ.ELLI) GO TO 161
      ASI(1) = ASMIN
      ASO(1) = SRATIO*ASMIN
      TBO(1) = TBI(1)
      GO TO 165
161   ASI(1) = ASMIN
      ASO(1) = 0.0
      TBO(1) = 0.0
```

```
C
C     DESIGN AND ANALYSIS ... ASSIGN CIRCULAR OR ELLIPTICAL STEEL CAGES.
C     NOTE THIS SECTION IS RE-ENTERED FOR EACH TRIAL DESIGN.
C
165   IF(PDIA.LT.0.0) GO TO 181
      IF(RSHAPE.EQ.ARBI) GO TO 180
      DELTB = 0.0
      IF(RSHAPE.NE.ELLI) GO TO 168
      XPPT = NPPT
      XPMAT = NPMAT
      DELTB = (PT-2.*TBI(1))/(XPMAT/2.)
168   CONTINUE
      DO 170 N=2,NPPT
      ASI(N) = ASI(N-1)
      ASO(N) = ASO(N-1)
      SIGN = 1.0
      IF(RSHAPE.NE.ELLI) GO TO 169
      XN = N
      TERM = XPPT/2. - XN + 1.
      IF(TERM.EQ.0.0) SIGN = 0.0
      IF(TERM.LT.0.0) SIGN=-1.
169   TBI(N) = TBI(N-1) + DELTB*SIGN
      TBO(N) = TBO(N-1)
170   CONTINUE
C
C     CALCULATE INTIAL SECTION PROPERTIES BASED ON UNCRACKED SECTION.
C     (GENERALLY THIS WILL PRODUCE MAXIMUM MOMENT AND THUST)
C
180   IF(ICOME.EQ.1) PDIA = PDIA + PT
181   CONTINUE
      DO 190 N=1,NPPT
      IF(PDIA.LT.0.0) PT=PTV(N)
      APNI = ASI(N)*(PN-1.)
      APNO = ASO(N)*(PN-1.)
      YTI = TBI(N)
      YTO = PT - TBO(N)
      PA = PT + APNI + APNO
      YBAR = (0.5*PT*PT + APNI*YTI + APNO*YTO)/PA
      PI = 0.3333*PT**3 - PT*PT*YBAR + PT*YBAR*YBAR
      PI = PI +APNI*(YTI-YBAR)**2 + APNO*(YTO-YBAR)**2
      RESULT(14,N) = PA
      RESULT(15,N) = PI
      RESULT(16,N) = YBAR
      PSTRNI(N) = 0.0
      PSTRNO(N) = 0.0
C
C     INITIAL VALUES FOR MAX. STRAINS AND STRESSES
C
      DO 188 IL=1,2
      PTSN(N,IL)=0.0
188   PTSS(N,IL)=0.0
      DO 189 II=1,11
      STNMAX(N,II)=-0.000001/PCENU
189   STSMAX(N,II)=-0.000001
190   CONTINUE
C
C     AVERAGE AREA AND MOMENT OF INERTIA AT ELEMENT CENTERS FOR INPUT
C     INTO SOLUTION THEORIES.
C
      DO 195 N=1,NPMAT
      N1 = N+1
      IF(N1.GT.NPPT) N1=1
      PIPMAT(1,N) = PCE
      PIPMAT(2,N) = PNU
      PIPMAT(3,N) = 0.5*(RESULT(14,N)+RESULT(14,N1))
      PIPMAT(4,N) = 0.5*(RESULT(15,N)+RESULT(15,N1))
      PIPMAT(5,N) = PDEN*PT/1728.
195   CONTINUE
      NTIMES = 0
      RETURN
C
C
C     SECOND ENTRY ... (1) RE-EVALUATE SECTION PROPERTIES BASED ON
C     CURRENT VALUES OF THRUST AND MOMENT. (2) EVALUATE STRUCTURAL
C     RESPONSES IN TERMS OF DESIGN CRITERIA. (3) IN DESIGN MODE
C     DETERMINE NEW STEEL AREA AND/OR CONCRETE THICKNESS AS REQUIRED.
C
200   CALL CONMAT(NPMAT,NPPT,ICON,ASI,ASO,TBI,TBO,RESULT,PIPMAT,STNMAT,
     X PSTRNI,PSTRNO,PT,PN,PCENU,PSENU,NONLIN,PDIA,PTV,NTIMES,SL)
      IF(ICON.EQ.1) GO TO 500
      IF(NTIMES.GE.NTMAX) GO TO 500
      NTIMES = NTIMES + 1
      IEXIT = -1
      WRITE(6,*)NTIMES
      RETURN
C
C     CHECK IF NONLINEAR SOIL MODEL HAS CONVERGED
C
500   NTIMES = 0
      IF(NINC.GT.0) GO TO 505
      IEXIT = -1
      RETURN
C
C     SAVE INNER AND OUTER PIPE STRAINS OF CONVERGED SOLUTION
C
505   DO 507 N=1,NPPT
      PSTRNI(N) = RESULT(19,N)
507   PSTRNO(N) = RESULT(20,N)
C
C     EVALUATE STRUTURAL RESPONSES. (IN DESIGN MODE ONLY ON LAST STEP)
C
C        BSMAX = MAX TENSILE STRESS IN STEEL
C        BCMAX = MAX COMPRESSIVE STRESS IN CONCRETE
C        VMAX = MAX SHEAR STRESS ON GROSS SECTION
C        DISP = MAX RELATIVE DISPLACEMENT
C        CRWD = MAX CRACK WIDTH
C
      IF(XMODE.EQ.ANALYS) GO TO 510
      IF(IA.EQ.NINC) GO TO 510
      IEXIT = 0
      RETURN
C
510   BSMAX = 0.000001*PFSY
      BCMAX = -0.000001*PFPC
      VMAX = 0.0000001*PFPC
      UYMAX = RESULT(4,1)+0.0000001*PT
      UYMIN = RESULT(4,1)
      CRWD = 0.0
      DO 540 N=1,NPPT
      IF(ABS(RESULT(10,N)).GT.BSMAX) BSMAX = ABS(RESULT(10,N))
      IF(ABS(RESULT(11,N)).GT.BSMAX) BSMAX = ABS(RESULT(11,N))
      IF(RESULT(12,N).LT.BCMAX) BCMAX = RESULT(12,N)
      VN = ABS(RESULT(13,N))
      IF(VN.LT.VMAX) GO TO 520
      VMAX = VN
520   IF(RESULT(17,N).GT.CRWD) CRWD = RESULT(17,N)
```

```
      IF(RESULT(4,N).GT.UYMAX) UYMAX = RESULT(4,N)
      IF(RESULT(4,N).LT.UYMIN) UYMIN = RESULT(4,N)
540   CONTINUE
      DISP = UYMAX-UYMIN
      BCMAX = ABS(BCMAX)
      IF(CRWD.EQ.0.0) CRWD =0.000001
C
C     COMPUTE ACTUAL SAFETY AND PERFORMANCE FACTORS, THE CRITERIA IS ...
C
C        (1) STEEL YIELD STRESS (PFSY)
C        (2) CONCRETE COMPRESSIVE STRENGTH (PFPC)
C        (3) SHEAR STRENGTH=TENSILE STRENGTH (VU=PFPT)
C        (4) ALLOWABLE CRACK WIDTH, 0.01 INCHES OR USER DEFINED, (ALCW)
C        (5) ALLOWABLE DISPLACEMENT, FROM LUM,NCHRP-116,PAGE 21.(DISLUM)
C
      VU = PFVT
C
C     DETERMINE RADIAL STRESS FOR BOW STINGING
C
      BOWSTR = 10.*BSMAX*ASI(1)/(PDIA/2.0)
      IF(BOWSTR.LE.0.0) BOWSTR = 0.001
C
      DISLUM = (PDIA-PT)**2/(1200.*PT)
C
C     ACTUAL SAFETY FACTORS AND PERFORMANCE FACTORS
C
      FSST = PFSY/BSMAX
      FSCC = PFPC/BCMAX
      FSV = VU/VMAX
      FSCW = CW01/CRWD
      FSDIS = DISLUM/DISP
      FSBOW = PFPT/BOWSTR
C
C     DESIGN UPDATE
C
      IF(XMODE.EQ.ANALYS) GO TO 900
      ITER = ITER + 1
      IF(ITER.GT.9) WRITE(6,2200) ITER,PT,ASI(1),ASO(1)
      IF(ITER.GT.9) GO TO 900
C
C     COMPUTE RATIOS OF ACTUAL TO DESIRED SAFETY FACTORS.
C
      SAFE(1) = FSST/PFS(1)
      SAFE(2) = FSCC/PFS(2)
      SAFE(3) = FSV/PFS(3)
      SAFE(4) = FSCW/PFSC
C
C     REVISE STEEL AREA
C
*     Revise intrinsic funtion to generic form. SCM 3-10-90
*
      SMIN = MIN(SAFE(1),SAFE(4))
*
      ICHKS = 1
      IF(ABS(SMIN-1.0).LT.0.05) GO TO 555
      ASNEW = ASI(1)/SMIN
      IF(ASNEW.LT.ASMIN) ASNEW = ASMIN
      IF(ASI(1).EQ.ASNEW) GO TO 555
      ICHKS = 0
      RATIO = ASNEW/ASI(1)
      ASI(1) = RATIO*ASI(1)
      ASO(1) = RATIO*ASO(1)
C
```

```
C     REVISE CONCRETE WALL THICKNESS
C
555   ICHKC = 1
      IF(NGOAL.EQ.3) GO TO 570
*
*     Revise intrinsic function to generic form.  SCM  3-10-90
*
      SMIN = MIN(SAFE(2),SAFE(3))
*
      IF(ABS(SMIN-1.0).LT.0.05) GO TO 570
      PTNEW = PT/SMIN
      IF(PTNEW.LT.PTMIN) PTNEW = PTMIN
      IF(PT.EQ.PTNEW) GO TO 570
      GO TO (561,562),NGOAL
561   IF(ITER.LE.2) PT = PTNEW
      PT = 0.5*(PT+PTNEW)
      ICHKC = 0
      GO TO 570
C
562   IF(PTNEW.GT.WALLA(ISIZE)) GO TO 563
      PTNEW = WALLA(ISIZE)
      GO TO 568
563   IF(PTNEW.GT.WALLB(ISIZE)) GO TO 564
      PTNEW = WALLB(ISIZE)
      GO TO 568
564   IF(PTNEW.GT.WALLC(ISIZE) .AND. ITER .GT. 4) GO TO 565
      PTNEW = WALLC(ISIZE)
      GO TO 568
C
565   WRITE(6,2016)
      NGOAL = 1
      GO TO 561
C
568   IF(PT.EQ.PTNEW) GO TO 570
      ICHKC = 0
      PT = PTNEW
C
C     TEST FOR CONVERGENCE OF BOTH STEEL AND CONCRETE SECTIONS ...
C
570   IF(ICHKS*ICHKC.EQ.1) GO TO 800
      IEXIT = -2
      GO TO 165
C
C     PRINT DESIGN RESULTS
C
800   WRITE(6,2310) ITER,PT,ASI(1),ASO(1)
C
C     PRINT STRUCTURAL RESULTS AND EVALUATION
C
900   CONTINUE
      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,(J,RESULT(1,J),RESULT(3,J),RESULT(8,J),
     X  RESULT(5,J),RESULT(7,J),RESULT(2,J),RESULT(4,J),RESULT(9,J),
     X  RESULT(6,J),RESULT(18,J),J=1,NPPT)
      WRITE(6,2810) IA,(J,RESULT(10,J),RESULT(11,J),RESULT(12,J),
     X RESULT(13,J),J=1,NPPT)
      WRITE(6,2820) (J,PSTRNI(J),PSTRNO(J),J=1,NPPT)
      IF(PDIA.LT.0.0) GO TO 580
```

```
      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, 38H (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 OPTI
     XON FOR VARIABLE CONCRETE THICKNESS. ***  ,//10X,75H***OPTION IS RE
     XSTRICTED 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 //
     X 15X,36H CONCRETE SHEAR FAILURE ............ , F15.4 )
2101  FORMAT(/15X,36H ALLOWABLE CRACK WIDTH (IN) ........ , F15.4 )
2102  FORMAT(/15X,36H SHAPE OF REBAR CAGE(S) ............ ,11X,A4//
     X 15X,36H MIN. COVER TO CENTER OF REBAR ...... ,F15.4 //
     X 15X,36H DESIGN GOAL FOR WALL THICKNESS .... ,12X,A3)
2103  FORMAT(/15X, 36H FIXED WALL THICKNESS ............... ,F15.4)
2104  FORMAT(/15X,36H OUTER-TO-INNER STEEL RATIO ........., F15.4)
2200  FORMAT(1H1,/// 10X,50H DESIGN ITERATIVE PROCEDURE DID NOT CONVERGE
     X AFTER ,I3/ 10X,55H ITERATIONS. THE LAST SOLUTION IS PRINTED AND E
     XVALUATED, / 10X,37H WHERE THE SECTION PROPERTIES ARE ... //
     X 15X,36H WALL THICKNESS ..................... , E15.5 //
     X 15X,36H INNER CAGE STEEL AREA (OR ELLIPSE)  , E15.5 //
     X 15X,36H OUTER CAGE STEEL AREA.............. , E15.5 )
2310  FORMAT(1H1,// 10X,19H DESIGN RESULTS AT ,I2,11H ITERATIONS //
     X 15X,36H *** WALL THICKNESS ................ , F10.5 //
     X 15X,36H *** INNER CAGE STEEL AREA (OR ELLI) , F10.5 //
     X 15X,36H *** OUTER CAGE STEEL AREA ......... , F10.5 )
2800  FORMAT(1H1,//10X,32H STRUCTURAL RESPONSE OF CULVERT ,
     X 20H FOR LOAD INCREMENT ,I5///
     X 58H COORDINATES, DISPLACEMENTS AND CRACK DEPTHS ARE IN INCHES /
     X 26H PRESSURES ARE IN LB/IN**2/28H MOMENTS ARE IN IN.*LB/IN.  /
     X 31H THRUST AND SHEAR ARE IN LB/IN.  ///
     X 5H NPPT,6X,9H X-COORD.,7X,8H X-DISP.,7X,8H N-PRES.,8X,7H MOMENT,
     X 9X,6H SHEAR/11X,9H Y-COORD.,7X,8H Y-DISP.,7X,8H S-PRES.,8X,
     X 7H THRUST,3X,12H CRACK DEPTH,(//I5,F15.2,4E15.5/5X,F15.2,4E15.5))
2810  FORMAT(1H1,// 10X,33H STRESSES IN CULVERT WALL  (PSI) ,
     X 20H FOR LOAD INCREMENT ,I5///
     X 9X, 11H ELLIP. OR /
     X 5H NPPT,4X,11H INNER CAGE ,4X,11H OUTER CAGE ,6X,9H CONCRETE ,
     X 9X,6H SHEAR /14X,6H STEEL ,9X,6H STEEL ,3X,12H COMPRESSION ,
     X 8X,7H STRESS //
     X (I5,4E15.5))
2820  FORMAT(1H0,/////2X,'STRAINS  IN  THE  INNER  AND  OUTER  FIBER  OF
     X  THE  CULVERT WALL',//2X,'(ONLY STRAINS FOR COMPRESSION ZONES HAV
     XE PHYSICAL MEANING)',///,' NPPT    INNER STRAIN    OUTER STRAIN',/
     X/(I5,2E15.5))
2900  FORMAT(1H1,////  5X,26H CALCULATED SAFETY FACTORS ,
     X 20H FOR LOAD INCREMENT ,I5///
     X 10X,50H STEEL YIELD STRESS / MAX. STEEL STRESS .......... ,F10.3//
     X 10X,50H CONCRETE STRENGTH / MAX. COMPRESSIVE STRESS ..... ,F10.3//
     X 10X,50H WALL SHEAR CAPACITY / MAX. SHEAR ................ ,F10.3//
     X  5X,20H PERFORMANCE FACTORS //
     X 10X,50H 0.01 INCH  / MAX. CRACK WIDTH.................... ,F10.3//
     X 10X,50H ALLOWABLE DISPLACEMENT(LUM) / MAX. DISP. ........ ,F10.3//
     X 10X,50H TENSILE STRENGTH / STRESS FROM BOW STRING ...... ,F10.3 )
2901  FORMAT(1H1,////  5X,26H CALCULATED SAFETY FACTORS ,
     X 20H FOR LOAD INCREMENT ,I5///
     X 10X,50H STEEL YIELD STRESS / MAX. STEEL STRESS ......... ,F10.3//
     X 10X,50H CONCRETE STRENGTH / MAX. COMPRESSIVE STRESS .... ,F10.3//
     X 10X,50H WALL SHEAR CAPACITY / MAX. SHEAR ............... ,F10.3//
     X  5X,20H PERFORMANCE FACTORS //
     X 10X,50H 0.01 INCH. / MAX. CRACK WIDTH................... ,F10.3 )
      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 CONMAT(NPMAT,NPPT,ICON,ASI,ASO,TBI,TBO,RESULT,PIPMAT,
     X STNMAT,PSTRNI,PSTRNO,PT,PN,PCE,PSE,NONLIN,PDIA,PTV,NOLIT,SL)
C
C     THIS ROUTINE IS ENTERED WITH THE TOTAL MOMENT AND THRUST ON EACH
```

```
C     PIPE NODE. THE OBJECTIVE IS TO DETERMINE THE FOLLOWING RESPONSES
C     CONSISTANT WITH THIS LOADING ...
C
C     (1) INNER AND OUTER FIBER STAINS ASSUMING LINEAR DISTRIBUTION.
C     (2) DEPTH OF CRACKING, DISTANCE TO BENDING AXIS.
C     (3) EFFECTIVE THRUST AREA AND EFFECTIVE MOMENT OF INERTIA.
C         (THESE ARE AVERAGED INTO PIPMAT VECTOR FOR NEXT SOLUTION)
C     (4) STEEL STRESSES, CONCRETE COMPRESSIVE STRESS, AND SHEAR STRESS.
C     (5) CRACK WIDTH ESTIMATE.
C
C     THE ABOVE RESPONSES ARE STORED IN THE VECTOR RESULT FOR EACH NODE
C     POINT AS DEFINED BELOW...
C
C        RESULT(10,N) = INNER CAGE STEEL STRESS
C        RESULT(11,N) = OUTER CAGE STEEL STRESS.
C        RESULT(12,N) = MAX CONCRETE COMPRESSIVE STRESS.
C        RESULT(13,N) = MAX SHEAR STRESS IN CONCRETE.
C        RESULT(14,N) = EFFECTIVE AREA OF SECTION
C        RESULT(15,N) = MOMENT OF INERTIA ABOUT YBAR.
C        RESULT(16,N) = DISTANCE TO BENDING AXIS (YBAR).
C        RESULT(17,N) = WIDTH OF CRACK.
C        RESULT(18,N) = LENGTH OF CRACK.
C        RESULT(19,N) = MOMENT INCREMENT, OR CURRENT INNER STRAIN
C        RESULT(20,N) = THRUST INCREMENT, OR CURRENT OUTER STRAIN
C        PSTRNI(N) = INNER WALL STRAIN OF PREVIOUS CONVERGED STEP.
C        PSTRNO(N) = OUTER WALL STRAIN OF PREVIOUS CONVERGED STEP.
C
*     Uncomment this statement in all program modules for double
*         precision version. SCM 3-10-90
*
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
*     Revise pipe dimension statements from 30 to 99.
*        SCM   5-22-89
*
      DIMENSION ASI(NPPT),ASO(NPPT),TBI(NPPT),TBO(NPPT),RESULT(20,NPPT),
     X PIPMAT(5,NPMAT),STNMAT(5),PSTRNI(99),PSTRNO(99),PTV(99),FE(11),
     X CMOM(99),CTHR(99)
*
      COMMON /PEDRO/ STSMAX(30,11),STNMAX(30,11),SM1,SM2,SM3,PCE2,FTC,
     X FYC,FCC,FSY,SM5,PTSN(30,2),PTSS(30,2)
      DATA LIMIT,TOLER1,TOLER2/3,0.01,0.03/
C
C     SET OUTER LOOP CONVERGENCE CODE, (FOR ITERATIONS WITHIN LOAD STEP)
C
      ICON = 1
C
C     BEGIN NODE LOOP TO CONSIDER EACH NODE INDIVIDUALLY
C
      DO 100 N=1,NPPT
      IF(NOLIT.GT.0) GO TO 81
      CDELM=0.0
      CDELN=0.0
      GO TO 82
81    CDELM=CMOM(N)
      CDELN=CTHR(N)
82    CONTINUE
      IF(PDIA.LT.0.0) PT=PTV(N)
      PA = RESULT(14,N)
      PI = RESULT(15,N)
      YBAR = RESULT(16,N)
      DELM = RESULT(19,N) + CDELM
      DELN = RESULT(20,N) + CDELN
      PDENOM = PSTRNO(N)-PSTRNI(N)
C
C     DETERMINE DISTANCE TO CENTER OF REBARS AND SET INNER LOOP COUNTER.
C
      YTI = TBI(N)
      YTO = PT-TBO(N)
      NTIMES = 0
C
C     BEGIN INNER LOOP ... DETERMINE INNER AND OUTER STRAINS OF PIPE.
C
1     THETA = DELM/(PI*PCE)
      STNORM = DELN/(PA*PCE)
      STRNI = STNORM + THETA*YBAR + PSTRNI(N)
      STRNO = STNORM + THETA*(YBAR-PT) + PSTRNO(N)
      IF (DELM .EQ. 0.0 .AND. DELN .EQ. 0.0) GO TO 99
      DENOM = STRNO - STRNI
      IF(DENOM.EQ.0.0) DENOM=0.00001*STNMAT(2)
      STRNEG=STRNI
      IF(STRNEG.GT.STRNO)STRNEG=STRNO
C
C     DETERMINE FE(I) AT INTEGRATION POINTS
C
      NINT=10
      NINTP=11
      DY=PT/FLOAT(NINT)
      YY=0.0
      DO 135 I=1,NINTP
      PSTAIN=PSTRNI(N)+PDENOM*YY/PT
C
C     SET MAX. VALUE OF STRESS AND STRAIN AT EACH POINT
C
      SSM=-ABS(STSMAX(N,I))
      SNM=STNMAX(N,I)
      IF(NOLIT.GT.0) GO TO 34
      IF(PSTAIN.GT.SNM) GO TO 27
      IF(PSTAIN.LT.SNM) STNMAX(N,I)=PSTAIN
      IF(SNM.LT.SM3)GO TO 33
      IF(SNM.LT.SM2)GO TO 32
      PSTESS=(PSTAIN-SNM)*PCE+SSM
      IF(PSTESS.GT.FYC) GO TO 31
      PSTESS=(PSTAIN-SM2)*PCE2+(SM2-SNM)*PCE+SSM
      IF(PSTESS.GT.FCC) GO TO 31
      GO TO 33
32    PSTESS=(PSTAIN-SNM)*PCE2+SSM
      IF(PSTESS.GT.FCC)GO TO 31
33    PSTESS=FCC+(PSTAIN-SM3)*0.00001
31    IF(PSTESS.LT.SSM .AND. STSMAX(N,I).LT.0.0) GO TO 29
      IF(PSTESS.LT.SSM) STSMAX(N,I)=-PSTESS
      GO TO 34
29    STSMAX(N,I)=PSTESS
      GO TO 34
27    PSTESS=(PSTAIN-SNM)*PCE + SSM
      IF(PSTESS.GT.FTC) STSMAX(N,I)=ABS(STSMAX(N,I))
      IF(PSTESS.GT.0.0 .AND. STSMAX(N,I).GT.0.0) STSMAX(N,I)=ABS(STSMAX(
     XN,I))
34    FE(I)=1.0
      ICK=1
      IF(STSMAX(N,I).GT.0.0) ICK=0
C
C     COMPUTE STRAINS OF THE POINT YY
C
      STAIN=STRNI+DENOM*YY/PT
      IF(NONLIN.LT.2) GO TO 43
```

```
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
C
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+FCC
      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
      IF(FE(I).EQ.0.0) STSMAX(N,I)=ABS(STSMAX(N,I))
35    CONTINUE
C
C     DETERMINE CRACK DEPTH USING STRESSES
C
      IF(I.GT.1) GO TO 12
C
C     * *  COMPUTE INITIAL VALUES OF STRESSES  **
C
      ICRK=0
      IF(FE(I).GT.0.0) ICRK=1
      KV=0
      IF(ICRK.EQ.1) GO TO 10
      HI=0.0
      STSHI=STESS
      GO TO 135
10    HF=0.0
      STSHF=STESS
      GO TO 135
12    IF(KV.EQ.1) GO TO 135
      IF(ICRK.GT.0) GO TO 13
      IF(FE(I).EQ.0.0) GO TO 14
      HF=(I-1)*DY
      STSHF=STESS
      SLIM=FTC
      IF(STSMAX(N,I).GT.0.0) SLIM=0.0
      HK=HI+((SLIM-STSHI)/(STSHF-STSHI))*DY
      XKD=HK
      Y1=HK
      Y2=PT
      KV=1
C
C     WE KNOW  XKD  AND  HK
C
      GO TO 135
14    HI=(I-1)*DY
      STSHI=STESS
      GO TO 135
13    IF(FE(I).GT.0.0) GO TO 16
      HI=(I-1)*DY
      STSHI=STESS
      SLIM=FTC
      IM1=I-1
      IF(STSMAX(N,IM1).GT.0.0) SLIM=0.0
      HK=HI-((SLIM-STSHI)/(STSHF-STSHI))*DY
      XKD=PT-HK
      Y1=0.0
      Y2=HK
      KV=1
C
C     WE KNOW  XKD  AND  HK
C
      GO TO 135
16    HF=(I-1)*DY
      STSHF=STESS
135   YY=YY+DY
      IF(KV.EQ.0 .AND. ICRK.EQ.1) GO TO 18
      IF(KV.EQ.0 .AND. ICRK.EQ.0) GO TO 19
      GO TO 20
19    XKD=PT
      HK=PT
      Y1=HK
      Y2=PT
```

```
      GO TO 20
18    XKD=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
C
      SUM1=0.0
      SUM2=0.0
      SUM3=0.0
      YY=0.0
      DO 55 I=1,NINT,2
      I2=I+1
      I3=I+2
      SUM1=FE(I)+4.0*FE(I2)+FE(I3)+SUM1
      YY1=YY
      YY2=YY+DY
      YY3=YY+2.0*DY
      SUM2=FE(I)*YY1+4.0*FE(I2)*YY2+FE(I3)*YY3+SUM2
      SUM3=FE(I)*YY1**2+4.0*FE(I2)*YY2**2+FE(I3)*YY3**2+SUM3
55    YY=YY3
C
C     COMPUTE PA, YBAR, PI
C
      PAP=PA
      PA=DY*SUM1/3.0+APNI+APNO
      YBAR=(DY*SUM2/3.0+APNI*YTI+APNO*YTO)/PA
      PI=(DY/3.0)*(SUM3-2.0*YBAR*SUM2+YBAR*YBAR*SUM1)
      PI=PI+APNI*(YTI-YBAR)**2+APNO*(YTO-YBAR)**2
      PAMIN=0.00001*PT
      PIMIN=0.00001*PT
      IF(PA.LT.PAMIN)PA=PAMIN
      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
C       COMPUTE CORRECTIONS OF MOMENTS AND THRUSTS FOR NEXT ITERATION
C
50    YY=0.0
      CORRM=0.0
      CORRN=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(IRAJ.EQ.0) GO TO 136
```

```
      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) = FCE*STRNEG
      IF(NONLIN.LT.2) GO TO 57
      RESULT(12,N)=STSIN
      IF(STSIN.GT.STSOU)RESULT(12,N)=STSOU
      IF(RESULT(12,N).GT.FTC) RESULT(12,N)=FTC
C
C     COMPUTE MAX SHEAR
C
57    SHAREA= PT-(PT-YTO+YTI)/2.0
      RESULT(13,N)=RESULT(7,N)/SHAREA
C
C     COMPUTE CRAC 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
```

```
      IF(CRWD.LT.0.0)  CRWD=0.0
      GO TO 74
71    CRWD=0.0
74    RESULT(17,N)=CRWD
C
C     SAVE INNER AND OUTER STRAINS AND CRACK LENGTH
C
      RESULT(18,N) = XKD
   99 CONTINUE
      RESULT(19,N) = STRNI
      RESULT(20,N) = STRNO
100   CONTINUE
C
C     AVERAGE PIPE PROPERTIES AT ELEMENT CENTERS FOR INPUT TO SOLUTIONS
C
      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))
150   PIPMAT(4,N) = 0.5*(RESULT(15,N)+RESULT(15,N1))
C
      RETURN
C
C**   FORMATS
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
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE CONVT(D1,D2)
C
C     THIS SUBROUTINE EXPRESSES ISOTROPIC PROPERTIES IN ANISOTROPIC FORM
C
*     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),NON(10)
      DO 10 I = 1, 3
      DO 10 K = 1, 3
   10 CP(I,K) = 0.0
      IF(D2 .EQ. 0.5)D2=.495
      CP(3,3)=D1*0.5/(1.0+D2)
      D1=D1/((1.0+D2)*(1.0-2.0*D2))
      CP(1,2)=D2*D1
      CP(2,1)=CP(1,2)
      D1=D1*(1.0-D2)
      CP(1,1)=D1
      CP(2,2)=D1
      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 DUNCAN(ST,STHARD,NEL,ICON,ITER,MN,NELEM,IA)
```

```
C
C     THIS SUBROUTINE GENERATES AN INCREMENTAL PLANE STRAIN
C     CONSTITUTIVE MATRIX BASED ON THE HYPERBOLIC SOIL MODEL
C     OF DUNCAN
C
*     SELIG HYPERBOLIC BULK MODULUS OPTION ADDED IN 1989.
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),
     $                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),
     $               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/
C
C     STHARD(1,NEL)=TANGENT YOUNG'S MODULUS OF PREVIOUS LOAD STEP
C
C
C     STHARD(2,NEL)=TANGENT BULK MODULUS OF PREVIOUS LOAD STEP
C
*     STHARD(4,NEL) = Convergence flag.     SCM  3-11-89
*                   = -1.0 (Not converged)
*                   =  1.0 (Converged)
C
C     STHARD(5,NEL)=TANGENT YOUNG'S MODULUS OF LAST ITERATION
C
C
C     STHARD(6,NEL)=TANGENT BULK MODULUS OF LAST ITERATION
C
C
C     RECOVER CURRENT STRESSES AND CONVERT TO PRINCIPAL
C
      SXX = -ST(4,NEL)
      SYY = -ST(5,NEL)
      SXY = ST(6,NEL)
C
      CENTR = (SXX + SYY)/2.0
      RADIUS = SQRT(SXY**2 + ((SXX - SYY)/2.0)**2)
      S1 = CENTR + RADIUS
      S3 = CENTR - RADIUS
C
C     ON FIRST ITERATION OF EACH STEP UPDATE PARAMETERS
C
      IF(ITER.NE.1) GO TO 5
      E2 = STHARD(5,NEL)
      E1 = E2
      STHARD(1,NEL) = E1
C
      B2=STHARD(6,NEL)
      B1=B2
      STHARD(2,NEL)=B1
C
C     FOR FIRST ITERATION OF A NEW ELEMENT ENTERING SYSTEM,
C     ASSIGN INITIAL GUESS-VALUES
C
      IF(E1.NE.0.0) GO TO 5
      POIS1 = VT(MN)
      IF(POIS1.EQ.0.0) POIS1 = 0.33
      S3 = 0.1*PATM
      S1 = S3*(1.0-POIS1)/POIS1
C
C     RECALL PREVIOSLY CONVERGED PARAMETERS ON SUBSEQUENT ITERATIONS
C
    5 IF(ITER.EQ.1) GO TO 10
      E1 = STHARD(1,NEL)
      B1 = STHARD(2,NEL)
C
C     SET WEIGHTING RATIOS FOR AVERAGING ENEW FROM E1 AND E2,
C     AND BNEW FROM B1 AND B2
C
   10 WTE = 0.5
      IF(E1.EQ.0.0) WTE = RATIO(MN)
C
      WTP = 0.5
      IF(E1.EQ.0.0) WTP = RATIO(MN)
C
C     SET LOWER LIMIT FOR S3PA
C
      S3PA=S3/PATM
      IF(S3PA.LT.0.1) S3PA=0.1
C
C     TENSION FAILURE CHECK
C
      IF(S1.GT.0.0) GO TO 20
      E2=0.0025*ZK(MN)*PATM*S3PA**ZN(MN)
      POIS=0.40
      IF(VT(MN).GT.0.0) POIS=VT(MN)
      B2=E2/(3.0-6.0*POIS)
      E2P=STHARD(5,NEL)
      ETCHEK=(E2-E2P)/E2
      STHARD(5,NEL)=E2
      STHARD(6,NEL)=B2
      GO TO 30
   20 CONTINUE
C
C     COMPUTE YOUNGS TANGENT MODULUS E2, THEN ETNEW
C
      XS3=PATM*S3PA
*
*     Revise intrinsic function to generic form.  SCM 3-10-90
*
      PHI = PHIO(MN) - DPHI(MN) * LOG10(S3PA)
*
      RFF = RF(MN) * (1.0 - SIN(PHI))/
     1      (2.0 * C(MN) * COS(PHI) + 2.0 * XS3 * SIN(PHI))
      DEV = RFF * (S1 - S3)
      IF (DEV.LT.0.0) DEV = 0.0
      IF(DEV.GT.0.95) DEV = 0.95
      EINIT = ZK(MN) * PATM * S3PA**ZN(MN)
      E2 = (1.0 - DEV)**2*EINIT
      E2A=E2/3.0
      E2B=E2*8.0
C
C     USE UNDER RELAXATION FOR E2 WHEN ITER .GT. 2
```

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

```
      CALL CONVT(CMOD,PNEW)
*
      MATNO = MN
*
C
C     PRINT-OUT RECORD OF NONLINEAR MODULI (NON(MN).LT.0)
C
*     Revise unit numbers, variables and formats for soil modulus
*          output. SCM 1-9-89
*
      IF(IWRT.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,ETCHEK,SLEV,
     X               ETNEW,BNEW,PNEW,CMOD
*
  100 CONTINUE
      RETURN
C
C     FORMATS
C
*
 6000 FORMAT(1H1,//,6X,23HCONSTRUCTION INCREMENT ,I2,5X,
     X       10HITERATION ,I2)
 6020 FORMAT(1H0,15X,47H* * * ELASTIC MODULI AND CONVERGENCE PARAMETERS,
     X28H FOR DUNCAN SUBROUTINE * * *,/
     X22X,69H(CONVERGENCE ERRORS (ERROR-EMOD) ARE RATIOS GIVEN BY, (NEW-
     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,6HK-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,PYIELD,RESULT,ICOME)
C
C     THIS ROUTINE DETERMINES NONLINEAR BENDING PROPERTIES FOR A
C     HOMOGENOUS MATERIAL WITH A BILINEAR STREES-STRAIN CURVE. THE
C     FOLLOWING RESULT MATRIX IS IDENTIFIED ...
C
C        RESULT(10,N) = BENDING STRESS IN OUTER FIBER
C        RESULT(11,N) = THRUST STRESS (AVERAGE OVER SECTION)
C        RESULT(12,N) = SHEAR STRESS (AVERAGE OVER SECTION)
C        RESULT(13,N) = FRACTION OF SECTION YIELDED
C        RESULT(14,N) = MODIFIED THRUST AREA (PASTAR)
C        RESULT(15,N) = MODIFIED MOMENT OF INERTIA (PISTAR)
C        RESULT(16,N) = DISTANCE TO BENDING AXIS (YBAR)
C        RESULT(17,N) = MAX BENDING STRESS ABOVE YIELD
C        RESULT(18,N) = MAXIMUM FIBER STRAIN/ YIELD STRAIN
C        RESULT(19,N) = MOMENT INCREMENT, OR CURRENT INNER STRAIN
C        RESULT(20,N) = THRUST INCREMENT, OR CURRENT OUTER STRAIN
C        PSTRNI(N) = INNER WALL STRAIN OF PREVIOUS CONVERGED STEP
C        PSTRNO(N) = OUTER WALL STRAIN OF PREVIOUS CONVERGED STEP
C        YEPR(1,N) = DISTANCE TO BOTTOM OF ELASTIC ZONE (PRE. CON. STEP)
C        YEPR(2,N) = DISTANCE TO TOP OF ELASTIC ZONE (PRE. CON. STEP)
C
*     Uncomment this statement in all program modules for double
*          precision version. SCM 3-10-90
*
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
*     Add XMOMPR and DELNM arrays to dimension statement.
*          MGK 2-19-89
*
*     Revise dimensions of PSTRNI, PSTRNO, YEPR, YMOMPR and DELNM
*          arrays from 30 to 99.   SCM  4-30-89
*
      DIMENSION PIPMAT(5,NPMAT),RESULT(20,NPPT),PSTRNI(99),PSTRNO(99),
     X          YEPR(4,99),XMOMPR(99),DELNM(99)
*
      DATA LIMIT,TOLER/10,0.05/
      GO TO (1,1,200),ICOME
C
C     ON FIRST ITERATION OF FIRST LOAD STEP, COMPUTE CONSTANTS
C
    1 IF(IA.GT.1) GO TO 10
      IF(ICON.GT.1) GO TO 10
      POIS = 1.-PNU*PNU
      PSE1 = PE1/POIS
      PSE2 = PE2/POIS
      IF(PSE2.EQ.0.0) PSE2 = PSE1/(1.E+10)
      RATIO = PSE2/PSE1
      EPSY = PYIELD/PSE1
      YTOP = 2.0*PI/PS
C
C     INITIALIZE PREVIOUS CONVERGED STEP RESPONSES
C
      DO 5 N=1,NPPT
*
*     Initialize new array XMOMPR.  MGK 2-19-89
*
      XMOMPR(N) = 0.0
*
      YEPR(1,N) = 0.0
      YEPR(2,N) = YTOP
      PSTRNI(N) = 0.0
    5 PSTRNO(N) = 0.0
C
C     BEGIN OUTER LOOP ITERATION FOR MOMENT AND THRUST CONVERGENCE
C
   10 ERRSUM = 0.0
      DO 100 N=1,NPPT
      PASTAR = RESULT(14,N)
      PISTAR = RESULT(15,N)
      YBAR = RESULT(16,N)
      DELM = RESULT(19,N)
      DELN = RESULT(20,N)
      PDENOM = PSTRNO(N)-PSTRNI(N)
      YT1 = YEPR(1,N)
      YT2 = YEPR(2,N)
```

```
      YE1 = YT1
      YE2 = YT2
      NTIMES = 0
*
*     Revise moment summation logic.  MGK  2-19-89
*
      DELNM(N) = DELN*(YTOP/2.0 - YBAR)
      RESULT(5,N) = RESULT(5,N) + XMOMPR(N) + DELNM(N)
*
C
C     BEGIN INNER LOOP FOR DETERMINING PASTAR,PISTAR AND YBAR BASED ON
C     CURRENT ASSUMED VALUES OF MOMENT AND THRUST.
C     (FIRST DTERMINE TOTAL STRAIN DISTRIBUTION)
C
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     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     DETERMINE UPPER AND LOWER TRANSITION ZONE WEIGHTING FACTORS
C
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
      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
C
      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
```

```
C
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) = YE1
      YEPR(4,N) = YE2
C
C     COMPUTE NONLINEAR CHARACTERISTICS (PRINTED OUT ON CONVERGENCE)
C        (1) FRACTION OF SECTION YIELDED
C        (2) BENDING STRESS ABOVE YIELD STRESS (AVERAGE)
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*PE1/(XMAT*PI)
C
C     RETURN TO PIPE ROUTINE
C
      RETURN
```

```
C
C     SECOND ENTRY,  SAVE CONVERGED STRAINS AND ELASTIC ZONE MEASURES
C
 200  DO 175 N=1,NPPT
*
*     Save accumulated moment correction.  MGK   2-19-89
*
      XMOMPR(N) = XMOMPR(N) + DELNM(N)
*
      PSTRNI(N) = RESULT(19,N)
      PSTRNO(N) = RESULT(20,N)
      YEPR(1,N) = YEPR(3,N)
 175  YEPR(2,N) = YEPR(4,N)
C
C     RETURN TO PIPE SUBROUTINE
C
      RETURN
C
*
*     Revise wording of message to include "PIPE WALL" to avoid
*         confusion with other nonlinear models.  SCM 3-18-89
*
 6500 FORMAT(1H1,////58(2H *)///112H NONLINEAR PIPE WALL MODULUS DID NOT
     X CONVERGE, SOLUTION WILL CONTINUE WITH A RATIO OF ERROR IN THE MODE
     XULUS OF     ,E15.5///61H IF ERROR RATIO IS LESS THAN 0.1, SOLUTION
     X MAY BE ACCEPTABLE   ///58(2H *))
      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 ESTAB
C
C     TRANSFORMATION RELATIONSHIPS BETWEEN ELEMENT AND SUB-ELEMENT
C           UNKNOWNS ESTABLISHED
*
*     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)
      DO 55 I=1,2
      DO 50 J=1,28
      ITRW(J,I)=1
   50 T(J,I)=1.0
      DO 55 J=1,17
   55 NETCL(J,I)=1
      DO 57 I=1,3
      ITRW(I+ 8,1)=I+6
      ITRW(I+12,1)=I+9
      ITRW(I+18,1)=I+6
      ITRW(I+21,1)=I+9
      ITRW(I+24,1)=I+6
      ITRW(I+ 9,2)=I+6
      ITRW(I+13,2)=I+9
      ITRW(I+19,2)=I+9
      ITRW(I+22,2)=I+6
   57 ITRW(I+25,2)=I+9
```

```
      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, KPUTCK,
     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),ETYPE(5)
      DATA ETYPE/4HQUAD,4HTRI.,4HBEAM,4HFACE,4HBAD /
      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
      DO 20 NE = 1,MAXEL
   20 LGEN(1,NE) = 0
   40 IF (LIMIT .EQ. L1H) RETURN
      READ (LUPREP,5015) LIMIT, NE, (IX(K), K = 1,7), KINDLG
      IF (NE .LE. NUMEL .OR. NE .GT. MAXEL) GO TO 520
      IF (NUMEL .EQ. 0 .AND. (NE .NE. 1 .OR. IX(1) .EQ. 0)) GO TO 540
C
C**   STORE DATA RELATING TO MATERIAL OR INCREMENT CHANGES
C
      IF (IX(1) .NE. 0) GO TO 60
      LGEN(1,NE) = NE
      LGEN(2,NE) = IX(5)
      LGEN(3,NE) = IX(6)
      LGEN(4,NE) = IX(7)
      IF (LIMIT .EQ. L1H) GO TO 560
      GO TO 40
C
C**   SETUP PARAMETERS FOR ELEMENT GENERATION
C
   60 INTRAL = KINDLG(1)
      NUMLAY = KINDLG(2)
      INTERL = KINDLG(3)
      IF (INTRAL .EQ. 0) INTRAL = 1
      IF (NUMLAY .LE. 1) GO TO 80
      LINLAY = (NE-LPRIOR +NUMLAY-1)/ NUMLAY
C
C**   DETERMINE TYPE OF GENERATION TO ACCOMPLISH
C
   80 NUMEL = NUMEL +1
      IF (NE .GT. NUMEL) GO TO 140
C
C**   INITIALIZE PARAMETERS FOR LAST CARD READ IX(1) .NE. 0
C
      KGEN = I1H
      IF (IX(5) + IX(6)+IX(7) .EQ. 0) KGEN = N1H
      IF (IX(5) .EQ. 0) IX(5) = MPRIOR
      IF (IX(6) .EQ. 0) IX(6) = IPRIOR
      IF (IX(7) .EQ. 0) IX(7) = JPRIOR
      MPRIOR = IX(5)
      IPRIOR = IX(6)
      JPRIOR = IX(7)
      LPRIOR = NUMEL
C
C**   DETERMINE BAND WIDTH
C
      MB = 0
      DO 120 K = 1,4
      DO 120 L = K,4
      IF (IX(K) .EQ. 0 .OR. IX(L) .EQ. 0) GO TO 120
      M = IABS (IX(K) -IX(L))
      IF (M .GT. MB) MB = M
  120 CONTINUE
      NDF = 2
      IF(IX(3).EQ.0) NDF = 3
      MB = NDF*(MB+1)
      IF (MB .GT. MBAND) MBAND = MB
C
C**   PLACE CONNECTIVITY INPUT INTO NOD ARRAY
C
      DO 130 K = 1,LSTOP
  130 NOD(K,NE) = IX(K)
      GO TO 170
C
```

```
C**   GENERATE AN ELEMENT                                               GENL0093
C                                                                       GENL0094
  140 KGEN = IBLANK                                                     GENL0095
      DO 150 K = 1,4                                                    GENL0096
      NOD(K,NUMEL) = NOD(K,NUMEL-1) +INTRAL                             GENL0097
      IF (NUMLAY .LE. 1) GO TO 145                                      GENL0098
      IF (MOD(NUMEL-LPRIOR,LINLAY) .NE. 0) GO TO 145                    GENL0099
      NOD(K,NUMEL) = NOD(K,LPRIOR) + INTERL*(NUMEL-LPRIOR)/LINLAY       GENL0100
  145 CONTINUE                                                          GENL0101
      IF (NOD(K,NUMEL) .LT. 0 .OR. NOD(K,NUMEL) .GT. NUMNP) GO TO 580   GENL0102
  150 IF (NOD(K,NUMEL-1) .EQ. 0) NOD(K,NUMEL) = 0                       GENL0103
C                                                                       GENL0104
C**   MAKE MATERIAL CHANGE FOR GENERATED ELEMENTS WHERE SPECIFIED       GENL0105
C                                                                       GENL0106
      IF (LGEN(1,NUMEL) .NE. NUMEL) GO TO 160                           GENL0107
      IF (LGEN(2,NUMEL) .GT. 0) MPRIOR = LGEN(2,NUMEL)                  GENL0108
      IF (LGEN(3,NUMEL) .GT. 0) IPRIOR = LGEN(3,NUMEL)                  GENL0109
      IF (LGEN(4,NUMEL) .GT. 0) JPRIOR = LGEN(4,NUMEL)                  GENL0110
      KGEN = M1H                                                        GENL0111
  160 NOD(5,NUMEL) = MPRIOR                                             GENL0112
      NOD(6,NUMEL) = IPRIOR                                             GENL0113
      NOD(7,NUMEL) = JPRIOR                                             GENL0114
C                                                                       GENL0115
C**   PRINT ELEMENT DATA                                                GENL0116
C                                                                       GENL0117
  170 IF (MGENPR .LT. 2) GO TO 190                                      GENL0118
      ELM = ETYPE(5)                                                    GENL0119
      IF(NOD(4,NUMEL).NE.0) ELM = ETYPE(1)                              GENL0120
      IF(NOD(4,NUMEL).EQ.0) ELM = ETYPE(2)                              GENL0121
      IF(NOD(3,NUMEL).EQ.0) ELM = ETYPE(3)                              GENL0122
      IF(NOD(7,NUMEL).NE.0) ELM = ETYPE(4)                              GENL0123
      IF (KGEN.EQ.I1H.OR.KGEN.EQ.N1H)WRITE(6,6015) KGEN,NUMEL,          GENL0124
     1   (NOD(K,NUMEL),K=1,6),ELM,MB,(KINDLG(K),K=1,3)                  GENL0125
      IF (KGEN .EQ. IBLANK .OR. KGEN .EQ. M1H) WRITE(6,6015) KGEN,NUMEL,GENL0126
     1   (NOD(K,NUMEL),K=1,6),ELM                                       GENL0127
  190 IF (NE .EQ. NUMEL) GO TO 40                                       GENL0128
      GO TO 80                                                          GENL0129
C                                                                       GENL0130
C**   WRITE ERROR MESSAGES FOR ELEMENT INPUT                            GENL0131
C                                                                       GENL0132
  520 WRITE (6,6025)                                                    GENL0133
      GO TO 600                                                         GENL0134
  540 WRITE (6,6030)                                                    GENL0135
      GO TO 600                                                         GENL0136
  560 WRITE (6,6035)                                                    GENL0137
      GO TO 600                                                         GENL0138
  580 WRITE (6,6040)                                                    GENL0139
  600 MB = 0                                                            GENL0140
      ELM = ETYPE(5)                                                    GENL0141
      WRITE(6,6015) KGEN,NE,(IX(K),K=1,6),ELM,MB,(KINDLG(K),K=1,3)      GENL0142
      KPUTCK = KPUTCK +1                                                GENL0143
      GO TO 40                                                          GENL0144
 5015 FORMAT (A1, I4, 10I5)                                             GENL0145
 6005 FORMAT (122H1ELEMENT     NODAL  CONNECTIVITY   MATERIAL   CONSTR. GENL0146
     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(1X,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
 6030 FORMAT ( 103H0FATAL DATA ERROR ON FOLLOWING ELEMENT CARD.   THE FIGENL0156
     1RST ELEMENT CARD MUST HAVE NE = 1 AND IX(1) .NE. 0/)              GENL0157
 6035 FORMAT (  94H0FATAL DATA ERROR ON FOLLOWING ELEMENT CARD.   IT IS GENL0158
     1ILLEGAL TO SPECIFY LIMIT = L IF IX(1) = 0/)                       GENL0159
 6040 FORMAT ( 117H0FATAL DATA ERROR ON FOLLOWING ELEMENT CARD.   A NODEGENL0160
     1 OF A CONNECTIVITY BEING GENERATED AS SPECIFIED IS OUT OF BOUNDS/)GENL0161
      END                                                               GENL0162
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * GENN0001
*     CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM                GENN0002
*                   VERSION 1.0, MARCH 1990                             GENN0003
*             PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY                     GENN0004
*               THE FEDERAL HIGHWAY ADMINISTRATION                      GENN0005
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * GENN0006
      SUBROUTINE  GENEND  (X, Y, NOD, NODB, BIV, KODE, NODTYP, XYGEN,   GENN0007
     1       NUMNP, NUMEL, NBPTC, MAXBC, KPUTCK, MGENPR, NSMAT , LSTOP, GENN0008
     1       NUMLI,LUPREP,MAXMAT,NXMAT)                                 GENN0009
*                                                                       GENN0010
*     Uncomment this statement in all program modules for double        GENN0011
*          precision version. SCM 3-10-90                               GENN0012
*                                                                       GENN0013
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               GENN0014
*                                                                       GENN0015
      DIMENSION  X(1), Y(1), NOD(LSTOP,1), NODB(1), BIV(3,1), KODE(1),  GENN0016
     1       NODTYP(1), XYGEN(3,1)                                      GENN0017
*                                                                       GENN0018
*     Revise array dimension from AN(2) to AN(3) for extended boundary  GENN0019
*          condition option.  SCM 10-25-89                              GENN0020
*                                                                       GENN0021
      DIMENSION NAMEG(33),AREA(4),IX(4),ANI(2),AN(3),IIFLG(3),BIVD(3)   GENN0022
      DIMENSION RBC(2)                                                  GENN0023
*                                                                       GENN0024
      COMMON /AFCALC/ XX(4), YY(4)                                      GENN0025
      DATA IBLANK ,L1H /1H , 1HL/                                       GENN0026
      DATA NAMEG /4HNOT ,4*4HSTRA,4HLAPL,4HINPU,2*4HSTRA,2*4HCURV  ,    GENN0027
     1              4HUSED,4*4HIGHT,4HACE ,4HT   ,2*4HIGHT,4HE (A,4HE (E,GENN0028
     3              4H****,8*4H    ,4HRC) ,4HLIP) /                     GENN0029
      DATA ANI,SAFE /4HF = ,4HD = ,1.E-6/                               GENN0030
*                                                                       GENN0031
*     Add data statement for extended boundary condition option.        GENN0032
*          SCM 10-25-89                                                 GENN0033
*                                                                       GENN0034
      DATA RBC /4H F  ,4H D  /                                          GENN0035
*                                                                       GENN0036
C                                                                       GENN0037
C**   GENERATE COORDINATES FOR UNSPECIFIED NODES USING LAPLACE SCHEME   GENN0038
C                                                                       GENN0039
C**   INITIALIZE PARAMETERS                                             GENN0040
C                                                                       GENN0041
      NMISNG = 0                                                        GENN0042
      DO 20 NNP = 3,NUMNP                                               GENN0043
      IF (NODTYP(NNP) .GT. 6) GO TO 20                                  GENN0044
      IF (NODTYP(NNP)/2 .NE. 2) X(NNP) = (X(NNP-2) +X(NNP-1))/ 2.0      GENN0045
      IF (NODTYP(NNP)/2 .NE. 1) Y(NNP) = (Y(NNP-2) +Y(NNP-1))/ 2.0      GENN0046
      NMISNG = NMISNG +6                                                GENN0047
   20 CONTINUE                                                          GENN0048
      IF (NMISNG .EQ. 0) GO TO 220                                      GENN0049
      IF (NMISNG .GT. NUMLI) NMISNG = NUMLI                             GENN0050
      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 (NOD(4,NE) .EQ. 0) LIMIT = 3
      IF (NOD(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) +2.0
   40 CONTINUE
   50 CONTINUE
      DO 60 NNP = 1,NUMNP
      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, NUMNP
      K  =NODTYP(NNP)
  250 WRITE(6,6020)  NNP,X(NNP),Y(NNP),NAMEG(K),NAMEG(K+11),NAMEG(K+22)
  260 CONTINUE
C
C**   COMPUTE ELEMENT AREAS   IF TRIANGLE, CHECK AREA AND RESTART LOOP
C
      DO 380 NE = 1,NUMEL
      IF (NOD(3,NE) .EQ. 0) GO TO 380
      IF(NOD(7,NE).NE.0) GO TO 380
      LIMIT = 4
      IF (NOD(4,NE) .EQ. 0) LIMIT = 3
      DO 320 L = 1,LIMIT
      K = NOD(L,NE)
      IX(L) = K
      XX(L) = X(K)
  320 YY(L) = Y(K)
      IF (LIMIT .EQ. 4) GO TO 330
      AREA(1) =  AF(3,1) +AF(2,3) -AF(2,1)
      IF(AREA(1).GT.SAFE) GO TO 380
      WRITE(6,6025) NE,AREA(1)
      KPUTCK = KPUTCK + 1
      GO TO 380
C
  330 AREA(1) = -AF(1,4) -AF(2,1) +AF(2,4)
      AREA(2) = AF(3,4) +AF(2,3) -AF(2,4)
      AREA(3) =  AF(3,1) +AF(2,3) -AF(2,1)
      AREA(4) = -AF(1,4) -AF(3,1) +AF(3,4)
C
C**   CHECK FOR NEGATIVE AREAS OF QUAD SUB ELEMENTS
C
      SIGN = 1.0
      SIGNA = 1.0
      IF (AREA(1) .LT. SAFE .OR. AREA(2) .LT. SAFE) SIGN = -SIGN
      IF (AREA(3) .LT. SAFE .OR. AREA(4) .LT. SAFE) SIGNA = -SIGNA
      IF (SIGN .GT. 0.0 .OR. SIGNA .GT. 0.0) GO TO 340
      WRITE (6,6025) NE, AREA
      KPUTCK = KPUTCK +1
      GO TO 380
C
C**   SWITCH ORIENTATION OF DIAGONAL OF QUAD IF PRODUCES BETTER ELEMENT
C
  340 L = 0
      IF (SIGN * ABS(AREA(1) +AREA(2)) .GT.
     1       SIGNA * ABS(AREA(3) +AREA(4)) ) GO TO 360
      NOD(1,NE) = IX(4)
      DO 350 K = 2,4
  350 NOD(K,NE) = IX(K-1)
      L = 2
  360 IF (AREA(L+1) .GT. 10.*AREA(L+2) .OR.
     1       AREA(L+1) .LT. AREA(L+2)/10.) WRITE (6,6030) NE, AREA
  380 CONTINUE
C
C**   SET FLAGS FOR OTHER THAN QUAD ELEMENTS
C
      DO 400 NE = 1,NUMEL
      IF (NOD(3,NE) .EQ. 0) NOD(3,NE) = NOD(2,NE)
  400 IF (NOD(4,NE) .EQ. 0) NOD(4,NE) = NOD(3,NE)
C
C**   CHECK MATERIAL DEFINITIONS, AND DETERMINE NUMBER OF MATERIALS.
C
      DO 410 J=1,MAXMAT
  410 NODTYP(J) = -1
      NSMAT = 0
C
      DO 450 NE=1,NUMEL
      IF(NOD(2,NE).EQ.NOD(3,NE)) GO TO 450
      IF(NOD(7,NE).NE.0) GO TO 450
      MAT = NOD(5,NE)
C
      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(NOD(7,NE).EQ.0) GO TO 470
      NOD(7,NE) = 1
      MATX = NOD(5,NE)
      IF(MATX.GT.0 .AND. MATX.LE.MAXMX) GO TO 466
      WRITE(6,6050) NE
      KPUTCK = KPUTCK + 1
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. L1H) 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
      DXF=0.0
      DYF=0.0
      IF(L .EQ. NM) GO TO 186
      DXF=X(KK+INCR)-X(KK)
      DYF=Y(KK+INCR)-Y(KK)
      PK=PJ+DP*(SQRT(DXF*DXF+DYF*DYF))
  186 BIV(1,I)=BIVD(1)-(2.0*(DYF+DYB)*PJ+DYB*PI+DYF*PK)/6.0
      BIV(2,I)=BIVD(2)+(2.0*(DXF+DXB)*PJ+DXB*PI+DXF*PK)/6.0
      DXB=DXF
      DYB=DYF
      PI=PJ
      PJ=PK
      BIV(3,I)=TH
      NODB(I)=KK*10000+IIFLG(1)*10+IIFLG(2) +IA*100
      DO 187 J=1,2
      IF (IIFLG(J) .EQ. 0) AN(J) = ANI(1)
      IF (IIFLG(J) .EQ. 1) AN(J) = ANI(2)
*
*     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          40H0NUMBER OF ITERATIONS -----------------,I5/
     2          40H0MAXIMUM NUMBER OF ITERATIONS ALLOWED---,I5/
     3          40H0LARGEST LAPLACE ERROR FACTOR---------- ,F15.6)
 6010 FORMAT (61H1NODE              X-COORDINATE   Y-COORDINATE   GENE
     1RATION  /14H NUMBER        ,37X,4HTYPE)
 6020 FORMAT (1X,I4,8X,E20.4,E15.4,3X,3A4)
 6025 FORMAT ( 131H0FATAL DATA ERROR.   NEGATIVE AREA FOR ELEMENT NUMBER
     1     AREA OF TRIANGLES AS INPUT     AREA OF TRIANGLES USING ALTERN
     2ATE DIAGONAL/   I51,  8X,2E12.3, 13X,2E12.3)
 6030 FORMAT (  75H0WARNING ONLY.   THE AREA OF ONE SUB ELEMENT IS MUCH
```

```
     1SMALLER THAN THE OTHER/  97H FOR ELEMENT NUMBER      AREA OF TRIANG
     2LES AS INPUT      AREA OF TRIANGLES USING ALTERNATE DIAGONAL/
     3   I17, 8X,2E12.3, 13X,2E12.3/)
 6035 FORMAT(/30H FATAL DATA ERROR. THERE WERE ,I3,  14H DECLARED BUT,
     X ,I3, 12H IS MAXIMUN. )
 6045 FORMAT (  65H0FATAL DATA ERROR.   NUMBER OF BOUNDARY CARDS CANNOT
     1EXCEED MAXBC)
6050  FORMAT(// 62H FATAL DATA ERROR FOR MATERIAL NUMBER OF INTERFACE EL
     XEMENT NO. ,I5)
*
*     Revise formats for boundary condtion output.  SCM  10-25-89
*
 915  FORMAT(1H1,1X,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,I5,5X,I5,4X,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,
     1        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,NM)
      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,NM)
      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,NM)
      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 (  82H1NODE     GENERATION   X-COORDINATE    Y-COORDINATE
     1  NODE          SPACING       ARC/  16H NUMBER      MODE,39X ,30HINCREM
     2ENT   PARAMETER   RADIUS)
 6010 FORMAT (I7, I8,  E19.4, E15.4,            I9, F16.3,F11.2)
 6015 FORMAT (  74H0FATAL DATA ERROR ON ABOVE NODE CARD.   NODE POINT NU
     1MBER IS OUT OF BOUNDS /)
 6020 FORMAT ( 78H0FATAL DATA ERROR ON ABOVE NODE CARD.   IF MODEG .GE.
     110, THEN RADIUS MUST = 0)
 6025 FORMAT ( 125H0FATAL DATA ERROR ON ABOVE NODE CARD.   A NODE HAS BE
     1EN RECALLED (I.E., MODEG = 1 OR 3) WHICH HAS NOT PREVIOUSLY BEEN D
     2EFINED  /)
 6030 FORMAT ( 117H0FATAL DATA ERROR ON ABOVE NODE CARD.   A NODE HAS BE
     1EN PRESCRIBED (MODEG = 0 OR 2) WHICH HAS PREVIOUSLY BEEN DEFINED/)
 6135 FORMAT (  83H0FATAL DATA ERROR ON ABOVE NODAL CARD.   THE NODE REL
```

```
     LATIVELY ADDRESSED IS UNDEFINED)                                  GENO0233
 6040 FORMAT ( 109H0FATAL DATA ERROR ON ABOVE NODE CARD.   INCORRECT GENGENO0234
     1ERATION SPECIFIED (POSSIBLY MODEG OR NPINCR IS IN ERROR) /)     GENO0235
 6045 FORMAT (  80H0FATAL DATA ERROR.   GENERATION SPECIFIED BY ABOVE NOGENO0236
     1DE CARD WILL CAUSE NODE  ( ,I5,  20H  )  TO BE REDEFINED  /)    GENO0237
 6050 FORMAT (109H0FATAL DATA ERROR.   THE RADIUS SPECIFIED FOR THE ABOVGENO0238
     1E ARC GENERATION IS LESS THAN HALF THE DISTANCE BETWEEN  /       GENO0239
     2    85H THE CURRENT AND PRIOR NODES OR THE CURRENT AND PRIOR NODESGENO0240
     3 HAVE THE SAME COORDINATES  /)                                   GENO0241
      END                                                              GENO0242
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * GEOM0001
*     CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM              GEOM0002
*                  VERSION 1.0, MARCH 1990                             GEOM0003
*          PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY                       GEOM0004
*            THE FEDERAL HIGHWAY ADMINISTRATION                        GEOM0005
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * GEOM0006
      SUBROUTINE GEOM(AA,D1,D2,D3)                                     GEOM0007
*                                                                      GEOM0008
*     Uncomment this statement in all program modules for double       GEOM0009
*          precision version. SCM 3-10-90                              GEOM0010
*                                                                      GEOM0011
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)                              GEOM0012
*                                                                      GEOM0013
      COMMON /ELEMNT/ AS(3),BS(3),C1(17,17),EST(4,3),RQ(4),RT(3),      GEOM0014
     1                STORE(17),Z4(12),ZQ(4),ZT(3),ZY(17),J1,RO        GEOM0015
      COMMON /ESTABL/ ITRW(28,2),NETCL(17,2),T(28,2)                   GEOM0016
      DIMENSION IS(5)                                                  GEOM0017
      DATA IS /1,2,3,1,2/                                              GEOM0018
C                                                                      GEOM0019
C     GEOMETRIC PROPERTIES OF ELEMENT CALCULATED                       GEOM0020
C                                                                      GEOM0021
      AA=0.0                                                           GEOM0022
      D1=0.0                                                           GEOM0023
      D2=0.0                                                           GEOM0024
      D3=0.0                                                           GEOM0025
      DO 110 I=1,3                                                     GEOM0026
      J=IS(I+1)                                                        GEOM0027
      K=IS(I+2)                                                        GEOM0028
      D4 = ZT(J) -ZT(K)                                                GEOM0029
      D5 = RT(K) - RT(J)                                               GEOM0030
      AA = AA + RT(I)*D4                                               GEOM0031
      D1 = D1 + D4*D4                                                  GEOM0032
      D2=D2+D4*D5                                                      GEOM0033
      D3=D3+D5*D5                                                      GEOM0034
      BS(I)=D4                                                         GEOM0035
  110 AS(I)=D5                                                         GEOM0036
C                                                                      GEOM0037
C     CONSTRAINT EQUATIONS NECESSARY FOR CONVERGENCE FORMED            GEOM0038
C                                                                      GEOM0039
      DU=1.0/AA                                                        GEOM0040
      IF(J1 .EQ. 2) GO TO 117                                          GEOM0041
      DO 112 I=1,3                                                     GEOM0042
      T(I+18,1) = ZT(I) * DU                                           GEOM0043
      T(I+24,1) = ZT(I) *DU                                            GEOM0044
  112 T(I+21,1) = -RT(I) * DU                                          GEOM0045
      GO TO 140                                                        GEOM0046
  117 DO 121 I=1,3                                                     GEOM0047
      T(I+19,2) = -RT(I) * DU                                          GEOM0048
      T(I+25,2) = -RT(I) * DU                                          GEOM0049
  121 T(I+22,2) = ZT(I) * DU                                           GEOM0050
  140 RETURN                                                           GEOM0051
      END                                                              GEOM0052
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * HARD0001
*     CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM              HARD0002
*                  VERSION 1.0, MARCH 1990                             HARD0003
*          PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY                       HARD0004
*            THE FEDERAL HIGHWAY ADMINISTRATION                        HARD0005
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * HARD0006
      SUBROUTINE HARDIN (ST,STHARD,NEL,ICON,ITER,MN,NELEM)             HARD0007
C                                                                      HARD0008
C     THIS SUBROUTINE GENERATES AN INCREMENTAL PLAIN STRAIN            HARD0009
C     CONSTITUTIVE MATRIX BASED ON THE EXTENDED-HARDIN SOIL MODEL.     HARD0010
C     REFERENCE ... AFWL-TR-72-201, MARCH 1973, BY B. HARDIN.          HARD0011
C                                                                      HARD0012
*     Uncomment this statement in all program modules for double       HARD0013
*          precision version. SCM 3-10-90                              HARD0014
*                                                                      HARD0015
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)                              HARD0016
*                                                                      HARD0017
      DIMENSION ST(6,NELEM),STHARD(6,NELEM)                            HARD0018
      COMMON /MATERL/ CP(3,3),DEN(10),E(10,10),GNU(10,10),H(10,10),    HARD0019
     1                ITYPE(10),NON(10)                                HARD0020
      DATA TOLER /0.05/                                                HARD0021
C                                                                      HARD0022
C     THE FOLLOWING VARIABLES ARE IDENTIFIED ...                       HARD0023
C                                                                      HARD0024
C          NEL = ELEMENT NUMBER                                        HARD0025
C          MN = MATERIAL NUMBER                                        HARD0026
C          ITER = ITERATION NUMBER WITHIN LOAD STEP                    HARD0027
C          ICON = CONVERGENCE CODE, 0 = NO, 1 = YES.                   HARD0028
C          H(MN,1) = HARDIN SOIL PARAMETER, 1230*F                     HARD0029
C          H(MN,2) = HARDIN SOIL PARAMETER, C1                         HARD0030
C          H(MN,3) = HARDIN SOIL PARAMETER, A                          HARD0031
C          H(MN,4) = EXTENDED-HARDIN PARAMETER, MIN POISSON RATIO      HARD0032
C          H(MN,5) = EXTENDED-HARDIN PARAMETER, MAX POISSON RATIO      HARD0033
C          H(MN,6) = EXTENDED HARDIN CURVE PARAMETER                   HARD0034
C          ST(I,NEL) = STRAIN STATE. I=1,2,3  (XX,YY,XY)               HARD0035
C          ST(I,NEL) = STRESS STATE. I=4,5,6  (XX,YY,XY)               HARD0036
C          STHARD(1,NEL) = MAXIMUM SHEAR AT PREVIOUS LOAD STEP         HARD0037
C          STHARD(2,NEL) = MAXIMUM SHEAR STRAIN AT PREVIOUS LOAD STEP  HARD0038
C          STHARD(3,NEL) = AVERAGE PRESSURE AT THE PREVIOUS LOAD STEP  HARD0039
C          STHARD(4,NEL) = VOLUME CHANGE AT THE PREVIOUS LOAD STEP     HARD0040
C          STHARD(5,NEL) = SECANT SHEAR MODULUS AT LAST ITERATION STEP HARD0041
C          STHARD(6,NEL) = SECANT BULK MODULUS AT LAST ITERATION STEP  HARD0042
C                                                                      HARD0043
C**   RECOVER SYSTEM RESPONSES                                         HARD0044
C                                                                      HARD0045
      AA = ST(1,NEL)*ST(4,NEL)-ST(2,NEL)*ST(5,NEL)                     HARD0046
      BB = ST(1,NEL)*ST(5,NEL) - ST(2,NEL)*ST(4,NEL)                   HARD0047
      XNU2= H(MN,4)                                                    HARD0048
      IF(BB.NE.0.0) XNU2= BB/(AA+BB)                                   HARD0049
      TEMP = ST(1,NEL) - ST(2,NEL)                                     HARD0050
      GAMMA2 = SQRT(TEMP**2+ST(3,NEL)**2)                              HARD0051
      TEMP = 0.5* (ST(4,NEL) - ST(5,NEL))                              HARD0052
      TAU2 = SQRT(TEMP**2 +ST(6,NEL)**2)                               HARD0053
      SIG0 = -(ST(4,NEL)+ST(5,NEL))*(1.+XNU2)/3.0                      HARD0054
      IF(SIG0.LE.0.0) SIG0=1.0                                         HARD0055
C                                                                      HARD0056
C**   COMPUTE HARDIN VARIABLES                                         HARD0057
C                                                                      HARD0058
      GMAX = H(MN,1)*SQRT(SIG0)                                        HARD0059
      GAMMAR = GMAX/H(MN,2)                                            HARD0060
      RATIO = GAMMA2/GAMMAR                                            HARD0061
      GAMMAP = H(MN,6)*RATIO                                           HARD0062
      TERM = H(MN,3)/EXP(RATIO**0.4)                                   HARD0063
      GAMMAH = RATIO*(1. + TERM)                                       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) +GAMMAP*H(MN,5))/(1.+GAMMAP)
      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**   CKECK 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 CONVT(ECORD,VCORD)
      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 HEROIC (BIV,DENSTY,IA,ICOME,IEXIT,LEVEL,NBPTC,MNO,
     1                  NELEM,NINC,NOD,NODB,NPMAT,NPPT,NPT,NQ,NSMAT,
     2                  NXMXX,PIPMAT,RESULT,SK,SM,ST,STHARD,STPIPE,
     3                  U,V,X,Y)
C
C
C     INCREMENTAL PLAIN STRAIN ANALYSIS
C     HEROIC = (HER)MANN (O)N (I)NCREMENTAL (C)ONSTRUCTION.
C
C
*     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),MNO(NELEM),NOD(4,NELEM),NODB(NBPTC),
     1         NQ(NPT+1),PIPMAT(5,NPPT),RESULT(20,NPPT),ST(6,NELEM),
     2         STHARD(6,NELEM),STPIPE(2,NPPT),U(NPT),V(NPT),X(NPT),
     3         Y(NPT)
C
      COMMON /SIZEPR/ BIVD(3),IX,NINCC
      COMMON /ELEMNT/ AS(3),BS(3),CI(17,17),EST(4,3),RQ(4),RT(3),
     1                STORE(17),Z4(12),ZQ(4),ZT(3),ZY(17),J1,RQ
      COMMON/SYSTEM/B(240),A(240,120)
      COMMON /MATERL/ CP(3,3),DEN(10),E(10,10),GNU(10,10),H(10,10),
     1                ITYPE(10),NON(10)
*
*     Expand array dimension statments from 30 to 99.
*          SCM  5-22-89
*
      COMMON/XFACE/NXMAT,ANGLE(99),FCOEF(99),TENSIL(99),ITMAX
*
*      Add common block.   SCM  10-23-89
*
      COMMON /OUTPUT/ IWRT,IPLOT,MAXIT,JCON
*
*     Revise MAXMAT dimension statement from 30 to 99.
*          SCM  5-22-89
*
      DATA LUDATA,MAXMAT /11,99/
*
C
      GO TO (100,110), ICOME
C
C     FIRST ENTRY ESTABLISHES CONSTANTS AND READS FINITE ELEMENT DATA.
C
100   CONTINUE
      NEQ = 1200
      NXMAT = NXMXX
C
C
C**   ALL INPUT VALUES FROM PREP ARE READ HERE FROM TAPE LUDATA
C
*
*     Revise I/O unit from LUDATA to Unit 14.
*          SCM 11-22-88
*
      REWIND 14
      READ (14) NPT,NELEM,NBPTC,NINC,NPUTCK
      READ (14) (X(N),Y(N),N=1,NPT),((NOD(K,N),K=1,4),MNO(N),
```

```
     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
      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(1,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 = IABS(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
      ITER = 1
C
155   CONTINUE
C
C     THE EQUATION POSITIONING MATRIX FOR THE SYSTEM MATRIX IS FORMED
C     AND THE ELEVATION OF THE TOP OF THE CONSTRUCTION INCREMENT ESTABLI
C
      DO 160 I=1,NPT
  160 NQ(I+1)=0
      DO 198 M=1,NELEM
      IAC=MOD(MNO(M),1000)/10
      IF(IAC.GT.IA) GO TO 198
      LS=200
      IF (NOD(2,M) .EQ. NOD(3,M)) LS = 300
      DO 180 J=1,4
      I = NOD(J,M) + 1
  180 IF(NQ(I).LT.300)NQ(I)=LS
  198 CONTINUE
      NQ(1)=100
      DO 200 I=1,NPT
  200 NQ(I+1)=NQ(I)+NQ(I+1)
C
C     THE BAND WIDTH-NPC- IS COMPUTED
C
      NPC=0
      DO 218 I=1,NELEM
      IAC = MOD(MNO(I),1000)/10
      IF(IAC.GT.IA) GO TO 218
      DO 202 J=1,4
      JJ = NOD(J,I)
      NQ(JJ)=NQ(JJ)+1
      DO 202 K=1,4
      KK = NOD(K,I) + 1
      IV=NQ(KK)/100-NQ(JJ)/100
      IF (NPC  .LT. IV) NPC=IV
  202 CONTINUE
  218 CONTINUE
C
C     COMPUTE MATRIX SPECIFICATIONS
C
      IF (NPC .LE. NEQ) GO TO 220
      WRITE(6,6010) NPC,NEQ
      RETURN
  220 NCOL = NPC
      NROW=NQ(NPT +1)/100-1
C
C     INITIALIZE SYSTEM MATRIX
C
      MN=2*NEQ
      DO 223 N=1,MN
      B(N) = 0.0
      DO 223 J=1,NEQ
  223 A(N,J) = 0.0
C
C     THE STIFFNESS MATRIX IS GENERATED IN BLOCKS AND STORED ON TAPE
C
*
*     Revise I/O unit from LUDATA to Unit 14.
*          SCM 11-22-88
*
      REWIND 14
*
      LB=0
```

```
      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 (NOD(1,I) .LT. 0) GO TO 355
      IAC=MOD(MNO(I),1000)/10
      MN=MNO(I)/1000
      IBEND = NOD(2,I) -NOD(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 = NOD(L,I)
      K1=NQ(KK)/100
      IF (K1 .LE. MB) GO TO 305
  300 CONTINUE
      GO TO 355
C
C     CHOOSE APPROPIATE 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
  349 CALL STIFNS(BIV,NOD,NODB,NQ,X,Y,MNO,U,V,ST,IA,LB1,MN,IAC,MNX,
     X            ITER,ICON,NEQ,NBPTC,NELEM,NPT)
      NELEMT=NELEMT+1
      NOD(1,I) = -NOD(1,I)
  355 CONTINUE
C
C     IF HAVE LESS THAN 1 BLOCK  OF EQUATIONS LEFT NO ADDITIONAL TAPE
C     STORAGE IS USED
C
      IF(NROW .LE. MB ) GO TO 560
      LB1=LB1+1
C
C     THE BLOCK OF EQUATIONS IS REDUCED AND PUT ON TAPE
C
      CALL REDUCI(NEQ,NCOL)
*
*     Revise I/O unit from LUDATA to Unit 14.
*          SCM  11-22-88
*
      WRITE(14)  (B(N),(A(N,M),M=1,NCOL),N=1,NEQ)
*
C
C     THE NUMBER TWO BLOCK OF EQUATIONS IS SHIFTED INTO THE NUMBER ONE
C     POSITION
C
      DO 555 L=1,NEQ
      LRH=L+NEQ
      B(L) = B(LRH)
      B(LRH) = 0.0
      DO 555  M=1,NCOL
      A(L,M) = A(LRH,M)
  555 A(LRH,M) = 0.0
      GO TO 288
  560 NT=NROW-LB1*NEQ
C
C     THE FINAL TWO BLOCKS OF EQUATIONS ARE REDUCED AND BACK-SUBSTITUTED
C
      CALL REDUCI(NT,NCOL)
      CALL BAKSUB(NT,NCOL,LB,NEQ)
      IF (LB1 .EQ. 0) GO TO 564
      DO 563 J=1,LB1
      DO 562 L=1, NEQ
      LL=L+NEQ
  562 B(LL) = B(L)
      L =LB1-J+1
C
C     ANY BLOCKS ON THE TAPE ARE RECALLED AND BACK-SUBSTITUTED
C
*
*     Revise I/O unit from LUDATA to Unit 14.
*          SCM 11-22-88
*
      BACKSPACE 14
      READ(14)  (B(N),(A(N,M),M=1,NCOL),N=1,NEQ)
      BACKSPACE 14
*
      CALL BAKSUB(NEQ,NCOL,L,NEQ)
  563 CONTINUE
  564 L=0
      DO 566 N=1,LB
      K=NEQ
      DO 566 KK=1,NEQ
      K=K+1
```

```
      L=L+1                                                             HERC0319
  566 B(L) = A(K,N)                                                     HERC0320
C                                                                       HERC0321
C     RECORD THE STRUCTURAL  RESPONSES FOR CONVERGED SOLUTIONS          HERC0322
C                                                                       HERC0323
      IF(IEXIT.EQ.-1) GO TO 580                                         HERC0324
      REWIND 13                                                         HERC0325
      WRITE(13)  (U(I),V(I),I=1,NPT),((ST(J,I),I=1,NELEM),J=1,6)        HERC0326
     1          ,((STPIPE(J,I),I=1,NPMAT),J=1,2)                        HERC0327
      GO TO 590                                                         HERC0328
C                                                                       HERC0329
C     READ PREVIOUSLY CONVERGED STRUCTURAL RESPONSES IF IEXIT = -1      HERC0330
C                                                                       HERC0331
  580 REWIND 13                                                         HERC0332
      READ(13)  (U(I),V(I),I=1,NPT),((ST(J,I),I=1,NELEM),J=1,6)         HERC0333
     1          ,((STPIPE(J,I),I=1,NPMAT),J=1,2)                        HERC0334
C                                                                       HERC0335
590   CONTINUE                                                          HERC0336
C                                                                       HERC0337
C     STRESSES AND STRAINS COMPUTED                                     HERC0338
C                                                                       HERC0339
      CALL STRESS (BIV,MNO,NOD,NODB,NQ,ST,STPIPE,U,V,IA,NBPTC,          HERC0340
     1              NELEM,NELEMT,NPMAT,NPPT,NPT,SK,SM)                  HERC0341
C                                                                       HERC0342
C     PIPE RESULTS COLLECTED                                            HERC0343
C                                                                       HERC0344
      CALL RESPIP (MNO,NOD,RESULT,ST,STPIPE,U,V,X,Y,NELEM,NPMAT,        HERC0345
     1              NPPT,NPT,IEXIT,2)                                   HERC0346
C                                                                       HERC0347
      IF(ICON.NE.1) NINC = -NINCC                                       HERC0348
*                                                                       HERC0349
*     Revise convergence limit and add JCON convergence flag for Duncan HERC0350
*          subroutine. SCM  1-15-89                                     HERC0351
*                                                                       HERC0352
*     IF(ITER.LE.20) RETURN                                             HERC0353
*                                                                       HERC0354
      IF(ITER.LT.MAXIT) RETURN                                          HERC0355
      IF(JCON.LT.1) WRITE(6,5000) IA,ITER                               HERC0356
*                                                                       HERC0357
      RETURN                                                            HERC0358
C                                                                       HERC0359
C**   FORMATS                                                           HERC0360
C                                                                       HERC0361
 6010 FORMAT(1H1,41HFATAL ERROR BANDWIDTH IS TOO LARGE         /        HERC0362
     1           30HBANDWIDTH IS .................        ,I5  /        HERC0363
     2           30HMAXIMUM ALLOWABLE BANDWIDTH IS        ,I5  )        HERC0364
5000  FORMAT(//23H *** WARNING, LOAD STEP,I3,23H DID NOT CONVERGE AFTER,HERC0365
     X I3,12H ITERATIONS.,23H PROGRAM WILL CONTINUE. )                  HERC0366
      NINC = NINCC                                                      HERC0367
      END                                                               HERC0368
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * HING0001
*     CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM                HING0002
*                VERSION 1.0, MARCH 1990                                HING0003
*          PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY                        HING0004
*            THE FEDERAL HIGHWAY ADMINISTRATION                         HING0005
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * HING0006
      SUBROUTINE HINGE(PIPMAT,NPMAT,PS,PDIA,YIELD,SM,SK,RESULT,NPPT,    HING0007
     X PHINGE)                                                          HING0008
C                                                                       HING0009
C     THIS ROUTINE CALCULATES A CORRECTIONAL SET OF STRUCTURAL RESPONSESHING0010
C     TO ACCOUNT FOR PLASTIC YIELDING AT THE QUARTER POINTS OF A        HING0011
C     CIRCULAR CULVERT.                                                 HING0012
C                                                                       HING0013
C     THE FOLLOWING HINGING EVALUATIONS ARE CALCULATED ...              HING0014
C                                                                       HING0015
C          PHINGE(1) = RATIO OF ELASTIC MOMENT TO YIELD MOMENT.         HING0016
C          PHINGE(2) = RATIO OF YIELDED SECTION TO TOTAL SECTION.       HING0017
C          PHINGE(3) = RATIO OF YIELDED CIRCUMFERENCE TO TOTAL.         HING0018
C          PHINGE(4) = RATIO OF PLASTIC DISPLACEMENT TO TOTAL.          HING0019
C          PHINGE(5) = RATIO OF TOTAL STRAIN TO YIELD STRAIN.           HING0020
C                                                                       HING0021
*     Uncomment this statement in all program modules for double        HING0022
*          precision version. SCM 3-10-90                               HING0023
*                                                                       HING0024
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               HING0025
*                                                                       HING0026
*     Revise dimensions of RESULT, DESULT and PIPMAT arrays from        HING0027
*          30 to 99.   SCM  4-30-89                                     HING0028
*                                                                       HING0029
      DIMENSION RESULT(20,99),XK(6,6),A(6),U(6,6),DESULT(8,99),FW(7)    HING0030
     X ,PHINGE(5),PIPMAT(5,99)                                          HING0031
*                                                                       HING0032
C                                                                       HING0033
C     SET CONSTANTS                                                     HING0034
C                                                                       HING0035
      DO 50 I=1,NPPT                                                    HING0036
      DO 50 J=1,7                                                       HING0037
   50 DESULT(J,I) = 0.0                                                 HING0038
      DO 55 I=1,5                                                       HING0039
   55 PHINGE(I) = 0.0                                                   HING0040
C                                                                       HING0041
      PE = PIPMAT(1,NPMAT)                                              HING0042
      PNU = PIPMAT(2,NPMAT)                                             HING0043
      PA = PIPMAT(3,NPMAT)                                              HING0044
      PI = PIPMAT(4,NPMAT)                                              HING0045
      PRAD = PDIA/2.                                                    HING0046
      PC = PI/PS                                                        HING0047
      NPO = (NPPT+1)/2                                                  HING0048
      NP1 = NPO-1                                                       HING0049
      NP2 = NPO-2                                                       HING0050
      IF(RESULT(5,NPO).EQ.0.0) RETURN                                   HING0051
      SIGN = RESULT(5,NPO)/ABS(RESULT(5,NPO))                           HING0052
      XMEO = RESULT(5,NPO)*SIGN                                         HING0053
      XME1 = RESULT(5,NP1)*SIGN                                         HING0054
      XME2 = RESULT(5,NP2)*SIGN                                         HING0055
      XMYLD = YIELD*PI/PC                                               HING0056
      XMULT = 1.5*XMYLD                                                 HING0057
      PHINGE(1) = XMYLD/XMEO                                            HING0058
      PHINGE(5) = XMEO/(PS*YIELD)                                       HING0059
      IF(XMEO .LT. 1.01*XMYLD) RETURN                                   HING0060
      SPK = SM*(1.-SK)/PRAD                                             HING0061
      ALPHA = PRAD**4*SPK*(1.-PNU**2)/(PE*PI)                           HING0062
      SQRAL = SQRT(ALPHA)                                               HING0063
      XP = 0.5*SQRT(2.*SQRAL-1.)                                        HING0064
      XQ = 0.5*SQRT(2.*SQRAL+1.)                                        HING0065
      C1 = 1.+SQRAL                                                     HING0066
      C2 = -1.+SQRAL                                                    HING0067
      C3 = SQRT(ALPHA-.25)                                              HING0068
      C4 = 0.5-ALPHA                                                    HING0069
      F1 = PI/(PA*PRAD**2)                                              HING0070
      F2 = PRAD**2*SPK*(1.-PNU**2)/(PE*PA) + 1.                         HING0071
      F3 = PE*PI/(PRAD**2*(1.-PNU**2))                                  HING0072
      F4 = YIELD*PRAD*(1.-PNU**2)/(PE*PC)                               HING0073
      HPI = 2.*ATAN(1.0)                                                HING0074
      DTHETA = 2.*HPI/(NPPT-1.)                                         HING0075
      BE = (-3.*XMEO+4.*XME1-XME2)/(2.*DTHETA)                          HING0076
```

```
      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
      CALL SETU(0.0,C1,C2,C3,C4,SQRAL,XP,XQ,U)
      DO 100 I=1,4
      XK(1,I) = U(5,I)
      XK(3,I) = U(4,I)
100   XK(5,I) = F1*U(5,I) + F2*U(1,I)
      CALL SETU(HPI,C1,C2,C3,C4,SQRAL,XP,XQ,U)
      DO 110 I=1,4
      XK(2,I) = U(5,I)
      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
      CALL INVER(XK,4)
      DO 120 I=1,4
120   A(I) = (XK(I,3)-XK(I,4))/F3
C
C     SOLVE FOR THE LAST TWO UNKOWN COEFFICIENTS.
C
      T1 = 0.0
      T2 = 0.0
      DO 130 I=1,4
      T1 = T1 + XK(5,I)*A(I)
130   T2 = T2 + XK(6,I)*A(I)
      A(5) = (T1-T2)/(F1*HPI)
      A(6) = T1
C
C     COMPUTE STRUCTURAL RESPONSES FOR A UNIT APLLIED END MOMENTS.
C
      THETA = HPI
      ROTAT = 0.0
      DO 150 N=1,NP0
      DESULT(3,N) = A(5)
      DESULT(4,N) = -A(5)*THETA*F2 + A(6)
      DESULT(5,N) = 0.0
      DESULT(6,N) = -A(5)*PRAD*SPK
      DESULT(7,N) = 0.0
      DESULT(8,N) = -A(5)*SPK
      CALL SETU(THETA,C1,C2,C3,C4,SQRAL,XP,XQ,U)
      DO 140 I=1,4
      IF(N.EQ.NP0) ROTAT = ROTAT + U(3,I)*A(I)/PRAD
      DESULT(3,N) = DESULT(3,N) + A(I)*U(2,I)
      DESULT(4,N) = DESULT(4,N) - A(I)*(F1*U(5,I)+F2*U(1,I))
      DESULT(5,N) = DESULT(5,N) + A(I)*F3*U(4,I)
      DESULT(6,N) = DESULT(6,N) - A(I)*(F3/PRAD*U(6,I)+PRAD*SPK*U(2,I))
      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)
      THETA = THETA-DTHETA
150   CONTINUE
C
C     COMPUTE PLASTIC HINGE ROTATION FOR THE ASSUMED END MOMENT, DM0.
C
      DMMAX = XME0-XMYLD
      DMMIN = XME0-XMULT
      IF(DMMIN.LT.0.0) DMMIN = 0.0
      DMHIGH = DMMAX
      DMLOW = DMMIN
      KOUNT = 0
C
200   DM0 = (DMHIGH+DMLOW)/2.
      XMD0 =-DM0*DESULT(5,NP0)
      XMD1 =-DM0*DESULT(5,NP1)
      XMD2 =-DM0*DESULT(5,NP2)
      BB = (-3.*(XME0+XMD0)+4.*(XME1+XMD1)-(XME2+XMD2))/(2.*DTHETA)
      CC = (XME0+XMD0-2.*(XME1+XMD1)+XME2+XMD2)/(2.*DTHETA**2)
C
C     COMPUTE LENGTH OF PLASTIC HINGE
C
      W1 = (-BB/CC+ SQRT((BB/CC)**2 -4.*(XME0+XMD0-XMYLD)/CC))/2.0
      W2 = (-BB/CC- SQRT((BB/CC)**2 -4.*(XME0+XMD0-XMYLD)/CC))/2.0
      WP = W1
      IF(W2.GT.0.0) WP = W2
C
C     DETERMINE PLASTIC ROTATION OVER HINGE LENGTH WITH 7 POINT SIMPSON
C     INTEGRATION FOR PART1, AND EXACT INTEGRATION FOR PART2.
C
      NWP = 7
      NWP1 =NWP-1
      DW = WP/NWP1
      WW = 0.0
      DO 250 I=1,NWP
      ARG = 1.-(XME0+XMD0+BB*WW+CC*WW*WW)/XMULT
      IF(ARG.LE.0.0) ARG = 0.0000001
      FW(I) = 1./SQRT(3.*ARG)
250   WW = WW + DW
      SIMP = 0.0
      DO 260 I=1,NWP1,2
260   SIMP = SIMP + FW(I) +4.*FW(I+1) + FW(I+2)
      PART1 = SIMP*DW/3.
      PART2 = WP/XMYLD*(XME0+BE*WP/2.+CE*WP*WP/3.)
      ROTAP = F4*(PART1-PART2)
C
C     COMPARE ROTATION FROM ELASTIC SOLUTION,ROTAE , WITH THE ROTATION
C     OF THE PLASTIC HINGE, ROTAP.
C
      ROTAE = -ROTAT*DM0
      RATIO = ROTAE/ROTAP
      IF(ABS(RATIO-1.).LT.0.01) GO TO 300
      IF(ROTAE.LT.ROTAP) DMLOW = DM0
      IF(ROTAE.GT.ROTAP) DMHIGH = DM0
      KOUNT = KOUNT + 1
      IF(KOUNT.GT.30) GO TO 400
      GO TO 200
C
C     COLLECT CONVERGED SOLUTION AND ADD TO ELASTIC SOLUTION
C
300   CONTINUE
      K = NPPT
      THETA = HPI
      DO 350 N=1,NP0
      RDISP = DESULT(3,N)*DM0
      TDISP = DESULT(4,N)*DM0
      DESULT(3,N) = RDISP*COS(THETA) - TDISP*SIN(THETA)
      DESULT(4,N) = RDISP*SIN(THETA) + TDISP*COS(THETA)
      DESULT(3,K) = DESULT(3,N)
      DESULT(4,K) = -DESULT(4,N)
      THETA = THETA-DTHETA
      DO 325 J=5,8
      DESULT(J,N) = DESULT(J,N)*DM0
      DESULT(J,K) = DESULT(J,N)
325   CONTINUE
```

```
      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 INTP1(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 OVERBURNDEN 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
C     PLACED ON TOP OF ELEMENT NEL.
C
      DPRES = 0.0
      DO 200 I=1,NELEM
      IAC = MOD(MNO(I),1000)/10
      IF(IAC.NE.IA) GO TO 200
      MAT = MNO(I)/1000
      MNX = MNO(I)-MAT*1000-IAC*10
      IF(MNX.NE.0) GO TO 200
      ND1 = IABS(NOD(1,I))
      ND1 = IABS(NOD(1,I))
      ND2 = NOD(2,I)
      ND3 = NOD(3,I)
      ND4 = NOD(4,I)
      IF(ND2.EQ.ND3) GO TO 200
*
*     Revise intrinsic funtion to generic form.  SCM 3-10-90
*
      XMIN = MIN(X(ND1),X(ND2),X(ND3),X(ND4))
      XMAX = MAX(X(ND1),X(ND2),X(ND3),X(ND4))
*
      IF(XNEL.LT.XMIN) GO TO 200
      IF(XNEL.GE.XMAX) GO TO 200
*
      YMAX = MAX(Y(ND1),Y(ND2),Y(ND3),Y(ND4))
*
      IF(YNEL.GT.YMAX) GO TO 200
C
C     SUM PRESSURE CONTRIBUTIONS OF EACH NEW ELEMENT ABOVE NEL.
C
      YDEPTH = 0.5*(ABS(Y(ND1)-Y(ND3)) + ABS(Y(ND2)-Y(ND4)))
      IF(I.EQ.NEL) YDEPTH = 0.5*YDEPTH
      GAM = DEN(MAT)
      DPRES = DPRES + GAM*YDEPTH
200   CONTINUE
C
C     TOTAL PRESSURE ON NEL DURING THIS INCREMENT IS PREVIOUS PRESSURE
C     PLUS ONE-HALF OF CURRENT OVERBURNDEN INCREMENT.
C
      PNEL = ABS(ST(5,NEL)) + 0.5*DPRES
C
C     FIND OVERBURDEN PRESSURE DATA POINTS ON EITHER SIDE OF PNEL.
C
      K1 = 1
      K2 = 2
      DO 300 J=2,NOMI
      J1 = J-1
      IF(PNEL.LT.H(MN,J1)) GO TO 300
      IF(PNEL.GE.H(MN,J)) GO TO 300
      K1 = J1
      K2 = J
      GO TO 350
300   CONTINUE
      IF(PNEL.LT.H(MN,NOMI)) GO TO 350
      K1 = NOMI-1
      K2 = NOMI
350   DH = H(MN,K2)-H(MN,K1)
C
C     COMPUTE CHORD MODULUS AND AVERAGE POISSON RATIO
C
      DHB = H(MN,K2) - H(MN,K1)*E(MN,K2)/E(MN,K1)
      IF(DHB.LE.0.0) DHB = 0.01*DH
      EI = E(MN,K2)*DH/DHB
```

```
      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
100   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
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         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-FAILTURE REGION)
C         IFLAG=7     FAILURE REGION
C         IFLAG=-1    UNLOADING,ELASTIC REGION
C
C     START BY SETTING ABSOLUTE VALUES TO CURRENT STRAIN OF
C     PREVIOUS CONVERGED LOAD STEP
C
*     Uncomment this statement in all program modules for double
*          precision version. SCM 3-10-90
*
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
      DIMENSION ANDA1(15),JFLAG(15),ANDA2(15)
      COMMON/LIMITS/PSLIP,SSTRN,PFAIL
      ASTRN=ABS(STRN)
      ASTRNP=ABS(STRNPR)
      ASTRS=ABS(STRES)
      ASTRSP=ABS(STRESP)
      IFLAG=0
C
C     CHECK IF UNLOADING IS OCCURING. IF SO, USE ELASTIC MODULUS
C
      IF(ASTRN.LT.ASTRNP)IFLAG=-1
      IF(IFLAG.EQ.0)GO TO 100
      PENEW=PE
      RETURN
C
C     CHECK IF LOADING IN ELASTIC REGION. IF SO, USE ELASTIC MODULUS.
C
  100 IF(ASTRS.LT.PSLIP) IFLAG=1
      IF(IFLAG.EQ.0) GO TO 200
      IF(JFLAG(NJ).EQ.2)STRES=STRES/ASTRS*PSLIP
      IF(JFLAG(NJ).EQ.2)ASTRS=ABS(STRES)
      IF(JFLAG(NJ).EQ.2) GO TO 200
      JFLAG(NJ)=IFLAG
      PENEW=PE
      RETURN
C
C     CHECK IF LOADING IN TRANSITION REGION ELASTIC-SLIP.
C     IF SO, USE WEIGHTED TRANSITION MODULUS.
C     WEIGHT IS DETERMINED ON FIRST ITERATION SUCH THAT THE
C     FIRST GUESS FOR PENEW IS 1.01*PES (THIS WORKS BEST
C     FOR CONVERGENCE)
C
  200 IF(ASTRS.GE.PSLIP .AND. PSLIP.GT.ASTRSP)IFLAG=2
      IF(IFLAG.EQ.0) GO TO 300
      IF(ASTRN.GT.SSTRN .AND. PSLIP.GT.ASTRSP) GO TO 900
      PENEW=(PE-PES)*(PSLIP-ASTRSP)*(1.-PNU**2)/PE/(ASTRN-ASTRNP)+PES
      IF(JCON.NE.1) GO TO 250
      ANDA=(PEOLD-1.01*PES)/(PEOLD-PENEW)
      IF(ANDA.LT.1.0)ANDA=1.0
      ANDA1(NJ)=ANDA
  250 PENEW=ANDA1(NJ)*PENEW+(1.0-ANDA1(NJ))*PEOLD
      IF(PENEW.GT.PE)PENEW=(PE+PEOLD)/2.
      IF(PENEW.LT.PES.AND.JCON.NE.1)PENEW=(PES+PEOLD)/2.
      IF(PENEW.LT.PES.AND.JCON.EQ.1)PENEW=1.01*PES
      JFLAG(NJ)=IFLAG
      RETURN
C
C     CHECK IF LOADING IN PURE SLIPPING REGION. IF SO,USE SLIPPING
C     MODULUS PES
C
  300 IF(SSTRN.GT.ASTRN .AND. ASTRSP.GE.PSLIP) IFLAG=3
      IF(IFLAG.EQ.0) GO TO 400
      IF(JFLAG(NJ).EQ.4 .AND. ASTRN.LT.SSTRN) STRN=STRN/ASTRN*SSTRN
      IF(JFLAG(NJ).EQ.4 .AND. ASTRN.LT.SSTRN) ASTRN=ABS(STRN)
      IF(JFLAG(NJ).EQ.4) GO TO 400
      PENEW=PES
```

```
      RETURN                                                            JMOD0093
C                                                                       JMOD0094
C     CHECK IF LOADING IN TRANSITION REGION SLIP-POSTSLIP. IF SO,       JMOD0095
C     USE WEIGHTED TRANSITION MODULUS                                   JMOD0096
C                                                                       JMOD0097
  400 IF(ASTRN.GE.SSTRN .AND. SSTRN.GT.ASTRNP)IFLAG=4                   JMOD0098
      IF(IFLAG.EQ.0) GO TO 500                                          JMOD0099
      JFLAG(NJ)=IFLAG                                                   JMOD0100
      PENEW=PEOLD*(ASTRN-ASTRNP)/((SSTRN-ASTRNP)*(1.-PES/PEP)           JMOD0101
     * +(ASTRN-ASTRNP)*PEOLD/PEP)                                       JMOD0102
*                                                                       JMOD0103
*     Revise intrinsic function to generic form.  SCM  3-10-90          JMOD0104
*                                                                       JMOD0105
      IF(JCON.EQ.1 .AND. PEOLD.EQ.PES)SRATIO=LOG10(PEP/PES)/3.*         JMOD0106
     * (SSTRN-ASTRNP)/(ASTRN-SSTRN)                                     JMOD0107
*                                                                       JMOD0108
      IF(PENEW.LT.PEOLD) GO TO 450                                      JMOD0109
      ANDA=2.*PENEW/PEOLD*SRATIO                                        JMOD0110
      IF(ANDA.LT.1.2)ANDA=1.2                                           JMOD0111
      IF((PENEW/PEOLD).LT.1.01)ANDA=1.0                                 JMOD0112
      IF(PES.GT.0.0003*PEP)ANDA=1.0                                     JMOD0113
      PENEW=ANDA*PENEW+(1.-ANDA)*PEOLD                                  JMOD0114
      RETURN                                                            JMOD0115
  450 ANDA=2.*PEOLD/PENEW                                               JMOD0116
      IF((PEOLD/PENEW).LT.1.01)ANDA=1.0                                 JMOD0117
      PENEW=ANDA*PENEW+(1.-ANDA)*PEOLD                                  JMOD0118
      RETURN                                                            JMOD0119
C                                                                       JMOD0120
C     CHECK IF LOADING IN POSTSLIP REGION. IF SO, USE POSTSLIP MODULUS  JMOD0121
C                                                                       JMOD0122
  500 IF(PFAIL.GT.ASTRS .AND. ASTRNP.GE.SSTRN)IFLAG=5                   JMOD0123
      IF(IFLAG.EQ.0)GO TO 600                                           JMOD0124
      IF(JFLAG(NJ).EQ.6)STRES=STRES/ASTRS*PFAIL                         JMOD0125
      IF(JFLAG(NJ).EQ.6)ASTRS=ABS(STRES)                                JMOD0126
      IF(JFLAG(NJ).EQ.6) GO TO 600                                      JMOD0127
      JFLAG(NJ)=IFLAG                                                   JMOD0128
      PENEW=PEP                                                         JMOD0129
      RETURN                                                            JMOD0130
C                                                                       JMOD0131
C     CHECK IF LOADING IN TRANSITION REGION POSTSLIP-FAILURE. IF SO,    JMOD0132
C     USE WEIGHTED TRANSITION MODULUS.                                  JMOD0133
C                                                                       JMOD0134
  600 IF(ASTRS.GE.PFAIL .AND. ASTRSP.LT.PFAIL) IFLAG=6                  JMOD0135
      IF(IFLAG.EQ.0) GO TO 700                                          JMOD0136
      IF(JFLAG(NJ).EQ.4) GO TO 660                                      JMOD0137
      PENEW1=(PEP-PEF)*(PFAIL-ASTRSP)*(1.-PNU**2)/PEP/(ASTRN-ASTRNP)+PEFJMOD0138
      IF(JCON.NE.1) GO TO 650                                           JMOD0139
      ANDA=(PEOLD-1.01*PEF)/(PEOLD-PENEW1)                              JMOD0140
      IF(ANDA.LT.1.0)ANDA=1.0                                           JMOD0141
      ANDA2(NJ)=ANDA                                                    JMOD0142
  650 PENEW=ANDA2(NJ)*PENEW1+(1.-ANDA2(NJ))*PEOLD                       JMOD0143
      IF(PENEW.LT.PEF.AND.JCON.NE.1)PENEW=(PEF+PEOLD)/2.                JMOD0144
      IF(PENEW.LT.PEF.AND.JCON.EQ.1)PENEW=1.01*PEF                      JMOD0145
      IF(PENEW.GT.PEP)PENEW=(PEP+PEOLD)/2.                              JMOD0146
      JFLAG(NJ)=IFLAG                                                   JMOD0147
      RETURN                                                            JMOD0148
C                                                                       JMOD0149
  660 IFLAG=5                                                           JMOD0150
      JFLAG(NJ)=IFLAG                                                   JMOD0151
      PENEW=PEP                                                         JMOD0152
      RETURN                                                            JMOD0153
C                                                                       JMOD0154
C     CHECK IF LOADING IN FAILURE REGION. IF SO, USE FAILURE MODULUS.   JMOD0155
C                                                                       JMOD0156
  700 IF(ASTRS.GT.PFAIL .AND. ASTRSP.GE.PFAIL)IFLAG=7                   JMOD0157
      IF(IFLAG.EQ.0) GO TO 800                                          JMOD0158
      IF(JFLAG(NJ).EQ.4) GO TO 900                                      JMOD0159
      PENEW=PEF                                                         JMOD0160
      RETURN                                                            JMOD0161
C                                                                       JMOD0162
C     CHECK IF IFLAG STILL EQUAL ZERO. IF SO,WE HAVE AN ERROR,          JMOD0163
C     I.E.,LOAD STEP IS TOO LARGE                                       JMOD0164
C                                                                       JMOD0165
  800 WRITE(6,2000)                                                     JMOD0166
      GO TO 950                                                         JMOD0167
  900 WRITE(6,2100)                                                     JMOD0168
  950 STOP                                                              JMOD0169
 2000 FORMAT(////,44H **FATAL ERROR: I.E.OR TOO LARGE LOAD STEP**)      JMOD0170
 2100 FORMAT(////,20H TOO LARGE LOAD STEP)                              JMOD0171
C***********************************************************************JMOD0172
      END                                                               JMOD0173
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * MODM0001
*     CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM                MODM0002
*                   VERSION 1.0, MARCH 1990                             MODM0003
*              PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY                    MODM0004
*                THE FEDERAL HIGHWAY ADMINISTRATION                     MODM0005
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * MODM0006
      SUBROUTINE MODMSH (BIV,NOD,NODB,X,Y,NBPTC,MAXBC,NELEM,NPT)        MODM0007
C                                                                       MODM0008
C**   ROUTINE ALLOWS THE USER TO MODIFITY THE LEVEL 2 MESH              MODM0009
C                                                                       MODM0010
*     Uncomment this statement in all program modules for double        MODM0011
*         precision version. SCM 3-10-90                                MODM0012
*                                                                       MODM0013
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               MODM0014
*                                                                       MODM0015
      DIMENSION BIV(3,NBPTC), NOD(7,NPT),NODB(NBPTC),X(NPT),Y(NPT)      MODM0016
*                                                                       MODM0017
*     Revise array dimension from AN(2) to AN(3) for generalized        MODM0018
*         boundary condition option.  SCM 10-25-89                      MODM0019
*                                                                       MODM0020
      DIMENSION AN(3),ANI(2),BV(3),IFLAG(2),ND(7)                       MODM0021
      DIMENSION RBC(2)                                                  MODM0022
*                                                                       MODM0023
      DATA ANI /4HF = ,4HD = /                                          MODM0024
*                                                                       MODM0025
*     Add data statement for generalized boundary condition option.     MODM0026
*         SCM 10-25-89                                                  MODM0027
*                                                                       MODM0028
      DATA RBC /4H F  ,4H D  /                                          MODM0029
*                                                                       MODM0030
C                                                                       MODM0031
C**   READ A CONTROL CARD TO DETERMINE WHAT PARAMETERS ARE TO BE CHANGEDMODM0032
C                                                                       MODM0033
      READ (5,5010) NEWXY, NEWEL, NEWBD                                 MODM0034
      WRITE(6,6010) NEWXY, NEWEL, NEWBD                                 MODM0035
      IF (NEWXY + NEWEL+ NEWBD .EQ. 0) GO TO 120                        MODM0036
      IF (NEWXY .EQ. 0) GO TO 30                                        MODM0037
      WRITE( 6,6020)                                                    MODM0038
C                                                                       MODM0039
C**   CHANGE THE NODE COORDINATES                                       MODM0040
C                                                                       MODM0041
      DO 20 N = 1, NEWXY                                                MODM0042
      READ(5,5015) NEWNOD, XCOORD, YCOORD                               MODM0043
      IF (NEWNOD .GT. 0 .AND. NEWNOD .LT. NPT) GO TO 10                 MODM0044
      WRITE(6,6030) NEWNOD                                              MODM0045
```

```
      GO TO 20                                                          MODM0046
   10 WRITE(6,6040) NEWNOD, X(NEWNOD),Y(NEWNOD),XCOORD,YCOORD           MODM0047
      X(NEWNOD) = XCOORD                                                MODM0048
      Y(NEWNOD) = YCOORD                                                MODM0049
   20 CONTINUE                                                          MODM0050
C                                                                       MODM0051
C**   READ IN MODIFIED ELEMENTS                                         MODM0052
C                                                                       MODM0053
   30 CONTINUE                                                          MODM0054
      IF (NEWEL .EQ. 0) GO TO 70                                        MODM0055
      WRITE(6,6050)                                                     MODM0056
      DO 60 N = 1, NEWEL                                                MODM0057
      READ(5,5030) NE,(ND(I),I=1,7)                                     MODM0058
      IF (NE.GT. 0 .AND. NE .LE. NELEM) GO TO 40                        MODM0059
      WRITE(6,6060) NE                                                  MODM0060
      GO TO 60                                                          MODM0061
   40 CONTINUE                                                          MODM0062
      DO 50 K = 1, 7                                                    MODM0063
      IF (ND(K) .EQ. 0) ND(K) = NOD(K,NE)                               MODM0064
      IF (ND(K) .EQ. -1) ND(K) = 0                                      MODM0065
   50 CONTINUE                                                          MODM0066
      WRITE(6,6070) NE, (NOD(I,NE),I=1,7),(ND(I),I=1,7)                 MODM0067
      DO 55 K = 1, 7                                                    MODM0068
   55 NOD(K,NE) = ND(K)                                                 MODM0069
   60 CONTINUE                                                          MODM0070
   70 CONTINUE                                                          MODM0071
C                                                                       MODM0072
C**   DEFINE NEW BOUNDARY CONDITIONS                                    MODM0073
C                                                                       MODM0074
      IF (NEWBD .EQ. 0) RETURN                                          MODM0075
      WRITE(6,6090)                                                     MODM0076
      DO 110 N = 1, NEWBD                                               MODM0077
      READ(5,5040) NU,IFLAG(1),BV(1),IFLAG(2),BV(2),BV(3),IA            MODM0078
      NBPTC = NBPTC + 1                                                 MODM0079
      IF (NBPTC .LE. MAXBC) GO TO 75                                    MODM0080
      WRITE(6,6120) MAXBC,NBPTC                                         MODM0081
      GO TO 110                                                         MODM0082
   75 IF (NU .GT.0.AND. NU .LE. NPT) GO TO 80                           MODM0083
      WRITE( 6,6030) NU                                                 MODM0084
      STOP 76                                                           MODM0085
   80 CONTINUE                                                          MODM0086
      DO 90  J = 1, 2                                                   MODM0087
      IF (IFLAG(J) .EQ. 0) AN(J) = ANI(1)                               MODM0088
      IF (IFLAG(J) .EQ. 1) AN(J) = ANI(2)                               MODM0089
*                                                                       MODM0090
*     Add extended boundary condition definition.   MGK 3-18-89         MODM0091
*                                                                       MODM0092
      IF (IFLAG(J) .EQ. 2) AN(J) = ANI(2)                               MODM0093
      IF (IFLAG(J) .EQ. 3) AN(J) = ANI(1)                               MODM0094
*                                                                       MODM0095
   90 CONTINUE                                                          MODM0096
*                                                                       MODM0097
*     Extend boundary condition output for rotational constraints.      MODM0098
*         SCM   10-25-89                                                MODM0099
*                                                                       MODM0100
      IF(IFLAG(1).EQ.0 .AND. IFLAG(2).EQ.0) THEN                        MODM0101
          AN(3) = RBC(1)                                                MODM0102
      ELSE IF(IFLAG(1).EQ.2 .AND. IFLAG(2).EQ.0) THEN                   MODM0103
          AN(3) = RBC(1)                                                MODM0104
      ELSE IF(IFLAG(1).EQ.0 .AND. IFLAG(2).EQ.2) THEN                   MODM0105
          AN(3) = RBC(1)                                                MODM0106
      ELSE IF(IFLAG(1).EQ.2 .AND. IFLAG(2).EQ.2) THEN                   MODM0107
          AN(3) = RBC(1)                                                MODM0108
      ELSE                                                              MODM0109
          AN(3) = RBC(2)                                                MODM0110
      END IF                                                            MODM0111
      WRITE(6,6080) NU,IA,AN(1),BV(1),AN(2),BV(2),                      MODM0112
     X                                 BV(3),AN(3)                      MODM0113
      BV(3)=BV(3)*0.0174533                                             MODM0114
*                                                                       MODM0115
      DO 100 I = 1, 3                                                   MODM0116
  100 BIV(I,NBPTC) = BV(I)                                              MODM0117
      NODB(NBPTC) = NU*10000 + IA*100 + IFLAG(1)*10+IFLAG(2)            MODM0118
  110 CONTINUE                                                          MODM0119
  120 CONTINUE                                                          MODM0120
      RETURN                                                            MODM0121
 5010 FORMAT (3I5)                                                      MODM0122
 5015 FORMAT (I5,5X,2F10.0)                                             MODM0123
 5030 FORMAT (8I5)                                                      MODM0124
5040  FORMAT (2I5,F10.0,I5,F10.0,F10.0,I5)                              MODM0125
 6010 FORMAT (1H1,10X,47H * * * CHANGES TO STANDARD LEVEL 2 MESH * * * *MODM0126
     1              /10X,40H * NUMBER OF NODES TO BE CHANGED ------- ,I7,  MODM0127
     1          1H* /10X,40H * NUMBER OF ELEMENTS TO BE CHANGED ---- ,I7,  MODM0128
     1          1H* /10X,40H * ADDITIONAL BOUNDARY CONDITIONS ------ ,I7,  MODM0129
     4           1H*/10X,47H * * * * * * * * * * * * * * * * * *  * * * *MODM0130
     5           ///)                                                   MODM0131
 6020 FORMAT (1H ,13X,11HNODE NUMBER,13X,15HOLD COORDINATES,20X,        MODM0132
     1               15HNEW COORDINATES /                               MODM0133
     2               40X,1HX,10X,1HY,25X,1HX,10X,1HY)                   MODM0134
 6030 FORMAT (//103H * * WARNING NODE NUMBER IS OUTSIDE OF ALLOWABLE RANMODM0135
     1GE (0 TO NPT) AND CARD WILL BE SKIPPED NODE READ = ,I5 ///)       MODM0136
 6040 FORMAT (15X,I5,10X,F10.2,10X,F10.2,15X,F10.2,10X,F10.2)           MODM0137
 6050 FORMAT (////10X,15H ELEMENT NUMBER,10X, 19H ELEMENT CONNECTIVE,   MODM0138
     1                5X,16H MATERIAL NUMBER,2X,19H CONSTRUCTION INCR. ,5X,MODM0139
     2                4HX(7)/ 33X,1HI,7X,1HJ,7X,1HK,7X,1HL/)            MODM0140
 6060 FORMAT (///10X,46H* * * WARNING ELEMENT NUMBER IS OUT OF BOUNDS.  MODM0141
     1                ,40H (1 TO NELEM) ELEMENT NUMBER IS ******* ,I5///)  MODM0142
 6070 FORMAT(14X,I5,7X,4(4X,I4),7X,I4,15X,I5,12X,I4,5X,                 MODM0143
     1          22HOLD ELEMENT PARAMETERS/26X,4(4X,I4),7X,I4,15X,I5,    MODM0144
     2        12X,I4,5X,22HNEW ELEMENT PARAMETERS /)                    MODM0145
*                                                                       MODM0146
*     Revise formats for generalized boundary condition option.         MODM0147
*         SCM 10-25-89                                                  MODM0148
*                                                                       MODM0149
 6090 FORMAT(1H1,1X,31H* * * BOUNDARY CONDITIONS * * */2X,              MODM0150
     X59H(FORCES = LBS; DISPLACEMENTS = INCHES; ROTATIONS = DEGREES)//  MODM0151
     X2X,8HBOUNDARY,6X,4HLOAD,6X,14HX-FORCE     OR,6X,14HY-FORCE     OR,MODM0152
     X6X,12HX-Y ROTATION,6X,15HBEAM ROTATIONAL/6X,4HNODE,6X,4HSTEP,6X,  MODM0153
     X14HX-DISPLACEMENT,6X,14HY-DISPLACEMENT,22X,                       MODM0154
     X19HBOUNDARY CONDITIONS//)                                         MODM0155
 6080 FORMAT(5X,I5,5X,I5,4X,A4,E12.4,4X,A4,E12.4,4X,E12.4,14X,A4)       MODM0156
*                                                                       MODM0157
 6120 FORMAT (1H1,50H FATAL ERROR MAX. NUMBER OF BOUNDARY POINTS IS ,I5/MODM0158
     1             50H NUMBER OF BOUNDARY POINTS REQUIRED BY INPUT   ,I5/ MODM0159
     2              50HTO RUN THIS PROBLEM INCREASE SIZE OF BLANK COMMON /MODM0160
     3              50H IN SUBROUTINE PREHERO AND RECOMPILE THE ROUTINE  )MODM0161
      END                                                               MODM0162
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * PLAS0001
*     CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM                PLAS0002
*                   VERSION 1.0, MARCH 1990                             PLAS0003
*             PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY                     PLAS0004
*               THE FEDERAL HIGHWAY ADMINISTRATION                      PLAS0005
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * PLAS0006
      SUBROUTINE PLASTI(IA,ICOME,IEXIT,LEVEL,NINC,NPMAT,NPPT,PDIA,      PLAS0007
     1                  PIPMAT,RESULT,SK,SM,XMODE)                      PLAS0008
C     SUBROUTINE FOR PLASTIC PIPE CULVERTS.                             PLAS0009
```

```
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)
C              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 pipe dimension statements from 30 to 99.
*          SCM   5-22-89
*
      DIMENSION FS(3),PFS(3),PIPMAT(5,99),RESULT(20,99)
*
      DATA E,XNU,YIELD/1600000.,0.3,25000./
      DATA NFS,FS/3,4.,3.,3./
      DATA FF /0.333/
      DATA ANALYS, DESIGN /4HANAL,4HDESI/
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
200   IEXIT = 0
C
C     CHECK FOR CONVERGENCE OF NONLINEAR SOIL LAW
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,NPPT
      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/NPPT)
      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
*
*     Revise intrinsic function to generic form.  SCM 3-10-90
*
      RMAX = MAX(DISPR, OUTFR, BUCKR)
*
      IF(ABS(1.0 - RMAX) .LE. 0.05) GO TO 250
```

```
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, DISPSF, OUTFSF, BUCKSF, 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 E
     XVALUATED, / 10X,37H WHERE THE SECTION PROPERTIES ARE ... //
     X 15X, 36H THICKNESS OF PIPEWALL (IN)......... , F10.5 //
     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 //)
2300  FORMAT(1H1,//10X,19H DESIGN RESULTS AT ,I2,11H ITERATIONS //
     X 15X, 35H *** REQUIRED WALL THICKNESS....... , F10.5 //
     X 15X, 35H *** REQUIRED THRUST AREA ......... , F10.5 //
     X 15X, 35H *** REQUIRED MOM. OR INERTIA ..... , F10.5 //
     X 15X, 35H *** REQUIRED SECTION MODULUS ..... , F10.5 ///)
2800  FORMAT(1H1,//10X,32H STRUCTURAL RESPONSES OF CULVERT ,
     X 3X, 12H LOAD STEP =  ,I2 //
     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 ,4X,9H M-STRESS,4X,9H T-STRESS,4X,9H S-STRESS ,
     X (//F13.2,5E13.3/F13.2,5E13.3))
2900  FORMAT(1H1,////10X,34H CALCULATED SAFETY FACTORS AT STEP, I3 ///
     X15X, 50H DISPLACEMENT ......... (DIAMETER*0.2/MAX-DISP.) =,F10.2//
     X15X, 50H OUTER-FIBER STRESS ... (ULTIMATE / MAXIMUM)      =,F10.2//
     X15X, 50H ELASTIC BUCKLING ....... (P-CRITICAL/P-AVERAGE) =,F10.2//
     X 10X, 21H PERFORMANCE FACTORS      //
     X15X, 50H HANDELING ......................... (FF*E*I/D**2) =,F10.2)
      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
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
*     Add RESULT array and NPPT variable to subroutine parameter list.
*         SCM   5-22-89
*
      SUBROUTINE PREP(FEDATA,PIPMAT,DENSTY,ISIZE,KPUTCH,LCODE,LUDATA,
     1                LUPLOT,MAXBD,MAXEL,MAXMAT,MAXNP,NPMAT,NPT,NSMAT,
     2                NXMAT,NBPTC,NELEM,PDIA,RESULT,NPPT)
*
*     Uncomment this statement in all program modules for double
*         precision version. SCM 3-10-90
*
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
*     Add RESULT array to dimension statement.
*         SCM  5-22-89
*
      DIMENSION FEDATA(ISIZE),PIPMAT(5,NPMAT),RESULT(20,NPPT),TITLE(17)
*
*     Add common statement for output control.  SCM 3-18-89
*
      COMMON /OUTPUT/ IWRT,IPLOT,MAXIT,JCON
*
      DATA LSTOP,NUMLI /7,200/
      DATA XPREP, XMOD /4HPREP, 4HMOD /
*
*     Add logic for evaluating LCODE parameter.  SCM 3-18-89
*
      IF(LCODE.GT.0) THEN
          LEVEL = LCODE
      ELSE
          LEVEL = IABS(LCODE/100)
          NCAN = IABS(LCODE + LEVEL*100)
      ENDIF
*
```

```
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,KPUTCH,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, IPLOT,
     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,IPLOT, 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
C**   SET UP STORAGE REQUIREMENTS FOR MESH GENERATOR
C
      NDNEL = MAX0(NPT,NELEM)
      LOCX = 1
      LOCY = LOCX + NPT
      LOCNGN  = LOCY + NPT
      LOCNTP = LOCNGN + 7*NDNEL
      LOCXYG = LOCNTP + NPT
C
C**   STORAGE FOR GENEL
C
      LOCNOD = LOCNGN
      LOCLGN = LOCXYG
C
C**   STORAGE FOR SAVED
C
      LOCNDB = LOCXYG + 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 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
   60 CALL GENNOD(FEDATA(LOCX),FEDATA(LOCY),FEDATA(LOCNGN),
     1              FEDATA(LOCNTP),FEDATA(LOCXYG),NPT,MAXNP,KPUTCK,
     2              MGENPR,LUPREP)
C
      CALL GENEL(FEDATA(LOCNOD),FEDATA(LOCLGN),NPT,NELEM,MAXEL,KPUTCK,
     1             MGENPR,MBAND,LSTOP,LUPREP)
C
      CALL GENEND(FEDATA(LOCX),FEDATA(LOCY),FEDATA(LOCNOD),
     1             FEDATA(LOCNDB),FEDATA(LOCBIV),FEDATA(LOCKDE),
     2             FEDATA(LOCNTP),FEDATA(LOCXYG),NPT,NELEM,NBPTCU,MAXBC,
     3             KPUTCK,MGENPR,NSMAT,LSTOP,NUML1,LUPREP,MAXMAT,NXMAT)
C
C**   ROUTINE MODMSH CHANGES THE CAN MESH TO USER SPECS.
C
      IF (WORD2 .EQ.XMOD) CALL MODMSH(FEDATA(LOCBIV),FEDATA(LOCNOD),
     1                       FEDATA(LOCNDB),FEDATA(LOCX),FEDATA(LOCY),
     2                       NBPTCU,MAXBC,NELEM,NPT)
C
C**   CHECK NUMBER OF BOUNDARY CONDITIONS
C
      IF (WORD2 .EQ. XMOD) NBPTC = NUMBC
      IF (NBPTCU .LE. NBPTC) GO TO 70
      WRITE(6,6018) NBPTCU, NBPTC
      KPUTCK = KPUTCK + 1
   70 NBPTC = NBPTCU
C
C**   SAVE GENERATED DATA
C
      WRITE (6,6020) TITLE, KPUTCK, NSMAT, NPMAT, NXMAT, MBAND
C
*     Add NPPT parameter to subroutine call.  SCM  3-10-90
*
      CALL SAVED(FEDATA(LOCX),FEDATA(LOCY),FEDATA(LOCNOD),FEDATA(LOCNDB)
     1             ,FEDATA(LOCBIV),FEDATA(LOCKDE),FEDATA(LOCXYG),TITLE,
     2            NPT,NELEM,IPLOT,NBPTC,NPUTCK,KPUTCK,LSTOP,NINC,
     3            LUDATA,LUPLOT,NSMAT,NPMAT,NPPT)
*
C
C**   ESTABLISH RESPONSE OUTPUT AND PLOTTING CONTROLS.
C
*
*     Revise ranges of output and plotting controls.  SCM 3-01-89
*
      IF(IPLOT.LT.0 .OR. IPLOT.GT.3) IPLOT = 1
      IF(IWRT.LT.0 .OR. IWRT.GT.3) IWRT = 1
*
      IOUT = -(LUPLOT*100 + IPLOT*10 + IWRT)
*
*     Add RESULT and NPPT parameters to subroutine call.
*          SCM  3-4-89
*
      CALL RESOUT (FEDATA(1),FEDATA(1),FEDATA(1),FEDATA(1),FEDATA(1),
     1               FEDATA(1),FEDATA(1),IOUT,NELEM,NPT,RESULT,NPPT)
*
C
      RETURN
 5010 FORMAT (A4,1X,17A4,3X,A4/8I5)
 6005 FORMAT ( 63H0FATAL DATA ERROR.   THE CONTROL WORD MUST BE *PREP *
     1, BUT IS *,A5, 1H*)
 6010 FORMAT (   43H1* * BEGIN PREP OF FINITE ELEMENT INPUT * */
     1     //31H0THE DATA TO BE RUN IS ENTITLED  //1H0,17A4///
     2          40H0NUMBER OF CONSTRUCTION INCREMENTS------,I5/
     3          40H0PRINT CONTROL FOR PREP OUTPUT----------,I5/
     4          40H0INPUT DATA CHECK-----------------------,I5/
```

```
     5          40HOPLOT TAPE GENERATION-------------------,I5/
     6          40HOENTIRE FINITE ELEMENT RESULTS OUTPUT---,I5/
     7          40HOTHE NUMBER OF NODES IS-----------------,I5/
     8          40HOTHE NUMBER OF ELEMENTS IS--------------,I5/
     9          40HOTHE NUMBER OF BOUNDARY CONDITIONS IS---,I5)
 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) ,I5,10X,      1H* ,//
     4         1H ,23H *REQUIRED SIZE (WORDS) ,I5, 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         40H * * * * NUMBER OF BOUNDARY CONDITIONS *  ,I5/
     2         40H * * * * MAXIMUM NUMBER ALLOWED * * * *   ,I5/
     3         40H * * * (SPECIFIED ON CARD C2. COL 36 -40   )
 6020 FORMAT ( 47H1* * * * * * FOR MESH GENERATION OF * * * * * */
     1         // 1H0,17A4 ///
     2          40HOTHE NUMBER OF DATA ERRORS IS-----------,I5/
     3          40HOTHE NUMBER OF SOIL MATERIALS IS--------,I5/
     4          40HOTHE NUMBER OF PIPE MATERIALS IS--------,I5/
     5           40HOTHE NUMBER OF INTERFACE MATERIALS IS---,I5/
     6          40HOBAND WIDTH ESTIMATE (MAX)--------------,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
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
*     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 SOULTIONS
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     Revise FEDATA and ISIZE parameters from 8000 to 10000.   by KS
CC
      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
*
      IF(LCODE.GT.0) THEN
          LEVEL = LCODE
      ELSE
          LEVEL = IABS(LCODE/100)
          NCAN = IABS(LCODE + LEVEL*100)
      ENDIF
*
C
      GO TO (100,110,140),ICOME
C
  100 CONTINUE
C
C**   PREPARE INPUT DATA FOR FINITE ELEMENT METHOD
C
      CALL ESTAB
      REWIND LUDATA
      REWIND LUPLOT
*
*     Add RESULT array and NPPT variable to subroutine parameter list.
*          SCM   5-22-89
*
      CALL PREP  (FEDATA,PIPMAT,DENSTY,ISIZE,KPUTCH,LCODE,LUDATA,
     1              LUPLOT,MAXBC,MAXEL,MAXMAT,MAXNP,NPMAT,NPT,NSMAT,
     2              NXMAT,NBPTC,NELEM,PDIA,RESULT,NPPT)
*
C
C**   DEFINE BLANK COMMON STORAGE FOR HEROIC
C
      LOCX = 1
      LOCY = LOCX + NPT
      LOCNOD = LOCY + NPT
      LOCMNO = LOCNOD + NELEM * 4
      LOCNOB = LOCMNO + NELEM
      LOCBIV = LOCNOB + NBPTC
      LOCNQ = LOCBIV + NBPTC * 3
      LOCU = LOCNQ + NPT + 1
      LOCV = LOCU + NPT
      LOCST = LOCV + NPT
      LOCSTP = LOCST + NELEM*6
      LOCSTH = LOCSTP + NPMAT*2 + 2
      ISZTST = LOCSTH + NELEM * 6
      IF (ISZTST .LE. ISIZE) GO TO 105
*
*     Reverse order of ISZTST and ISIZE in WRITE statement below.
*          SCM  10-23-89
*
      WRITE(6,6020) ISIZE, ISZTST
*
      STOP 11
C
C**   ZERO BLANK COMMON
C
  105 DO 120 NZERO = 1, ISIZE
  120 FEDATA(NZERO) = 0.0
  110 CONTINUE
C
C**   HEROIC PERFORMS 2 AND 3 SOLUTIONS   (THE FINITE ELEMENT METHOD)
C
      CALL HEROIC (FEDATA(LOCBIV),DENSTY,IA,ICOME,IEXIT,LEVEL,NBPTC,
     1                 FEDATA(LOCMNO),NELEM,NINC,FEDATA(LOCNOD),
     2                 FEDATA(LOCNOB),NPMAT,NPPT,NPT,FEDATA(LOCNQ),NSMAT,
     3                 NXMAT,PIPMAT,RESULT,SK,SM,FEDATA(LOCST),
     4                 FEDATA(LOCSTH),FEDATA(LOCSTP),FEDATA(LOCU),
```

```
     5            FEDATA(LOCV),FEDATA(LOCX),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(LOCU),FEDATA(LOCV),FEDATA(LOCX),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,I5,5HWORDS,
     4        1H ,32H * REQUIRED SIZE OF BLANK COMMON,I5,5HWORDS,//
     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
*               THE FEDERAL HIGHWAY ADMINISTRATION
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE READM( NSMAT,MAXMAT,DENSTY,LEVEL)
*
*     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),NON(10)
*
*     Expand array dimension statements from 30 to 99.
*          SCM  5-22-89
*
      COMMON/XFACE/ NXMAT,ANGLE(99),FCOEF(99),TENSIL(99),ITMAX
*
*
*     Add IBULK(10) to COMMON /HYPER/.  MGK 9-29-88
*
      COMMON /HYPER/ C(10),PHIO(10),DPHI(10),ZK(10),ZN(10),RF(10),
     $               VT(10),BK(10),BM(10),RATIO(10),IBULK(10)
*
      DIMENSION MATNAM(5)
*
*     Add dimension statement for Selig parameters. SCM 3-3-89
*
      DIMENSION ISELIG(15),SXC(15),SXPHIO(15),SXDPHI(15),SXZK(15),
     *          SXZN(15),SXRF(15),SXBI(15),SXEU(15)
*
      DIMENSION IDUNC(12),XC(12),XPHIO(12),XDPHI(12),XZK(12),
     *          XZN(12),XRF(12),XBK(12),XBM(12)
      DIMENSION IOVER(6),XE(6,8),XH(8)
      DATA ITRIA, IGRAN, ICOHE, IMIX /4HTRIA,4HGRAN,4HCOHE,4HMIXE /
      DATA  XNUMIN,XNUMAX,XQ   ,VOIDRG,VOIDRM, VOIDRC,SATM,SATC,PIM,PIC
     X   /  0.1,    0.49,  .26, 0.6, 0.5,    1.0,    0.5, .90,.05,0.20  /
      DATA IBLANK, L1H /1H , 1HL/
*
*     Add data for Selig model parameters.  SCM 3-3-89
*
      DATA PI /3.14159265/
      DATA ISELIG/4HSW95,4HSW90,4HSW85,4HSW80,4HSW61,
     *            4HML95,4HML90,4HML85,4HML80,4HML49,
     *            4HCL95,4HCL90,4HCL85,4HCL80,4HCL45/
      DATA SXC/0.0,0.0,0.0,0.0,0.0,4.0,3.5,3.0,2.5,0.0,
     *         9.0,7.0,6.0,5.0,0.0/
      DATA SXPHIO/48.0,42.0,38.0,36.0,29.0,34.0,32.0,30.0,
     *            28.0,23.0,15.0,17.0,18.0,19.0,23.0/
      DATA SXDPHI/8.0,4.0,2.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,
     *            4.0,7.0,8.0,8.5,11.0/
      DATA SXZK/950.0,640.0,450.0,320.0,54.0,440.0,200.0,110.0,
     *          75.0,16.0,120.0,75.0,50.0,35.0,16.0/
      DATA SXZN/0.60,0.43,0.35,0.35,0.85,0.40,0.26,0.25,
     *          0.25,0.95,0.45,0.54,0.60,0.66,0.95/
      DATA SXRF/0.70,0.75,0.80,0.83,0.90,0.95,0.89,0.85,
     *          0.80,0.55,1.00,0.94,0.90,0.87,0.75/
      DATA SXBI/187.0,102.0,31.8,15.3,4.3,120.8,46.0,23.8,
     *          12.8,3.3,53.0,25.5,13.0,8.8,1.8/
      DATA SXEU/0.014,0.036,0.057,0.078,0.163,0.043,0.071,0.100,
     *          0.134,0.305,0.092,0.121,0.149,0.178,0.391/
*
      DATA IDUNC/4HCA10,4HCA95,4HCA90,4HSM10,4HSM90,
     *           4HSM85,4HSC10,4HSC90,4HSC85,4HCL10,
```

```
     *          4HCL90,4HCL85/
      DATA XC/0.0,0.0,0.0,0.0,0.0,0.0,0.0,3.472,2.083,
     *        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,4HGFAI,4HMGOO,4HMFAI,4HCGOO,4HCFAI/
      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/
C
C     MATERIAL PROPERTIES FOR SOIL-SYSTEM (CONTINUUM ELEMENTS)
C
      WRITE(6,2000)
      KPUTCK = 0
      MAT = 0
      MATX = 0
      LIMIT = IBLANK
    2 IF (LIMIT .EQ. L1H) GO TO 500
      READ(5,5005) LIMIT,I,ITYP,DENX,(MATNAM(K),K=1,5)
      IF(ITYP.EQ.6) GO TO 200
      ITYPE(I) = ITYP
      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. ITYP.GT.5) GO TO 30
      GO TO 40
   30 WRITE(6,6070) ITYP,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
C
C               LINEAR ISOTROPIC PROPERTIES
C
   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)   )
      WRITE(6,904) E(I,1),GNU(I,1),SMH,SKH
      CALL CONVT(E(I,1),GNU(I,1))
      GO TO 65
C
C               ANISOTROPIC PROPERTIES
C
   60 READ(5,5010) D2,D3,D4,D5,D6
      WRITE(6,905) D2,D3,D4,D5,D6
      CALL ANISP(D2,D3,D4,D5,D6)
C
   65 L=0
      DO 70 J=1,3
      DO 70 K=J,3
      L=L+1
   70 H(I,L)=CP(J,K)
      GO TO 2
C
C               DUNCAN'S HYPERBOLIC SOIL MODEL
C
*
*     Add Selig IBULK(I) variable to read statement.
*          MGK  9-29-1988
*
   80 READ(5,300) NON(I),RATIO(I),IBULK(I)
*
*     Revise soil model iteration limit.  SCM
*
      IF(NON(I).EQ.0.0)NON(I) = 20
*
      IF(RATIO(I).EQ.0.0) RATIO(I) = 0.5
*
*     Add control statements to print out heading for
*          appropriate model (Duncan or Selig).
*          SCM 10-28-88
*
      IF (IBULK(I).EQ.0) THEN
           WRITE(6,305) NON(I),RATIO(I)
           ELSE
           WRITE(6,306) NON(I),RATIO(I)
      ENDIF
*
C
C     DETERMINE IF MATNAM IS REQUESTING STANDARD PROPERTIES,
C     OTHERWISE ASSUME MATNAM=USER
C
      J=0
*
*     Add logic to select Selig standard parameters. SCM 3-3-89
*
      IF (IBULK(I).EQ.0) THEN
*
      DO 81 N=1,12
      IF(MATNAM(1).EQ.IDUNC(N)) J=N
   81 CONTINUE
*
      ELSE
      DO 82 N=1,15
      IF(MATNAM(1).EQ.ISELIG(N)) JS=N
   82 CONTINUE
      END IF
      IF(J.EQ.0 .AND. JS.EQ.0) GO TO 85
      IF(IBULK(I).NE.0) GO TO 83
*
```

```
C
C     ESTABLISH STANDARD PARAMETER VALUES
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)=SXBI(JS)
      BM(I)=SXBU(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
*
*     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)
      IF (VT(I) .EQ. 0.0 .AND. IBULK(I) .NE. 0) WRITE(6,323) BK(I),BM(I)
*
      IF(VT(I).GT.0.0) WRITE(6,325) VT(I)
      GO TO 2
C
C                 OVERBURDEN PRESSURE DEPENDENT PROPERTIES
C                 (FIRST DETERMINE IF STANDARD PROPERTIES ARE
C                 REQUESTED)
C
   90 J=0
      DO 91 N=1,6
      IF(MATNAM(1).EQ.IOVER(N)) J=N
   91 CONTINUE
      IF(J.EQ.0) GO TO 93
C
C     ESTABLISH STANDARD PROPERTY TABLES
C
      NOMI=8
      DO 92 N=1,NOMI
      H(I,N)=XH(N)
      E(I,N)=XE(J,N)
   92 GNU(I,N)=0.333
      GO TO 98
C
C     USER DEFINED OVERBURDEN TABLE
C
   93 N = 0
   94 N = N+1
      READ(5,5010) D1,D2,D3
      IF(D1+D2 .EQ. 0.0) GO TO 95
      H(I,N) = D1
      E(I,N) = D2
      GNU(I,N) = D3
      GO TO 94
   95 NOMI = N-1
      IF(NOMI.GE.2 .AND. NOMI.LE.10) GO TO 98
      WRITE(6,6060) NOMI
      KPUTCK = KPUTCK + 1
   98 WRITE(6,923)
      WRITE(6,916) (H(I,N),E(I,N),GNU(I,N),N=1,NOMI)
      NON(I) = NOMI
      GO TO 2
C
C ... PROCESS EXTENDED-HARDIN MODEL, DEPENDENT ON SOIL CLASS...
C     GRANULAR, MIXED, COHESIVE, OR USER DEFINED FROM TRIAXIAL TEST.
C
100   READ(5,5010) D0, D1, D2, D3, D4, D5, NON(I)
C
C     CHECK ON PERMISSIBLE VALUES OF MAX AND MIN POISSON RATIOS FOR
C     EXTENDED HARDIN MODEL. THIS APPLYS TO ALL SOIL CLASSES AND USER.
C
      IF(NON(I).LE.0) NON(I) = 5
      IF(D0.LE.0.0 .OR. D0.GE.0.5) D0 = XNUMIN
      IF(D1.LE.0.0 .OR. D1.GE.0.5) D1 = XNUMAX
      IF(D2.LE.0.0) D2 = XQ
      H(I,4) = D0
      H(I,5) = D1
      H(I,6) = D2
C
      IF(MATNAM(1) .EQ. ITRIA) GO TO 150
C
C     SET DEFAULT VALUES FOR GRANULAR SOIL CLASS
C
```

```
      IF(MATNAM(1) .NE. IGRAN) GO TO 110
      IF(D3.LT. 0.1 .OR. D3 .GT. 1.0) D3 = VOIDRG
      D4 = 0.0
      D5 = 0.0
      A = 3.20
      GO TO 130
C
C     SET DEFAULT VALUES FOR COHESIVE SOIL CLASS
C
110   IF(MATNAM(1).NE. ICOHE) 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)
C
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
C     WRITE OUT INPUT/DEFAULT DATA FOR SOIL CLASSES.
C
130   WRITE(6,6040) MATNAM(1),MATNAM(2),D0,D1,D2,D3,D4,D5,NON(I)
C
C     COMPUTE MATERIAL CONSTANTS IN HARDINS EQUATIONS.(NOTE THE VARIABLE
C     A WAS BASED ON HARDIN EQUATIONS WITH T = 10 AND N=1.
C
      F = (2.973-D3)**2/(1.+D3)
      R = (1100.-600.*D4)
      H(I,1) = 1230.*F
      H(I,2) =     F*F*R*B/(0.6 - 0.25*D5**0.6)
      H(I,3) = A
C
      GO TO 2
C
C     USER DEFINED PROPERTIES FOR HARDINS RELATIONS FROM TRIAXIAL TESTS.
C
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(I,2) = D4
      H(I,3) = D5
      GO TO 2
C
C     INPUT PROPERTIES FOR INTERFACE ELEMENTS
C
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)
*
*     Revise default value for TENSIL.  SCM 4-25-89
*
      IF(TENSIL(I).EQ.0.0) TENSIL(I) = 1.0
*
      WRITE(6,955) I,ANGLE(I),FCOEF(I),TENSIL(I)
*
*     Revise logic to set ITMAX to highest number specified
*          for interface elements.    SCM  3-10-89
*
      IF(ABS(DENX).GT.ITMAX) ITMAX = ABS(DENX)
*
*     Disable maximum iteration limit.  SCM 3-3-89
*     Revise minimum iteration limit to 20.   SCM  3-18-89
*
*     IF (ITMAX.GT.10) ITMAX=10
*
*      IF (ITMAX.EQ. 0) ITMAX=5
*
      IF (ITMAX.EQ.0) ITMAX=20
*
      GO TO 2
C
  500 CONTINUE
      IF(MAT.GT.NSMAT) WRITE(6,6080)MAT,NSMAT
      IF(MATX.GT.NXMAT) WRITE(6,6080) MATX,NXMAT
      IF(MAT.GE.NSMAT .AND. MATX.GE.NXMAT) GO TO 550
      WRITE(6,6090) NSMAT
      WRITE(6,6090) NXMAT
      KPUTCK = KPUTCK + 1
550   IF(LEVEL.NE.2) GO TO 600
      ICKFIL = 3
      DENSTY = DENSTY/1728.
      IFILL = 2*(NSMAT/ICKFIL) + 1
      IF(DENSTY .NE. DEN(IFILL))WRITE(6,6095)DENSTY,DEN(IFILL)
600   IF(KPUTCK.EQ.0) RETURN
      WRITE(6,7000) KPUTCK
      STOP
C
C**   FORMATS
C
2000  FORMAT(1H1,//10X,37H MATERIAL CHARACTERIZATION FOR SOILS.   )
 5005 FORMAT(A1,I4,I5,F10.0,5A4)
 5010 FORMAT(6F10.0,I5)
6040  FORMAT(// 5X, 42H HARDIN MATERIAL, STANDARD SOIL PROPERTIES//
     X 10X, 27H SOIL CLASSIFICATION ...... ,2A4 /
     X 10X, 27H MINIMUN POISSON RATIO .... ,F7.3 /
     X 10X, 27H MAXIMUM POISSON RATIO .... ,F7.3 /
     X 10X, 27H POISSON PARAMETER Q ...... ,F7.3 /
     X 10X, 27H VOID RATIO ............... ,F7.3 /
     X 10X, 27H PERCENT OF SATURATION .... ,F7.3 /
     X 10X, 27H PLASTICITY INDEX ......... ,F7.3 /
     X 10X, 27H MAX ITERATIONS PER STEP ..,I5)
6050  FORMAT(//5X,38H HARDIN MATERIAL, TRIAXIAL  DATA INPUT  //
     X 10X, 31H MINIMUM POISSON RATIO .........,F10.3 /
     X 10X, 31H MAXIMUM POISSON RATIO .........,F10.3 /
     X 10X, 31H POISSON RATIO PARAMETER Q ...., F10.3 /
     X 10X, 31H HARDIN PARAMETER, S1 ..........,F10.3/
     X 10X, 31H HARDIN PARAMETER, C1 ..........,E10.3/
     X 10X, 31H HARDIN PARAMETER, A  ..........,F10.3/
     X 10X, 31H MAX ITERATIONS PER STEP .......,I5)
6055  FORMAT(///25H *** FATAL DATA ERROR *** //
     X59H SOME OF THE ABOVE TRIAXIAL DATA IS  LESS OR EQUAL TO ZERO,/
     X42H OR SECANT MODULUS IS GREATER THAN GMAX.   )
```

```
 6060 FORMAT(/ 79H FATAL DATA ERROR. NUMBER OF NONLINEAR DATA POINTS NOTREDM0446
     X BETWEEN 2 AND 10, NOMI = ,I5)                                    REDM0447
 6070 FORMAT(/73H FATAL DATA ERROR. MATERIAL NUMBER AN/OR ITYP ARE OUT-OREDM0448
     XF-BOUNDS. I,ITYPE=,2I5)                                           REDM0449
6080  FORMAT(///46H WARNING MORE MATERIAS 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 DENSTY WITHIN MESH,  ,F10.5 )       REDM0453
7000  FORMAT(/// 48H ***FATAL DATA ERRORS IN MATERIAL INPUT, ERRORS=,I3)REDM0454
  907 FORMAT(1H0//5X,23HPROPERTIES 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,1P3E20.2)                                           REDM0464
  923 FORMAT(1H0,9X,31HELASTIC PROPERTIES - OVERBURDEN//20X,            REDM0465
     1         10HOVERBURDEN,10X,14HYOUNGS MODULUS,6X,14HPOISSONS RATIO) REDM0466
950   FORMAT(//// 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(I5,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
     $ 10X,30H 10-FOLD REDUCTION IN PHIO... ,F12.4//                    REDM0495
     4 10X,30H SCALED MODULUS NUMBER  ZK .. ,F12.4//                    REDM0496
     5 10X,30H MODULUS EXPONENT  ZN ....... ,F12.4//                    REDM0497
     6 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/B(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
*                                                                       RESO0007
*     Add RESULT and NPPT parameters to subroutine statement.           RESO0008
*          SCM 3-01-89                                                  RESO0009
*                                                                       RESO0010
      SUBROUTINE RESOUT(MNO,NOD,ST,U,V,X,Y,IA,NELEM,NPT,RESULT,NPPT)    RESO0011
*                                                                       RESO0012
*     Uncomment this statement in all program modules for double        RESO0013
*          precision version. SCM 3-10-90                               RESO0014
*                                                                       RESO0015
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               RESO0016
*                                                                       RESO0017
*     Add character variable.  SCM 3-3-89                               RESO0018
*                                                                       RESO0019
      CHARACTER RECORD*133                                              RESO0020
*                                                                       RESO0021
*     Add RESULT variable to dimension statement.  SCM 3-4-89           RESO0022
*                                                                       RESO0023
      DIMENSION MNO(NELEM),NOD(4,NELEM),ST(6,NELEM),U(NPT),V(NPT),      RESO0024
     1          X(NPT),Y(NPT),STATE(4),RESULT(20,NPPT)                  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,RO
      COMMON /SIZEPR/ BIVD(3),IX,NINC
*
*     Expand array dimension statements from 30 to 99.
*          SCM  5-22-89
*
      COMMON/XFACE/NXMAT,ANGLE(99),FCOEF(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
C***********WRITE STATEMENTS ELIMINATED, JUNE 78***********************
C
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*********************
*
*     WRITE(LUPLOT) IA,(U(N),V(N),N=1,NPT)
*     WRITE(LUPLOT) IA,((ST(J,N),J=1,6),N=1,NELEM)
*
      WRITE(LUPLOT,1000) IA
      DO 60 N=1,NPT
      WRITE(LUPLOT,1010) U(N),V(N)
60    CONTINUE
      DO 70 N=1,NELEM
      WRITE(LUPLOT,1020) (ST(J,N), J=1,6)
70    CONTINUE
      IF(IPLOT.LT.2) GO TO 100
80    WRITE(30,1000) IA
      DO 90, N=1,NPPT
          WRITE(30,1110) (RESULT(J,N), J=1,9)
90        CONTINUE
*
C
C*********************************************************************
C
C
      IF(IA.EQ.NINC) END FILE LUPLOT
100   IF(IWRT.LE.1) GO TO 120
*
*     Revise control logic to add printing of Duncan data.
*          SCM 3-1-89
*
      IF(IWRT.LE.1) GO TO 120
      REWIND 16
      NREC = 0
      MSUMR = 0
105   READ(16,'(I1)',ERR=500,END=107) IPAGE
      NREC = NREC + 1
      IF(IPAGE.EQ.1) THEN
          READ(16,6000)
          READ(16,6000) IAX,ITERX
              IF (IAX.EQ.IA) THEN
                  MSUMR = MSUMR + NREC + 2
                  NREC = 0
              ELSE
                  NREC = NREC + 2
              END IF
      END IF
      GO TO 105
107   REWIND 16
      IF(MSUMR.EQ.0) GO TO 120
      DO 110, NR=1,MSUMR+9
          READ(16,*,ERR=500,END=110)
110       CONTINUE
      MCT = 40
      DO 118, NN=1,NREC-9
          IF(MCT.LT.40) GO TO 115
          MCT = 0
          WRITE(6,6010) IAX,ITERX
          WRITE(6,6020)
115       MCT = MCT + 1
          READ(16,'(A133)',ERR=500,END=118) RECORD
          WRITE(6,'(A133)') RECORD
118       CONTINUE
*
120   CONTINUE
C
C**   PRINT OUT DISPLACEMENTS
C
```

```
      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.IA) GO TO 300
      IF (NOD(2,I) .NE. NOD(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.IA) 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)
      IAC = MOD(N,1000)/10
      IF(IAC.GT.IA) GO TO 400
      IF (NOD(2,I) .EQ. NOD(3,I)) GO TO 400
      N = N/1000
      MNX = MNO(I)-N*1000-10*IAC
      IF (MNX .NE. 0) GO TO 400
      IF(MCT.LT.40) GO TO 325
       MCT = 0
      WRITE(6,2000) IA
      WRITE(6,4100)
325   MCT = MCT + 1
      N1 = NOD(1,I)
      N2 = NOD(2,I)
      N3 = NOD(3,I)
      N4 = NOD(4,I)
      XC = 0.25*(X(N1)+X(N2)+X(N3)+X(N4))
      YC = 0.25*(Y(N1)+Y(N2)+Y(N3)+Y(N4))
      DO 350 K=1,3
  350 RQ(K) = ST(K+3,I)
C
C     PRINCIPAL STRESSES CALCULATED
C
      CALL PRINC(RQ,ZQ)
      WRITE(6,4200) I,XC,YC,(ST(K,I),K=1,6),(ZQ(K),K=1,3)
400   CONTINUE
      RETURN
*
*     Add ERROR STOP for I/O errors on Unit 16.  SCM  3-1-89
*
500   WRITE(6,6030)
      STOP 15
*
C
C**   FORMATS
C
*
*     Revise formats for plot and print output.  SCM  11-24-88
*
1000  FORMAT(I5)
1010  FORMAT(E12.4,1H,,E12.4)
1020  FORMAT(E12.4,5(1H,,E12.4))
1110  FORMAT(E12.4,8(1H,,E12.4))
*
2000  FORMAT(1H1,/// 5X,23H CONSTRUCTION INCREMENT, I4 ///)
2100  FORMAT(//5X,5H NODE,7X,8HX-COORD.,7X,8HY-COORD.,8X,7HX-DISP.,
     X 8X,7HY-DISP. )
2200  FORMAT(5X,I3,2X,4(2X,E13.4))
3100  FORMAT(5X, //36H BENDING ELEMENT RESULTANT FORCES .. //5X,5HELMNT,
     X 6X,8HX-CENTER,6X,8HY-CENTER,8X,6HTHRUST,9X,5HSHEAR,8X,6HMOMENT/)
3200  FORMAT(I10,5(1X,E13.4))
3500  FORMAT(// 5X,43H INTERFACE ELEMENT FORCES AND DISPLACEMENTS  //
     X 5X,5HELMNT,5H MAT.,3X,5HSTATE,7X,7HN-FORCE,7X,7HS-FORCE,2X,
     X 3X, 9HN-DISP(I),5X,9HS-DISP(I),5X,9HN-DISP(J),5X,9HS-DISP(J))
3600  FORMAT(5X,2I5,4X,A4,6(1X,E13.4))
4100  FORMAT(5X, 7HELEMENT, 45X,
     1               28HELEMENT STRAINS AND STRESSES / ,6X,3HNO.,8X
     1  ,1HX,10X,1HY,6X,9HEPSILON-X, 2X, 9HEPSILON-Y, 3X, 7HGAMA-XY, 4X,
     2 7HSIGMA-X, 4X, 7HSIGMA-Y,4X,6HTAU-XY,5X,7HSIGMA-1, 4X, 7HSIGMA-2,
     3   6X, 3HANG, /)
4200  FORMAT(I7 ,4X,1P11E11.2)
*
*     Add formats for Duncan output.  SCM 3-1-89
*
```

```
6000  FORMAT(29X,I2,15X,I2)                                               RESO0279
6010  FORMAT(1H1,//,6X,23HCONSTRUCTION INCREMENT ,I2,15X,                 RESO0280
     X31H(CONVERGENCE ACHIEVED AFTER ***,I4,29H  *** ITERATIONS IF ICON RESO0281
     X= 1))                                                              RESO0282
6020  FORMAT(1H0,15X,47H* * * ELASTIC MODULI AND CONVERGENCE PARAMETERS,RESO0283
     X28H FOR DUNCAN SOIL MODEL * * *,/                                  RESO0284
     X22X,69H(CONVERGENCE ERRORS (ERROR-EMOD) ARE RATIOS GIVEN BY, (NEW-RESO0285
     XOLD)/NEW ),/                                                       RESO0286
     X22X,42HK-TYPE = 1 ; DUNCAN POWER LAW BULK MODULUS,                 RESO0287
     X10X,45HSTRESS-LEV <= -1.0 ; COMPLETE TENSION FAILURE,/             RESO0288
     X22X,42HK-TYPE = 2 ; SELIG HYPERBOLIC BULK MODULUS,                 RESO0289
     X10X,34HSTRESS-LEV >= +1.0 ; SHEAR FAILURE,/                        RESO0290
     X22X,37HK-TYPE = 3 ; CONSTANT POISSON'S RATIO,//                    RESO0291
     X3X,5HELE-N,4X,5HMAT-N,4X,6HK-TYPE,4X,4HICON,4X,                    RESO0292
     X10HERROR-EMOD,4X,10HSTRESS-LEV,4X,10HYOUNGS-MOD,6X,8HBULK-MOD,     RESO0293
     X4X,10HPOIS-RATIO,4X,12HCONFINED-MOD,//)                            RESO0294
6030  FORMAT(1H0,21H***FATAL I/O ERROR***/1H ,55HDUNCAN SUBROUTINE OUTPURESO0295
     XT FILE NOT PRESENT OR UNREADABLE/)                                 RESO0296
*                                                                        RESO0297
      END                                                                RESO0298
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  RESP0001
*     CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM                 RESP0002
*                   VERSION 1.0, MARCH 1990                              RESP0003
*             PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY                      RESP0004
*               THE FEDERAL HIGHWAY ADMINISTRATION                       RESP0005
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  RESP0006
      SUBROUTINE RESPIP (MNO,NOD,RESULT,ST,STPIPE,U,V,X,Y,               RESP0007
     1                  NELEM,NPMAT,NPPT,NPT,IEXIT,ICOME)                RESP0008
C                                                                        RESP0009
C     THIS SUBROUTINE ORDERS PIPE ELEMENTS AND PUTS PIPE RESPONSES       RESP0010
C     INTO MATRIX RESULTS.                                               RESP0011
C                                                                        RESP0012
C         RESULT(1,N) = X-COORDINATE ON PIPE (X=Y=0, AT PIPE CENTER, FOR RESP0013
C         RESULT(2,N) = Y-COORDINATE ON PIPE           LEVEL = 2 )       RESP0014
C         RESULT(2,N) = Y-COORDINATE ON PIPE       LEVEL = 2)            RESP0015
C         RESULT(3,N) = X-DISPLACEMENT                                   RESP0016
C         RESULT(4,N) = Y-DISPLACEMENT                                   RESP0017
C         RESULT(5,N) = BENDING MOMENT (POSITIVE = INNER FIBER TENSION)  RESP0018
C         RESULT(6,N) = THRUST FORCE (COMPRESSION = NEGATIVE)            RESP0019
C         RESULT(7,N) = SHEAR FORCE                                      RESP0020
C         RESULT(8,N) = NORMAL PRESSURE ON PIPE (POSITIVE OUTWARD NORMAL)RESP0021
C         RESULT(9,N) = TANGENTIAL PRESSURE ON PIPE (POSITIVE CLOCKWISE) RESP0022
C                                                                        RESP0023
C           THE REMAINING RESULT ARRAY IS DEPENDENT ON PIPE TYPE ..      RESP0024
C                                                                        RESP0025
*     Uncomment this statement in all program modules for double         RESP0026
*         precision version. SCM 3-10-90                                 RESP0027
*                                                                        RESP0028
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                RESP0029
*                                                                        RESP0030
      DIMENSION MNO(NELEM),NOD(4,NELEM),RESULT(20,NPPT)                  RESP0031
     1          ,ST(6,NELEM),STPIPE(2,NPPT ),U(NPT),V(NPT),X(NPT),Y(NPT)RESP0032
*                                                                        RESP0033
*     Revise dimensions of XMONT and XTHRU arrays from 30 to 99.         RESP0034
*         SCM  4-30-89                                                   RESP0035
*                                                                        RESP0036
      DIMENSION XMONT(99),XTHRU(99)                                      RESP0037
*                                                                        RESP0038
      COMMON /SIZEPR/ BIVD(3),IX,NINC                                    RESP0039
      COMMON /MATERL/ CP(3,3),DEN(10),E(10,10),GNU(10,10),H(10,10),      RESP0040
     1                ITYPE(10),NON(10)                                  RESP0041
*                                                                        RESP0042
*     Revise dimensions of NELPIP and NELIST arrays from 30 to 99.       RESP0043
*         SCM  4-30-89                                                   RESP0044
*                                                                        RESP0045
      DIMENSION NELPIP(99),NELIST(99)                                    RESP0046
*                                                                        RESP0047
C                                                                        RESP0048
      GO TO (99,299) , ICOME                                             RESP0049
C                                                                        RESP0050
C     FIRST ENTRY                                                        RESP0051
C     COLLECT CULVERT ELEMENTS AND FIND TOP ELEMENT                      RESP0052
C                                                                        RESP0053
99    CONTINUE                                                           RESP0054
      YMAX = -1.E+10                                                     RESP0055
      NEP = 0                                                            RESP0056
      DO 100 I=1,NELEM                                                   RESP0057
      IF (NOD(2,I) .NE. NOD(3,I)) GO TO 100                              RESP0058
      NEP = NEP + 1                                                      RESP0059
      NELPIP(NEP) = I                                                    RESP0060
      ND1 = NOD(1,I)                                                     RESP0061
      ND2 = NOD(2,I)                                                     RESP0062
*                                                                        RESP0063
*     Revise selection procedure to choose Material Number 1 as          RESP0064
*     first beam element in top, right position from centerline.         RESP0065
*         SCM  2-25-90                                                   RESP0066
*                                                                        RESP0067
*     IF(Y(ND1)+Y(ND2) .LE. YMAX) GO TO 100                              RESP0068
*     YMAX = Y(ND1) + Y(ND2)                                             RESP0069
*                                                                        RESP0070
      IF(Y(ND2).LE.YMAX) GO TO 100                                       RESP0071
      YMAX = Y(ND2)                                                      RESP0072
*                                                                        RESP0073
      NETOP = I                                                          RESP0074
100   CONTINUE                                                           RESP0075
      IF(NEP.EQ.0) RETURN                                                RESP0076
C                                                                        RESP0077
C     FIND BEAM ELEMENTS CONNECTED CLOCKWISE FROM TOP ELEMENT.           RESP0078
C                                                                        RESP0079
      KK = 1                                                             RESP0080
      NEL = NETOP                                                        RESP0081
150   NELIST(KK) = NEL                                                   RESP0082
      IF(KK.EQ. NEP) GO TO 250                                           RESP0083
      NA1 = NOD(1,NEL)                                                   RESP0084
      DO 200 I=1,NEP                                                     RESP0085
      NEL = NELPIP(I)                                                    RESP0086
      NB2 = NOD(2,NEL)                                                   RESP0087
      IF(NA1.NE.NB2) GO TO 200                                           RESP0088
      KK = KK +1                                                         RESP0089
      GO TO 150                                                          RESP0090
200   CONTINUE                                                           RESP0091
C                                                                        RESP0092
C     FIND BEAM ELEMENTS CONNECTED COUNTRE-CLOCKWISE FROM TOP ELEMENT.   RESP0093
C                                                                        RESP0094
      DO 220 I=1,KK                                                      RESP0095
220   NELIST(NEP-I+1) = NELIST(KK-I+1)                                   RESP0096
C                                                                        RESP0097
      NEL = NETOP                                                        RESP0098
225   NELIST(NEP-KK+1) = NEL                                             RESP0099
      IF(KK.EQ.NEP) GO TO 250                                            RESP0100
      NA2 = NOD(2,NEL)                                                   RESP0101
      DO 230 I=1,NEP                                                     RESP0102
      NEL = NELPIP(I)                                                    RESP0103
      NB1 = NOD(1,NEL)                                                   RESP0104
      IF(NA2.NE.NB1) GO TO 230                                           RESP0105
      KK = KK + 1                                                        RESP0106
```

```
      GO TO 225
230   CONTINUE
C
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. I) 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
      WRITE(6,2000)   (NELIST(I),I=1,NPMAT)
C
      RETURN
C
C     SECOND ENTRY
C     PUT STRUCTURAL RESPONSES INTO RESULT VECTOR.
C
299   CONTINUE
      IF(NEP.EQ.0) RETURN
      DO 300 I=1,NPMAT
      NE = NELIST(I)
      ND1 = NOD(1,NE)
      ND2 = NOD(2,NE)
      RESULT(1,I) = X(ND2)
      RESULT(2,I) = Y(ND2)
      RESULT(3,I) = U(ND2)
300   RESULT(4,I) = V(ND2)
C
      IF(ICURVE.EQ.0) GO TO 310
      RESULT(1,NPPT) = X(ND1)
      RESULT(2,NPPT) = Y(ND1)
      RESULT(3,NPPT) = U(ND1)
      RESULT(4,NPPT) = V(ND1)
310   CONTINUE
C
C**   SAVE THE PREVIOUS CONVERGED SOLUTION FOR MOMENT AND THRUST
C
      IF(IEXIT.EQ.-1) GO TO 360
      DO 350 NUD = 1, NPPT
      XMONT(NUD) = RESULT(5,NUD)
  350 XTHRU(NUD) = RESULT(6,NUD)
  360 CONTINUE
C
C     DISTRIBUTE ELEMENT RESPONSES TO NODES
C
      I = 1
      I1 = I+1
400   NE1 = NELIST(I)
      NE2 = NELIST(I1)
      ND1 = NOD(2,NE1)
      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,NE1) + ST(6,NE2))/2.0
      RESULT(6,I1) = (ST(1,NE1) + ST(4,NE2))/2.0
      RESULT(7,I1) = (ST(2,NE1) + 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. I) 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)
```

```
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,NPMAT) /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,58H ORDERING IS CLOCKWISE FROM TOP TO BOTTOM OR LEFT TO RIGHT
     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 PROPER
     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,/
     2            41H NUMBER OF ACTUAL CONNECTED BEAM ELEMENTS, I5)
4200  FORMAT (/// 40H * * * * * FATAL DATA ERROR * * * * *      ,/
     1         67H UNCONNECTED BEAM MATERIAL NUMBER SMALLER THAN NPMA
     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,CODE,TITLE,NPT,NELEM,
     1                 IPLOT,NBPTC,NPUTCK,KPUTCK,LSTOP,NINC,LUDATA,
     2                 LUPLOT,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)
*
      DIMENSION X(1), Y(1), NOD(LSTOP,1), NODB(1), BIV(3,1), KODE(1),
     1          CODE(1), TITLE(17)
C     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
      NODE = NODB(N)/10000
      IF (MOD(NODB(N),100) .GT. 1) CODE(NODE) = 1.0
      IF (MOD(NODB(N),10) .EQ. 1)  CODE(NODE) = 2.0
      IF (MOD(NODB(N),100) .EQ. 11) CODE(NODE) = 3.0
   30 CONTINUE
      DO 40 N = 1, NELEM
      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
   40 CONTINUE
C
C**   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 LUPLOT from unformated to formatted.
*          SCM 11-24-88
***********WRITE STATEMENTS PERMANENTLY INCORPORATED (JUNE 78)**********
*     WRITE(LUPLOT)(TITLE(K),K=1,17),NPT,NELEM,NUMMAT,NPUTCK,NINC
*     WRITE(LUPLOT)(CODE(N),X(N),Y(N),N=1,NPT)
*     WRITE(LUPLOT)((NOD(K,N),K=1,4),KODE(N),NOD(6,N),N=1,NELEM)
*
      WRITE(LUPLOT,6010) (TITLE(K),K=1,17)
      WRITE(LUPLOT,6015) NPT,NELEM,NUMMAT,NPUTCK,NBPTC,NINC
      DO 100 N=1,NPT
      WRITE(LUPLOT,6020) N,X(N),Y(N)
  100 CONTINUE
      DO 110 N=1,NELEM
      WRITE(LUPLOT,6025) N,(NOD(K,N),K=1,4),KODE(N),NOD(6,N)
  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
C
C
C**********WRITE STATEMENTS PERMANENTLY ELIMINATED (JUNE 78)************
C     WRITE (LUPLOT) (TITLE(K), K = 1,8), NPT, NELEM, NUMMAT, NPUTCK,
C    1               FZERO, NINC, IZERO, FZERO, FZERO, TPSCAN, TPSCAN,
C    2               (IZERO,K=1,12)
C     WRITE (LUPLOT) (CODE(N), X(N), Y(N), N = 1,NPT),
C    1               ((NOD(K,N),K=1,4),KODE(N),N=1,NELEM),(IZERO,K=1,4)
C
```

```
C*************************************************************************
C
   50 CONTINUE
      DO 45 N = 1, NELEM
      NOD(6,N) = 1000*NOD(5,N) +10*NOD(6,N) +NOD(7,N)
   45 CONTINUE
      IF(IPLOT.LT.0) END FILE LUPLOT
C
*
*     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)
*
C
      WRITE (6,6005)
      RETURN
 6005 FORMAT ( 53H0* * * * * * * MESH DATA HAS BEEN SAVED * * * * * * *)
 6010 FORMAT(1H',17A4,1H')
 6015 FORMAT(I5,5(1H,,I5))
 6020 FORMAT(I5,2(1H,,E12.4))
 6025 FORMAT(I5,6(1H,,I5))
 6030 FORMAT(I5,2(1H,,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,CODE,TITLE,NPT,NELEM,
     1                  IPLOT,NBPTC,NPUTCK,KPUTCK,LSTOP,NINC,LUDATA,
     2                  LUPLOT,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)
*
      DIMENSION X(1), Y(1), NOD(LSTOP,1), NODB(1), BIV(3,1), KODE(1),
     1          CODE(1), TITLE(17)
C     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
      NODE = NODB(N)/10000
      IF (MOD(NODB(N),100) .GT. 1) CODE(NODE) = 1.0
      IF (MOD(NODB(N),10) .EQ. 1)  CODE(NODE) = 2.0
      IF (MOD(NODB(N),100).EQ. 11) CODE(NODE) = 3.0
   30 CONTINUE
      DO 40 N = 1, NELEM
      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
   40 CONTINUE
C
C**   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 LUPLOT from unformated to formatted.
*          SCM 11-24-88
***********WRITE STATEMENTS PERMANENTLY INCORPORATED (JUNE 78)*********
*     WRITE(LUPLOT)(TITLE(K),K=1,17),NPT,NELEM,NUMMAT,NPUTCK,NINC
*     WRITE(LUPLOT)(CODE(N),X(N),Y(N),N=1,NPT)
*     WRITE(LUPLOT)((NOD(K,N),K=1,4),KODE(N),NOD(6,N),N=1,NELEM)
*
      WRITE(LUPLOT,6010) (TITLE(K),K=1,17)
      WRITE(LUPLOT,6015) NPT,NELEM,NUMMAT,NPUTCK,NBPTC,NINC
      DO 100 N=1,NPT
      WRITE(LUPLOT,6020) N,X(N),Y(N)
  100 CONTINUE
      DO 110 N=1,NELEM
      WRITE(LUPLOT,6025) N,(NOD(K,N),K=1,4),KODE(N),NOD(6,N)
  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
C
C
C**********WRITE STATEMENTS PERMANENTLY ELIMINATED (JUNE 78)************
C     WRITE (LUPLOT) (TITLE(K), K = 1,8), NPT, NELEM, NUMMAT, NPUTCK,
C    1              FZERO, NINC, IZERO, FZERO, FZERO, TPSCAN, TPSCAN,
C    2              (IZERO,K=1,12)
C     WRITE (LUPLOT) (CODE(N), X(N), Y(N), N = 1,NPT),
C    1              ((NOD(K,N),K=1,4),KODE(N),N=1,NELEM),(IZERO,K=1,4)
C
```


```
C*************************************************************************
C
   50 CONTINUE
      DO 45 N = 1, NELEM
      NOD(6,N) = 1000*NOD(5,N) +10*NOD(6,N) +NOD(7,N)
   45 CONTINUE
      IF(IPLOT.LT.0) END FILE LUPLOT
C
*
*     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)
*
C
      WRITE (6,6005)
      RETURN
 6005 FORMAT ( 53H0* * * * * * * MESH DATA HAS BEEN SAVED * * * * * * *)
 6010 FORMAT(1H',17A4,1H')
```

```
 6015 FORMAT(I5,5(1H,,I5))
 6020 FORMAT(I5,2(1H,,E12.4))
 6025 FORMAT(I5,6(1H,,I5))
 6030 FORMAT(I5,2(1H,,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 SAVEG (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 ( 74H1LISTING OF NODES (EXCEPT LAPLACE GENERATED NODES) AS
     1GENERATED FROM INPUT/       65H0NODE        GENERATION
     2 X-COORDINATE     Y-COORDINATE/31H NUMBER         MODE            )
 6002 FORMAT ( I6, 24X, E19.3, E16.3)
 6004 FORMAT ( 2H I,I4, 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)
      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
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     * * * * * * * * SUBROUTINE SLPJNT * * * * * * * * * * * * * * * *
C     NEW ADDITION SUBROUTINE IN 1983 FOR JOINT SLIPPING BEHAVIOR
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 'PENEW'
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.
```

```
*         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,XRPEP,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,JSLTLR
      IF(LEVEL.NE.1)READ(5,1100)(JLOC(N),N=1,NUMJ)
      IF(LEVEL.NE.1 .AND. JSLTLR.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=XRPES
      IF(RPEP.LE.0.0)RPEP=XRPEP
      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     EALSTIC 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
C     INITIALIZE PREVIOUS VALUES
C
      PENEW=PE
      DO 110 NJ=1,NUMJ
      PEOLDX(NJ)=PE
      PSTRES(NJ)=0.0
  110 PRSTRN(NJ)=0.0
C
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(JSLTLR.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)
```

```
      STRES=STRESP+DSTRNJ*PEOLD/(1.-PNU**2)
      CALL JMOD(STRN,STRNPR,PE,PES,PEP,PEF,STRES,STRESP,PEOLD,PENEW,PNU
     *,IFLAG,JCON,NJ)
      STRN1(NJ)=STRN
      STRES1(NJ)=STRES
      IF(JOINT.EQ.2) WRITE(6,2750)NJ,STRN,STRES,STRNPR,STRESP,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
      AVERR=ERRSUM/FLOAT(NUMJ)
      IF(AVERR.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
```

```
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,I5,
     *            18H JOINT LOCATION IS ,I5,/)
 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..........,I5,//)
 2200 FORMAT(///,13H JOINT NUMBER,10X,14HJOINT LOCATION,10X,
     * 13H JOINT LENGTH,//,
     * (6X,I5,18X,I5,13X,E13.4,/))
 2300 FORMAT(1H1,///,41H SLOTTED JOINTS RESPONSES AT LOAD STEP  =,I5,///
     *          5X,7H STRAIN,5X,7H STRESS,5X,20H JOINT CLOSURE RATIO,
     *          5X,6H IFLAG,//,2E13.3,10X,E13.3,5X,I5)
 2400 FORMAT(1H1,///,41H SLIPPING JOINTS RESPONSES AT LOAD STEP =,I5,///
     *5X,8HJOINT NO,5X,7H STRAIN,5X,7H STRESS,5X,19HJOINT CLOSURE RATIO,
     *5X,6H IFLAG,//,(8X,I5,2E13.3,9X,E13.3,5X,I5))
 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  =,I5,
     * //,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 STRESP,6X,7H PEOLD ,6X,7H PENEW ,4X,6H IFLAG,4X,5H JCON,4X,
     * 5H ICON,//)
 2650 FORMAT(6E13.5,3(4X,I5))
 2700 FORMAT(1H1,//,6X,41H SLOTTED JOINT ITERATIONS AT LOAD STEP  =,I5,
     * //,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 STRESP,6X,7H PEOLD ,6X,7H PENEW ,4X,6H IFLAG,4X,
     * 5H JCON,4X,5H ICON,//)
 2750 FORMAT(6X,I5,6E13.5,3(4X,I5))
 2800 FORMAT(//)
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
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE STEEL(IA,ICOME,IEXIT,LEVEL,NINC,NPMAT,NPPT,PDIA,PIPMAT,
     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)
```

```
C         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),FF(2),FS(3),PALIST(35),PFS(3),PHINGE(5),
     1          PILIST(35),PIPMAT(5,99),PSLIST(35),RESULT(20,99)
*
C
C     NEW ADDITION COMMON STATEMENT IN 1983 FOR JOINT SLIPPING BEHAVIOR
C
*
*     Expand array dimension statements from 30 to 99.
*          SCM  5-22-89
*
      COMMON/XJOINT/DSTRN(99),XLONG(99),JOINT
*
C
      DATA E,XNU,YIELD/30000000.,0.3,33000./
      DATA NFS,FS/3,3.,4.,2./
      DATA FF,FIMIN,GAGE20/0.0433,0.02,0.05,0.038/
      DATA ANALYS,DESIGN/4HANAL,4HDESI/
      DATA CPLIST,BXB,NUMCP,NUMGA/ 4H 3/2,4H  2 ,4H 8/3 ,4H   3,4H   6,
     1     4H1/4    , 2*4H1/2 ,4H1   ,4H2   ,3H X  ,5,7/
      DATA PALIST/
     X     0.03800,0.05066,0.06341,0.07916,0.11091,0.14266,0.17441,
     X     0.04075,0.05433,0.06791,0.08491,0.11900,0.15316,0.18741,
     X     0.03875,0.05158,0.06458,0.08066,0.11300,0.14533,0.17775,
     X     0.04450,0.05925,0.07416,0.09275,0.13000,0.16733,0.20483,
     X     0.12966,0.16691,0.20408,0.22825,0.26658,0.30483,0.34325/
      DATA PILIST/
     X     0.00025,0.00034,0.00044,0.00056,0.00085,0.00120,0.00163,
     X     0.00144,0.00153,0.00194,0.00245,0.00354,0.00471,0.00599,
     X     0.00112,0.00150,0.00189,0.00239,0.00342,0.00453,0.0057 ,
     X     0.00515,0.00689,0.00865,0.01088,0.01545,0.02017,0.02508,
     X     0.06041,0.07816,0.09616,0.10800,0.12691,0.14616,0.16583/
      DATA PSLIST/
     X     0.00177,0.00230,0.00283,0.00349,0.00483,0.00627,0.00787,
     X     0.00427,0.00560,0.00693,0.00854,0.01171,0.01485,0.01805,
     X     0.00419,0.00549,0.00676,0.00831,0.01133,0.01428,0.01724,
     X     0.00995,0.01315,0.01634,0.02025,0.02798,0.03557,0.04308,
     X     0.05741,0.07325,0.08883,0.09891,0.11466,0.13016,0.14575/
C
      GO TO (100,200) , ICOME
C
C     FIRST ENTRY ESTABLISHES PIPE PROPERTIES, AND SAFETY FACTORS
C
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
  100 READ(5,1000) NONLIN,PDIA,PE,PNU,PYIELD,PDEN,PE2,JOINT
      IF(PE.LE.0.0) PE = E
      IF(PNU.LE. 0.0) PNU = XNU
      IF(PYIELD.LE.0.0) PYIELD = YIELD
      IF(NONLIN.NE.1 .AND. NONLIN.NE.2) NONLIN = 0
      PIMIN = PDIA**2/(PE*FF(1))
      IF(PIMIN.GT.FIMIN) PIMIN = PIMIN*FF(1)/FF(2)
C
      WRITE(6,2000) PDIA,PE,PNU,PYIELD,PDEN,NONLIN
      IF(NONLIN.EQ.2) WRITE(6,2001) PE2
      PHINGE(1) = 0.0
      IF(XMODE.EQ.DESIGN) GO TO 110
      READ(5,1010) PA,PI,PS
      WRITE(6,2010) PA,PI,PS
      GO TO 130
C
  110 READ(5,1010) (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
      PAMIN = GAGE20
      PA = PAMIN
      PI = PIMIN
C     NOTE...SECTION MODULUS FOR STANDARD CORRUGATIONS HAS THE RELATION,
      PS = PI/(1.465*SQRT(PI/PA) + 0.375*PA)
      ITER = 0
C
C     INITIALIZE PIPE PROPERTIES ALL AROUND PIPE TO BE THE SAME
C
C
C     NEW ADDITION STATEMENTS IN 1983 FOR JOINT SLIPPING BEHAVIOR.
C     INPUT JOINT PROPERTIES IF JOINT .GT. 0
C
  130 IF(JOINT.EQ.0) GO TO 135
      CALL SLPJNT(ICOME,LEVEL,NPMAT,NPPT,PDIA,PIPMAT,PE,PNU,
     *PA,IA,JCON,ICON)
  135 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
C
      DO 145 N=1,NPPT
      RESULT(14,N) = PA
      RESULT(15,N) = PI
  145 RESULT(16,N) = PI/PS
      ICON = 0
      JCON = 0
      PESUM = PE
      RETURN
C
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
  200 IEXIT = 0
C
      IF(NONLIN.EQ.1 .AND. IA.EQ.NINC) CALL HINGE(PIPMAT,NPMAT,PS,PDIA,
     X PYIELD,SM,SK,RESULT,NPPT,PHINGE)
C
C     IF NONLIN = 2, UPDATE NONLINEAR BEAM PROPERTIES (EFFECTIVE BENDING
C     AND AXIAL STIFFNESS). CONVERGENCE IS SIGNALED BY SUBROUTINE EMOD
C     RETURNING ICON=-1 . ITERATION LIMIT SETS ICON=-2.
C
      IF(NONLIN.NE.2) GO TO 201
```

```
      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 SIGNALED 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.NE.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
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
      IF(ABS((PANEW-PA)/PANEW).LT.0.05) ICHKA = 1
      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
      PINEWD = PFS(2)*DISP*PI/(0.2*PDIA)
      PINEWB = (PFS(3)*PAVG)**2*PBUCK
*
*     Revise intrinsic funtion to generic form.  SCM 3-10-90
*
      PINEW = MAX(PINEWD,PINEWB)
*
      IF(PINEW.LT.PIMIN) PINEW = PIMIN
      IF(ABS((PINEW-PI)/PINEW).LT.0.05) ICHKI = 1
      PI = PINEW
      PS = PI/(1.465*SQRT(PI/PA)+0.375*PA)
C
C     UPDATE SECTION PROPERTIES AND CHECK FOR CONVERGENCE.
C
      IF(ICHKA*ICHKI.EQ.1) GO TO 250
      IF(ITER.GT.5) WRITE(6,2200) ITER,PA,PI,PS
      IF(ITER.GT.5) GO TO 300
      IEXIT = -2
      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,NUMCP
      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
```

```
      IJBEST = IJ
      NGBEST = NGAGE
  280 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
  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)
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 = PCRIT/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.-2) WRITE(6,2995)
      IF(ICON.EQ.-2) STOP
      ICON = 0
      JCON = 0
      RETURN
C
C     FORMATS
C
 1000 FORMAT(I5,6F10.0,I5)
 1010 FORMAT(3F10.0)
 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 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 /
     X 17X,34H NONLIN=2, IMPLIES BILINEAR CURVE  /)
 2001 FORMAT( /15X,36H MODULUS OF UPPER BI-CURVE ..........,E15.5 /)
 2010 FORMAT(// 10X, 31H SECTION PROPERTIES OF PIPE ...  //
     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 THRUST STRESS YIELDING ............ , F10.5 //
     X 15X,36H DEFLECTION COLLAPSE,(0.2*D) ....... , 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 E
     XVALUATED, / 10X,37H WHERE THE 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 //)
 2300 FORMAT(1H1,//10X,19H DESIGN RESULTS AT ,I2,11H ITERATIONS //
     X 15X, 35H *** REQUIRED THRUST AREA ......... , F10.5 //
     X 15X, 35H *** REQUIRED MOM. OR INERTIA ..... , F10.5 //
     X 15X, 35H *** REQUIRED SECTION MODULUS ..... , F10.5 ////
     X 10X, 49H THE FOLLOWING STEEL CORRUGATED SECTIONS MEET THE  /
     X 10X, 37H ABOVE REQUIREMENTS WITH MINIMUM AREA  //
     X15X,12H CORRUGATION,3X,5H GAGE,3X,12H THRUST AREA,3X,13H MOM. INER
     XTIA /)
 2400 FORMAT(17X,3A4,3X,22H (NO ADEQUATE SECTION) )
 2500 FORMAT(17X,3A4,6X,I2,5X,F10.5,5X,F10.5)
 2600 FORMAT(//// 10X,37H NO STANDARD CORRUGATION IS ADEQUATE. /
     X 10X, 45H AN ANALYSIS OF THE REQUIRED SECTION FOLLOWS. )
 2700 FORMAT(//// 10X, 38H THE CORRUGATION WITH MINIMUM AREA AND /
     X10X,21H MOMENT OF INERTIA IS,3A4,5H GAGE,I3,/
     X 10X, 41H AN ANALYSIS OF THIS CORRUGATION FOLLOWS. )
 2800 FORMAT(1H1,//10X,32H STRUCTURAL RESPONSES OF CULVERT ,
     X 3X, 12H LOAD STEP =  ,I2 //
     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 ,4X,9H M-STRESS,4X,9H T-STRESS,4X,9H S-STRESS ,
     X (//F13.2,5E13.3/F13.2,5E13.3))
 2900 FORMAT(1H1,////10X,34H CALCULATED SAFETY FACTORS AT STEP, I3 ///
     X15X,50H THRUST STRESS ....... (YIELD-STRESS/MAX-STRESS) =,F10.2//
     X15X,50H DISPLACEMENT ......... (DIAMETER*0.2/MAX-DISP.) =,F10.2//
     X15X,50H ELASTIC BUCKLING ....... (P-CRITICAL/P-AVERAGE) =,F10.2///
     X 10X, 20H PERFORMANCE FACTORS  ///
     X15X,50H MAXIMUM STRESS ...... (YIELD-STRESS/MAX-STRESS) =,F10.2 //
     X15X,50H HANDLING REQUIREMENT ............. (FF*EI/D**2) =,F10.2 /)
 2950 FORMAT(/// 10X, 27H PLASTIC HINGING EVALUATION ///
     X 15X,42H   RATIO OF LOAD AT INITIAL YIELDING ..... , F10.5 //
     X 15X,42H   RATIO OF MAXIMUM HINGE PENETRATION .... , F10.5 //
```

```
     X 15X,42H   RATIO OF CIRCUMFERENCE IN YIELDING .... , F10.5 //
     X 15X,42H   RATIO OF MAX-DISP. DUE TO YIELDING .... , F10.5 //
     X 15X,42H   RATIO OF MAX-STRAIN/YIELD-STRAIN ...... , F10.5 //)
2990 FORMAT(1H1,///10X,40H NONLINEAR CHARACTERISTICS,  LOAD STEP =,I3//
     X5X,73H NODE       INNER-FIBER       OUTER-FIBER      STRAIN RATIO
     X  FRACTION OF /21X,57HSTRAIN              STRAIN      MAX-TO-YIELD
     X WALL YIELDED // (5X,I5,2E17.5,7X,F10.5,7X,F10.5))
2995 FORMAT(1H0,////37(2H *)///73H THE SOLUTION WILL NOT CONTINUE DUE T
     XO THE RATIO OF ERROR IN THE MODULUS   ///47H IT IS RECOMMENDABLE T
     XO USE SMALLER LOAD STEPS   ///37(2H *))
2999 FORMAT(//,5X,6H JCON=,I5,/,5X,6H ICON=,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 STFSUB
C
C     THIS SUBROUTINE FORMS THE ELEMENT MATRIX FOR A TRIANGULAR
C     ELEMENT
C
*
*     Uncomment this statement in all program modules for double
*          precision version. SCM 3-10-90
*
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
      DIMENSION AL(3,3),BT(3,3),CT(12,12),CTP(12,17),GA(3,3),ZT(12)
      COMMON /ELEMNT/ AS(3),BS(3),C1(17,17),EST(4,3),RQ(4),RT(3),
     1                STORE(17),Z4(12),ZQ(4),ZDUM(3),ZY(17),J1,RO
      COMMON /SIZEPR/ BIVD(3),IX,NINC
      COMMON /MATERL/ CP(3,3),DEN(10),E(10,10),GNU(10,10),H(10,10),
     1                ITYPE(10),NON(10)
      COMMON /ESTABL/ ITRW(28,2),NETCL(17,2),T(28,2)
      CALL GEOM(AA,D1,D2,D3)
      DO 155 N=1,3
      BN=BS(N)/12.0
      AN=AS(N)/12.0
      DO 150 I=1,3
      AL(N,I)=2.0*BN   *BS(I)
      BT(N,I)=AN   *BS(I)+AS(I)*BN
  150 GA(N,I)=2.0*AN   *AS(I)
      AL(N,N)=D1/12.0
      BT(N,N)=D2/12.0
  155 GA(N,N)=D3/12.0
      DU=0.5/AA
      C11=CP(1,1)*DU
      C12=CP(1,2)*DU
      C13=CP(1,3)*DU
      C22=CP(2,2)*DU
      C23=CP(2,3)*DU
      C33=CP(3,3)*DU
      FY = -RO*AA/8.
C
C     CHECK FOR DATA ERROR
C
      IF(AA.LE.0.0) WRITE(6,911) IX
  911 FORMAT (// 7X,21HDATA ERROR IN ELEMENT  , I4//)
C
C     FORM ELEMENT STIFFNESS MATRIX
C
      DO 250 N=1,3
      DU1 = 1.0
      IF(N*J1.LT.2 .OR. N*J1.GT.4) DU1 = 2.0
      BN=BS(N)
      AN=AS(N)
      ZT(N  )=0.0
      ZT(N+3) = FY*DU1
      ZT(N+6)=0.0
      ZT(N+9) = 0.0
      DO 250 I=1,3
      D2=-1.0/3.0
      D4=AL(I,N)
      D5=BT(I,N)
      D6=GA(I,N)
      BI=BS(I)
      AI=AS(I)
      X1=BI*BN
      X2=BI*AN
      X3=AI*BN
      X4=AI*AN
      X5=X2+X3
      CT(N,I)=    C11*X1+C13* X5    +C33*X4
      CT(N,I+3)=    C12*X3+C13*X1+C23*X4+C33*X2
      CT(N,I+6)=D2*CT(N,I)
      CT(N,I+9)=D2*CT(N,I+3)
      CT(N+3,I+3)=    C22*X4+C23* X5    +C33*X1
      CT(N+3,I+6)=D2*(C13*X1+C12*X2+     C33*X3+C23*X4)
      CT(N+3,I+9)=D2*CT(N+3,I+3)
      CT(N+6,I+6)=C11*D4+2.0*C13*D5+C33*D6
      CT(N+6,I+9)=C13*D4    +(C12+C33)*D5+C23*D6
  250 CT(N+9,I+9)=C22*D6+2.0*C23*D5+C33*D4
      DO 300 N=1,12
      DO 300 I=N,12
  300 CT(I,N)=CT(N,I)
C
C     SUB-ELEMENT UNKNOWNS TRANSFORMED TO ELEMENT UNKNOWNS
C
      KB=1
      MC=1
      DO 400 K=1,17
      KA=NETCL(K,J1)
      DO 320 MM=1,KA
      M=ITRW(KB,J1)
      ZY(K)=ZY(K)+ZT(M)*T(KB,J1)
  320 KB=KB+1
      DO 380 L=1,12
      CTP(L,K)=0.0
      MP=MC
      DO 380 MM=1,KA
      M=ITRW(MP,J1)
      CTP(L,K)=CTP(L,K)+CT(L,M)*T(MP,J1)
  380 MP=MP+1
  400 MC=MC+KA
      MC=1
      DO 410 I=1,17
      KA=NETCL(I,J1)
      DO 405 K=1,17
      MP=MC
      DO 405 MM=1,KA
      M=ITRW(MP,J1)
      C1(I,K)=C1(I,K)+CTP(M,K)*T(MP,J1)
  405 MP=MP+1
  410 MC=MC+KA
      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 STIFNS(BIV,NOD,NODB,NQ,X,Y,MNO,U,V,ST,IA,LB1,MN,IAC,
     X                MNX,ITER,ICON,NEQ,NBPTC,NELEM,NPT)
C
C     THIS SUBROUTINE FORMS THE ELEMENT MATRIX FOR A QUADRILATERAL
C     ELEMENT
C
*     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(4,NELEM),NODB(NBPTC),NQ(NPT),
     1          X(NPT),Y(NPT),U(NPT),V(NPT),MNO(NELEM),ST(6,NELEM)
      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 /SIZEPR/ BIVD(3),IX,NINC
      COMMON/SYSTEM/B(240),A(240,120)
      COMMON /MATERL/ CP(3,3),DEN(10),E(10,10),GNU(10,10),H(10,10),
     1                ITYPE(10),NON(10)
      DIMENSION IIFLG(3)
C
C     INITIALIZE ELEMENT STIFFNESS AND LOADING
C
      DO 60 J=1,17
      DO 50 K=1,17
   50 C1(J,K)=0.0
   60 ZY(J)=0.0
C
C     THE CORNER POINT COORDINATES ARE FOUND
C
      DO  90 J=1,4
      K = NOD(J,IX)
      RQ(J)=X(K)
   90 ZQ(J)=Y(K)
      NU=2
C
C     BENDING ELEMENT STIFFNESS DEVELOPED
C
      IF(NOD(3,IX).NE.NOD(2,IX)) GO TO 95
      CALL BEAMEL
      NU=3
      GO TO 313
C
C     INTERFACE ELEMENT CONSTRAINT AND LOAD DEVELOPMENT
C
   95 IF(MNX.EQ.0) GO TO 97
      CALL XFACES(MN,IAC,MNX,IX,ICON,ITER,NELEM,NPT,MNO,NOD,ST,U,V,IA)
      GO TO 313
C
C     THE SUB-ELEMENT MATRICES ARE FOUND AND ADDED INTO THE ELEMENT
C     MATRIX, JTRY=1 IMPLIES ONE TRIANGLE, JTRY = 2, IMPLIES TWO .
C
   97 JTRY=2
      IF (NOD(4,IX) .EQ. NOD(3,IX) ) JTRY = 1
C
      DO 300 J=1,JTRY
      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 UNKNOWNS 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))
      K1=KK/10000
      IF (NRQ .NE. K1) 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  3-18-89
*
      IF (IIFLG(1) .EQ. 2) IIFLG(3) = 0
      IF (IIFLG(2) .EQ. 2) IIFLG(3) = 0
*
```

```
      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
  320 DO 330 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
      NCCC=0
      DO 345 L=1,4
      JJ = NOD(L,IX)
      NCN=NQ(JJ)/100-NR-M
      DO 345 N=1,NU
```

```
      NCCC=NCCC+1
      NCN=NCN+1
      IF (NCN.LT. 1) GO TO 344
      A(NRM,NCN) = A(NRM,NCN) +C1(NRCC,NCCC)
  344 CONTINUE
  345 CONTINUE
  350 B(NRM) = B(NRM) + ZY(NRCC)
      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 STRESS (BIV,MNO,NOD,NODB,NQ,ST,STPIPE,U,V,IA,NBPTC,
     1                   NELEM,NELEMT,NPMAT,NPPT,NPT,SKAVG,SMAVG)
*
*     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),MNO(NELEM),NOD(4,NELEM),NODB(NBPTC),NQ(NPT)
     1          ,ST(6,NELEM),STPIPE(2,NPPT ),U(NPT),V(NPT)
      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)
      COMMON/SYSTEM/B(240),A(240,120)
      COMMON /ESTABL/ ITRW(28,2),NETCL(17,2),T(28,2)
C
C     BOUNDARY POINTS TRANSFORMED TO X-Y COORDINATE SYSTEM
C
      DO 600 K=1,NBPTC
      KK=IABS(NODB(K))
      NODB(K)=KK
      ANG = BIV(3,K)
      IF(ANG .EQ. 0.0) GO TO 600
      K1=KK/10000
      JJ=NQ(K1)/100
      IF(JJ.EQ.NQ(K1+1)/100) GO TO 600
      CT=COS(ANG)
      SA=SIN(ANG)
      D1 = B(JJ)
      D2 = B(JJ+1)
      B(JJ) = D1 *CT -D2*SA
      B(JJ+1) = D1*SA +D2*CT
  600 CONTINUE
C
C     FOR EACH ELEMENT FIND STRAINS AND STRESSES
C
      SMAVG = 0.0
      SKAVG = 0.0
      SMKNEL = 0.0
      REWIND 12
      DO 726 III=1,NELEMT
C
C     RECALL ELEMENT STIFFNESS MATRIX
C
      READ(12) (ZY(M),STORE(M),(C1(M,N),N=1,17),M=1,17),
     1         ((CP(J,K),K=1,3),J=1,3)
      I=STORE(17)+0.1
```

```
      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 UNKNOWNS
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)
  610 L= L+2
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
  622 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
C     CALCULATE SUB-ELEMENT EXPANSION COEFFICIENTS
C
      DO  715 J1=1,JTRY
      DO  670 K=1,12
  670 Z4(K)=0.0
      M=6*J1-7
      DO 675 K=1,3
      M=M+2
      RT(K)=STORE(M)
  675 ZT(K)=STORE(M+1)
      J1P=J1
      DU=0.5
      IF (JTRY .EQ. 2)GO TO 677
      DU=1.0
      J1P=2
  677 CALL GEOM(AA,D1,D2,D3)
      DU=DU/AA
      MP=1
      DO  680 K=1,17
      KA=NETCL(K,J1)
      DO  680 MM=1,KA
      M=ITRW(MP,J1)
      Z4(M)=Z4(M)+ZY(K)*T(MP,J1)
  680 MP=MP+1
C
C     STRAINS CALCULATED FOR SUBELEMENTS
C
      EX=0.
      EY=0.0
      GAMXY=0.0
      DO 701 K=1,3
      D1=AS(K)
      D2=BS(K)
      EX=EX+Z4(K)*D2
      EY=EY+Z4(K+3)*D1
  701 GAMXY=GAMXY+Z4(K)*D1+Z4(K+3)*D2
      RQ(1)=EX*DU+RQ(1)
      RQ(2)=EY*DU+RQ(2)
      RQ(3)=GAMXY*DU+RQ(3)
      DU=(1.0-DU*AA)*0.5/AA
      IF(J1 .EQ. 2)GO TO 705
      RQ(1)=RQ(1)-DU*BS(1)*(Z4(7)-Z4(8)-Z4(9))
      RQ(2)=RQ(2)-DU*AS(1)*(Z4(10)-Z4(11)-Z4(12))
      RQ(3)=RQ(3)-DU*(AS(1)*(Z4(7)-Z4(8)-Z4(9)) +BS(1)*(Z4(10)-Z4(11)
     1     -Z4(12)))
      GO TO 715
  705 RQ(1)=RQ(1)-DU   *BS(3)*(Z4(9)-Z4(8)-Z4(7))
      RQ(2)=RQ(2)-DU   *AS(3)*(Z4(12)-Z4(11)-Z4(10))
      RQ(3)=RQ(3)-DU   *(AS(3)*(Z4(9)-Z4(8)-Z4(7)) +BS(3)*(Z4(12)-Z4(11)
     1     -Z4(10)))
  715 CONTINUE
C
C     INCREMENTAL STRESSES CALCULATED
C
      DO 717 L=1,3
      DU=0.0
      DO 716 J=1,3
  716 DU=DU+RQ(J)*CP(L,J)
  717 ZQ(L)=DU
C
C     COMPUTE AVERAGE SOIL PROPERTIES FOR BUCKLING AND HINGING
C
      RT(1) = 0.0
```

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

```
      WRITE(LURITE,6110) IX,NX,MODX(N),XCX(N),YCX(N),IZERO,FZERO,FZERO  XCAN0087
  200 CONTINUE                                                          XCAN0088
C                                                                       XCAN0089
C     READ, MODIFY AND SAVE ELEMENT GENERATION INPUT                    XCAN0090
C                                                                       XCAN0091
      INCRT = 0                                                         XCAN0092
  300 READ(LUREAD,6115) LIMIT,NEL,N1,N2,N3,N4,NUMMAT,INCR,IZERO,IZERO,  XCAN0093
     X IZERO,IZERO                                                      XCAN0094
      IF (NEL .EQ. 90 .AND. INCR .EQ. 1) INCRT = 1                      XCAN0095
      IQUAD = 2                                                         XCAN0096
      IF(NEL.LE.10) IQUAD = 0                                           XCAN0097
      NX1 = ISWAS(N1,1) + IQUAD*ISWAS(N1,2)                             XCAN0098
      IF(N1.EQ.0) NX1 = 0                                               XCAN0099
      NX2 = ISWAS(N2,1) + IQUAD*ISWAS(N2,2)                             XCAN0100
      IF(N2.EQ.0) NX2 = 0                                               XCAN0101
      NX3 = ISWAS(N3,1) + IQUAD*ISWAS(N3,2)                             XCAN0102
      IF(N3.EQ.0) NX3 = 0                                               XCAN0103
      NX4 = ISWAS(N4,1) + IQUAD*ISWAS(N4,2)                             XCAN0104
      IF(N4.EQ.0) NX4 = 0                                               XCAN0105
      WRITE(LURITE,6115) IMORE,NEL,NX1,NX2,NX3,NX4,NUMMAT,INCR,IZERO,   XCAN0106
     X IZERO,IZERO,IZERO                                                XCAN0107
      IF(LIMIT.NE.ILAST) GO TO 300                                      XCAN0108
C                                                                       XCAN0109
      NEL = NUMEL-11                                                    XCAN0110
      IX = IMORE                                                        XCAN0111
      IONE = 1                                                          XCAN0112
      DO 350 N=1,11                                                     XCAN0113
      IF(N.EQ.11) IX = ILAST                                            XCAN0114
      NEL = NEL +1                                                      XCAN0115
      NX1 = NPX(N)                                                      XCAN0116
      NX2 = NX1 + 2                                                     XCAN0117
      NX3 = NX1 + 1                                                     XCAN0118
      INCF = INCC(N)                                                    XCAN0119
      IF (INCRT .EQ. 1) INCF = 1                                        XCAN0120
      WRITE(LURITE,6115) IX,NEL,NX1,NX2,NX3,IZERO,N,INCF,IONE,IZERO,    XCAN0121
     X IZERO,IZERO                                                      XCAN0122
  350 CONTINUE                                                          XCAN0123
C                                                                       XCAN0124
C     READ, MODIFY AND SAVE BOUNDARY CONDION INPUT                      XCAN0125
C                                                                       XCAN0126
  400 READ(LUREAD,6120) LIMIT,ND1,ICX,FZERO,ICY,FZERO,FZERO,INC,ND2,    XCAN0127
     X NNPINC,PLOAD,PLOAD                                               XCAN0128
      NX1 = ISWAS(ND1,1)                                                XCAN0129
      NX2 = 0                                                           XCAN0130
      IF(ND2.GT.0) NX2= ISWAS(ND2,1)                                    XCAN0131
      WRITE(LURITE,6120) IMORE,NX1,ICX,FZERO,ICY,FZERO,FZERO,INC,NX2,   XCAN0132
     X NNPINC,PLOAD,PLOAD                                               XCAN0133
      IF(LIMIT.NE.ILAST) GO TO 400                                      XCAN0134
C                                                                       XCAN0135
      IL = IMORE                                                        XCAN0136
      DO 450 N = 1, NADDB                                               XCAN0137
      IF (N .EQ. NADDB) IL = ILAST                                      XCAN0138
      NX1 = NBADD(N)                                                    XCAN0139
      ICX = IONE                                                        XCAN0140
      ICY = IZERO                                                       XCAN0141
      IF (N .GT. 2) ICX = IZERO                                         XCAN0142
      IF (N .GT. 2) ICY = IONE                                          XCAN0143
  450 WRITE(LURITE ,6120) IL,NX1,ICX,FZERO,ICY,FZERO,FZERO,IONE,        XCAN0144
     1                    IZERO,IZERO,FZERO,FZERO                       XCAN0145
C                                                                       XCAN0146
C     REDEFINE INPUT TAPE UNIT                                          XCAN0147
C                                                                       XCAN0148
      REWIND LUREAD                                                     XCAN0149
      REWIND LURITE                                                     XCAN0150
      LUREAD = 11                                                       XCAN0151
      RETURN                                                            XCAN0152
C                                                                       XCAN0153
C     FORMATS                                                           XCAN0154
C                                                                       XCAN0155
 6105 FORMAT (A4,1X,17A4,3X,A4/8I5)                                     XCAN0156
 6110 FORMAT (A1, I4, I5, 2F10.3, I5, 5X,2F10.3)                        XCAN0157
 6115 FORMAT ( A1, I4, 10I5)                                            XCAN0158
 6120 FORMAT ( A1, I4, I5, F10.2, I5, 2F10.2, 3I5, 2F10.2)              XCAN0159
      END                                                               XCAN0160
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * XCN20001
*     CANDE-89 (C)ULVERT (AN)ALYSIS AND (DE)SIGN PROGRAM                XCN20002
*                 VERSION 1.0, MARCH 1990                               XCN20003
*            PUBLIC DOMAIN SOFTWARE CONTRIBUTED BY                      XCN20004
*              THE FEDERAL HIGHWAY ADMINISTRATION                       XCN20005
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * XCN20006
      SUBROUTINE XCAN2(LUREAD,IGO)                                      XCN20007
C                                                                       XCN20008
C     MODIFY CAN LUDATA TAPE FOR INTERFACE ELEMENTS ALONG TRENCH FACE   XCN20009
C                                                                       XCN20010
*     Uncomment this statement in all program modules for double        XCN20011
*          precision version. SCM 3-10-90                               XCN20012
*                                                                       XCN20013
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               XCN20014
*                                                                       XCN20015
      DIMENSION  ISWAS(110,2),TITLE(17),NPX(11),MODX(11),XCX(11),YCX(11)XCN20016
      DIMENSION IADDN(7),INCC(7)                                        XCN20017
      DIMENSION NELNX(8)                                                XCN20018
      DATA NELNX /23,24,25,26,71,78,85,92/                              XCN20019
      DATA IADDN /51,62,71,82,90,98,106/                                XCN20020
      DATA INCC /2,3,3,4,3*5/                                           XCN20021
      DATA IMORE,ILAST/1H ,1HL/                                         XCN20022
C                                                                       XCN20023
      LUREAD = 10                                                       XCN20024
      IF(IGO.EQ.0) RETURN                                               XCN20025
      LURITE = 11                                                       XCN20026
      REWIND LURITE                                                     XCN20027
      REWIND LUREAD                                                     XCN20028
C                                                                       XCN20029
C     ESTABLISH IS-WAS LIST FOR NODES TO INCLUDE INTERFACE NODES.       XCN20030
C                                                                       XCN20031
      ISWAS(1,1) = 1                                                    XCN20032
      ISWAS(1,2) = 0                                                    XCN20033
      DO 50 N=2,110                                                     XCN20034
      IZONE = 0                                                         XCN20035
      DO 20 K = 1, 7                                                    XCN20036
   20 IF (IADDN(K) .EQ. N) IZONE = 1                                    XCN20037
      ISWAS(N,1) = ISWAS(N-1,1) + 1 + 2*ISWAS(N-1,2)                    XCN20038
      ISWAS(N,2) = IZONE                                                XCN20039
      WRITE(6,6081) N,ISWAS(N,1),ISWAS(N,2),IZONE                       XCN20040
 6081 FORMAT (10X,6I10)                                                 XCN20041
   50 CONTINUE                                                          XCN20042
C                                                                       XCN20043
C     READ, MODIFY AND SAVE CONTROL INPUT                               XCN20044
C                                                                       XCN20045
      READ (LUREAD,6105) PREP,TITLE,WORDZ,NINC,MGENPR,NPUTCK,IPLOT,IWRT,XCN20046
     X NUMNP,NUMEL,NBPTC                                                XCN20047
      NUMNP = NUMNP + 14                                                XCN20048
      NUMEL = NUMEL + 7                                                 XCN20049
      NBPTC = NBPTC + 2                                                 XCN20050
      WRITE(LURITE,6105) PREP,TITLE,WORDZ,NINC,MGENPR,NPUTCK,IPLOT,IWRT,XCN20051
     X NUMNP,NUMEL,NBPTC                                                XCN20052
```

```
C
C     READ, MODIFY AND SAVE NODAL GENERATION INPUT
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
      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
      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
      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. NELNX(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
      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 (INCRT .EQ. 1) INCF = 1
      WRITE(LURITE,6115) IX,NEL,NX1,NX2,NX3,IZERO,N,INCF,IONE,IZERO,
     X IZERO,IZERO
350   CONTINUE
C
C
400   READ(LUREAD,6120) LIMIT,ND1,ICX,FZERO,ICY,FZERO,FZERO,INC,ND2,
     X NNPINC,PLOAD,PLOAD
      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,NX2,
     X NNPINC,PLOAD,PLOAD
      WRITE(6,6120) IMORE,NX1,ICX,FZERO,ICY,FZERO,FZERO,INC,
     1       NX2,NNPINC,PLOAD,PLOAD
      IF(LIMIT.NE.ILAST) GO TO 400
C
C     REDEFINE INPUT TAPE UNIT
C
      REWIND LUREAD
      REWIND LURITE
      LUREAD = 11
      RETURN
C
C     FORMATS
C
 6105 FORMAT (A4,1X,17A4,3X,A4/8I5)
 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
```

```
*
*     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
*     Expand array dimension statements from 30 to 99.
*          SCM  5-22-89
*
      COMMON/XFACE/NXMAT,ANGLE(99),FCOEF(99),TENSIL(99),ITMAX
*
      COMMON /ELEMNT/ AS(3),BS(3),CI(17,17),EST(4,3),RQ(4),RT(3),
     1                STORE(17),Z4(12),ZQ(4),ZT(3),ZY(17),J1,RO
*
*     Add common statement for output control.  SCM 3-18-89
*
      COMMON /OUTPUT/ IWRT,IPLOT,MAXIT,JCON
*
      DIMENSION MNO(NELEM),NOD(4,NELEM),ST(6,NELEM),U(NPT),V(NPT)
*
*     Add dimension statement to force FREE-FREE state.  SCM 10-18-89
*
      DIMENSION XFFLAG(700)
*
      DATA CON/0.0174533/
C
C     THE FOLLOWING VARIABLES ARE IDENTIFIED FOR AN INTERFACE ELEMENT
C     COMPOSED OF NODES I AND J.
C
C            ST(1,NEL) = TOTAL NORMAL INTERFACE FORCE
C            ST(2,NEL) = TOTAL SHEAR INTERFACE FORCE
C            ST(3,NEL) = LATEST INCREMENT OF NORMAL FORCE
C            ST(4,NEL) = LATEST INCREMENT OF SHEAR FORCE
C            ST(5,NEL) = RELATIVE X-DISPLACEMENT INCREMENT, DU(J)-DU(I)
C            ST(6,NEL) = RELATIVE Y-DISPLACEMENT INCREMENT, DV(J)-DV(I)
C
C     CURRENT STATE OF INTERFACE ELEMENT IS GIVEN BY.....
C
C            MNX = 1, IMPLIES NODES ARE FIXED IN BOTH NORMAL AND
C                     TANGENT DIRECTION.
C            MNX = 2, IMPLIES NODES FIXED IN NORMAL DIRECTION
C                     TANGENT SLIPPING WITH RDUS .LT. 0.0
C            MNX = 3, IMPLIES NODES FIXED IN NORMAL DIRECTION
C                     TANGENT SLIPPING WITH RDUS .GT. 0.0
C            MNX = 4, IMPLIES FREE IN NORMAL DIRECTION AND FREE
C                     IN TANGENT DIRECTION
C
C     TESTING PARAMETERS FOR CHANGING STATE ARE .....
C
C            PN = TOTAL NORMAL FORCE
C            PS = TOTAL SHEAR FORCE
C            RDUS = RELATIVE TANGENT DISPLACEMENT INC., DUS(J)-DUS(I)
C            SIGN = DIRECTION OF FRICTIONAL RESISTANCE ON NODE J
C            UN = TOTAL NORMAL DISPLACEMENT
C
*
*     Lines added to force a FREE-FREE state.  SCM 10-18-89
*
      IF(IA.EQ.1 .OR. IA.EQ.0) IAXF = 0
      IF((IA-IAXF).NE.0) THEN
         DO 1,J=1,NELEM
            XFFLAG(J) = 0
    1       CONTINUE
         END IF
      IAXF = IA

C
C     EQUILIBRATE DISPLACEMENTS WHEN INTERFACE FIRST ENTERS SYSTEM
C
      IF(ITER.NE.1) GO TO 2
      IF(IAC.NE.IA) 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
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)
      PNPRE = ST(1,NEL)-ST(3,NEL)
      PSPRE = 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
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 addded to force a FREE-FREE state.  SCM 10-18-89
*
      IF(XFFLAG(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
C     MNX =1, FIX-FIX ON ENTRY, ANY STATE POSSIBLE ON EXIT
```

```
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(FCOEF(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
*
*     Add relaxation function to enhance convergence. SCM 3-20-90
*
      IF(ITER.GT.2) PS2 = 0.5*PS2 + 0.5*ST(4,NEL)
*
      IF(PS2.EQ.0.0) GO TO 100
      CHECK = ABS((PS2-ST(4,NEL))/PS2)
      IF(CHECK.GT.0.05) LKCHNG = 1
      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
150   C1(J,I) = C1(I,J)
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
```

```
C                                                                       XFAC0264
C     PRINT-OUT RECORD OF INTERFACE ITERATIONS IF ITMAX IS .LT. ZERO    XFAC0265
C                                                                       XFAC0266
      ITMAXA=IABS(ITMAX)                                                XFAC0267
*                                                                       XFAC0268
      IF(ITMAXA.GT.MAXIT) MAXIT = ITMAXA                                XFAC0269
*                                                                       XFAC0270
*     Disable ITMAX as print control.   SCM 3-4-89                      XFAC0271
*                                                                       XFAC0272
*     IF (ITMAX.GT.0) GO TO 200                                         XFAC0273
*                                                                       XFAC0274
*     Revise unit number for XFACES trace output. SCM 1-9-89            XFAC0275
*     Use IWRT for print control.  SCM  3-4-89                          XFAC0276
*                                                                       XFAC0277
*     IF (IA-IAP.NE.0) WRITE(6,6000) IA                                 XFAC0278
*                                                                       XFAC0279
      IF(IWRT.LT.3) GO TO 200                                           XFAC0280
      IF (IA-IAP.NE.0) WRITE(17,6000) IA                                XFAC0281
*                                                                       XFAC0282
      IAP=IA                                                            XFAC0283
*                                                                       XFAC0284
*     Revise iteration output variables and format. SCM 1-9-89          XFAC0285
*                                                                       XFAC0286
*     WRITE(6,6005)ITER,NEL,MNXP,MNX,ST(3,NEL),ST(4,NEL),RDUN,RDUS      XFAC0287
*                                                                       XFAC0288
      WRITE(17,6005)ITER,NEL,MNXP,MNX,ST(3,NEL),ST(4,NEL),RDUN,RDUS,    XFAC0289
     X BREAK,APS,RESIST,CHECK                                           XFAC0290
*                                                                       XFAC0291
  200 CONTINUE                                                          XFAC0292
C                                                                       XFAC0293
C     STORE CURRENT STATE OF ELEMENT AND CHECK CONVERGENCE              XFAC0294
C                                                                       XFAC0295
      MNO(NEL) = 1000*MN + 10*IAC + MNX                                 XFAC0296
      IF(LKCHNG.NE.0) ICON = 0                                          XFAC0297
      IF(ITER.EQ.1) ICON = 0                                            XFAC0298
*                                                                       XFAC0299
*     Revise test for printing warning.   SCM 3-11-89                   XFAC0300
*                                                                       XFAC0301
*     IF (ITER .GT. ITMAXA) WRITE(6,6010) NEL                           XFAC0302
*                                                                       XFAC0303
      IF(ITER.GE.ITMAXA .AND. LKCHNG.NE.0) WRITE(6,6010) NEL            XFAC0304
*                                                                       XFAC0305
      IF(ITER.GE.ITMAXA) ICON = 1                                       XFAC0306
C                                                                       XFAC0307
      RETURN                                                            XFAC0308
*                                                                       XFAC0309
*     FORMATS                                                           XFAC0310
*                                                                       XFAC0311
 6000 FORMAT(1H1,46HINTERFACE-ITER.-TRACE (INCREMENT VALUES) STEP=,I2,//XFAC0312
     X2X,4HITER,2X,4HELEM,3X,6HOLD-ST,1X,6HNEW-ST,4X,7HN-FORCE,5X,      XFAC0313
     X7HT-FORCE,5X,7HN-SEPAR,5X,7HT-SEPAR,7X,5HBREAK,7X,3HAPS,8X,       XFAC0314
     X6HRESIST,7X,5HCHECK,/)                                            XFAC0315
 6005 FORMAT(1X,I5,3I7,8F12.4)                                          XFAC0316
*                                                                       XFAC0317
 6010 FORMAT(//,10X,59H *** WARNING INTERFACE DID NOT CONVERGE FOR ELEMEXFAC0318
     XNT NUMBER ,I3, 25H.  PROGRAM WILL CONTINUE.)                      XFAC0319
      END                                                               XFAC0320
```