NONLINEAR BEHAVIOR OF REINFORCED CONCRETE BEAMS BY THE FINITE ELEMENT METHOD

By

ALEXANDER JEAN LASSKER

Diploma, Civil Engineering

Swiss Federal Institute of Technology

Zurich, Switzerland

December, 1964

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
May, 1972

NONLINEAR BEHAVIOR OF REINFORCED

CONCRETE BEAMS BY THE FINITE

ELEMENT METHOD

Thesis Approved:

William P. Dawkins

Thesis Adviser

R. Bennett

A. E. Kelly

Donald Boyd

N. Durham

Dean of the Graduate College

To my wife,

Elizabeth

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to the members of my committee:

To Dr. W. P. Dawkins, who served as my major advisor, for his invaluable guidance, interest and friendship throughout the preparation of this dissertation;

To Dr. D. E. Boyd for his many suggestions, his helpful advice and encouragement;

To Drs. R. K. Munshi and A. E. Kelly for their sound instruction.

The author takes this opportunity to thank Dr. M. Abdel-Hady, F. G. Amrhein, V. G. Tracey, and E. Hardy for their friendship. Thanks are also due to J. H. Hepp for his invaluable assistance with the computer program.

In addition, I would like to thank Mr. and Mrs. J. Eichhorn and Mr. and Mrs. G. Elliott for their encouragement and support.

Finally, I am deeply indebted to my family, especially to my wife, Elizabeth, my son, Urs, and my parents for their patience, understanding and support.

TABLE OF CONTENTS

Chapter | Page

LIST OF FIGURES

NOMENCLATURE

A	area of concrete element
$[a]$	rectangular matrix: function of position coordinates of a point within an element
$[b]$	strain displacement transformation matrix
$[D]$	square matrix of elastic constants
E	modulus of elasticity
$\{e\}$	column vector of total strains within an element
G	shear modulus
H'	strain hardening parameter (slope of effective stress-effective plastic strain function)
I_2	second stress invariant
J_2	second strain invariant
$[R]$	coordinate transformation matrix for concrete elements
S	surface on the boundary of a continuum
$[T]$	coordinate transformation matrix for steel and bond elements
$\{U\}$	column vector of nodal displacements in the global coordinate system
$\{U\}$	column vector of nodal displacements in the local system
u	displacement function
V	volume of concrete elements
$\left.\begin{matrix}x\\y\end{matrix}\right\}$	coordinates of nodes

δ	variational operator
Δ	symbol for incremental values
$\{\Delta\epsilon_{o}, p\}$	initial strain increments
$\{\Delta\epsilon_{p}\}$	plastic strain increments
$\{\epsilon\}$	column vector elastic strains within an element
ϵ_{o}	uniaxial strains
ϵ_{eff}	effective strains
ν	Poisson's ratio
$\{\sigma\}$	column vector of stresses within an element
σ_{o}	uniaxial stress
σ_{eff}	effective stress
τ_{oct}	octahedral shear stress

CHAPTER I

INTRODUCTION

1.1 General Discussion

The behavior of reinforced concrete structures subjected to various types of loads has been studied extensively during the past few decades. In spite of many efforts, no basic analytical approach has been developed to determine accurately the stress distribution in the concrete and the steel. This is mainly due to the fact that the constitutive relations for concrete depend on a number of factors such as the size and shape of the structure, the size, the material properties and the composition of the aggregate, and the rate and duration of loading. Furthermore, the tensile strength of concrete is much lower than its compressive strength. Therefore, additional difficulties arise from the continuing change in structural configuration caused by cracks in the concrete.

It is even more difficult to express the many different geometric shapes of the stress-strain curves for steel in analytical form. Here the manufacturing process and the choice of alloys have the most significant influence on the material properties. Finally, time-dependent effects on concrete strains, steel relaxation and complicated laws of interaction between concrete and reinforcements render a closed-form solution practically impossible. It is, therefore, necessary to utilize empirical laws obtained from extensive test data.

Present methods of analysis or design are based on assumptions which allow the application of the fundamental principles of continuum mechanics and empirical or simplified constitutive equations. Two different approaches are commonly used in the design of reinforced concrete structures. Both methods assume a perfect bond between steel and concrete and neglect the tensile resistance of concrete. The first assumption allows the use of the classical Navier-Bernoulli stipulation for planes perpendicular to the member axis. According to this assumption, these planes remain plane and perpendicular to the centroidal axis during the entire load history. Experiments on reinforced concrete beams have confirmed that the assumed strain distribution actually deviates very little from the real strain condition, provided that good bonding exists.

The main difference in the two methods lies in the choice of stress-strain relations. The "Working Stress Method" (1) utilizes linear material laws. Since concrete behaves elastically only as long as the maximum compressive stress is less than about half the ultimate strength, this approach has failed to give correct pictures of the stress distribution at high loads.

The "Ultimate Strength Method" (1), on the other hand, is based upon stress conditions just before failure occurs. It may appear essential to use realistic constitutive relations at these high stress levels. However, this is not the case because the geometric shape of the stress distribution has little effect on the location and magnitude of the resultant compressive force in the concrete. The real stress situation is therefore usually approximated by an equivalent rectangular or trapezoidal stress block.

Although both methods are of chief importance in design, neither one is of much help in studies of the nonlinear behavior of reinforced concrete beams. Such investigations are extremely involved due to previously stated reasons. Any reliable approach must therefore resort to numerical methods. With the introduction of the finite element technique to be discussed subsequently, such an analysis procedure has been established for the solution of complex problems of continuum mechanics. The application of this method results in a large system of linear, simultaneous equations which can be solved very efficiently on digital computers. Nonlinear problems introduce no new difficulties, since they can be treated either by iteration or as a sequence of consecutive linear problems.

1.2 Purpose and Scope of This Study

The purpose of this study is to develop a reliable tool for the analytical study of reinforced concrete members through their entire elastic, inelastic, and ultimate ranges.

The main emphasis is placed on the behavior in the inelastic range. Consequently, the problem approach is based upon nonlinear constitutive relations for steel, concrete, and bond. Nonlinearities introduced through the change in geometry are not included since the beams are assumed to have failed long before large displacements develop. Also, time dependent effects on concrete strains (such as creep and relaxation of reinforcements) are neglected. The loading history is restricted to monotonically increasing static loads.

After each load increment, the stress and strain distributions will be calculated. The arrangement of steel components is kept flexible in order to allow the study of various types of reinforcements.

1.3 Historical and Literature Review

The successful application of matrix analysis methods to materially nonlinear framed structures by Wilson (2) in 1960, and Goldberg and Richard (3) in 1963 demonstrated the feasibility of the finite element method for the solution of nonlinear problems. Wilson subsequently extended the incremental load procedure to a class of two-dimensional, nonlinear structures (4) in 1963. In the same report an iterative technique similar to the Newton-Raphson Method was applied to in-plane loaded thin plates with bilinear constitutive relations.

Argyris (5) and Denke (6), in 1964, adapted the matrix force method to elasto-plastic problems. Comprehensive presentations of the elasto-plastic displacement method were given by Pope (7) in 1965 for plane stress and plane strain states and by Argyris (8, 9) for three-dimensional states of stress. Both publications distinguish clearly between the two basic incremental procedures referred to as "Initial Strain Method" and "Tangent Modulus Method."

The "Initial Strain Method" was developed in matrix form by Argyris (8, 9). It involves approximating the change in plastic strain during each load increment. These plastic strains are then used as initial strains to reevaluate the stress distribution. Therefore, this procedure requires iterations in each loading step.

The "Tangent Modulus Technique" makes use of incremental stiffness matrices which are derived from well-known incremental stress-strain relations. For strain-hardening material, the stiffness matrices must be modified after each load increase. A partial stiffness method for elasto-plastic problems based on the "Tangent Modulus Approach" was first proposed by Marcal (10) in 1965 and later modified for use in the finite element method by Marcal and King (11). These papers state the necessary equations in matrix form and suggest the sequence of steps suitable for digital computation.

In his classical treatise, Zienkiewicz (12) presents an excellent summary of these fundamental matrix methods and also presents Wilson's "Direct Iterative Approach" as a third basic technique. The amount of research conducted in the area of nonlinear analysis by finite elements has increased rapidly since these initial efforts. Therefore, only the most significant publications pertinent to this study will be mentioned. In general, recent investigations have only refined the earlier formulations of the elasto-plastic problem.

Felippa's paper (13) can be considered as one of the early attempts to introduce refinements into the matrix methods for linear and nonlinear analysis of two-dimensional structures. Other planar problems were solved by Akyuz (14) and Akyuz and Merwin (15). Special attention was given in these publications to the computational difficulties arising from the repeated solution of simultaneous equations. A half-step method related to the Runge-Kutta procedure was applied to improve the accuracy. The comparative study by Marcal (16) in 1968 revealed that the Initial Strain Method fails for the case of elastic-perfectly-plastic material. Otherwise, the two incremental techniques were found to provide very similar results. Another contribution to the topic was presented by Marcal (17) in 1969. At the same time, Yamada (18) gave a general review of Japanese developments in the field of elasto-plastic matrix analysis. His paper contains an incremental stress-strain matrix for anisotropic materials and shows several practical applications of the step-by-step approach.

A variation of the Initial Strain Method based on known stress functions was proposed by Yamada et al. (19) in 1968. However, because the publication of their paper was delayed until 1969, it appears that Zienkiewicz, Valliappan and King (20) should earn full credit for the development of the so-called "Initial Stress Method." This new technique makes use of the fact that plastic strain increments prescribe uniquely the stress system, even in the case of an ideally plastic material. With this in mind, Zienkiewicz et al. were able to retain the advantages of the Initial Strain Method for which the matrix of elastic constants remains unchanged during the loading history. Probably the most comprehensive survey concerning nonlinear structural analysis techniques was made by Oden (21) in 1969. The main solution methods for both geometrically and materially nonlinear structures are discussed and presented in tensor form. Furthermore, the incremental stiffness approach first suggested by Pope (7) is generalized. The paper also includes an extensive list of selected references.

Despite the fact that finite element methods are highly suited for stress analyses, relatively few studies have adopted these techniques to investigate the behavior of concrete structures. Rashid (22) reported in 1966 the results of a two-dimensional finite element scheme used to analyze a prestressed concrete pressure vessel. In order to obtain a realistic model of this composite, heterogeneous, axisymmetric structure, three kinematically dissimilar elements were introduced to simulate the concrete, reinforcements and the steel liner. The program was later modified by Rashid (23) to include cracks in the concrete and the effects of plastic deformation in the steel components. As a special feature, Rashid proposed to treat the influence of a crack as a mechanism that changes the behavior of continuous elements from isotropic to orthotropic.

An alternative approach, suggested in 1967 by Scordelis and Ngo (24), introduced complete crack patterns by separating interelement boundaries. This study also included the simulation of bonds between reinforcement and concrete. Finally, the disadvantage of two-dimensional approximations has been overcome through the implementation of the "SAFE-3D" computer program, developed by Cornell et al. (25). This program was used by Corum and Krishnamurthy (26) to investigate a series of models of prestressed reactor vessels. It uses tetrahedral concrete elements, uniaxial bars, and triangular membrane steel components. As expected, the three-dimensional model provided much better results. However, a significant increase in computer time resulted from use of three-dimensional elements. Quite a different approach was taken by Cervenka (27) in his study entitled "Inelastic Finite Element Analysis of Reinforced Concrete Panels under In-Plane Loading." No individual reinforcement bars were considered. Instead, the total steel area was distributed over the quadrilateral element. The cracked state then could be visualized as a planar lattice structure.

1.4 Problem Approach

The solution method used herein is a combined iterative and step-by-step procedure based upon the matrix displacement method. The structure is analyzed as a plane stress problem. For each load increment, repeated elastic solutions are performed until the displacements meet a specified tolerance.

The mathematical model consists of an assemblage of triangular concrete plate elements, steel bar elements and bond links. The displacement fields are assumed to be linear for all three parts. The elastic constants (i. e., modulus of elasticity, Poisson's ratio, etc.) which are needed in the derivation of the elemental stiffness matrices are extrapolated from the pertinent uniaxial stress-strain curves. For all elements, these functions are approximated by piecewise linear polygons. The appropriate values of the material constants are found by entering the stress-strain diagram at the corresponding values of the principal strains.

A standard Gauss Elimination procedure is used to solve the equilibrium equations. Two computer programs were written to implement the method. Both provide stresses and strains in each element and the nodal displacements at all specified load levels.

A comparative study was made with the solutions presented by Scordelis for a simply supported beam (24). A second, more realistic problem was investigated to show the feasibility of the method to study crack propagation.

CHAPTER II

FINITE ELEMENT PROPERTIES

2.1 General

The finite element analysis of a continuum consists of three fundamental steps. First, the real structure is replaced by a suitable mathematical model. This is usually accomplished by dividing the original continuum into an assemblage of discrete elements. All elements are assumed to be interconnected at a discrete number of nodal points situated at the intersections of their fictitious boundaries. The second step is the formulation of the finite element characteristics.

In the matrix displacement approach, the material properties are described in the form of the elemental stiffness matrices. In recent years extensive research has been done in order to improve the various derivation procedures. Methods based upon energy theorems and related variational principles have been found to be the most satisfactory techniques. The foundation for such derivations is the assumption that an energy functional derived for the continuous system is equal to the same functional determined from the finite element model. The element properties can then be obtained by minimizing the functional through well-known variational methods.

Once the element properties of all the elements have been defined, the discrete system can be analyzed as a conventional structural problem. Hence, the last phase consists of a standard analysis of a structural system by means of suitable computer programs. In this chapter, the first two steps (i. e., the structural idealization and the evaluation of element characteristics) will be discussed. The mathematical models for concrete, steel reinforcements, and bond between the two materials are developed in section 2.4. The derivation of the solution procedure for linearly elastic problems and the necessary modifications for the nonlinear case will be presented in Chapter III.

2.2 Structural Idealization

In general, a reinforced concrete member must be considered as a three-dimensional, nonhomogeneous, nonisotropic, composite structure. The difficulties encountered in the solution of such structural systems have already been described in section 1.1. Clearly, a series of assumptions must be introduced in any solution procedure. The choice of suppositions is governed by the type of structure under consideration, the character of the results desired, and the numerical method utilized. In the case of reinforced concrete beams, simplifications concerning the type of structure are the most critical group. In order to obtain a reliable approximation, the model must include all physical constituents of the real composite structure. In addition, special attention should be given to the simulation of the interaction between the parts. A list of the necessary assumptions for the construction of a relevant model is set forth below.

2.2.1 General Assumptions

The following stipulations may be regarded as preparatory requirements for a possible application of the finite element method.

Real Structure	Assumed Structure
a. Three-dimensional;	a. Two-dimensional (of the plane stress type);
b. Nonhomogeneous components;	b. Homogeneous components;
c. Nonisotropic components;	c. Isotropic or orthotropic components;
d. Random change in structural configuration due to cracking;	d. Cracking predicted by principal tensile stresses in the concrete;
e. Continuous bond between concrete and steel reinforcements; bond-slip;	e. Discrete attachment between steel and concrete via bond links;
f. Influence of time-dependent effects such as creep and relaxation.	f. Neglected.

The next set of assumptions concerns the selection of the finite elements and their individual properties.

Figure 1 shows a typical, singly-reinforced concrete beam under an arbitrary static, in-plane loading condition. The finite element idealization relevant to this study is displayed in Figure 2 in an exaggerated view.

Three kinematically and geometrically dissimilar elements have been chosen as basic components of the model. The entire concrete

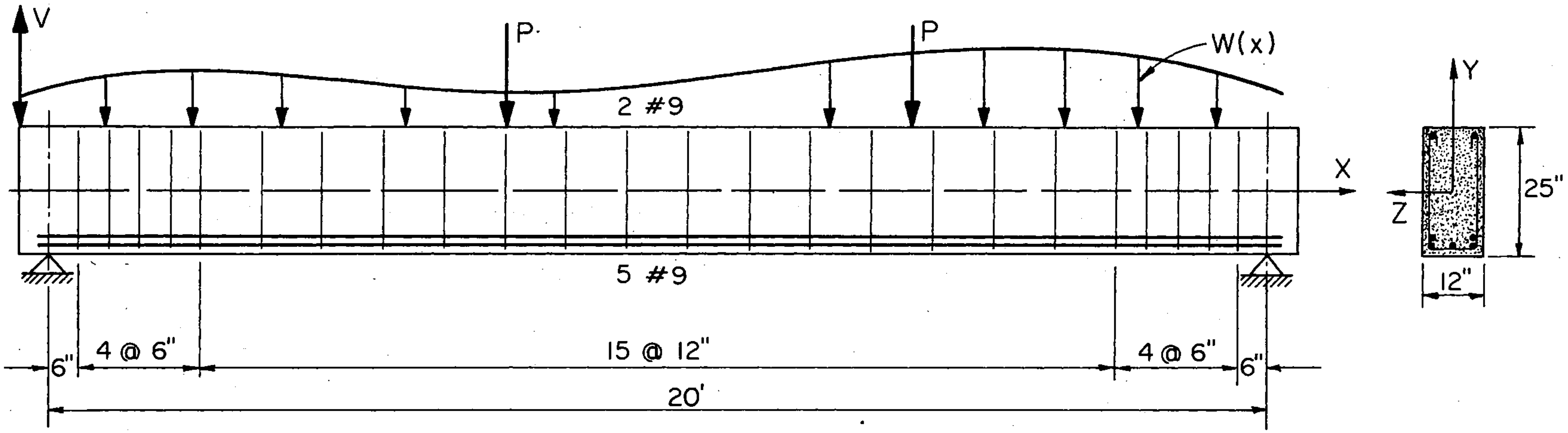


Figure 1. Example of a Singly Reinforced Concrete Beam

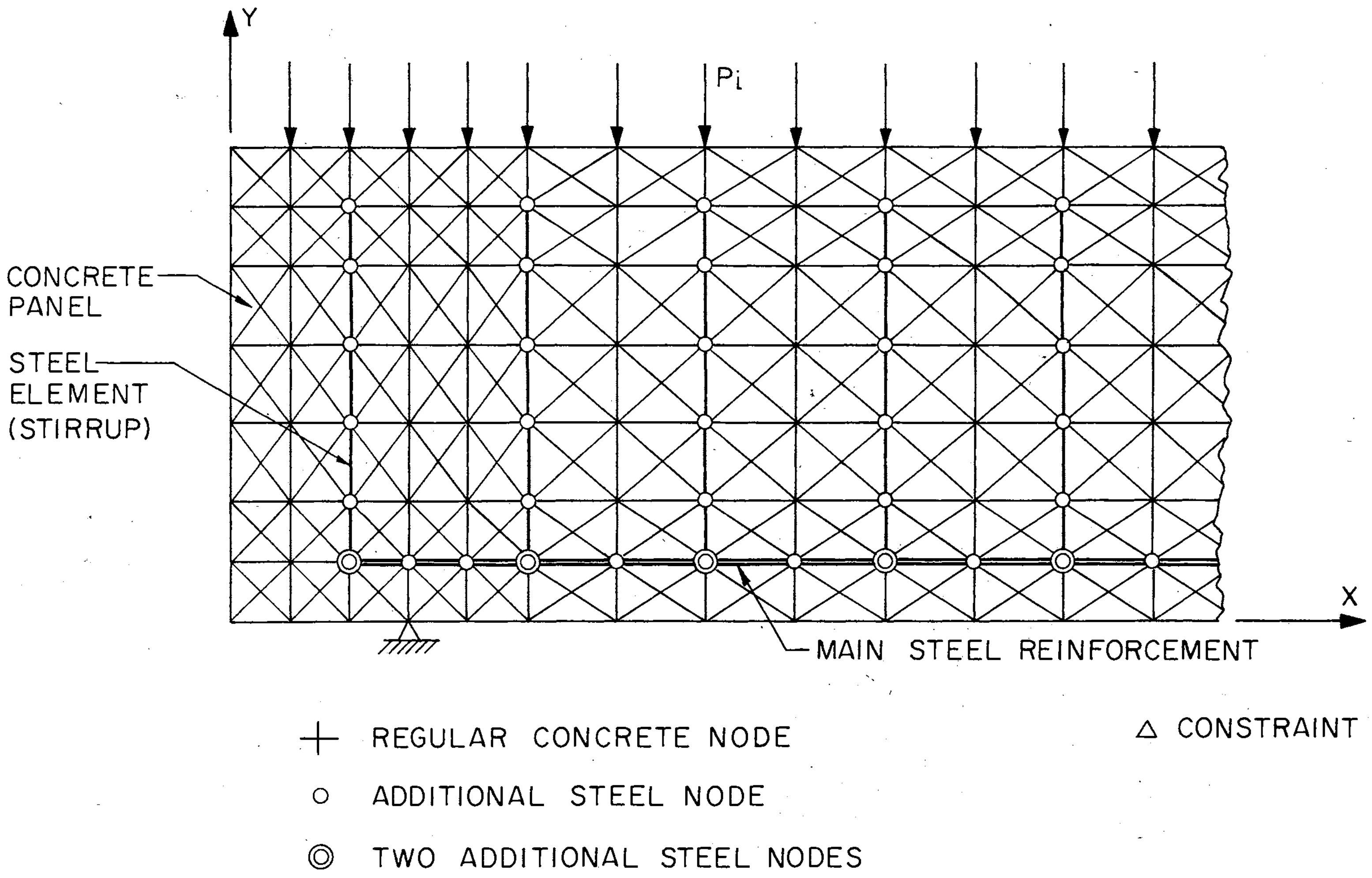


Figure 2. Finite Element Assemblage of a Singly Reinforced Beam

body is divided into flat, triangular panels. Combined with the steel segments (represented by "two-force" members), they constitute the material part of the composite structure. The complicated phenomenon of bond interaction between concrete and steel is simulated by a dimensionless connecting device, called a linkage element. According to Scordelis (24), these bond links can be conceptually thought of as linear springs. Both steel reinforcements and the connecting elements have been extracted in Figure 4 for illustrative purposes. In the real assemblage, the nodes of the steel bars and the connecting springs originally occupy the same geometrical position as their corresponding concrete joints. Therefore, these nodes have the same global coordinates. However, topologically they must be treated as separate joints.

2.2.2 Concrete Elements

The concrete body can be subdivided in a number of ways. The most commonly used configurations are triangular, rectangular, and quadrilateral meshes. Rectangular elements provide slightly better results. However, triangular panels are preferred for problems with irregular boundaries. In early publications the stiffness matrices were derived by the so-called direct approach (28). Recently, descriptions of a number of refined elements have been published as a result of the implementation of variational techniques. An excellent summary may be found in Reference 29. In this study, the traditional, constant-strain, triangular panel (Figure 3a) has been adopted for two reasons. First, it is desirable to decrease the size of elements in the vicinity of large stress gradients. A gradual change in size can easily be accomplished in the case of triangular panels. The second criterion for

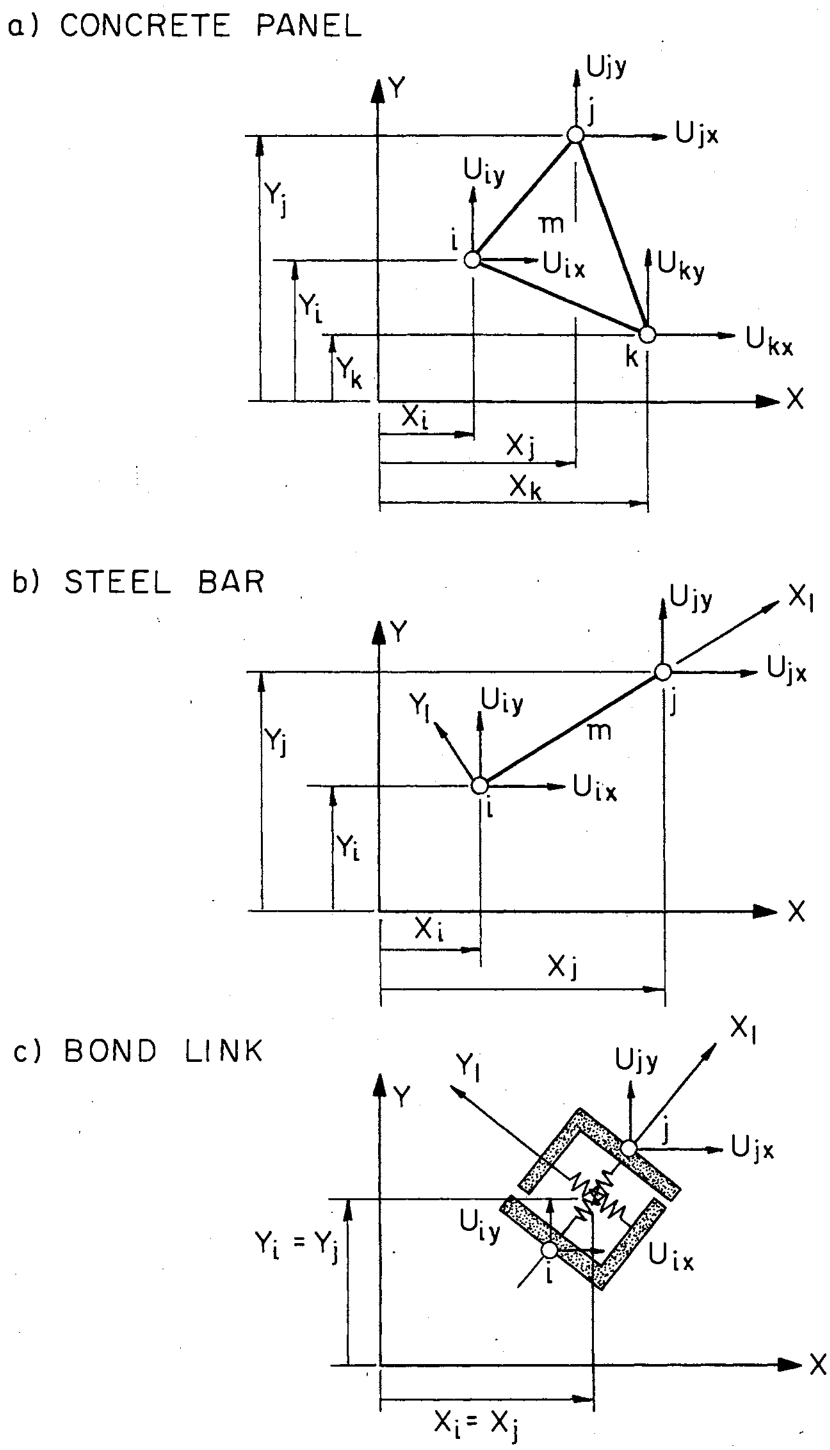


Figure 3. Components of Finite Element Model

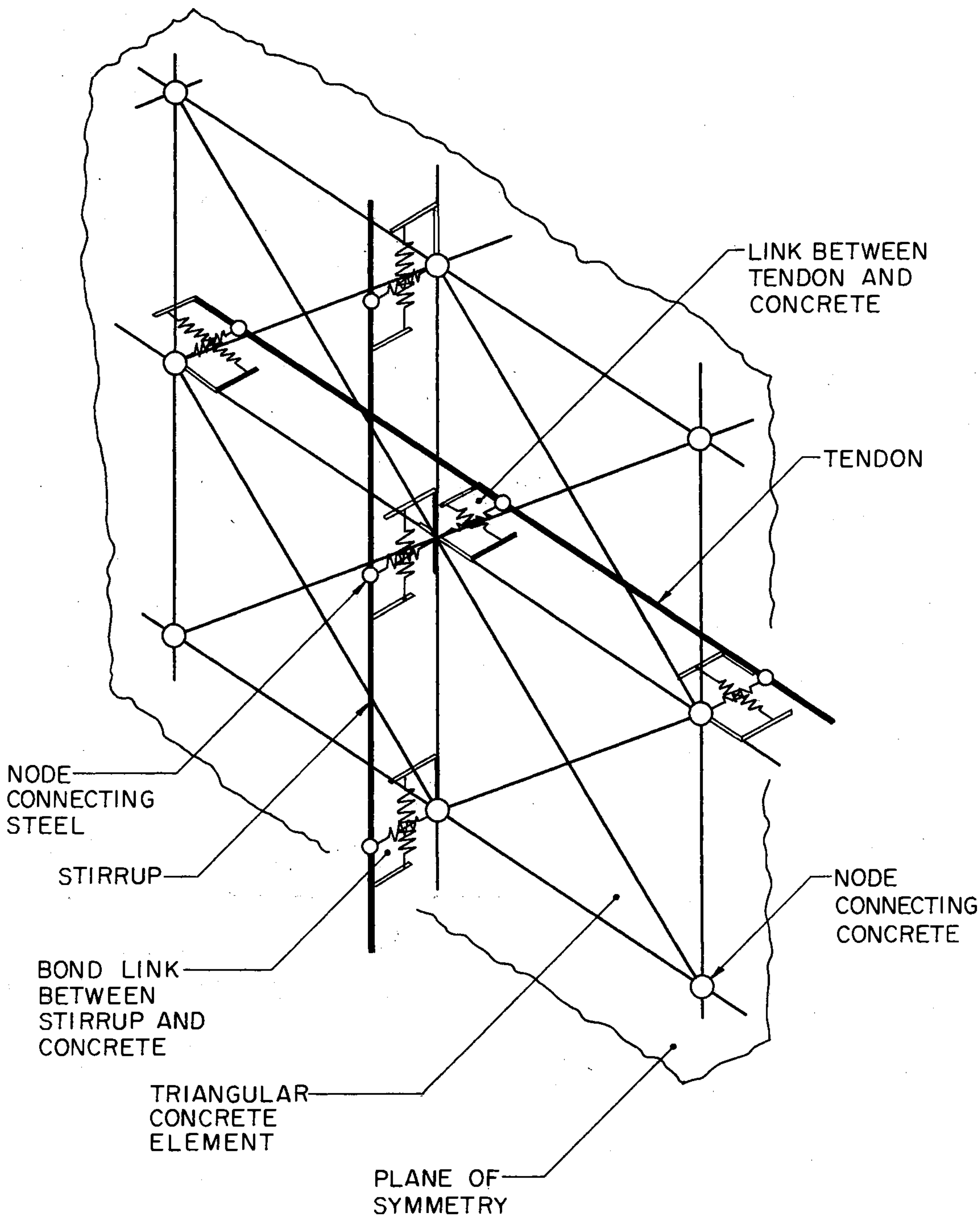


Figure 4. Configuration of Bond Links

selecting constant-strain elements is reflected by the fact that yielding takes place throughout the whole element. Elements with nonuniform stress distributions are subject to local yielding which results in additional complications in determining the state of stress.

2.2.3 Steel Elements

The reinforcement occupies a relatively small volume compared to that of the concrete. It is therefore justifiable to idealize the steel tendons by simple two-force members (Figure 3b). The triangular model used by Scordelis was abandoned mainly because the very small vertical reinforcements would require a large number of additional elements or extremely slender triangles which are known to behave unsatisfactorily (30).

2.2.4 Bond Links

To account for bond slip, the steel must be attached to the concrete by a special connection mechanism. The bond link (Figure 3c) is designed to allow for relative displacements between the steel bars and the concrete panels. As pointed out earlier, these elements are dimensionless because only their mechanical properties are of importance. Nevertheless, additional nodes must be provided to permit relative displacements between adjacent concrete and steel joints.

2.2.5 Displacement Functions

After the shape of an element has been chosen, all geometric relations can be established. The next logical step is to decide upon a suitable displacement function representing the deformation of the element. It should be noted that the degree of approximation which can be achieved depends very heavily on the element shape and the chosen deflection pattern. To ensure convergence, the assumed displacement function should resemble the real displacement distribution. According to Zienkiewicz (12), good deflection functions are obligated to satisfy the following five requirements:

1. Internal and interelement compatibility;
2. Linear dependence on nodal displacements;
3. Inclusion of rigid body displacements;
4. Uniform strain state;
5. Independence of the external frame of reference.

For all three elements utilized in this study, these criteria are satisfied by assumed linear displacement functions of the form

$$\left.\begin{aligned} u_x &= c_1 x + c_2 y + c_3 \\ u_y &= c_4 x + c_5 y + c_6 \end{aligned}\right\} \quad \text{for concrete panels} \qquad (2.1)$$

$$u_1 = c_7 x + c_8 \qquad \text{for steel and bond elements} \qquad (2.2)$$

It can easily be shown that the assumed displacements vary linearly along the edges of the concrete panels and that they depend only on the displacement of the two vertices on that particular edge (31). This ensures displacement compatibility along the common boundary of two triangular elements.

The assumed linear deflection pattern for the "two-force" steel bars results not only in a compatible but also in an exact strain distribution, since the elements are one-dimensional.

On the basis of these chosen deformation functions, the kinematic relations (i. e., the strain-displacement equations) are derivable through ordinary differentiation. To complete the preparations for the development of the stiffness matrices, the material laws for each element will be reviewed in the next section.

2.3 Constitutive Relations

2.3.1 General

The behavior of a material is characterized in the way it deforms under an imposed stress condition. It is therefore customary to express the material laws in the form of stress-strain curves. Two typical plots for a uniaxial, stress condition of nominal stress versus conventional strain for mild steel and concrete are shown in Figure 5. Both curves illustrate the complex, nonlinear character of the constitutive relations. Actually, the material characteristics can become even more complicated if the effects of time and temperature upon the rate of change of strains are included. It is therefore necessary to replace these empirical curves by mathematically defined expressions. The selection of an idealized stress-strain relationship depends upon several factors such as the nature of the problem, the kind of material, the type of load, desired accuracy, etc.

The most commonly used expression is the simple idealization known as Hooke's Law. In matrix form,

$$\{\sigma\} = [D]\{e\} \qquad (2.3)$$

The $\{e\}$ is the vector of total strains, $\{\sigma\}$ designates the stress vector, and $[D]$ is a square matrix containing the elastic constants. This linear relationship is, of course, very popular in engineering practice; however, its restriction to linearly elastic behavior must be remem-

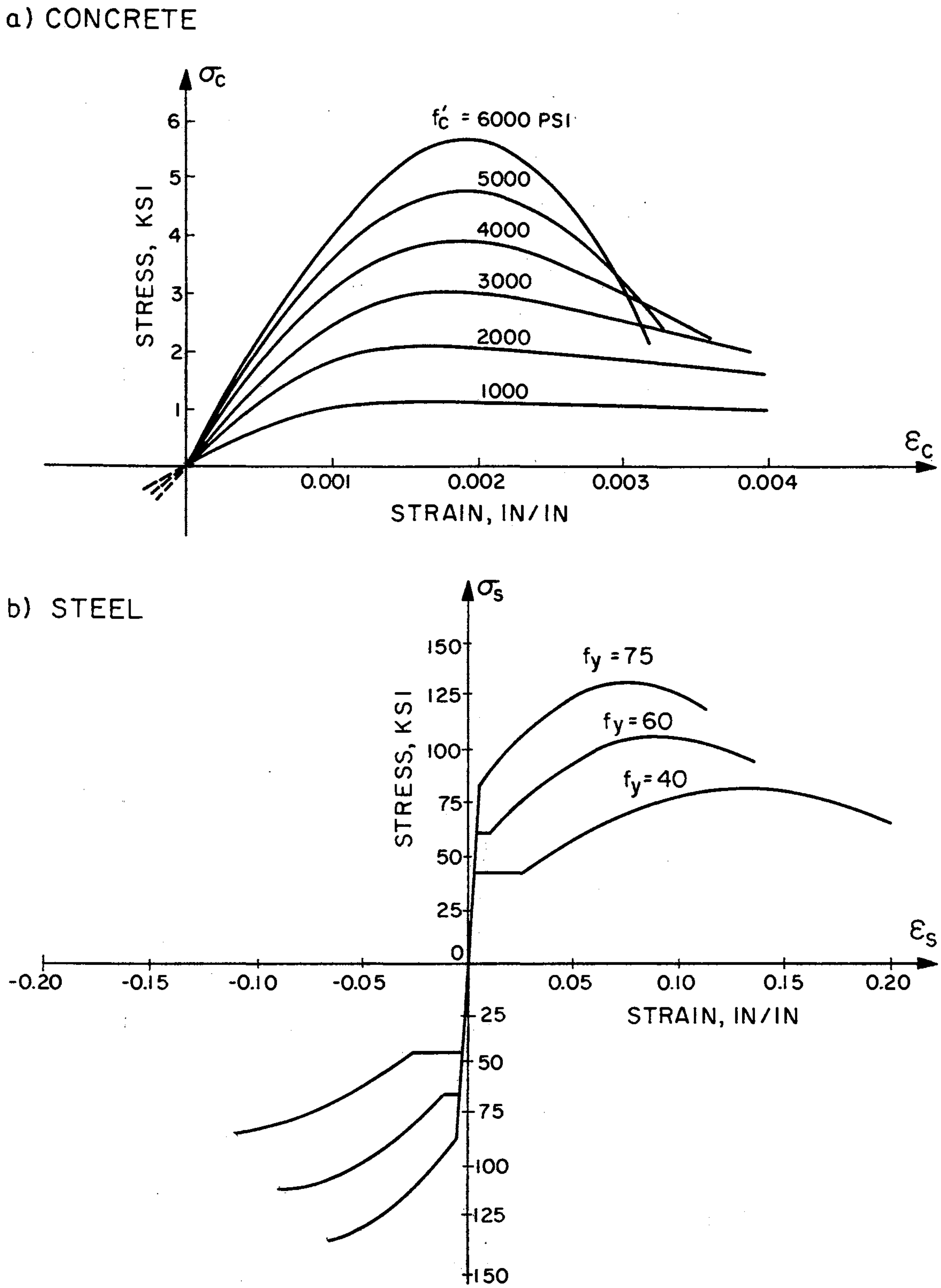


Figure 5. Typical Stress-Strain Curves for Steel and Concrete

bered. In an investigation concerning the nonlinear behavior, the entire stress-strain curve will be needed for the evaluation of the element characteristics. Analytical inelastic theories require that the constitutive relations may be replaced by reasonably simple continuous functions. The Ramberg-Osgood Law and the Bi-Linear Law are typical examples of such expressions (32). If more accurate idealizations are desired, the possibility of using a curve-fitting scheme always exists. In a numerical procedure, on the other hand, empirical data may be used directly in table-form. Values between discrete data points are easily calculated by means of suitable interpolation formulas. The approach adopted in this study is based upon linear interpolation; i. e., the stress-strain curves are replaced by a polygon (Figure 6).

After the idealized constitutive relations have been established, the elastic constants are available at any load level. Since the matrix $\left[D\right]$ contains "elastic constants" only, its derivation appears to be a straightforward procedure. This is true for the elastic interval. However, the evaluation of $\left[D\right]$ in the inelastic range presents some difficulties because of the biaxial state of stress in the concrete.

By methods well known from strength of materials, the biaxial state of stress can be reduced to two principal stresses acting at right angles to each other on an appropriately oriented elementary cube. Either, or both, of the principal stresses can be tension or compression.

In most cases only the uniaxial stress-strain relation of concrete is known from simple tests. To predict the elastic constants for a structure under a combined stress situation, it is necessary to relate the material properties to the uniaxial test parameters. Six different

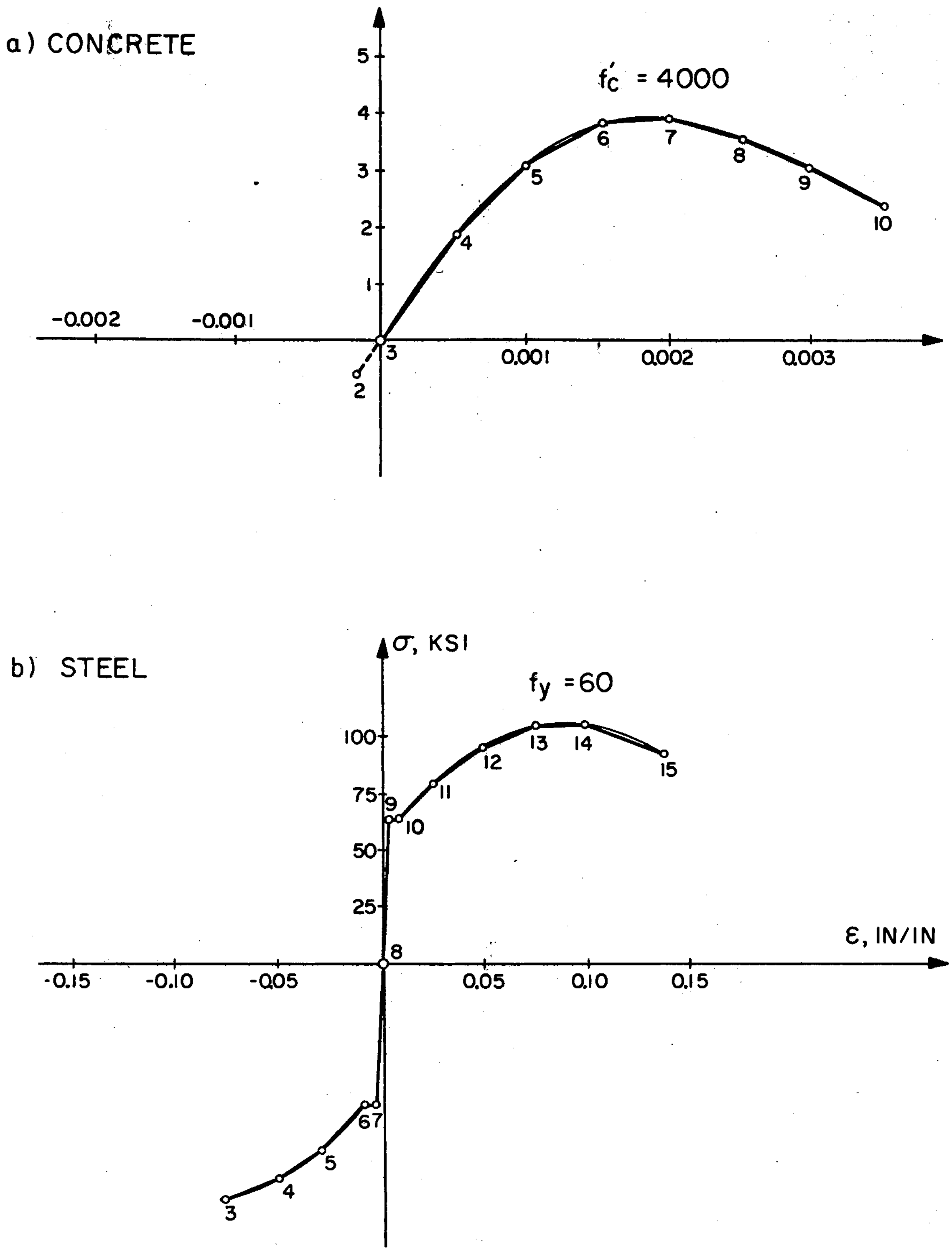


Figure 6. Idealized Stress-Strain Curves

quantities (i. e., maximum principal stress, maximum shearing stress, maximum strain, total strain energy, strain energy of distortion, and octahedral shearing stress) are available to compare the multiaxial state of stress with a tensile specimen. When the specimen starts to yield (or fracture), all six quantities reach their limiting values simultaneously. In members under biaxial or triaxial states of stress, the limits usually do not occur at the same time. Since the type of failure of a concrete member is dependent upon many variables (i. e., state of stress, shape and size of structure, type and duration of loading, etc.), it is extremely difficult to choose the proper failure criterion. In spite of extensive and continuing research, no reliable theory for the selection of the proper failure mode has yet emerged. The highly nonhomogeneous nature of concrete and the phenomenon of microcracking are possible reasons for the insufficient reliability of these theories. The most commonly used criteria are the Maximum-Tension-Stress, the Mohr, and the Octahedral Shear Stress theories (33).

2.3.2 Proposed Idealization of Stress-Strain Characteristics

Concrete. In order to arrive at the proper $\left[D\right]$ matrix for concrete, this study used the following approach. Consider an element (shown in Figure 7) under an arbitrary strain condition ϵ_u and ϵ_v. From the stress-strain curve, Figure 8, it appears that two different Young's moduli, E_u and E_v, can be associated with the strains ϵ_u and ϵ_v, respectively. The material behavior is obviously different in these two directions; in other words, the structure may be thought of as anisotropic. It should be noted that this anisotropy is different from the term used in the theory of elasticity. There, an anisotropic body

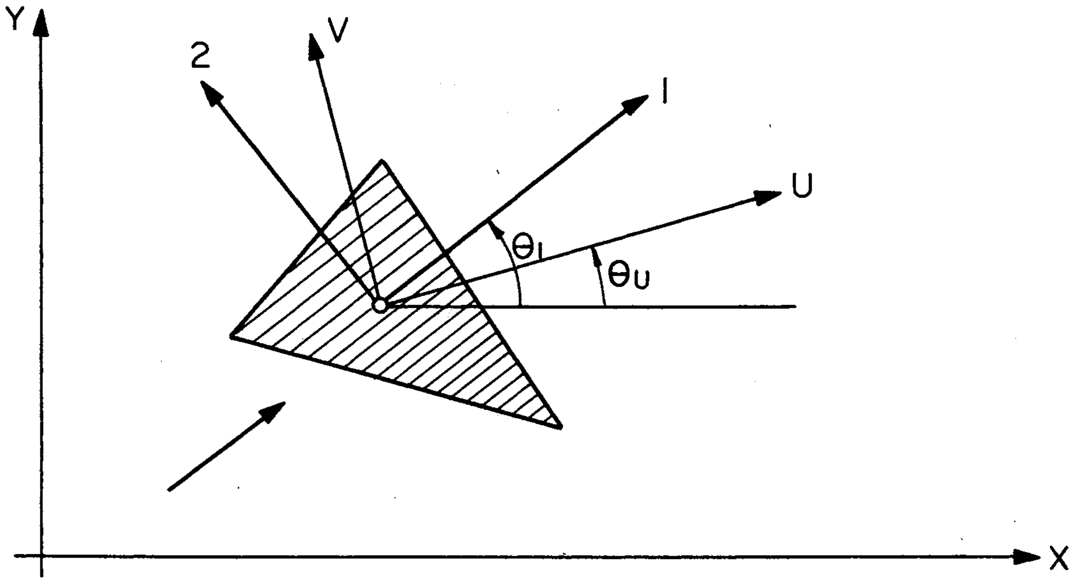


Figure 7. Anisotropic Triangular Element

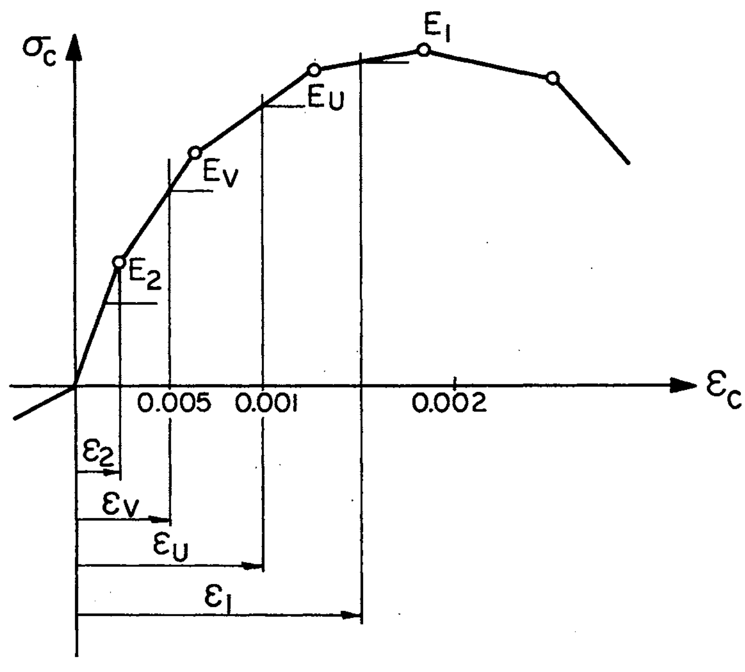


Figure 8. Material Constants for Anisotropic Element

is defined as a continuum with different values for E in at least two distinct directions. However, the elastic constants at each point in the structure are the same for one particular direction. Here, the behavior is assumed to change with the state of stress or strain at a point. If ϵ_u and ϵ_v are known, the corresponding values for the moduli of elasticity can be determined from the uniaxial stress-strain curve.

The constitutive relations for an anisotropic body are, in general, of the form

$$\{\sigma\} = [D_g] \{e\} \tag{2.3a}$$

where $[D_g]$ is a symmetric matrix containing six independent, non-zero constants. For the principal axes of anisotropy, they reduce to four independent coefficients. The material is then referred to as "orthotropic" with respect to the axes u and v. Once again, the standard definition of orthotropy does not apply to the structures considered in this study.

For the principal axes of anisotropy, the constitutive equations become

$$\begin{aligned} \sigma_u &= \frac{1}{1 - \nu_{uv}\nu_{vu}} (E_u \epsilon_u + \nu_{vu} E_u \epsilon_v) \\ \sigma_v &= \frac{1}{1 - \nu_{uv}\nu_{vu}} (\nu_{uv} E_v \epsilon_u + E_v \epsilon_v) \qquad (2.4) \\ \sigma_{uv} &= G_{uv} \epsilon_{uv} \end{aligned}$$

Under the assumption that the structure behaves locally as an orthotropic structure, one can assume that the principal axes of stress and strain coincide. Furthermore, these axes (1 and 2 in Figure 7) are taken as principal axes of anisotropy. With E_1, ν_{21} being the material constants in the direction of the first principal axis and E_2, ν_{12} being the values for the second, Equation (2.4) now reduces to

$$\sigma_1 = \frac{1}{1 - \nu_{12}\nu_{21}} (E_1\epsilon_1 + \nu_{21}E_1\epsilon_2)$$

$$\sigma_2 = \frac{}{1 - \nu_{12}\nu_{21}} (\nu_{21}E_2\epsilon_1 + E_2\epsilon_2) \qquad (2.5)$$

$$\sigma_{12} = 0$$

Once the principal strains have been calculated, the corresponding maximum stresses may be evaluated directly as

$$\{\sigma\} = [D_a] \{e\} \qquad (2.6)$$

where

$$[D_a] = \frac{1}{1 - \nu_{12}\nu_{21}} \begin{bmatrix} E_1 & \nu_{21}E_1 & 0 \\ \nu_{12}E_2 & E_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad (2.7)$$

It must be kept in mind that such an approach is contingent upon the assumption that the principal axes exist. The conditions for the existence of principal directions are stated in Reference 34. Since the computer solution does not calculate the stresses in any other but the principal directions, no stress transformations are performed. Thus, the transformation of elastic constants for new coordinate systems can be omitted, and transformation of coordinate systems is performed on the entire element stiffness matrix.

Finally, it should be noted that Equation (2.5) is subject to an additional condition. The four material constants, E_1, E_2, ν_{12} and ν_{21}, are not independent. The additional relation may be obtained by comparison of the total work done on a differential element for two different loading sequences. The resulting supplementary equation relates the Poisson's ratios to the moduli of elasticity as follows:

$$\frac{\nu_{12}}{\nu_{21}} = \frac{E_2}{E_1} \tag{2.8}$$

Substituting Equation (2.8) into Equation (2.5) yields the constitutive laws

$$\begin{aligned} \sigma_1 &= \frac{E_1}{1 - \nu_{12}\nu_{21}} (\epsilon_1 + \nu_{21}\epsilon_2) \\ \sigma_2 &= \frac{E_1}{1 - \nu_{12}\nu_{21}} (\nu_{21}\epsilon_1 + n\epsilon_2) \\ \sigma_{12} &= 0 \end{aligned} \tag{2.9}$$

where

$$n = \frac{E_2}{E_1} \tag{2.10}$$

In matrix form

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ 0 \end{bmatrix} = \frac{E_1}{1 - \nu_{12}\nu_{21}} \begin{bmatrix} 1 & \nu_{21} & 0 \\ \nu_{12} & n & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ 0 \end{bmatrix} \tag{2.11}$$

or $\{\sigma\}$ $[D_a]$ $\{\epsilon\}$

<u>Steel and Bond Links</u>. Since in this study the steel elements will be considered as two-force members only, the uniaxial stress-strain curve may be used directly to determine the material properties. It should be mentioned that any other element would require more sophisticated tools for the evaluation of the material constants. A

proper yield criterion and plastic stress-strain relations, such as the Prandtl-Reuss equations (32) would have to be adopted.

Here the incremental stress $\Delta\sigma_x$ in the longitudinal direction is calculated by

$$\Delta\sigma_x = E_s \Delta\epsilon_x \tag{2.12}$$

The appropriate modulus of elasticity is read from the uniaxial stress-strain curve for steel at the location of the total strain ϵ_x. In the case of an elastic-perfectly-plastic material, the incremental stresses will become zero beyond the yield stress. However, the total stress σ_x is still available from the stress-strain curve.

A similar situation exists for the bond links. Again, the uniaxial stress-strain relations provide the material constant E_b directly. The same approach as used for steel members yields the stress increments $\Delta\tau_b$ and the total bond stress τ_b at any load level.

$$\Delta\tau_b = E_b \Delta_r \tag{2.13}$$

Δ_r denotes the relative displacement between a steel and the corresponding concrete node. Figure 9 displays some possible stress-relative displacement curves (from Reference (35)).

2.4 Development of Element Stiffness Matrices

2.4.1 General

A number of alternative methods are available for the calculation of element stiffness matrices. The variational approach based on the principle of minimum potential energy is adopted here. Since these methods are well established, a comprehensive repetition of the

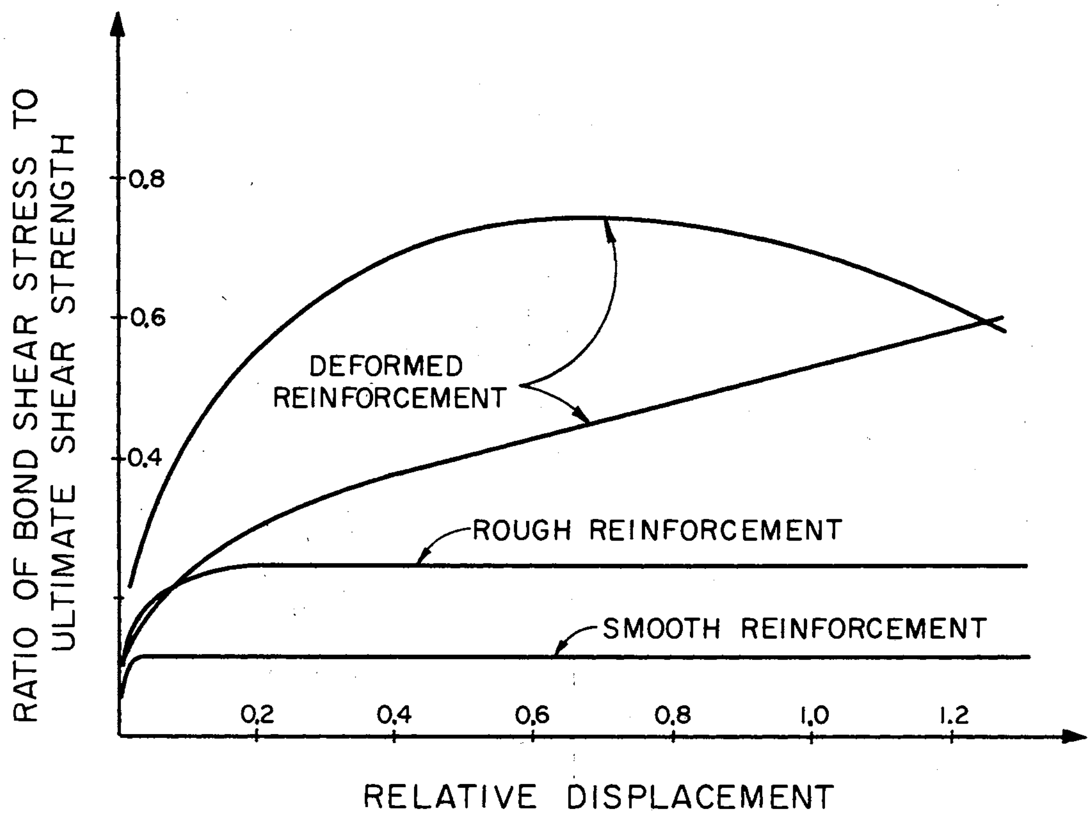


Figure 9. Stress-Relative Displacement Relations for Bond

procedure is omitted. References (31) and (36) contain excellent introductions to the variational treatment of the energy methods.

2.4.2 Matrix Formulation for the Plane Stress Case

The first step in determining the properties of the idealized element is to assume that the interior displacements $\{u\}$ at any point are expressible in terms of the nodal displacements $\{U\}$ by a set of equations given as

$$\{u\} = [a] \{U\} \tag{2.14}$$

$[a]$ is a rectangular matrix which is a function of the coordinates of the point under consideration. For discrete element systems, the matrix $[a]$ is an approximate expression. The total strain distribution $\{e\}$ ($\{e\}$ may include initial strains $\{\epsilon_o\}$) within a particular element is obtained by differentiating Equation (2.14) which leads to the matrix equation

$$\{e\} = \{\epsilon\} + \{\epsilon_o\} = [b] \{U\} \tag{2.15}$$

This expression replaces the kinematic relations used in the ordinary theory of elasticity.

Under the assumption that a unique matrix $[b]$ exists, the stresses may be determined from any conceivable constitutive relationship of the form

$$\{\sigma\} = [D] \left(\{e\} - \{\epsilon_o\}\right) \tag{2.16}$$

By substitution of Equation (2.15) into Equation (2.16)

$$\{\sigma\} = [D] [b] \{U\} - [D] \{\epsilon_o\} \tag{2.17}$$

$\{\sigma\}$, of course, represents the three stress components, σ_x, σ_y, σ_{xy} and $[D]$ is a square matrix containing the elasticity constants, E, G, ν, etc.

It is now possible to express the total energy functional in matrix form

$$\Pi = \sum_{n=1}^{N} \iiint_{V_n} \left(\frac{1}{2} \{e_n\}^t \{\sigma_n\} - \{f_{b,n}\}^t \{u_n\} \right) dV - \sum_{n=1}^{N} \iint_{S_{\sigma_1 n}} \{p_n\}^t \{u_n\} dS \tag{2.18}$$

where

$f_{b,n}$ = prescribed body forces

n = element index

p_n = prescribed surface tractions

V_n = volume of element n

$S_{\sigma,n}$ = portion of element surface over which the surface tractions p_n are prescribed.

Substituting Equations (2.15) and (2.17) into Equation (2.18) yields the required expression for Π.

$$\Pi = \sum_{n=1}^{N} \iiint_{V_n} \left(\frac{1}{2} \{U_n\}^t [b_n]^t [D_n] [b_n] \{U_n\} - \{U_n\}^t [b]^t [D_n] \{\epsilon_o\} - \{U_n\}^t [a]^t \{f_{n,o}\} \right) dV - \sum_{n=1}^{N} \iint_{S_{\sigma_1 n}} \{U_n\}^t [a]^t \{p_n\} dS \tag{2.19}$$

Application of the principle of minimum potential energy

$$\delta\Pi = 0 \tag{2.20}$$

to Equation (2. 19) will result in the desired stiffness matrix $\left[K\right]$

$$\left[K\right] = \iiint \left[b_n\right]^t \left[D_n\right] \left[b_n\right] dV \tag{2.21}$$

and three equivalent force vectors due to initial strain conditions, prescribed body forces and surface tractions (29).

Consideration will now be given to the three specific elements as shown in Figure 3. If the strain-displacement transformation matrix $\left[b_n\right]$ and the matrix of elastic constants $\left[D_n\right]$ are known, the stiffness matrix $\left[K_n\right]$ can be determined by evaluation of expression (2. 21).

2.4.3 Triangular Concrete Panels

It is possible in this case to obtain the stiffness matrix $\left[K_n\right]$ directly in terms of global coordinates x and y. The assumed displacement function will be taken as

$$\begin{aligned} u_x &= c_1 x + c_1 y + c_3 \\ u_y &= c_4 x + c_5 y + c_6 \end{aligned} \tag{2.22}$$

The six arbitrary constants, $c_1, \ldots c_6$, result from six boundary conditions involving the three vertices of the triangle, Figure 10. Upon substitution of the vertex coordinates into Equation (2. 22), the displacement functions are obtained as follows:

$$\begin{aligned} u_x(x, y) = \frac{1}{2A_n} &\left[y_{32}(x - x_2) - x_{32}(y - y_2)\right] U_1 \\ &+ \left[-y_{31}(x - x_3) + x_{31}(y - y_3)\right] U_3 \\ &+ \left[y_{21}(x - x_1) - x_{21}(y - y_1)\right] U_5 \end{aligned} \tag{2.23a}$$

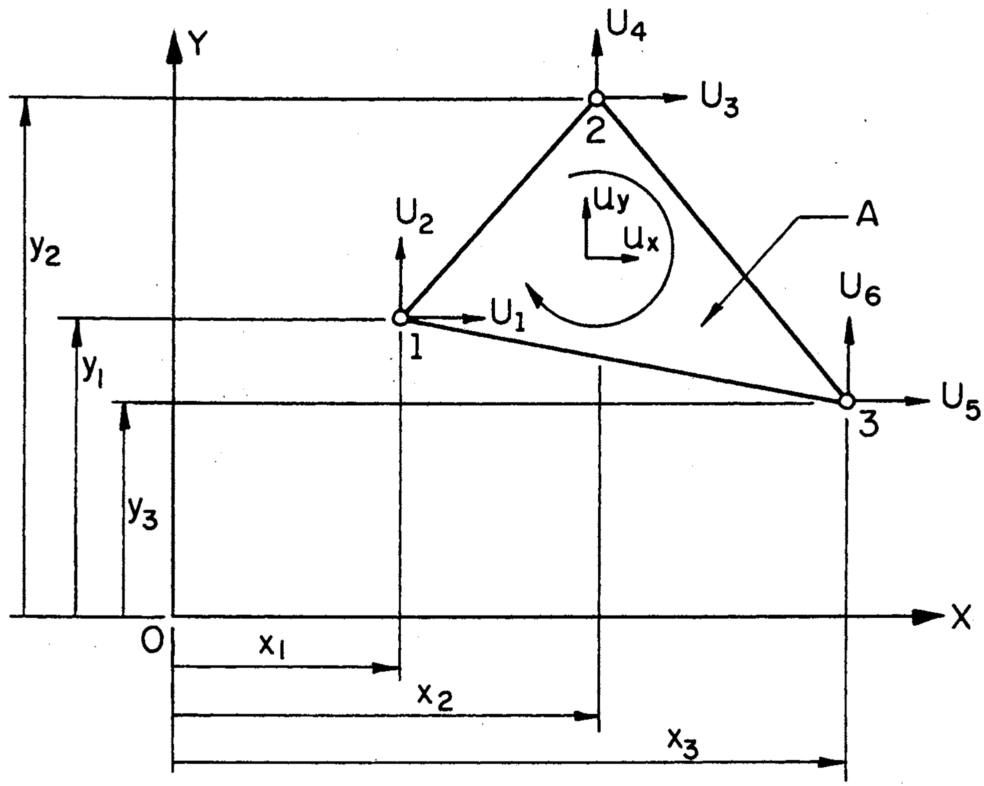


Figure 10. Arbitrary Triangular Concrete Element

$$u_y(x, y) = \frac{1}{2A_n} \left[y_{32}(x - y_2) - x_{32}(y - y_2)\right] U_2 + \left[-y_{31}(x - x_3) + x_{31}(y - y_3)\right] U_4 + \left[y_{21}(-x_1) - x_{21}(y - y_1)\right] U_6 \tag{2.23b}$$

where

$$A_n = x_{32}y_{21} - x_{21}y_{32} \tag{2.24}$$

and

$$x_{ij} = x_i - x_j \qquad y_{ij} = y_i - y_j \tag{2.25}$$

Thus, matrix $\left[b_n\right]$ is a function of the vertex coordinates only and therefore is unique. The strain-displacement transformation matrix is obtained by differentiating Equation (2.23). Hence,

$$\begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_{xy} \end{bmatrix} = \frac{1}{2A_n} \begin{bmatrix} y_{32} & 0 & -y_{31} & 0 & y_{21} & 0 \\ 0 & -x_{32} & 0 & x_{31} & 0 & -x_{21} \\ -x_{32} & y_{32} & x_{31} & -y_{31} & -x_{21} & y_{21} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \end{bmatrix} \tag{2.26}$$

or

$$\left\{e\right\} = \left[b_n\right] \left\{U\right\} \tag{2.27}$$

The assumption of linear displacement functions results, in this particular case, in a constant strain field. The compatibility equations are therefore satisfied within each element. Furthermore, displacements along the interelement boundaries are linear functions of the corresponding vertices and are identical for adjacent edges.

Since the elements under consideration are of unit thickness, the expression (2. 21) reduces to

$$\left[K_n\right] = \iint \left[b_n\right]^t \left[D_n\right] \left[b_n\right] dxdy \qquad (2.28)$$

the integration being carried over the area of the triangle. Both matrices, $\left[b_n\right]$ and $\left[D_n\right]$, are independent of the integration parameters and can be taken out of the integral sign. The integration then simply reduces to a matrix product of the form

$$\left[K_n\right] = \left[b_n\right] \left[D_n\right] \left[b\right] (1) A \qquad (2.29)$$

where A denotes the area of the triangle 1, 2, 3.

The resulting stiffness matrices for different matrices $\left[D_n\right]$ are tabulated in Appendix A.

2.4.4 Steel Bars

The derivation of $\left[K_n\right]$ for the linear steel elements is considerably less involved. Only one displacement function in the direction of the member axis is needed. It has been mentioned before that the assumption of linear functions of the form

$$\bar{u}_{x_1} = \bar{U}_1 + \frac{x_1}{L}(\bar{U}_3 - \bar{U}_1) \qquad (2.30)$$

will provide the exact strain distribution. $\bar{U}_1$ designates the displacements in local coordinates, Figure 11.

Upon differentiation of Equation (2. 30), the longitudinal strain ϵ_{x_1} becomes

$$\epsilon_{x_1} = \frac{\partial \bar{u}_{x_1}}{\partial x_1} = \frac{1}{L}(\bar{U}_3 - \bar{U}_1) \tag{2.31}$$

$$\begin{bmatrix} \epsilon_{x_1} \\ \epsilon_{y_1} \end{bmatrix} = \frac{1}{L} \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{2.32}$$

and therefore,

$$\left[b_s\right] = \frac{1}{L} \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \tag{2.33}$$

For one-dimensional elements, $\left[D_s\right]$ reduces to one term, E_s. The stiffness matrix then, after integrating over the length, can be written as

$$\left[K_s\right] = \frac{AE_s}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.34}$$

Young's modulus, naturally, must be chosen according to the prevailing strain condition in the member.

Although the stiffness matrix could be stated in the datum coordinate system directly, it is more convenient to develop the relations in local coordinates first and subsequently rotate the entire matrix into the global axes. According to Reference (2), the appropriate transformation is expressed by the matrix equation

$$[K]_g = [T]^t [K]_L [T] \tag{2.35}$$

where $[T]$ is an orthogonal matrix which relates the nodal displacements $\{U\}$ in the global system to the local deflections $\{\bar{U}\}$ in the following manner. θ denotes the angle between the local and global x-axis (Figure 11).

$$\{\bar{U}\} = [T] \{U\} \tag{2.36}$$

$$[T] = \left[\begin{array}{cc|cc} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ \hline 0 & 0 & \cos\theta & \sin\theta \\ 0 & 0 & -\sin\theta & \cos\theta \end{array}\right] \tag{2.37}$$

2.4.5 Bond Links

Finally, the derivation of the bond link stiffness matrix follows closely the procedure set forth in section 2.4.4. Consider a linkage element oriented at an arbitrary angle θ relative to the global axes x and y, Figure 12. Let the springs in the x_1 and y_1 directions have stiffness coefficients k_1 and k_2. Hence, the stress-strain relations in matrix notation simply become

$$\{\sigma\} = [D_b] \{\epsilon\} \tag{2.38}$$

$$\begin{bmatrix} \sigma_{x_1} \\ \sigma_{y_1} \end{bmatrix} = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} \epsilon_{x_1} \\ \epsilon_{y_1} \end{bmatrix} \tag{2.39}$$

where ϵ_{x_1} and ϵ_{x_2} are the relative displacements between the adjacent steel and concrete nodes. It can easily be verified that the strain-

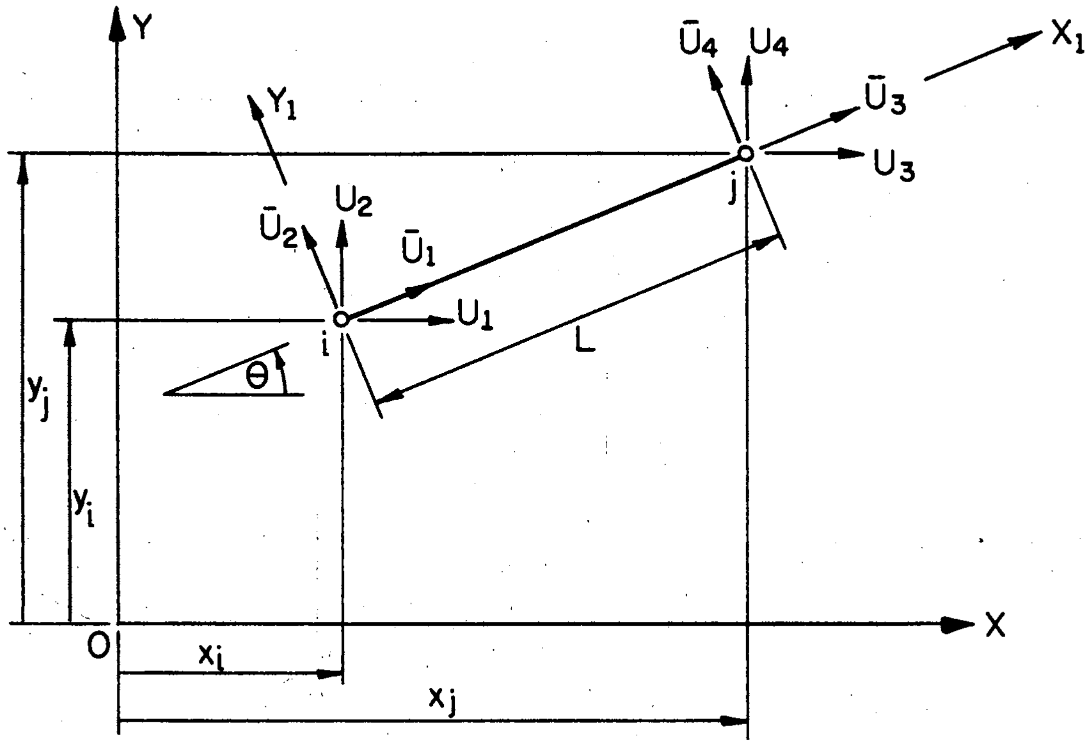


Figure 11. Steel Bar Elements

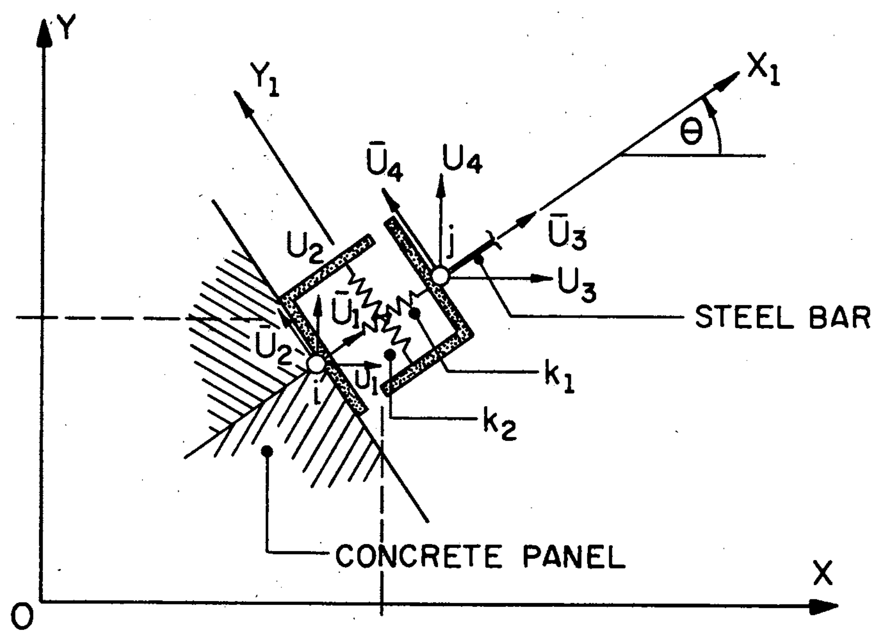


Figure 12. Bond Link

displacement relations

$$\epsilon_{x_1} = \overline{U}_3 - \overline{U}_1$$
$$\epsilon_{y_1} = \overline{U}_4 - \overline{U}_2 \qquad (2.40)$$

yield a $\left[b_b\right]$ matrix identical to Equation (2.33). Because of the conceptual similarity between the steel and bond elements, the remaining steps are analogous to those in section 2.4.4 and need not be repeated.

Scordelis has found in his study of reinforced concrete beams (24) that this type of bond mechanism simulates the interaction between concrete and steel quite accurately. It should be mentioned that the linkage element neglects frictional bond, local stress concentrations along the ribs of deformed bars and dowel action.

Research has shown that the redistribution of compressive stresses at the ribs of bars may cause small tensile stresses in the concrete (37). However, for all standard deformed bars the concrete is capable of sustaining these local disturbances. Therefore, these effects are neglected in the design. Frictional bond may be significant, especially near cracks; but it is again neglected since the coefficient of friction is extremely difficult to predict.

Dowel action is usually significant in the corners of bent or curved reinforcements. This study considers straight bars only. Therefore, a bond link which does not account for this effect is justified.

2.4.6 Cracked Concrete Element

It is well known that the tensile strength of concrete is only a fraction of its compressive strength. This rather unpleasant property leads to cleavage failure (tension cracking) at relatively small loads.

At any load larger than that which causes the concrete to crack, the reinforcements are called upon to resist the entire tensile force. This type of behavior plays an important role in the nonlinear analysis of reinforced concrete.

In the solution procedure presented here, the influence of a crack on a continuous triangular concrete element is treated in a similar way as proposed by Rashid (23) in 1968. The element is cut in the direction perpendicular to the principal tensile stress σ_1. In this new state the element no longer has any stiffness normal to the crack surface (Figure 13).

Consequently, the concrete may be considered as a uniaxial stress condition parallel to the second principal axis. This assumption results in the following stress-strain relationship:

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & E_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ 0 \end{bmatrix} \tag{2.41}$$

or, in matrix formulation,

$$\{\sigma\} = [D_{cr}] \{\epsilon\} \tag{2.42}$$

The stiffness matrix in local (principal) coordinates may now be derived on the basis of Equation (2. 29). Hence,

$$[k_{cr, e}] = [b]^t [D_{cr}] [b] A \tag{2.43}$$

For the assembly of the total stiffness matrix, the local matrix (Equation (2. 19)) must be expressed in terms of global coordinates. Ordinarily, this is accomplished by a matrix triple product of the following form:

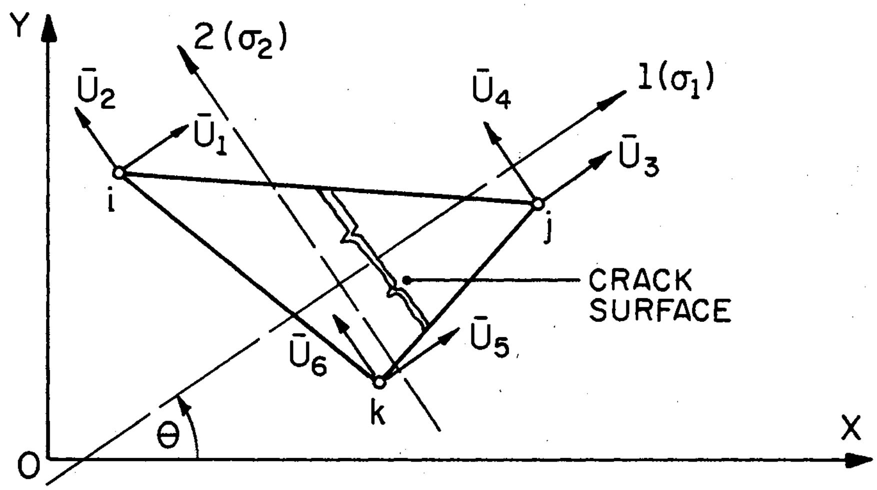


Figure 13. Cracked Element

$$\left[k_{cr,\,g}\right] = \left[R\right]^t \left[k_{cr,\,e}\right] \left[R\right] \tag{2.44}$$

where

$$\left[R\right] = \left[\begin{array}{cc|cc|cc} \cos\phi & \sin\phi & & & & \\ -\sin\phi & \cos\phi & & & & \\ \hline & & \cos\phi & \sin\phi & & \\ & & -\sin\phi & \cos\phi & & \\ \hline & & & & \cos\phi & \sin\phi \\ & & & & -\sin\phi & \cos\phi \end{array}\right]$$

It should be noted at this point that similar transformations must be performed on all anisotropic, uncracked elements, whose material characteristics follow Equation (2.11).

CHAPTER III

MATRIX ITERATIVE PROCEDURES

3.1 Review of Iterative Procedures for Problems with Nonlinear Material Properties

3.1.1 General

Nonlinear structures are usually classified according to the cause of nonlinear behavior. Since all solution procedures in solid mechanics involve equilibrium, kinematic and constitutive equations, nonlinearities may arise from either of these three sets of fundamental relations. In case of large displacements, the geometric configuration of the assembly may change sufficiently under load to influence the equilibrium relations. Large deflections also cause nonlinear terms in the kinematic relations. It appears then that nonlinearities may be due to either the geometry or the material properties or both. Thus, the following three categories contain all possible sources of nonlinear conditions:

1. Geometric nonlinearity caused by nonlinear kinematic relations.

2. Material nonlinearity which arises from complex material laws.

3. Combined geometric and material nonlinearity.

The matrix analysis methods developed for linear structures can be extended to include the above mentioned complications. Because of the presence of nonlinear terms, the solution to the governing matrix equations can no longer be obtained explicitly. Consequently, the use of iterative procedures is inevitable. Most of the early applications handle nonlinearities by calculating corrections to linear solutions. A common method used in the solution of geometrically nonlinear systems is due to Turner et al. (38). The structure is solved as a sequence of elastic problems in which corrective stiffness matrices are generated to update the geometry. A comprehensive review of such methods and subsequently developed procedures can be found in Oden's paper on nonlinear structural analysis (21).

Similar iterative schemes have also been adopted in the study of inelastic structures. Among the earliest applications were investigations concerning thermal effects and creep (39, 5). The most significant developments are connected with research on elasto-plastic problems. Basically, four methods have emerged from such investigations:

1. Direct iterative approach;
2. Initial strain approach;
3. Variable elasticity approach;
4. Initial stress approach.

The key to these different methods is the formulation of the matrix of elastic constants $\left[D\right]$. Since the coefficients of these matrices are functions of the state of stress or strain, they must be re-evaluated after each cycle. In the case of a uniaxial state of stress, the modulus of elasticity may be read from the stress-strain curve directly. More generally, under multiaxial stress conditions, the elastic constants will depend on the stress or strain invariants. It is reasonable to assume that the effective stress σ_{eff} is equal to the value of the second invariant J_2 of the stress deviator tensor. Similarly, the effective strain e_{eff} corresponds to the second strain deviator invariant I_2, where:

$$J_2 = \frac{1}{6}\left\{(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\sigma_{xy}{}^2 + \sigma_{yz}{}^2 + \sigma_{zx}{}^2)\right\} \tag{3.1}$$

$$I_2 = \frac{1}{6}\left\{(\epsilon_x - \epsilon_y)^2 + (\epsilon_y - \epsilon_z)^2 + (\epsilon_z - \epsilon_x)^2 + 6(\epsilon_{xy}{}^2 + \epsilon_{yz}{}^2 + \epsilon_{zx}{}^2)\right\} \tag{3.2}$$

Or, in terms of principal values,

$$J_2 = \frac{1}{6}\left\{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right\} \tag{3.3}$$

$$I_2 = \frac{1}{6}\left\{(\epsilon_1 - \epsilon_2)^2 + (\epsilon_2 - \epsilon_3)^2 + (\epsilon_3 - \epsilon_1)^2\right\} \tag{3.4}$$

The effective or equivalent stress is introduced for convenience as

$$\sigma_{eff} = \frac{1}{\sqrt{2}}\left\{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right\}^{\frac{1}{2}} \tag{3.5}$$

It is closely related to the frequently used octahedral shear stress

$$\tau_{oct} = \frac{1}{3}\left\{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right\}^{\frac{1}{2}} \tag{3.6}$$

through the following relationship:

$$\sigma_{eff} = \frac{3}{\sqrt{2}}\tau_{oct} \tag{3.7}$$

Most authors prefer to use either one of these quantities in place of the rather abstract term J_2:

$$\sigma_{eff} = (3J_2)^{\frac{1}{2}} \tag{3.8}$$

$$\tau_{oct} = \left(\frac{2}{3}J_2\right)^{\frac{1}{2}} \tag{3.9}$$

The convenience of the above definitions and the corresponding strain terms becomes apparent in the formulation of yield criteria. For example, the von Mises condition relates the second stress invariant of the multi-dimensional state of stress to the uniaxial case as follows:

$$3J_2 = \sigma_{eff}^2 = \sigma_o^2 \tag{3.10}$$

where σ_o is the uniaxial tensile or compressive stress. Thus, the three-dimensional situation may be expressed in terms of one parameter, σ_o, only. Furthermore, if a unique relationship between σ_o and ϵ_o exists, one is able to determine the material constants for the three-dimensional continuum at any load level from the uniaxial stress-strain curve:

$$\sigma_o = f(\epsilon_o)$$

$$\sigma_{eff} = f(\epsilon_{eff}) \tag{3.11}$$

3.1.2 Direct Iterative Approach

The direct iterative technique is based upon repeated elastic solutions, where for each cycle the full load is applied. Initially, all elements are assigned a modulus of elasticity, E_o, and a Poisson's ratio, ν_o, corresponding to zero stress. Subsequently, the elastic constants are redefined for each new solution. They depend on the state of stress (or strain) reached in the previous step. According to Zienkiewicz (12), an adequate solution requires three to four iterations.

Unfortunately, this simple method has several disadvantages. It is, for example, impossible to include an unloading cycle in a problem. Clearly, during a load decrease the plastic strains should remain constant. Since the procedure is entirely based upon total effective strains, a reduction in load may cause a change in plastic deformation. Furthermore, it is difficult to obtain reasonable, consistent representations of the equivalent Poisson's ratio.

Gallagher et al. (39), and Argyris (8) realized that both difficulties can be avoided by incremental procedures. These step-by-step methods have the additional advantage that they permit the use of incremental stress-strain characteristics such as the Prandtl-Reuss equations.

3.1.3 Initial Strain Approach

The procedure here consists of applying the load in small increments. For any such load interval the incremental stresses and strains are computed. Total stresses and strains may be obtained by adding the current incremental values to the total stresses (or strains) reached during the previous step. Clearly, the evaluation of the elastic strain increments is straightforward. However, the change in plastic strain depends on both the initial and final stress condition and cannot be determined directly.

The total incremental strain $\{\Delta e\}$ in any interval may, in general, consist of elastic, plastic, thermal and creep strain increments. Throughout this study the latter two contributions will be neglected. Hence, the total strain increments reduce to

$$\{\Delta e\} = \{\Delta \epsilon_e\} + \{\Delta \epsilon_p\} \tag{3.12}$$

If the plastic strain increments are known, they may be treated as initial strains $\{\Delta \epsilon_{o,p}\}$ similar to those caused by temperature changes. Consequently, the stress increments can be determined through a standard elastic analysis.

$$\{\Delta\sigma\} = [D]\left(\{\Delta\epsilon_e\} - \{\Delta\epsilon_{o,\,p}\}\right) \tag{3.13}$$

The difficulites involved in establishing the plastic strain increments depend on the degree of sophistication desired. Any plastic constitutive relation may be implied, including time or temperature dependent material characteristics. One of the most commonly used set of equations, the Prandtl-Reuss flow rule, will be elaborated as an example. In matrix notation they take on the form

$$\{\Delta\epsilon_p\} = C_p\,[D_o]^{-1}\{\sigma\} \tag{3.14}$$

where C_p is a function of the effective stress and the effective plastic strain increment, $\Delta\epsilon^p_{eff}$.

$$\Delta\epsilon^p_{eff} = \frac{\sqrt{3}}{2}\left\{\left(\Delta\epsilon^p_1 - \Delta\epsilon^p_2\right)^2 + \left(\Delta\epsilon^p_2 - \Delta\epsilon^p_3\right)^2 + \left(\Delta\epsilon^p_3 - \Delta\epsilon^p_1\right)^2\right\}^{\frac{1}{2}} \tag{3.15}$$

The matrix $[D_o]^{-1}$ contains $\nu = 0.5$; thus, for the three-dimensional case

$$[D_o]^{-1} = \frac{1}{E}\left[\begin{array}{ccc|ccc} 1 & -.5 & -.5 & & & \\ -.5 & 1 & -.5 & & 0 & \\ -.5 & -.5 & 1 & & & \\ \hline & & & 1.5 & & \\ & 0 & & & 1.5 & \\ & & & & & 1.5 \end{array}\right] \tag{3.16}$$

and

$$C_p = \frac{\Delta\epsilon^p_{eff}}{\sigma_{eff}} \tag{3.17}$$

To obtain the effective incremental plastic strain, Argyris (8) suggested two different methods. First, the so-called direct incremental approach makes use of results obtained in the preceding step. Assume that upon completion of increment i-1, the total and incremental stresses and strains are available. The values of $\Delta\epsilon^{p}_{eff}$ and σ_{eff} are readily determined from Equations (3.15) and (3.5), respectively. The modulus of elasticity, E_o, and Poisson's ratio, ν_o, for the zero stress condition are used throughout the entire solution. With E_o and ν_o known, the constitutive equations needed for the formulation of the stiffness matrices are defined as

$$\left\{\Delta e\right\}_i = \left[D\right]^{-1}\left\{\Delta\sigma\right\}_i + C_{p,i}\left[D_o\right]_i^{-1}\left\{\Delta\sigma\right\}_i \qquad (3.18)$$

This procedure may be improved by performing, for each load step, an initial elastic solution and a series of subsequent iterations. The increments of stresses and strains of the current cycle are used to obtain a new estimate of the plastic strain increment for the next iteration. According to Argyris (8), this iterative-incremental method usually converges after five iterations.

It should be mentioned that both methods require special precaution when unloading occurs. During a load decrease, the structure must behave in a purely elastic fashion which may be accomplished by specifying the factor C_p as zero. Likewise, upon reloading, C_p must remain zero until the current σ_{eff} is found to exceed the highest effective stress achieved during the previous increment.

3.1.4 Variable Elasticity Approach

For elastic-perfectly plastic and ideally plastic material, the methods of sections 3.1.3 and 3.1.4 break down. This is due to the fact that large plastic strain increments may result even from very small load augmentations. Pope suggested a method in 1965 (7) which adjusts the stress-strain relationship in every load increment to take into account plastic deformations. The works of Marcal and King (11), Akyuz and Merwin (15) fall into the same category.

For the elastic strain increment the expression remains as

$$\left\{\Delta\epsilon_e\right\} = \left[D_e\right]^{-1}\left\{\Delta\sigma\right\} \tag{3.19}$$

However, the Prandtl-Reuss equations (which express the plastic strain increments in terms of actual accumulated stresses $\left\{\sigma\right\}$) must be replaced by a relationship of the form

$$\left\{\Delta\epsilon_p\right\} = \left[D_p\right]^{-1}\left\{\Delta\sigma\right\} \tag{3.20}$$

To derive $\left[D_p\right]^{-1}$, let H' denote the slope of the effective stress-effective plastic strain function, which, again, will be assumed to be available through experiments. The strain-hardening criterion in discrete form then becomes

$$\Delta\sigma_{eff} = H'\Delta\epsilon^p_{eff} \tag{3.21}$$

By differentiating the von Mises yield condition, a second expression for $\Delta\sigma_{eff}$ may be obtained as follows:

$$\Delta\sigma_{eff} = \frac{3}{2\sigma_{eff}}\left\{\sigma'_x\Delta\sigma_x + \sigma'_y\Delta\sigma_y + \sigma'_z\Delta\sigma_z\right\} + \frac{3}{\sigma_{eff}}\left\{\sigma_{xy}\Delta\sigma_{xy} + \sigma_{yz}\Delta\sigma_{yz} + \sigma_{zx}\Delta\sigma_{zx}\right\} \tag{3.22}$$

where

$$\sigma'_x = \frac{1}{3}(2\sigma_x - \sigma_y - \sigma_z) \quad \text{(cyclic substitution)} \tag{3.23}$$

The above term can now be written in matrix form as

$$\Delta\sigma_{eff} = \left\{S\right\}^t \left\{\Delta\sigma\right\} \tag{3.24}$$

and substituted into Equation (3.21).

$$\Delta\epsilon^p_{eff} = \frac{1}{H'} \left\{S\right\}^t \left\{\Delta\sigma\right\} \tag{3.25}$$

Upon substitution of Equation (3.25) into the Prandtl-Reuss relations (3.14), one arrives at the desired incremental stress-strain rule.

$$\left\{\Delta\epsilon_p\right\} = \frac{\Delta\epsilon^p_{eff}}{\sigma_{eff}} \left[D_o\right]^{-1} \left\{\sigma\right\}$$

$$= \frac{1}{H'\sigma_{eff}} \left[D_o\right]^{-1} \left\{\sigma\right\} \left\{S\right\}^t \left\{\Delta\sigma\right\} \tag{3.26}$$

or

$$\left\{\Delta\epsilon_p\right\} = \left[D_p\right]^{-1} \left\{\Delta\sigma\right\} \tag{3.27}$$

Hence, $\left[D_p\right]$ depends upon the current state of stress $\left\{\sigma\right\}$ and the strain-hardening history through the parameter H'. Combined with the elastic constitutive relations, the total strain increments become

$$\left\{\Delta e\right\} = \left\{\Delta\epsilon_e\right\} + \left\{\Delta\epsilon_p\right\} = \left(\left[D_e\right]^{-1} + \left[D_p\right]^{-1}\right) \left\{\Delta\sigma\right\} \tag{3.28}$$

and the corresponding change in stress is

$$\left\{\Delta\sigma\right\} = \left(\left[D_e\right]^{-1} + \left[D_p\right]^{-1}\right)^{-1} \left\{\Delta\epsilon\right\} \tag{3.29}$$

This particular method is known to converge very rapidly. Furthermore, unloading can be treated by simply inserting an elastic $\left[D\right]$ matrix in the increment following an unloading interval. From a computational point of view, the variable elasticity approach has one disadvantage; at each solution step, the stiffness of the structure is changed. Thus, for every iteration the whole structural stiffness matrix must be reassembled, which naturally results in excessive computer time.

3.1.5 Initial Stress Method

The most recent method of elasto-plastic analysis was introduced by Zienkiewicz et al. (20) as an alternative approach to the "variable elasticity" procedure. This "initial stress" method makes use of the fact that the total strain increments uniquely define the corresponding stress situation throughout the entire load history. This holds true for any type of stress-strain relationship including those for ideally plastic structures. Therefore, it seems more reasonable to treat the stress increments as initial values rather than the strains. The change in stress derived from the corresponding strain increment will, in general, be incorrect. Consequently, the initial stress approach must again rely on an iteration scheme.

Once more, the first step in each load increment consists of solving the problem elastically. Both the strain increments $\left\{\Delta\epsilon_e\right\}$ and the associated change in stress $\left\{\Delta\sigma_e\right\}$ are computed. Since the calculated values for $\left\{\Delta\sigma_e\right\}$ deviate from the true stress increments $\left\{\Delta\sigma\right\}$, the equilibrium conditions are violated. In order to maintain equilibrium, a set of "body forces" equal and opposite to the initial stress system $\left\{\Delta\sigma_b\right\}$ must be introduced.

$$\left\{\Delta\sigma_b\right\} = \left\{\Delta\sigma_e\right\} - \left\{\Delta\sigma\right\} \tag{3.30}$$

In the computation, the unknown, true stress increments are replaced by approximative values $\left\{\Delta\bar{\sigma}\right\}$ determined from the second iteration cycle.

Before proceeding to the second solution step, each element is examined for its type of behavior. For this purpose the calculated stress increments are added to the total stresses $\left\{\sigma_o\right\}$ reached during the preceding increment to establish the current stresses $\{\sigma\}$.

$$\{\sigma\} = \{\sigma_o\} + \{\Delta\sigma_e\} \tag{3.31}$$

These values and the corresponding strain-hardening parameter k are substituted into a suitable yield criterion, $F(\{\sigma\}, k)$. The resulting numerical value determines whether the element exhibits elastic or plastic behavior. From the theory of plasticity, it is known that for a strain-hardening material, the following four cases may be distinguished:

$$\begin{array}{lll} \text{a.} & F < 0 & \text{elastic behavior} \\ \text{b.} & F = 0 \text{ and } \Delta F < 0 & \text{unloading, elastic behavior} \\ \text{c.} & F = 0 \text{ and } \Delta F = 0 & \text{neutral loading, plastic behavior} \\ \text{d.} & F = 0 \text{ and } \Delta F > 0 & \text{loading, plastic behavior} \end{array} \tag{3.32}$$

where

$$\Delta F = \frac{\partial F}{\partial \{\sigma\}} \{\Delta\sigma\} \tag{3.33}$$

Zienkiewicz states the same conditions in a more computer-oriented form (29).

Clearly, no further iteration is required if, after the beginning elastic cycle, the first or second condition is satisfied throughout the entire structure. Otherwise, the solution is continued by computing new stress increments $\{\Delta\bar{\sigma}\}$.

$$\{\Delta\bar{\sigma}\} = [D_{ep}] \{\Delta\epsilon_e\} \tag{3.34}$$

where $[D_{ep}]$, the matrix of material constants, is a rather involved expression. Its coefficients are dependent upon the yield condition, the strain-hardening parameter and the stress-strain curve. In the case of elasto-perfectly-plastic material, the matrix still exists since the slope of the stress-strain curve H' appears as a single term in the numerator (20).

All subsequent iterations must now be based on augmented load conditions (initial, externally applied load increments plus equilibrating nodal forces). For the ith solution cycle, they are

$$\left\{\overline{P}_i\right\} = \left[b\right]^t \left\{\Delta\sigma_i^*\right\} dV \qquad (3.35)$$

where

$$\left\{\Delta\sigma^*\right\} = \left\{\Delta\overline{\sigma}_i\right\} - \left\{\Delta\overline{\sigma}_{i-1}\right\} \qquad (3.36)$$

3.2 Proposed Iteration Procedure

The method presented here is based upon an iterative, incremental load approach. For each load increment, the whole structure is repeatedly solved as an elastic problem until closure. Consider an arbitrary concrete element during load increment i. Assume that at the end of the previous step the principal strains $\left\{e_{p,\ i-1}\right\}$ and stresses $\left\{\sigma_{p,\ i-1}\right\}$ have been established. Based upon these values the element may be in any one of the following conditions:

a. Type 1: Elastic, isotropic;

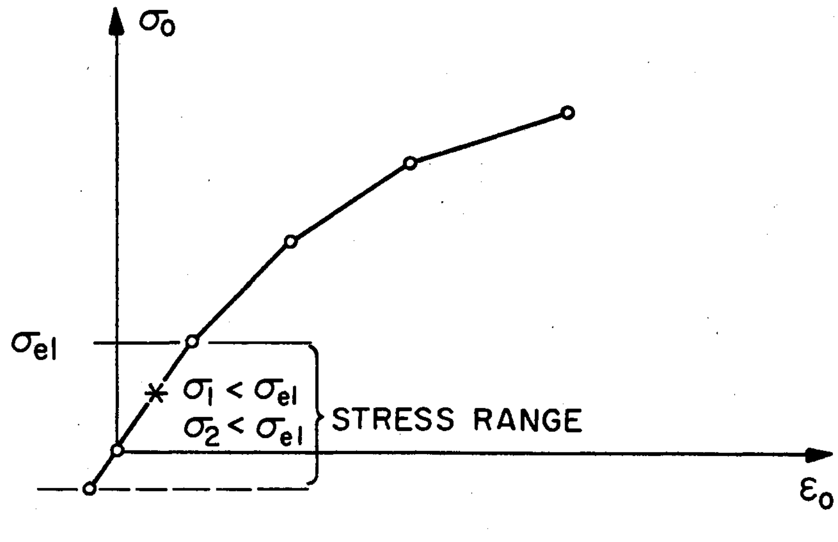
b. Type 2: Elastic, anisotropic;

c. Type 3: Inelastic, anisotropic;

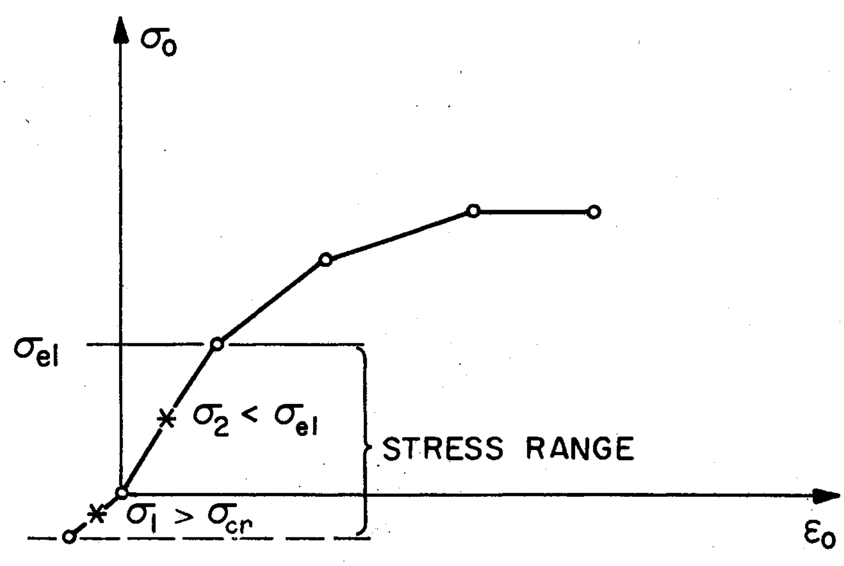
d. Type 4: Cracked.

The four cases can be visualized diagrammatically in Figures 14 and 15.

In the present computer program, principal strains $\left\{e_{i-1}\right\}$ are used to determine the relevant material constants. After the proper modulus of elasticity, E_i, and Poisson's ratio, ν_1, have been found for

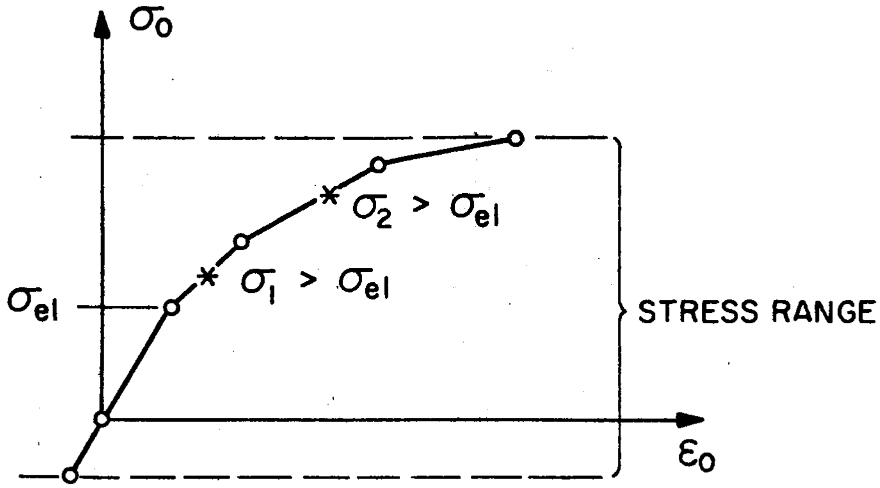


(a) TYPE 1

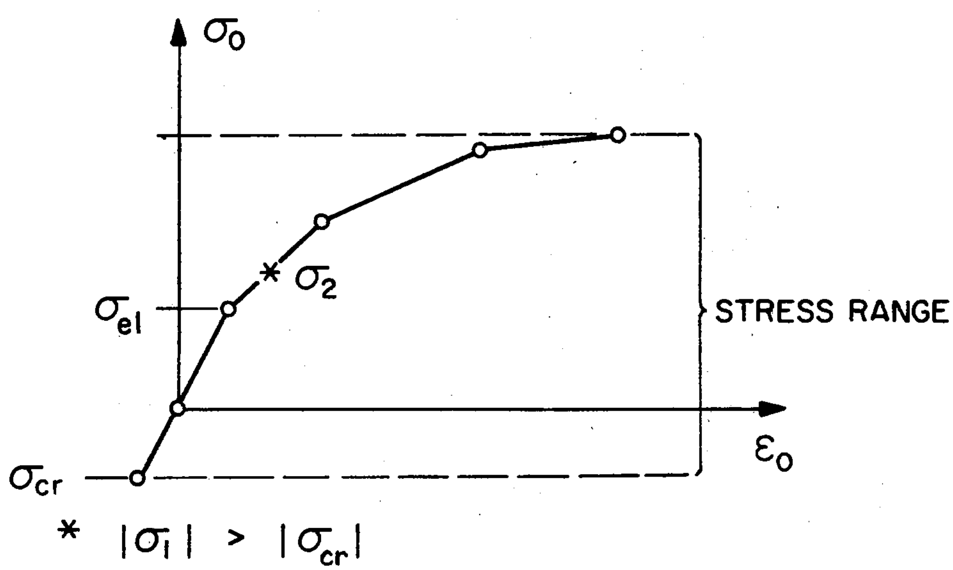


(b) TYPE 2

Figure 14. Classification of Concrete Elements in the Elastic Range



(a) TYPE 3



(b) TYPE 4

Figure 15. Classification of Concrete Elements in the Inelastic Range

each element, the $[D]$ matrices are generated. To recapitulate, the appropriate equation shall be summarized below:

Type 1:

$$[D] = \frac{E_1}{(1 - \nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad (3.37)$$

Types 2 and 3:

$$[D_a] = \frac{E_1}{1 - \nu_{12}\nu_{21}} \begin{bmatrix} 1 & \nu_{12} & 0 \\ \nu_{12} & n & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.38)$$

(for principal axes)

Type 4:

$$[D_{cr}] = E_2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.39)$$

(for principal axes)

The elemental stiffness matrices $[k_i]$ follow immediately as

$$[k_i] = [b]^t [D] [b] \quad (3.40)$$

or, for cases 2, 3, and 4,

$$[k_i] = [R^t] [b]^t [D_j] [b] [R] \quad (3.41)$$

Next, the total stiffness matrix is assembled and solved for the incremental displacements $\{\Delta U\}$. The discussion of the procedure used will be postponed until Chapter IV. The incremental strains are now evaluated as

$$\left\{\Delta e_i\right\} = \left[b_i\right] \left\{\Delta U_i\right\} \tag{3.42}$$

and added to the total strains $\left\{e_{i-1}\right\}$ of the preceding step to give the new total strains

$$\left\{e_i^1\right\} = \left\{e_{i-1}\right\} + \left\{\Delta e_i\right\} \tag{3.43}$$

These values constitute a new strain situation with a corresponding new set of principal strains $\left\{e_{p,\,i}\right\}$. The material properties of the following iteration cycle are again extrapolated from the stress-strain curve. The iteration is stopped after a specified tolerance is reached. Before proceeding to the next increment, all total stress and strain values are updated and stored. Similar treatment is imposed upon the reinforcements and bond links. However, the procedure here is much less involved since the matrix $\left[D\right]$ reduces to a single term E_1.

CHAPTER IV

COMPUTER PROGRAM

4.1 General

The iteration procedure described in Chapter III has been programmed for solution on a digital computer. Two programs, NARCOS-1 and NARCOS-2,[1] were written for the IBM 360-65 model operated by the Oklahoma State University Computer Center. The standard ASA FORTRAN language was used.

Both programs generate all necessary mesh data from a minimum of input information. NARCOS-1 was based on the element arrangement used by Scordelis (24). This first version had to be abandoned at an early stage because of unsatisfactory results. A convergence study revealed that the symmetric mesh of NARCOS-2 converges much more rapidly, Figure 16.

4.2 Computer Idealization of the Beam

The finite element representation of the beam is arranged into rectangles. Each rectangular unit is subdivided into four triangular elements of equal area. The corners are numbered in a clockwise direction, Figure 17. To ensure small band widths, the joint numbers

---

[1] NARCOS is the abbreviation for "Numerical Analysis of Reinforced Concrete Structures."

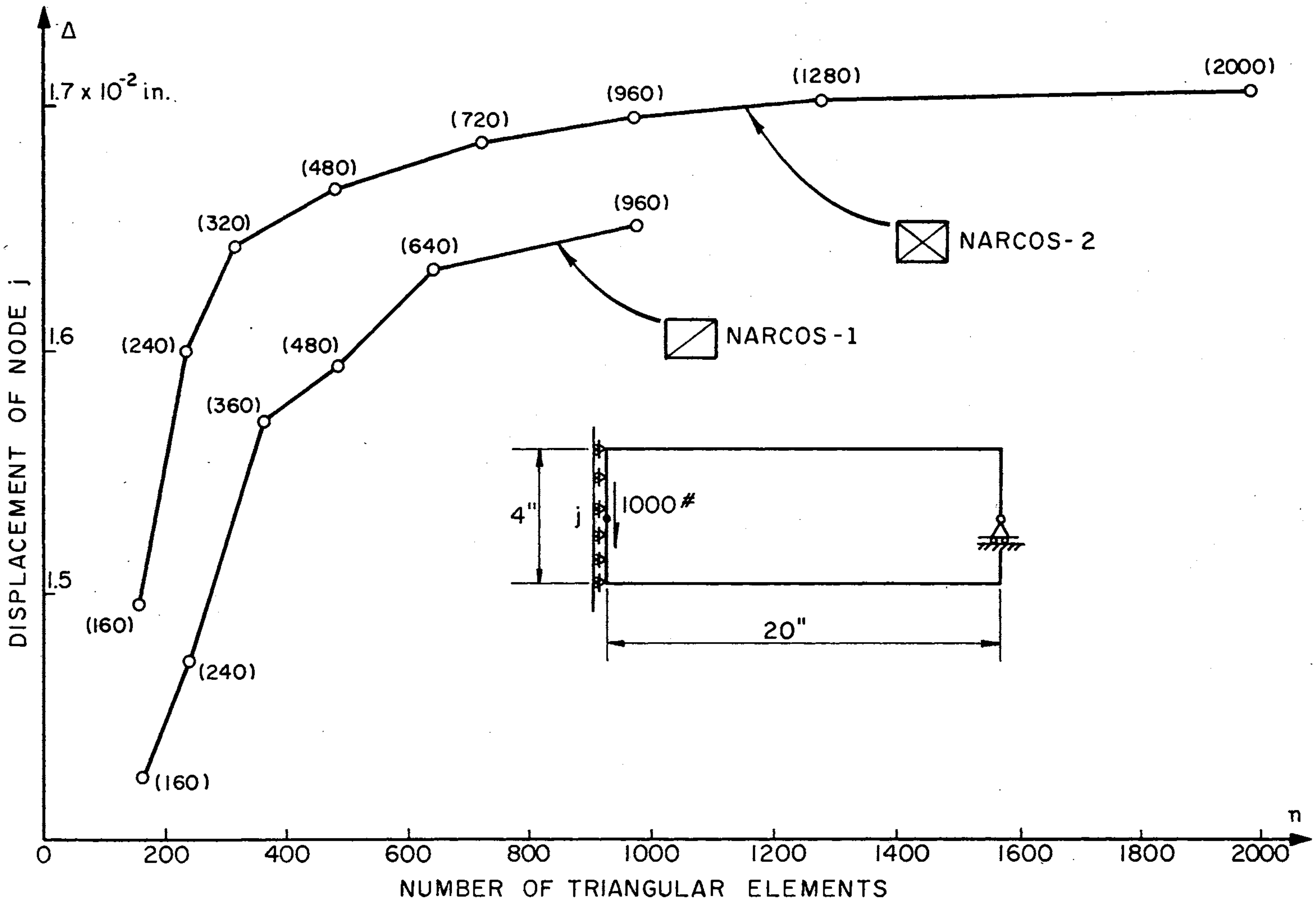
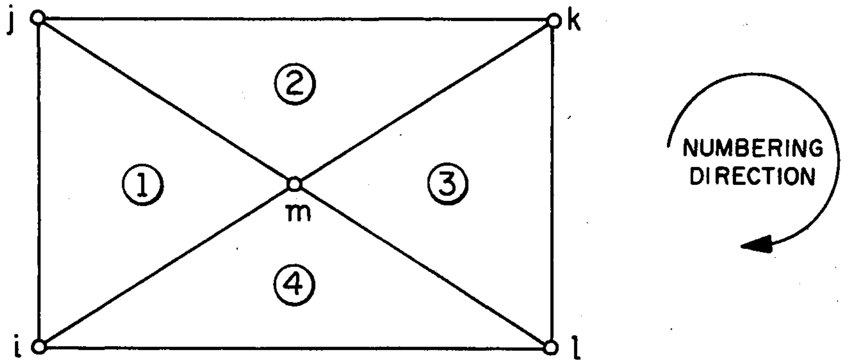
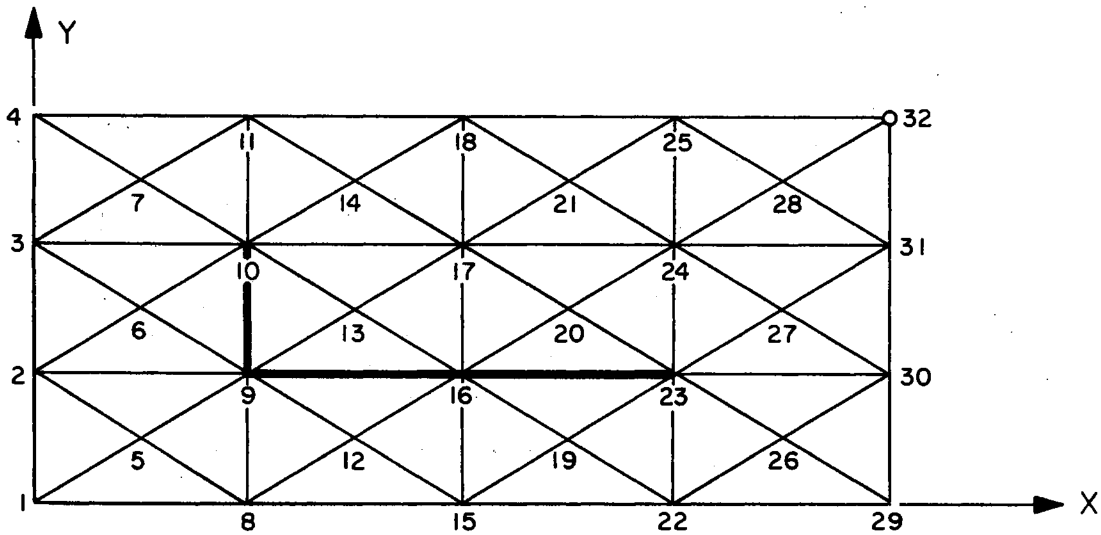


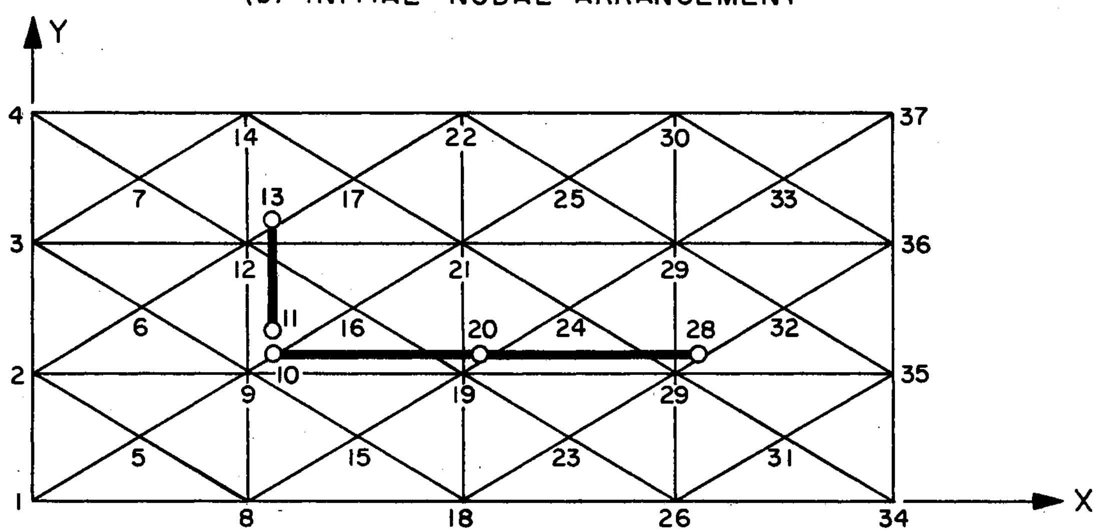
Figure 16. Results of Convergence Study



(a) ELEMENT AND NODAL ARRANGEMENT



(b) INITIAL NODAL ARRANGEMENT



(c) CUMULATIVE NODAL ARRANGEMENT

Figure 17. Example of Nodal Arrangements

are arranged in the direction of the least number of nodes. Reinforcements can be connected at the corner joints of the rectangles only. Inclined reinforcing (such as shear reinforcements under 45° wrt. beam axis) is not allowed. The program connects all steel nodes by means of bond links automatically. As mentioned earlier, this requires a revision of the nodal list. All input information must be specified in terms of the original nodal arrangement, Figure 17a. At the end of the data input, the program generates a cumulative nodal list which includes the additional steel nodes, Figure 17b.

4.3 Flow Chart

Figure 18 shows a summary flow chart of the program NARCOS-2, using symbols shown in Figure 19. The detailed listing is given in Appendix B.

4.4 Solution of Equilibrium Equations

With the nodal arrangement discussed above, the stiffness matrix for the finite element assembly will be banded. This type of matrix lends itself well to direct solution by Gauss elimination. Since the matrix is symmetric, only the upper half of the band is stored in a rectangular array. The assembly and solution of this array is done blockwise, Figure 20. The first step consists of a forward elimination. Each reduced block is stored on a disk. With the last reduced block still in core, the backward elimination is performed in reverse order. Although several load vectors may be included in this procedure, only one loading case is considered in the present program.

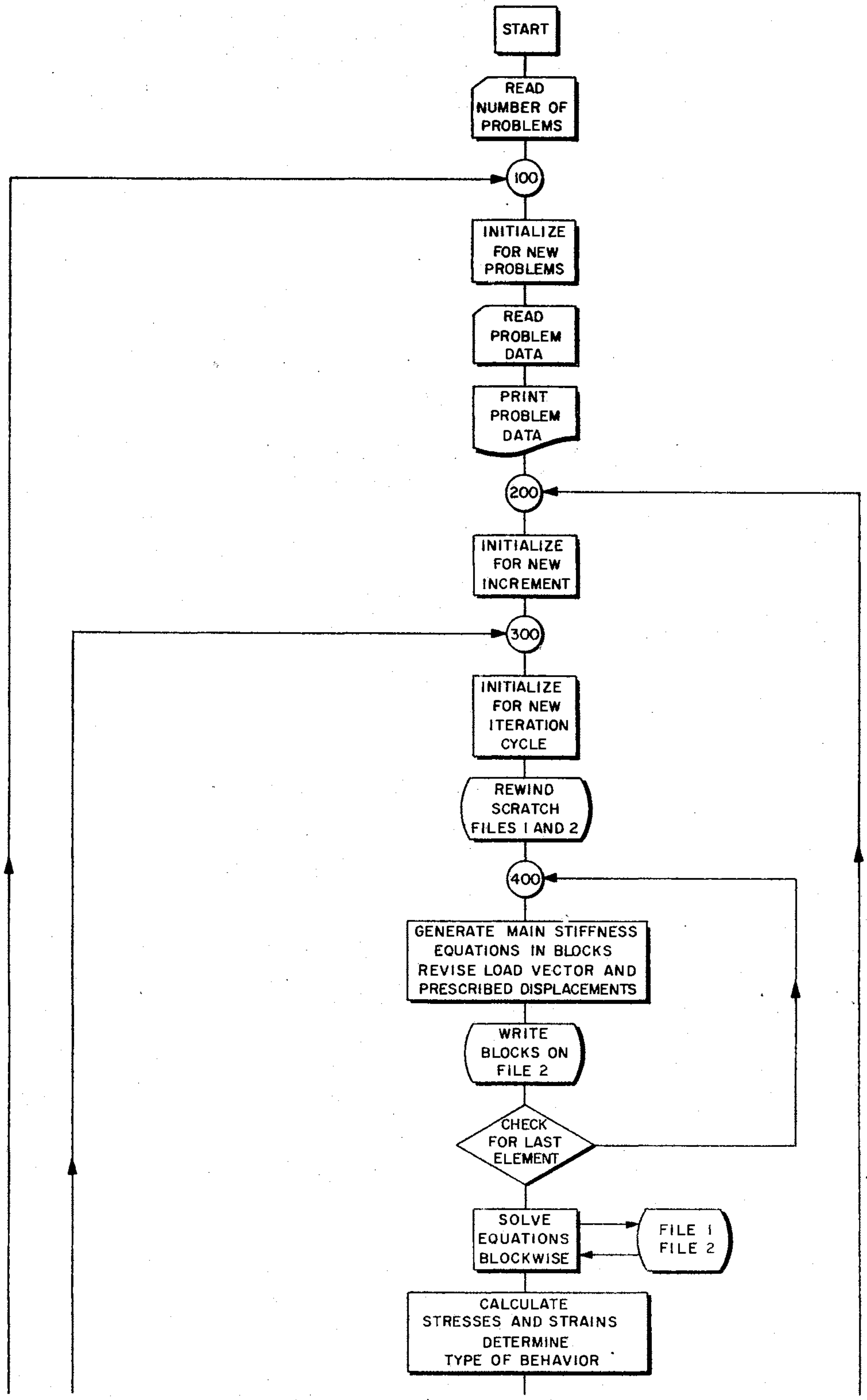


Figure 18. General Flow Chart

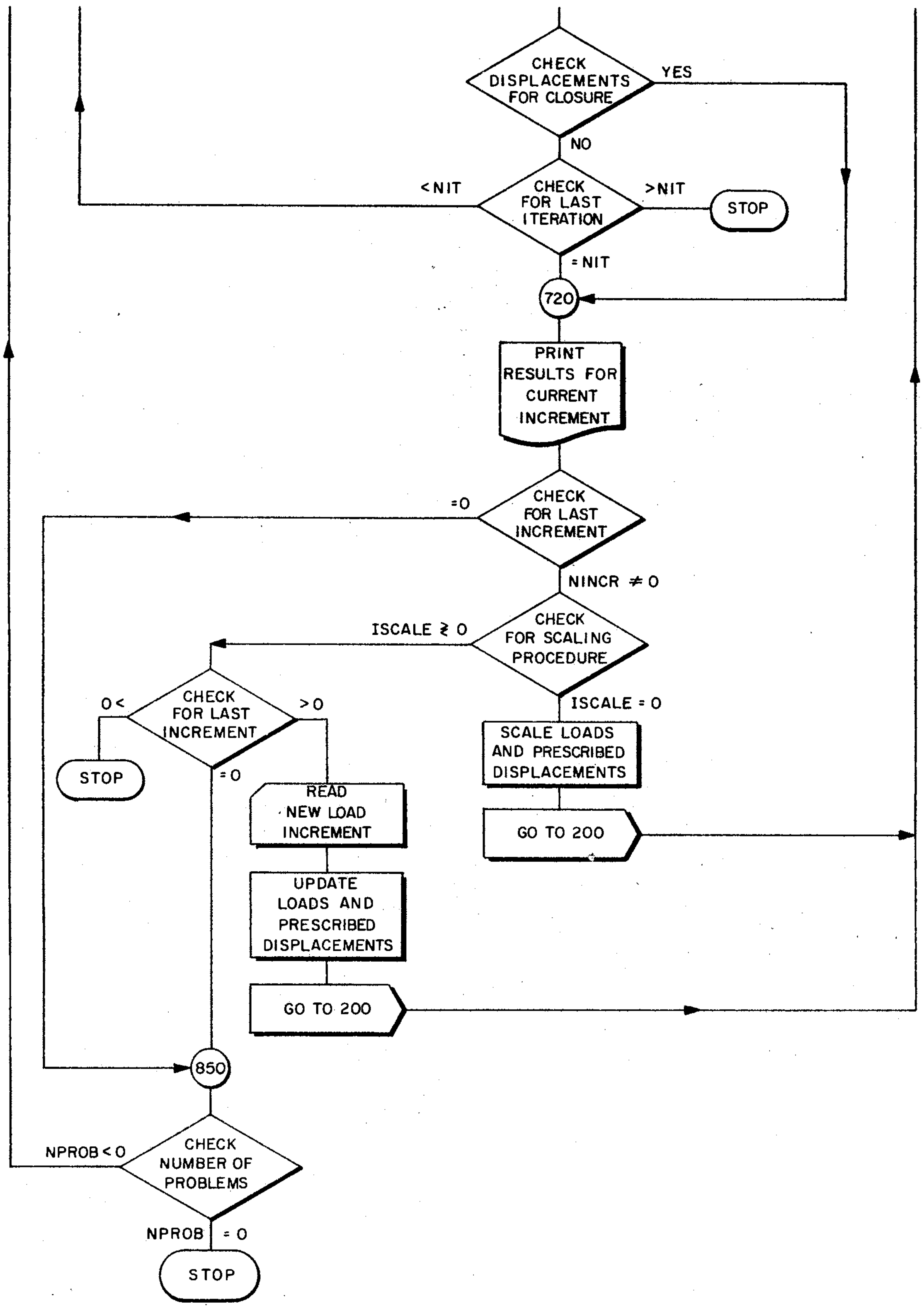


Figure 18. General Flow Chart (continued)

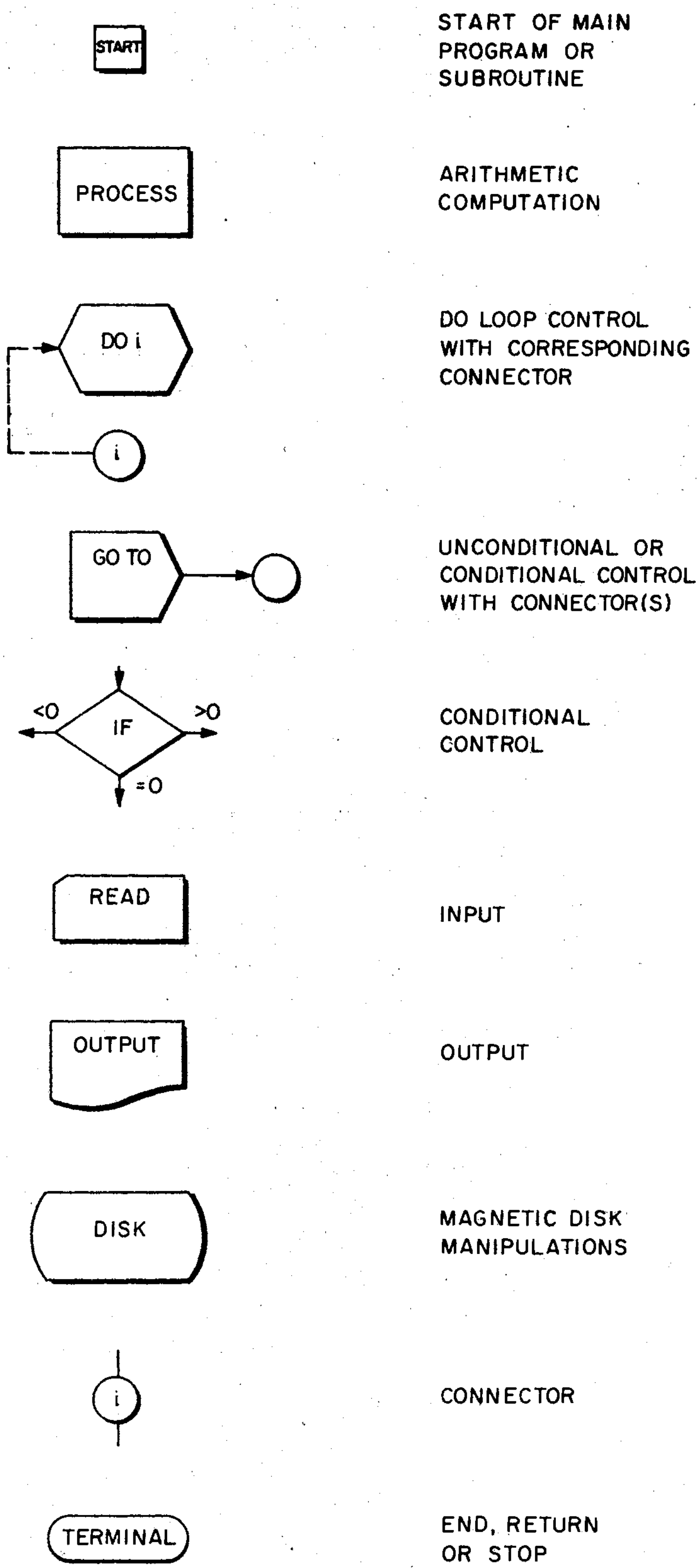


Figure 19. Flow Chart Symbols

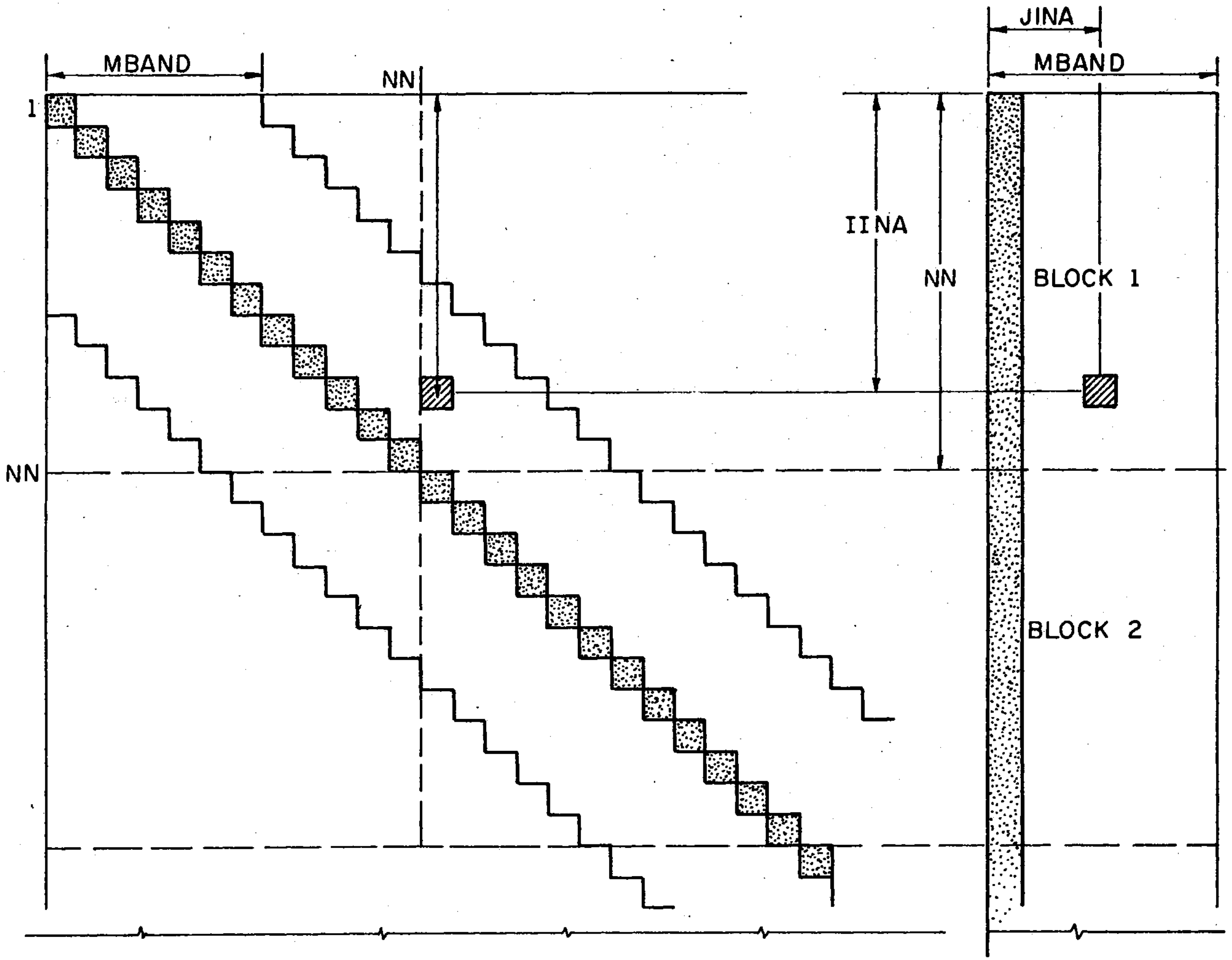


Figure 20. Block Arrangement of Main Stiffness Matrix

The solution of the stiffness matrix is done in subroutine BANSOL. A detailed description of the standard Gauss procedure shall be omitted. Complete information on this method may be found in Reference 40.

4.5 Input List

The input data are arranged in tabular form. Topological and geometrical properties make up the first block. The second block of information consists of the material properties. Block 3 contains the list of reinforcements. Loads and boundary conditions are specified in the last block. Each table is identified by a block header card as shown in Figure 21. The specific format of each type of data statement is given in Appendix C. In the following the general input sequence will be described in detail:

1. Number of problems: The first card must specify the number of problems to be solved.

2. Problem title: One alphanumeric card initializes and identifies a new problem.

3. Control card for first block: Topology and geometry. NARCOS-2 offers three modes for the input of geometric and topological properties. The mode is specified, together with the number of nodes in the horizontal and vertical direction and the number of reinforcements, on the first control card.

Mode 1. Equal spacing:

Under this mode all rectangular elements are of equal size. The program divides the length and depth of the beam into a specified number of intervals, respectively.

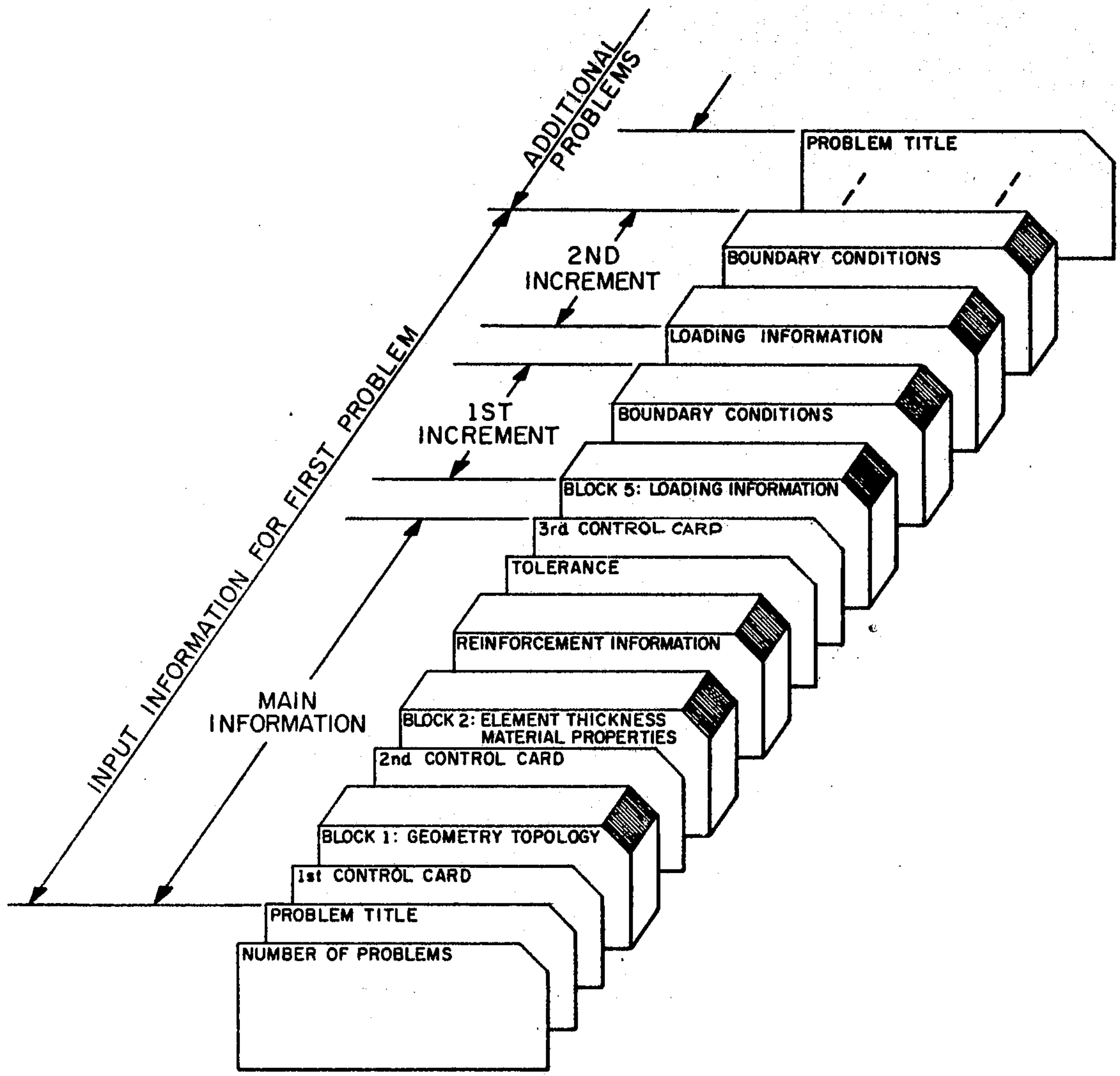


Figure 21. Arrangement of Input Data

The geometry may in this case be given by the coordinates of the upper right corner (for example, node 32 in Figure 17b).

Mode 2. Unequal spacing:

This input option allows for the variation of the element size in horizontal and/or vertical direction. Example 1 of Appendix D demonstrates this input mode.

Mode 3. Individual input:

Mode 3 requires the declaration of the geometry and topology (i. e., node and element numbers) for each triangular element. This option was included to make the program available for irregular shapes.

4. Joint coordinates: Data giving the location of either the corner node (mode 1) or of nodes which specify the interval length of unequally spaced elements (mode 2) follow the first header card. Input under mode 2 may best be explained by means of Example 1 in Appendix D. The coordinates of the interval points (1, 3, 9) in y-direction (cards 4 to 6) are given first. The number of the joints must correspond to the actual node number (Figure 32). The computer divides the intervals into the correct number of equal segments.

5. Control card for second block: Material properties. This data statement initializes the input list of the stress-strain characteristics and specifies the number of elements with irregular thickness.

6. Element thickness: The program assumes a unit thickness over the entire continuum. However, specification of other than unit thicknesses of the rectangular elements is permitted.

7. Material properties: The stress-strain curves for concrete, steel and bond are given by a set of points on the curve. For steel, the maximum number of stations is not to exceed 20. The stress-strain characteristics for concrete and bond are given by a maximum of ten points. The point, stress equal to zero and strain equal to zero, must also be specified (if the curve passes through the origin). Following the input of the stress-strain graphs, the Poisson's ratios for each region between two points on the curves must be declared. The input of Poisson's ratios is required for concrete and steel only. The last card in this block contains the bond link stiffness, K_v. This coefficient expresses the strain-relative displacement characteristic of the linkage element perpendicular to the reinforcement. If the numerical value is omitted (blank card), the program assumes a number which is 10^6 times the value of the first coefficient on the diagonal of the main stiffness matrix.

8. Reinforcement information: There is no special header card for this table, since the number of reinforcement cards was given in the control statement for the first block. Each reinforcement input specifies one or more bars. The cross-sectional area may be given directly or in the standard form as bar number according to the ACI Code. The position of reinforcements is specified by the number of the start and end node. The computer automatically divides the bar into two-force members of the same length as the corresponding rectangular concrete elements and assigns the proper bond links.

9. Tolerance: The number on this card declares the percentage error tolerated on the largest displacement.

10. Control card of last block: Loading and boundary conditions. The number of loads, boundary conditions, increments and iterations are specified. If the number of increments is equal to zero, the program performs an elastic solution without iterations. Automatic scaling is done if the parameter NINCR is equal to 1. In this case, the load is taken as total load. During the first cycle, the problem is solved elastically and all results are scaled until the element strains in the extreme element correspond to the yield values. The loads are adjusted accordingly. The difference of the total given load and the load at yield is divided into 20 increments.

11. Loading information: One card per joint load, i.e., x- or y- component or both, must be supplied.

12. Boundary conditions: Only one specific boundary condition may be stated on one card. The direction is identified in alphanumeric form (x or y). Prescribed displacements may be introduced by simply adding the numerical value of the induced delfection after the direction parameters, x and y.

To complete this section, a few additional remarks concerning the data input seem necessary. It should be mentioned that the entire input for a particular problem must be consistent with regard to dimensions. The program does not allow for mixed units.

When no reinforcements are specified, several portions of the input sequence are skipped. In this case, the user must omit stress-strain curves for steel and bond, "perpendicular" bond stiffness, and reinforcement cards.

For problems with fixed boundary nodes, it is not necessary to restate these boundary conditions for each load increment. If the program encounters a blank card after the input of a load increment, the reading of boundary conditions is suppressed. The conditions of the first (initial) load step are assumed to apply throughout the entire loading history.

A special feature has been introduced in the program in the form of a CHECK EQUILIBRIUM card. If such a card is included at the end of the first set of boundary conditions, the program checks the equilibrium for each vertical strip of elements. The shear equilibrium and bending stress equilibrium (including forces in steel tendons) are checked. The average shear and normal stresses for each rectangular unit and the residual force on the cross section is output at the end of each load increment.

4.6 Output Information

The complete list of input data is printed in tabular form. The calculated topological and geometrical properties refer to the updated cumulative nodal list. Results are provided after each successfully completed load increment. The results consist of a complete list of nodal displacements and stresses and strains in all elements. Several supplementary messages are included to clarify the large output. In addition, two "print error" subroutines report the most common input mistakes. A few are automatically corrected. The corrections made are recorded as nonfatal error messages. Sample output is included in Appendix D.

CHAPTER V

NUMERICAL RESULTS

5.1 General

Three singly reinforced concrete beams were solved with NARCOS-2. All examples were chosen to be simply supported and acted upon by concentrated loads as shown in Figures 22, 23, 27 and 30.

Because of the symmetric loading, boundary and geometric conditions, the solution could be performed for half beams only. The output consisted of stresses in the concrete and reinforcements together with the principal stresses and their directions for each triangular element. Bond forces as well as the relative displacements between steel and concrete nodes were printed. Several indicators, such as the condition of the concrete elements (e.g., cracked) and equilibrium checks, were included in the listing to facilitate the interpretation of the extensive computer output. Selected results will be discussed in the following sections.

5.2 Example Problem 1: Scordelis' Beam A-1

Several purely elastic problems were considered to serve as a check on the computational procedure and the program development. Existing programs for the analysis of in-plane loaded plates were primarily used for this purpose.

Next, Scordelis' beam A-1 was solved with NARCOS-2. The results reported in his paper (24) could be considered as a general guide only because of several differences in the problem setup. First, artificial cracks were introduced in beam A-1 at the beginning of the loading process. The procedure here was to let cracks develop in the direction perpendicular to the principal tensile stresses when the maximum allowable tensile stress was reached. Second, Scordelis did not allow for any tensile resistance in the concrete. In this study, the modulus of rupture was assumed to be 300 psi. As a result of this, the beam under investigation showed a higher loading capacity before cracking took place. Thus, Scordelis' initial load of 1000 pounds had to be modified to P_o = 7000 pounds. Cracking was initiated at a load level of 7200 pounds. Several cracks developed simultaneously at the bottom edge of the beam. Obviously, this type of crack pattern was to be expected with the moment being constant between the support and the point of application of the load (support at midspan). Subsequently, the load was increased by 200 pound increments up to 8800 pounds. The cracks continued to develop in the same direction (i. e., perpendicular to the edge) and additional cracks occurred along the bottom edge. The stress patterns, including the bond forces and crack propagation, are shown for different load levels in Figures 24 through 26. Figure 22 contains the information about the stress-strain characteristics used. Poisson's ratios of 0. 3 for the reinforcements and of 0. 15 for the concrete were chosen to be constant throughout the entire load history. The values for the bond stiffness corresponded to those given by Scordelis ($K_h = 2.2 \times 10^6$ psi).

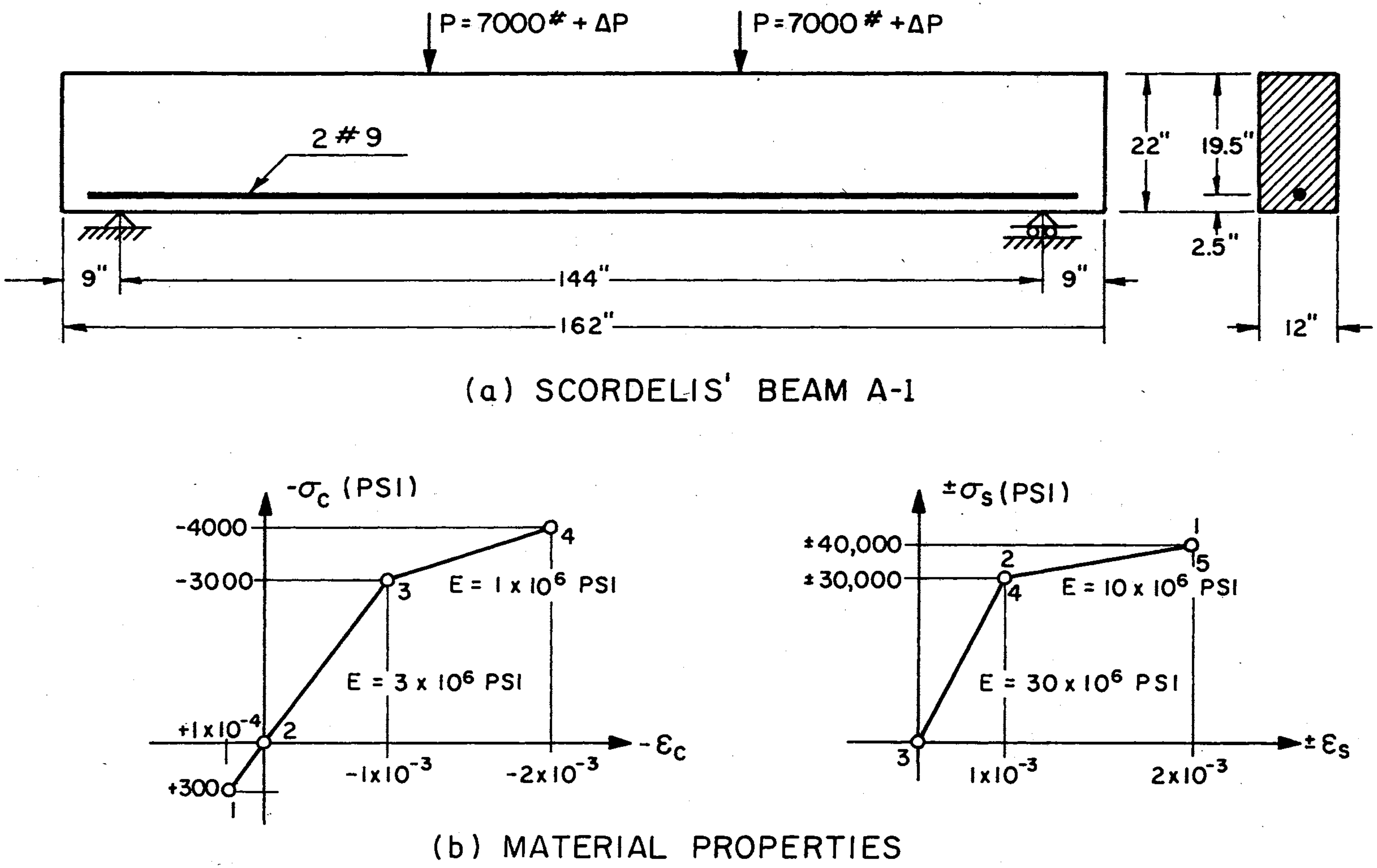
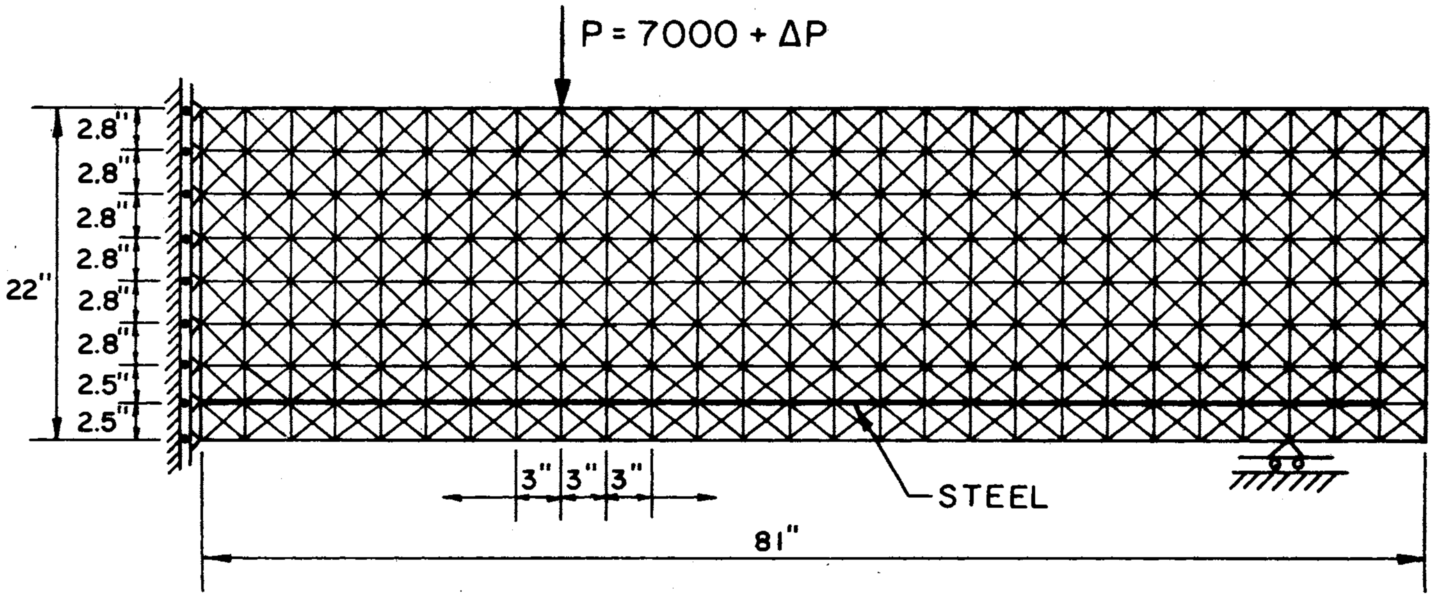


Figure 22. Example Problem 1: Scordelis' Beam A-1



NUMBER OF TRIANG. ELEMENTS	NTEL	=	864
NUMBER OF NODES (INCL. STEEL)	NUMNOD	=	495
NUMBER OF LOADS	NLOAD	=	1
NUMBER OF BOUND. COND.	NBOUND	=	10

Figure 23. Mathematical Model of Beam A-1

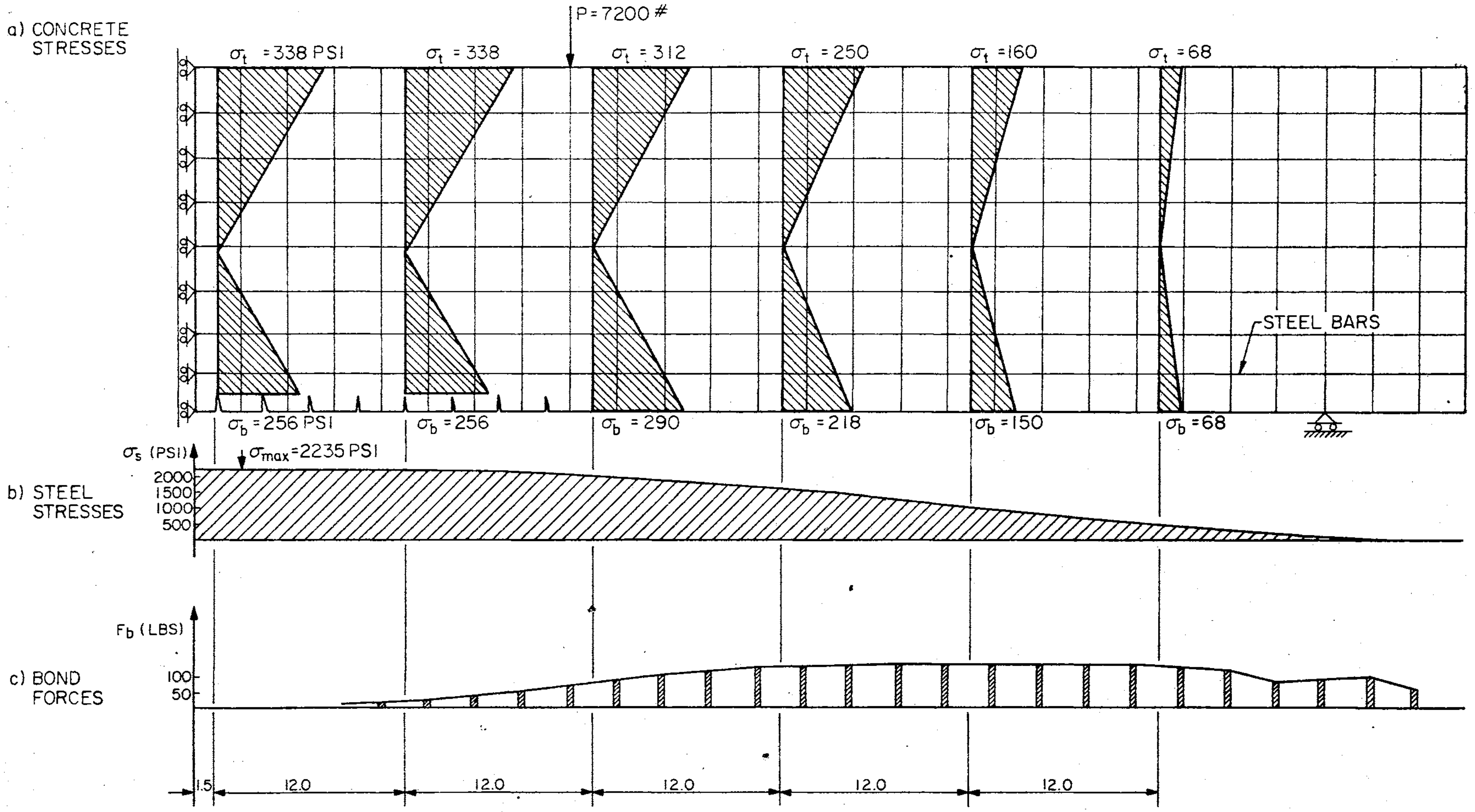


Figure 24. Example Problem 1: Stresses at P = 7200 Lbs.

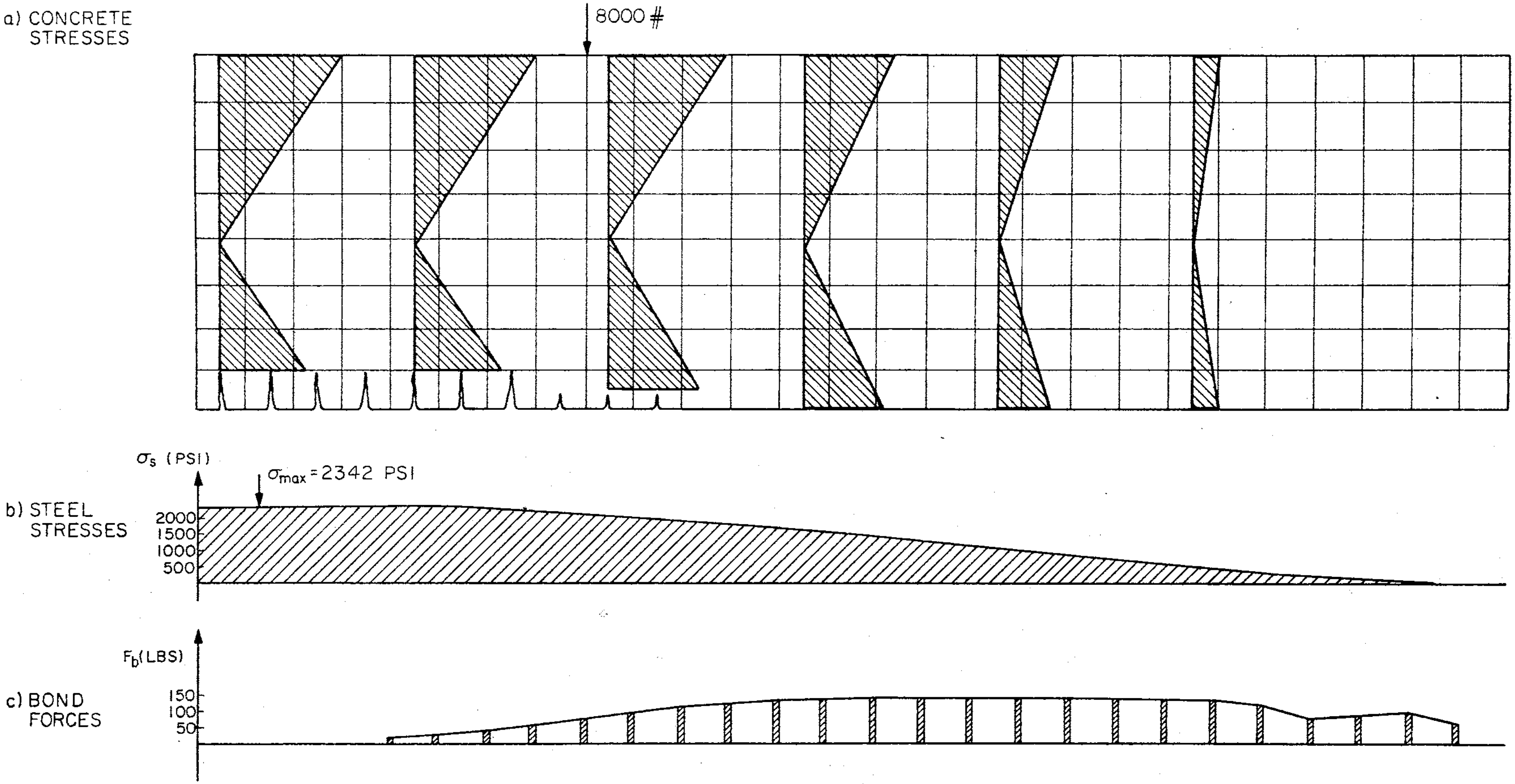


Figure 25. Example Problem 1: Stresses at P = 8000 Lbs.

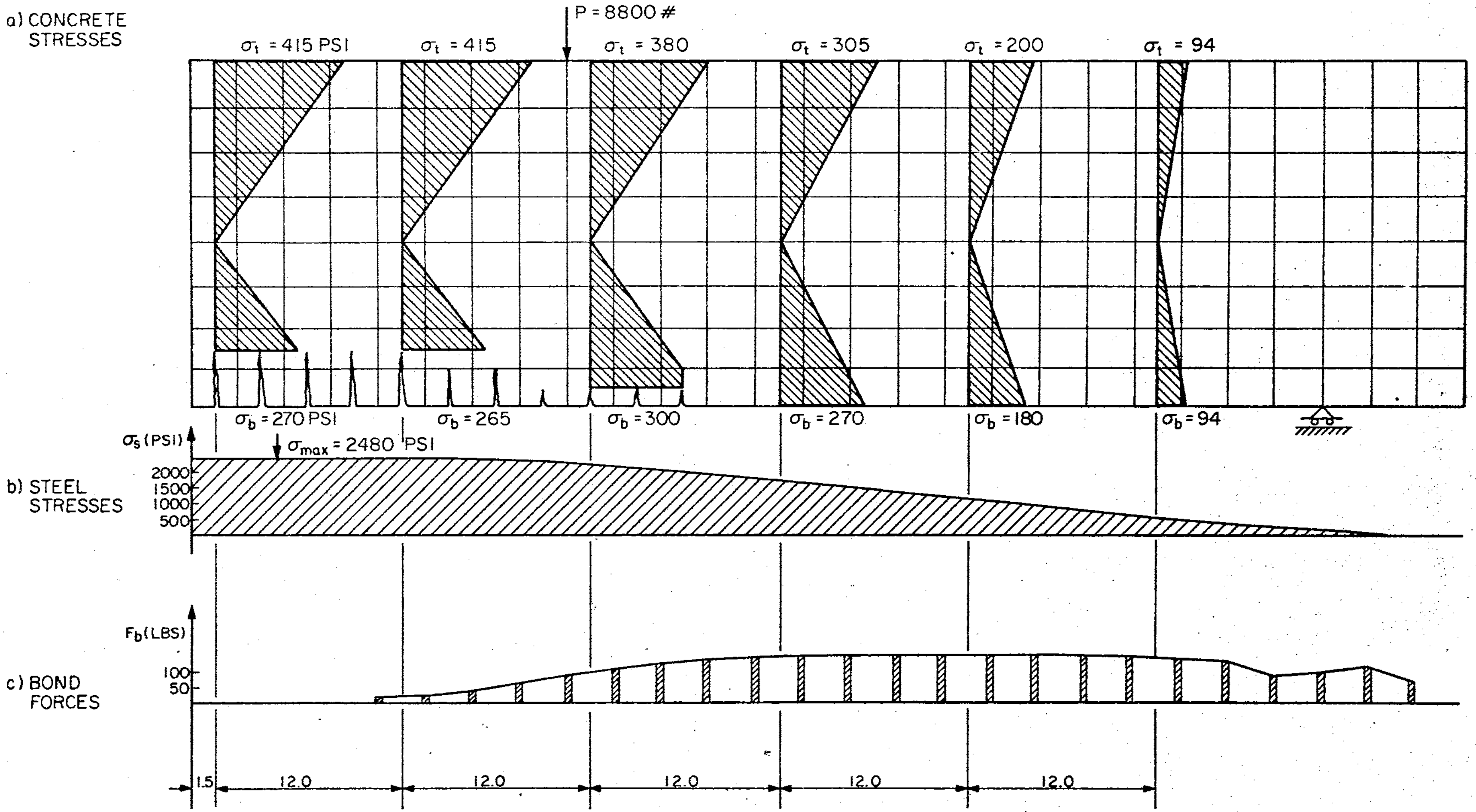


Figure 26. Example Problem 1: Stresses at P = 8800 Lbs.

As mentioned earlier, the introduction of new nodes for the reinforcement required the renumbering of all nodes. To illustrate the difference between the intitial mesh and modified nodal arrangement, the two numbering systems are shown in Appendix D accompanied by the listing of the data input statements.

5.3 Example Problem 2: Simple Beam Loaded at Midspan

A simply supported beam acted upon by a concentrated load at midspan was chosen to study the crack propagation in the concrete, Figure 27. This type of structure and load configuration seemed particularly suited for such an investigation, because relatively large cracks should be expected to develop near the center of the beam. An initial load of P = 1500 pounds was applied. The cracks appeared during the first load increment of ΔP = 100 lbs (i. e., at P = 1600 lbs). The cracking moment obtained from the simple beam theory was found to correspond to a load of P_c = 1564 lbs.

Figures 28 and 29 illustrate the crack pattern at different load levels. The cracked elements have been identified by shading. Small vertical cracks first appear in the bottom elements of the beam. During the next few load increments, new cracks occur in the elements above those which have already cracked. In addition, small vertical cracks appear along the bottom edge farther away from the load. For higher load levels, the cracked elements tend to group in the vicinity of the center of the beam. Near the bottom edge the cracks remain practically vertical. However, Figure 29 clearly indicates that the directions of the cracks in the higher elements begin to point toward the load. Local disturbances in the crack pattern may be observed in

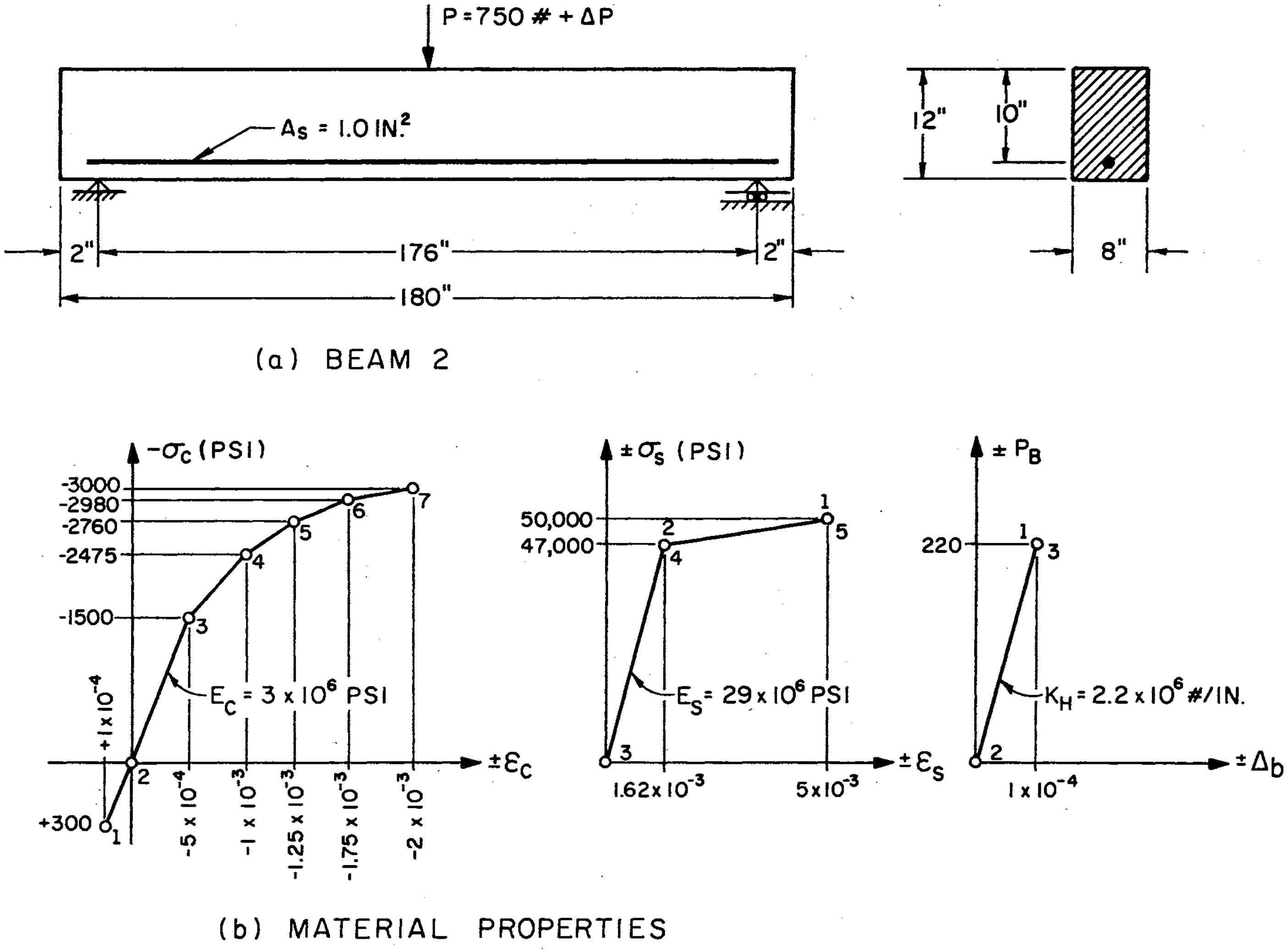


Figure 27. Example Problem 2: Simple Beam Loaded at Midspan

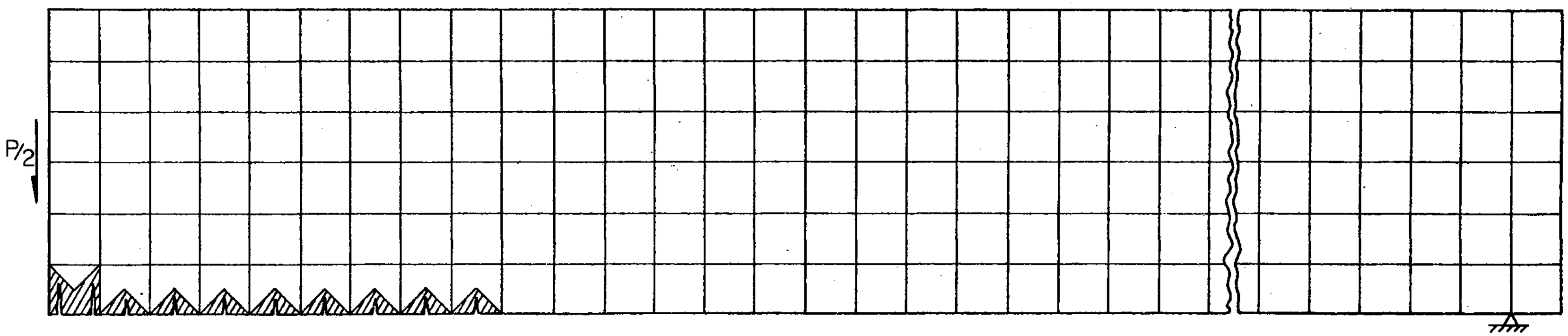


Figure 28. Example Problem 2: Crack Pattern at P = 2000 Lbs.

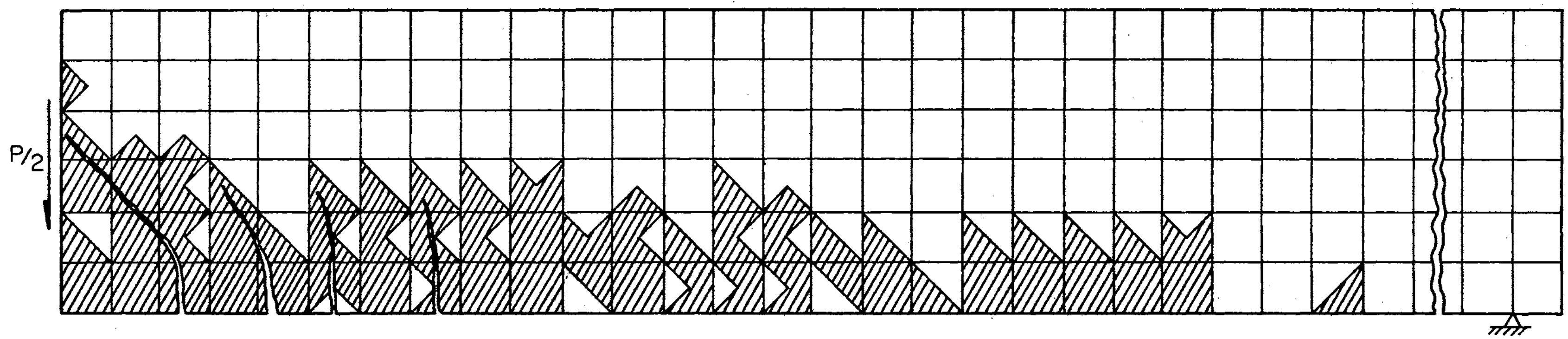


Figure 29. Example Problem 2: Crack Pattern at P = 4000 Lbs.

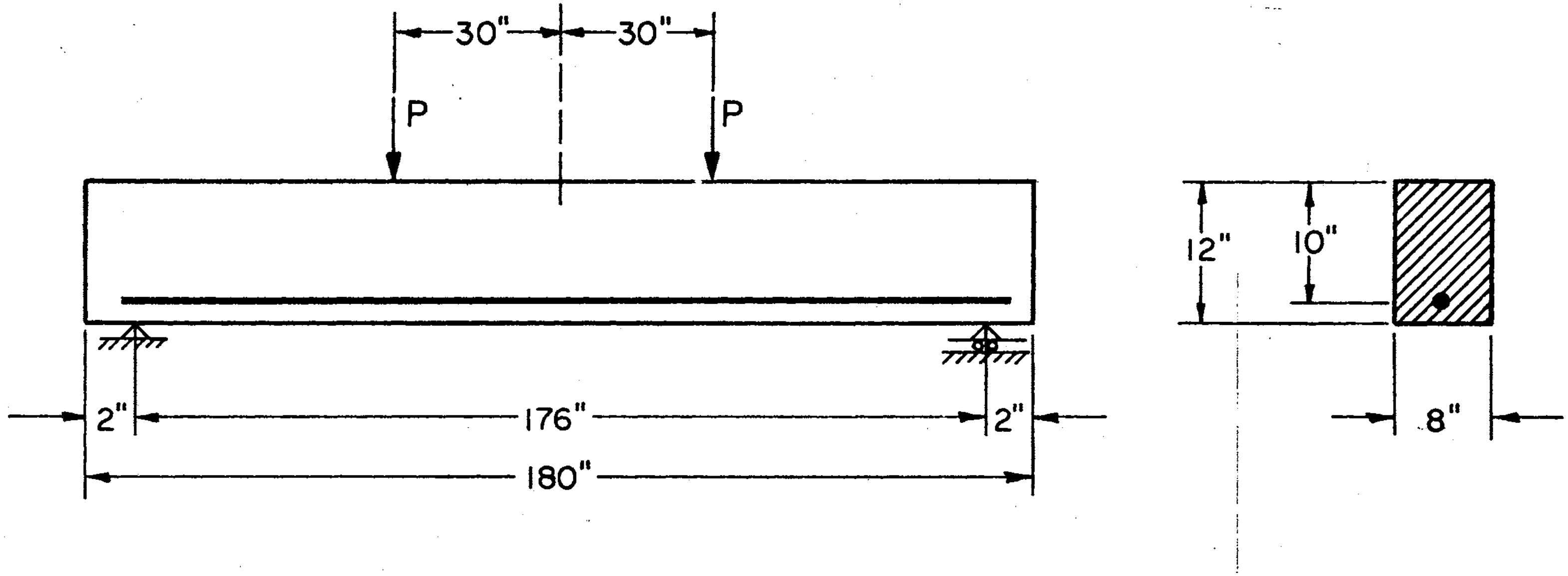
the neighborhood of the supports, especially the two triangular elements, six and seven, exhibit no cracks at all but experience relatively high compressive stresses in the x-direction. This irregularity may be caused by the vertical load at node four (the total concentrated load has been distributed over the nodes at the supports). Also, the steel reinforcement is fixed in the x-direction at support joint three. Thus, the applied vertical load at joint four must be transferred to the steel through the two concrete elements in question, causing high compressive stresses in these elements.

Several other elements at the bottom edge of the beam did not crack because of local stress redistributions. Element 124, for example, is obviously situated between two cracks. There the tensile stresses seem to have decreased enough in order not to cause cracking. Farther out in the beam the crack distribution becomes more regular with uncracked elements occurring more frequently.

5.4 Example Problem 3: Simple Beam Loaded Symmetrically by Two Concentrated Loads

A third, simple beam problem was solved to study the nonlinear stress distribution in the compressive zone of the concrete beam after cracking has taken place. The same structural model as in Problem 2 was loaded symmetrically by two concentrated loads at 30 inches from midspan, Figure 30.

Cracks developed again at the bottom edge and continued to extend vertically. The stress-distribution for only one load level will be reported to demonstrate the stress distribution in the compression zone. Due to the small number of relatively large elements in this problem, the stress distribution must be regarded as a crude approxi-



FOR MATERIAL PROPERTIES, SEE FIGURE 27.

Figure 30. Example Problem 3: Simple Beam Loaded Symmetrically by Two Concentrated Loads

mation. However, the results clearly show the nonlinear character of the concrete stress block above the neutral axis, Figure 31.

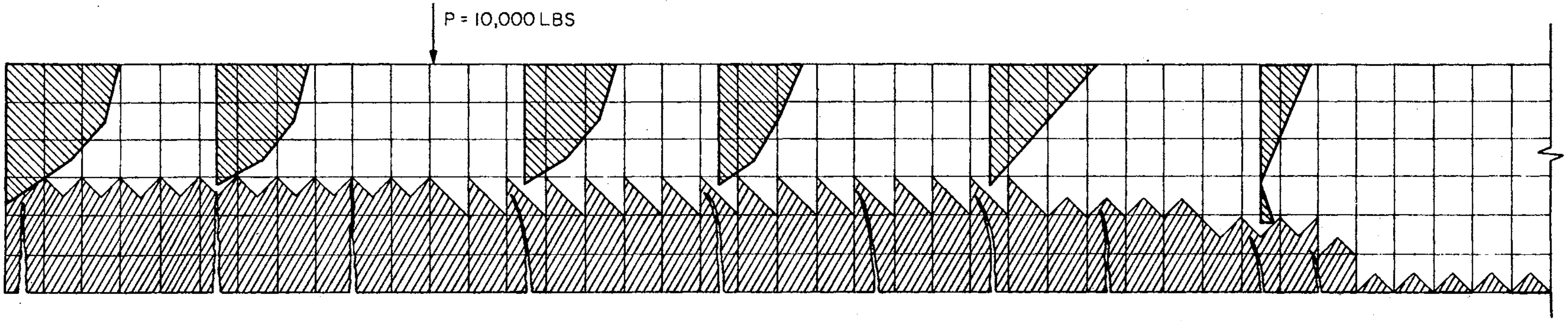


Figure 31. Example Problem 3: Crack Pattern and Stress Distribution at P = 10,000 Lbs.

CHAPTER VI

SUMMARY AND CONCLUSIONS

6.1 Summary

The feasibility of the finite element method in the investigation of reinforced concrete beams composed of Hookean material has been established by Scordelis (24). The objective of this thesis was to evaluate the potential of the finite element approach in the study of the nonlinear behavior of reinforced concrete beams under static loads and to provide a tool for the investigation of such structures.

The actual situation was approximated by a structural model of the plane stress type using a finite number of triangular, constant-strain, concrete elements, linear steel bars, and a mechanism to simulate the interaction between the two materials. In selecting the proper material constants for the concrete elements, the principal stresses were used to determine whether an element behaves isotropically, anisotropically or develops a crack. All material characteristics were replaced by piecewise linear stress-strain curves. The finite element approach was implemented in the form of a combined step-by-step, iterative procedure. Three example problems were solved on a digital computer.

6.2 Conclusions and Recommendations

The proposed finite element, step-by-step, iterative procedure is a feasible method to analyze reinforced, concrete beams. The simulation of the inelastic behavior by quasi-anisotropic, finite elements has shown satisfactory results. Likewise, the bond link models appear to approximate the bond slip phenomenon quite accurately. Two major difficulties had to be overcome in connection with the computer solution. First, the introduction of additional steel nodes presented some assembly problems. The use of a cumulative nodal list for the assemblage of such mathematical models proved to be extremely helpful. This technique allows the introduction of any number of additional nodes within any basic nodal configuration. Second, some difficulties were experienced with the bond link stiffness coefficients. To suppress relative displacements between steel and concrete nodes perpendicular to the steel bars, large numerical values for the vertical stiffness coefficients had to be used. These values must be selected with great caution. Extremely large numbers may cause completely erroneous solutions.

The convergence of the iterative process has been found to be slow for large load increments. On the other hand, too many small increments will result in excessive computer time. It would be advantageous to employ the solution procedure proposed by Zienkiewicz (20) which does not require the assemblage of the structural stiffness matrix for each iteration.

The present method may be recommended for extension to include time dependent effects such as creep or initial stress conditions resulting from temperature or prestressing forces. The method could be modified for three-dimensional assemblies. However, the need for larger digital machines becomes even more apparent for such models.

There is considerable doubt that the method could be used for dynamic loading conditions. Additional iteration cycles would probably increase the computer time tremendously. In addition, the program would have to be modified to include the possibility of unloading conditions.

BIBLIOGRAPHY

(1) Winter, George et al. Design of Concrete Structures, 7th ed. New York: McGraw-Hill Book Company, 1968.

(2) Wilson, E. L. "Matrix Analysis of Nonlinear Structures." Proceedings, Second Conference on Electronic Computation, Pittsburgh, Pa., September 8-9, 1960, 415-428.

(3) Goldberg, J. E. and R. M. Richard. "Analysis of Nonlinear Structures." Journal of the Structural Division, ASCE, Vol. 89, No. ST4 (August, 1963), 333-351.

(4) Wilson, E. L. "Fine Element Analysis of Two-Dimensional Structures." Structural Engineering Laboratory Report No. 63-2, University of California at Berkeley, June, 1963.

(5) Argyris, J. H. et al. "Matrix Methods of Structural Analysis--A Precis of Recent Developments." Matrix Methods in Structural Analysis--AGARDograph 72. Ed. Fraeijs de Veubeke, Oxford: Pergamon Press, 1964.

(6) Denke, P. H. "Digital Analysis of Nonlinear Structures by the Force Method." Matrix Methods in Structural Analysis--AGARDograph 72. Ed. Fraeijs de Veubeke. Oxford: Pergamon Press, 1964.

(7) Pope, G. G. "The Application of the Matrix Displacement Method in Plane Elasto-Plastic Problems." Proceedings, Conference on Matrix Methods in Structural Mechanics. Ed. J. S. Przemieniecki et al. AFFDL TR 66-80, Wright-Patterson AFB, Ohio, November, 1966, 635-654. (Conference held in October, 1965.)

(8) Argyris, J. H. "Elasto-Plastic Displacement Analysis of Three-Dimensional Continua." Journal of the Royal Aeronautical Society, Vol. 69 (September, 1965), 633-636.

(9) Argyris, J. H. "Continua and Discontinua." Proceedings, Conference on Matrix Methods in Structural Mechanics. Ed. J. S. Przemieniecki et al. AFFDL TR 66 80, Wright-Patterson AFB, Ohio, November, 1966, 11-189. (Conference held in October, 1965.)

(10) Marcal, P. V. "A Stiffness Method for Elastic-Plastic Problems." International Journal of Mechanical Science, Vol. 7 (1965), 229-238.

(11) Marcal, P. V. and I. P. King. "Elastic-Plastic Analysis of Two-Dimensional Stress Systems by the Finite Element Method." International Journal of Mechanical Sciences, Vol. 9 (1967), 143-155.

(12) Zienkiewicz, O. C. and Y. K. Cheung. The Finite Element Method in Structural and Continuum Mechanics. London: McGraw-Hill Book Company, Ltd., 1967.

(13) Felippa, C. A. "Refined Finite Element Analysis of Linear and Nonlinear Two-Dimensional Structures." Structures and Materials Research Report No. 66-22, the University of California at Berkeley, October, 1966.

(14) Akyuz, F. A. "On the Solution of Two-Dimensional Problems by the Finite Element and Direct Stiffness Method." Proceedings, Fifth AIAA Aerospace Science Meeting, AIAA Paper No. 67-144, January 23-26, 1967.

(15) Akyuz, F. A. and J. E. Merwin. "Solution of Nonlinear Problem of Elasto-Plasticity by the Finite Element Method." AIAA Journal, Vol. 6 (October, 1968), 1825-1831.

(16) Marcal, P. V. "A Comparative Study of Numerical Methods of Elasto-Plastic Analysis." AIAA Journal, Vol. 6 (January, 1968), 157-158.

(17) Marcal, P. V. "Finite Element Analysis with Material Nonlinearities." Proceedings, Japan-U.S. Seminar on Matrix Methods in Structural Analysis and Design, 1969, 257-282.

(18) Yamada, Y. "Recent Japanese Developments in Matrix Displacement Methods for Elasto-Plastic Problems." Proceedings, Japan-U.S. Seminar on Matrix Methods in Structural Analysis and Design, 1969, 283-316.

(19) Yamada, Y. et al. "Analysis of the Elastic-Plastic Problems by the Matrix Displacement Method." Proceedings, Second Conference on Matrix Methods in Structural Mechanics, Wright-Patterson AFB, Ohio, 1968, 1271-1299.

(20) Zienkiewicz, O. C. et al. "Elasto-Plastic Solutions of Engineering Problems, 'Initial stress' Finite Element Approach." International Journal of Numerical Methods in Engineering, Vol. 1 (1969), 75-100.

(21) Oden, J. T. "Finite Element Applications in Nonlinear Structural Analysis." Proceedings, Symposium on the Application of Finite Element Methods in Civil Engineering, Nashville, Tenn., November 13-14, 1969, 419-456.

(22) Rashid, Y. R. "Analysis of Axisymmetric Composite Structures by the Finite Element Method." Nuclear Engineering and Design, Vol. 3 (1966), 163-182.

(23) Rashid, Y. R. "Ultimate Strength Analysis of Prestressed Concrete Pressure Vessels." Nuclear Engineering and Design, Vol. 7 (1968), 334-344.

(24) Scordelis, A. C. and D. Ngo. "Finite Element Analysis of Reinforced Concrete Beams." ACI Journal, Vol. 64 (March, 1967), 152-163.

(25) Cornell, D. C. et al. "SAFE-3D; A Computer Program for the Three-Dimensional Stress Analysis of Composite Structures." Atomic Energy Commission Research and Development Report No. GA-7855, September, 1967.

(26) Corum, J. M. and N. Krishnamurthy. "A Three-Dimensional Finite Element Analysis of a Prestressed Concrete Reactor Vessel Model." Proceedings, Symposium on the Application of the Finite Element Method in Civil Engineering, ASCE, Vanderbilt University, Nashville, Tenn., November 13-14, 1969, 63-94.

(27) Cervenka, V. "Inelastic Finite Element Analysis of Reinforced Concrete Panels Under In-Plane Loading." Ph. D. Dissertation, University of Colorado, 1970.

(28) Clough, R. W. "The Finite Element in Plane Stress Analysis." Proceedings, Second ASCE Conference on Electronic Computation, Pittsburgh, Pa., September, 1960.

(29) Pian, T. H. H. and P. Tong. "Basis of Finite Element Methods for Solid Continua." International Journal for Numerical Methods in Engineering, Vol. 1 (1969), 3-28.

(30) Zienkiewicz, O. C. and G. S. Holister. Stress Analysis, Recent Development in Numerical and Experimental Methods. London: John Wiley and Sons, Ltd., 1965.

(31) Przemieniecky, T. S. Theory of Matrix Structural Analysis. New York: McGraw-Hill Book Company, 1968.

(32) Mendelson, A. Plasticity; Theory and Application. New York: The MacMillan Company, 1968.

(33) Richart, F. E. et al. "A Study of the Failure of Concrete Under Combined Compressive Stresses." Bulletin 185, Engineering Experiment Station, University of Illinois, 1958.

(34) Lekhnitskii, S. G. Anisotropic Plates. New York: Gordon and Breach, Science Publishers, 1968.

(35) Rehm, G. "Über die Grundlagen des Verbundes Zwischen Stahl und Beton." Deutscher Ausschuss für Stahlbeton, Heft Nr. 138, Berlin, 1961.

(36) Washizu, K. Variational Methods in Elasticity and Plasticity. Oxford: Pergamon Press, 1968.

(37) Peattie, K. R. and T. A. Pope, "Tests of the Bond Between Concrete and Steel." Civil Engineering and Public Works Review 51, 1956.

(38) Turner, M. J. et al. "Large Deflections of Structures Subjected to Heating and External Loads." Journal of Aerospace Sciences, Vol. 27 (1960), 97-102.

(39) Gallagher, R. H. et al. "Stress Analysis of Heated Complex Shapes." Journal of the American Rocket Society, Vol. 32 (1962), 700-707.

(40) Zurmühl, R. Praktische Mathematik für Ingenieure und Physiker. Fünfte Auflage. Berlin: Springer-Verlag, 1965.

APPENDIX A

STIFFNESS MATRICES IN TABLE FORM

A.1 Isotropic Stiffness Matrix for Triangular Concrete Panels

Using the numbering system shown in Figure 10 and the strain-displacement transformation matrix from Equation (2.32), the following stiffness matrix results from Equation (2.35). For convenience, the matrix is separated into two component matrices (Equations (A.1) and (A.2)): $[K_s]$ represents the stiffness due to shear and $[K]$ contains terms due to normal stresses only.

A.2 Anisotropic Stiffness Matrix for Triangular Concrete Panels

The stiffness matrix (Equation (A.3) below) is in terms of local coordinates, the axes being u and v, Figure 7. Again, the nodes are numbered in clockwise directions. The matrix becomes much less complex for the principal axes, since $\overline{G}$ vanishes.

A.3 Stiffness Matrix of Steel Bars

Linear bar stiffness matrices are conveniently given in global coordinates directly, i.e., after the transformation (Equation 2.41) has been performed.

$$[K_s]g = \frac{EA}{L}\begin{bmatrix} c_x^2 & & & \\ c_x c_y & c_y^2 & & \\ -c_x^2 & -c_x c_y & c_x^2 & \\ -c_x c_y & -c_y^2 & c_x c_y & c_y^2 \end{bmatrix} \qquad (A.6)$$

where

$$
\left[K\right] = \frac{Et}{4A(1-\nu^2)} \begin{bmatrix}
y_{32}^2 & & & & & \\
-\nu y_{32}x_{32} & x_{32}^2 & & & & \\
-y_{32}y_{31} & \nu x_{32}y_{31} & y_{31}^2 & & & \\
\nu y_{32}x_{31} & -x_{32}x_{31} & -\nu y_{31}x_{31} & x_{31}^2 & & \\
y_{32}y_{21} & -\nu x_{32}y_{21} & -y_{31}y_{21} & \nu x_{31}y_{21} & y_{21}^2 & \\
-\nu y_{32}x_{21} & x_{32}x_{21} & \nu y_{31}x_{21} & -x_{31}x_{21} & -\nu y_{21}x_{21} & x_{21}^2
\end{bmatrix} \quad (A.1)
$$

$$
\left[K_s\right] = \frac{Et}{8A(1+\nu)} \begin{bmatrix}
x_{32}^2 & & & & & \\
-x_{32}y_{32} & y_{32}^2 & & & & \\
-x_{32}x_{31} & y_{32}x_{31} & x_{31}^2 & & & \\
x_{32}y_{31} & -y_{32}y_{31} & -x_{31}y_{31} & y_{31}^2 & & \\
x_{32}x_{21} & -y_{32}x_{21} & -x_{31}x_{21} & y_{31}x_{21} & x_{21}^2 & \\
-x_{32}y_{21} & y_{32}y_{21} & x_{31}y_{21} & -y_{31}y_{21} & -x_{21}y_{21} & y_{21}^2
\end{bmatrix} \quad (A.2
$$

$$
\left[K_a\right] = \frac{E_1 t}{4A(1-\nu_{12}\nu_{21})}
\begin{bmatrix}
\begin{array}{l} y_{32}^2 + \\ x_{32}^2\bar{G} \end{array} & & & & & \\
\begin{array}{l} -x_{32}y_{32}\nu_{21} \\ -x_{32}y_{32}\bar{G} \end{array} & \begin{array}{l} nx_{32}^2 \\ +y_{32}^2\bar{G} \end{array} & & & & \\
\begin{array}{l} -y_{32}y_{31} \\ -x_{32}x_{31}\bar{G} \end{array} & \begin{array}{l} x_{32}y_{31}\nu_{21} \\ +x_{31}y_{32}\bar{G} \end{array} & \begin{array}{l} y_{31}^2 \\ +x_{31}^2\bar{G} \end{array} & & & \\
\begin{array}{l} x_{31}y_{32}\nu_{21} \\ +x_{32}y_{31}\bar{G} \end{array} & \begin{array}{l} -x_{31}x_{32}n \\ -y_{31}y_{32}\bar{G} \end{array} & \begin{array}{l} -x_{31}y_{31}\nu_{21} \\ -x_{31}y_{31}\bar{G} \end{array} & \begin{array}{l} x_{31}^2 n \\ +y_{31}^2\bar{G} \end{array} & & \\
\begin{array}{l} y_{21}y_{32} \\ +x_{21}x_{32}\bar{G} \end{array} & \begin{array}{l} -x_{32}y_{21}\nu_{21} \\ -x_{21}y_{32}\bar{G} \end{array} & \begin{array}{l} -y_{21}y_{31} \\ -x_{21}x_{31}\bar{G} \end{array} & \begin{array}{l} x_{31}y_{21}\nu_{21} \\ +x_{21}y_{31}\bar{G} \end{array} & \begin{array}{l} y_{21}^2 \\ +x_{21}^2\bar{G} \end{array} & \\
\begin{array}{l} -x_{21}y_{32}\nu_{21} \\ -x_{32}y_{21}\bar{G} \end{array} & \begin{array}{l} x_{32}x_{21}n \\ +y_{32}y_{21}\bar{G} \end{array} & \begin{array}{l} x_{21}y_{31}\nu_{21} \\ +x_{31}y_{21}\bar{G} \end{array} & \begin{array}{l} -x_{21}x_{31}n_{21} \\ -y_{21}y_{31}\bar{G} \end{array} & \begin{array}{l} -x_{21}y_{21}\nu_{21} \\ -x_{21}y_{21}\bar{G} \end{array} & \begin{array}{l} x_{21}^2 n \\ +y_{21}^2\bar{G} \end{array}
\end{bmatrix}
\tag{A.3}
$$

where

$$
G = \frac{\bar{G}_{21}}{E_1}(1 - \nu_{12}\nu_{21}) \quad \text{and} \quad G_{21} = \left\{\left[\frac{E_1}{2(1+\nu_{21})}\right]\left[\frac{E_2}{2(1+\nu_{12})}\right]\right\}^{\frac{1}{2}} \tag{A.4}
$$

$$
n = \frac{E_2}{E_1} \tag{A.5}
$$

$$c_x = \cos\theta \tag{A.7}$$

$$c_y = \sin\theta \tag{A.8}$$

A.4 Bond Link Stiffness Matrix

Similar to the "two-force" members, the bond link matrices are given in the datum system directly.

$$[K_b]g = \begin{bmatrix} c_x^2k_1 + s_x^2k_2 & & & \\ s_xc_xk_1 - s_xc_xk_2 & s_x^2k_1 + c_x^2k_2 & & \\ -c_x^2k_1 - s_x^2k_2 & -s_xc_xk_1 + s_xc_xk_2 & c_x^2k_1 + s_x^2k_2 & \\ -s_xc_xk + s_xc_xk & -s_x^2k_1 - c_x^2k_2 & s_xc_xk - s_xc_xk_1 & s_x^2k_2 + c_x^2k_2 \end{bmatrix} \tag{A.9}$$

A.5 Cracked Concrete Element

$$[K_u] = \begin{bmatrix} 0 & & & & & \\ 0 & & & & & \text{symmetric} \\ 0 & x_{32}^2 & 0 & & & \\ 0 & -x_{32}x_{31} & 0 & & & \\ 0 & 0 & 0 & & 0 & \\ 0 & x_{21}x_{32} & 0 & -x_{21}y_{31} & 0 & x_{21}^2 \end{bmatrix} \tag{A.10}$$

APPENDIX B

LISTING OF COMPUTER PROGRAM

```
C   ..............................................................
C   .                                                            .
C   .   PROGRAM NARCOS                                           .
C   .   NONLINEAR ANALYSIS OF REINFORCED CONCRETE STRUCTURES     .
C   .                                                            .
C   .   LANGUAGE USED       :  FORTRAN IV                        .
C   .   DIGITAL MACHINE     :  IBM 360-65                        .
C   .   PROGRAMMER          :  ALEXANDER J. LASSKER              .
C   .   DATE OF COMPLETION  :  JUNE 30, 1971                     .
C   .                                                            .
C   .   DESCRIPTION OF PROGRAM                                   .
C   .                                                            .
C   . THIS PROGRAM SOLVES REINFORCED CONCRETE STRUCTURES OF THE  .
C   . PLANE STRESS TYPE. THE FINITE ELEMENT METHOD IS USED IN AN .
C   . ITERATIVE PROCEDURE. AT EACH LOAD STEP THE PROBLEM IS SOLVED .
C   . AS AN ELASTIC PROBLEM. DETAILED INFORMATION CAN BE FOUND IN: .
C   . NONLINEAR ANALYSIS OF REINFORCED CONCRETE BEAMS UNDER STATIC .
C   . LOADS. PH.D. DISSERTATION BY A.J. LASSKER, OSU, AUGUST 1971. .
C   .                                                            .
C   ..............................................................
C   .                                                            .
C   .   PROBLEM PARAMETERS USED IN THIS PROGRAM                  .
C   .                                                            .
C   . NN      = BLOCK LENGTH                                     .
C   . MA      = BLOCK WIDTH                                      .
C   . NH      = LENGTH OF WORKING AREA                           .
C   . NREL    = NUMBER OF RECTANGULAR ELEMENTS                   .
C   . NTEL    = NUMBER OF TRIANGULAR ELEMENTS                    .
C   . NODV    = NUMBER OF NODES VERTICALLY                       .
C   . NODH    = NUMBER OF NODES HORIZONTALLY                     .
C   . NDF     = NUMBER OF DEGREES OF FREEDOM AT ONE NODE         .
C   .                                                            .
C   . X(NUMNOD)          = X - COORDINATES                       .
C   . Y(NUMNOD)          = Y - COORDINATES                       .
C   . THETA(NTEL)        = ANGLE OF PRINCIPAL AXES               .
C   . JTOP(NREL,4)       = TOPOLOGY OF CONCRETE ELEMENTS         .
C   .                      CONTAINS NODE NUMBERS IN CLOCKWISE DIRECTION .
C   . THICK(NREL)        = THICKNESS OF RECTANGULAR ELEMETS      .
C   . ITYPE(NTEL)        = TYPE OF BEHAVIOR OF CONCRETE ELEMENTS .
C   . SEC(10,2)          = STRESS - STRAIN VECTOR FOR CONCRETE   .
C   .                      CONTAINS STRESSES IN COLUMN 1         .
C   .                      CONTAINS STRAINS IN COLUMN 2          .
C   . ES(9)              = MODULI OF ELASTICITY FOR CONCRETE     .
C   . XNUC(9)            = POISSON'S RATIOS FOR CONCRETE         .
C   . EPS(NTEL,3)        = STRAINS FOR TRIANGULAR ELEMENTS       .
C   . EPR(NTEL,2)        = PRINCIPAL STRAINS FOR TRIANGULAR ELEMENTS .
C   . STR(NTEL,3)        = STRESSES IN TRIANGULAR ELEMENTS       .
C   . STRP(NTEL,2)       = PRINCIPAL STRESSES FOR TRIANGULAR ELEMENTS .
C   . NLOAD  = NUMBER OF LOADING CARDS                           .
C   . NBCS   = NUMBER OF BOUNDARY CONDITIONS SPECIFIED ( X,Y DIR.) .
C   . XLOAD(NLOAD,3)     = LOADING VECTOR                        .
C   .                      CONTAINS NODE NUMBER IN COLUMN 1      .
C   .                      CONTAINS X - COMPONENT IN COLUMN 2    .
C   .                      CONTAINS Y - COMPONENT IN COLUMN 3    .
C   . XBOUND(NBCS,3)     = PRESCRIBED DISPLACEMENT VECTOR        .
C   . U(NUMNOD*NDF)      = NODAL DISPLACEMENTS FOR A LOAD INCREMENT .
C   . TU(NUMNOD*NDF)     = TOTAL NODAL DISPLACEMENTS             .
C   . UB(NUMNOD*NDF)     = AUXILIARY DISPLACEMENT VECTOR         .
C   . UI(NUMNOD*NDF)     = AUXILIARY TOTAL DISPLACEMENTS AT BEGINNING OF .
C   .                                A NEW LOAD INCREMENT        .
C   .                                                            .
C   . ITOP(NUMBAR,4)     = TOPOLOGY OF REINFORCEMENT BARS        .
C   .                      CONTAINS NODE NUMBERS FOR REINFORCEMENTS .
C   . ISTYPE(NUMBAR)     = TYPE OF BEHAVIOR OF STEEL ELEMENTS    .
C   . SES(20,2)          = STRESS - STRAIN VECTOR FOR STEEL      .
C   . ES(19)             = MODULI OF ELASTICITY FOR STEEL        .
C   . XNUS(19)           = POISSON'S RATIUS FOR STEEL            .
C   . SAREA(NUMBAR)      = STEEL AREA                            .
C   . ESPS(NUMBAR)       = STRAINS IN STEEL REINFORCEMENTS       .
C   . STRS(NUMBAR)       = STRESSES IN STEEL REINFORCEMENTS      .
C   .                                                            .
C   . A(NH,MA)           = WORKING AREA FOR MAIN STIFFNESS MATRIX .
C   . B(NH)              = CORRESPONDING WORKING SPACE FOR LOADS .
C   .                                                            .
C   .   CURRENT SIZE OF PROGRAM                                  .
C   .                                                            .
C   . NN     =   54                                              .
C   . NH     =  108                                              .
C   . MA     =   54         (MBAND IS MAX. 54)                   .
C   . NUMNOD =  620                                              .
C   . NREL   =  288                                              .
C   . NTEL   = 1152                                              .
C   . NUMBAR =   50                                              .
C   . MAXNDF = 1300                                              .
C   .                                                            .
C   . STRESS - STRAIN CURVE GIVEN BY NO MORE THAN 10 POINTS      .
C   . MODULI OF ELASTICITY GIVEN BY NO MORE THAN 9 POINTS        .
C   . POISSON'S RATIOS      GIVEN BY NO MORE THAN 9 POINTS       .
C   . NUMBER OF LOADS GIVEN IS LESS THAN 5                       .
C   . NUMBER OF BOUNDARY CONDITIONS GIVEN IS LESS THAN 5         .
C   .                                                            .
C   ..............................................................
C
      IMPLICIT  REAL*8(A-H,O-Z)
      REAL*8 DATAN2,DSIN,DCOS,DABS
      COMMON S(10,10),VKH,VKV
      COMMON DXX,DYY,E1,E2,CNU,CNU12,CNU21,EP1,EP2,SNU,PI
      COMMON NN,NH,MA,NODV,NODH,NUMNOD,NREL,NTEL,NUMBAR,MAXNDF,MBAND
      COMMON NINCR,KINCR,NCURVC,NCURVS,NZC,NZS,ISCALE,NDF,IAUTO
      COMMON NUMDF,NUMTDF,NIT,IT,NCURVB,NZB,NOD2
C
      COMMON/ELEM  / X(650),Y(650),THETA(1152,2),U(1300),TU(1300)
      COMMON/ELEM1 / JTOP(288,4),THICK(288)
      COMMON/POOL  / A(108,54),B(108),NUMBLK
      COMMON/TYP   / ITYPE(1152),ISTYPE(50),IBTYPE(100),KDIR
      COMMON/MODULC/ SEC(10,2),EC(9),XNUC(9),TOL
      COMMON/MODULB/ SEB(10,2),EB(9)
      COMMON/MODULS/ SES(20,2),ES(19),XNUS(19)
      COMMON/LOADS / XLOAD( 20,2),XBOUND( 20,2)
      COMMON/LOADS1/ ALOAD( 20,2),ABOUND( 20,2)
      COMMON/LOADS2/ ILOAD( 20),IBOUND( 20),NLOAD,NBCS
      COMMON/REINF / SAREA( 50),JCNL(650,2),ITOP( 50,6),NREINF,IS1,IS2
      COMMON/CONSTR/ STR(1152,3),STRP(1152,2),EPS(1152,3),EPR(1152,2)
      COMMON/CONSTI/ TSTR(1152,3),TEPS(1152,3)
      COMMON/STLSTR/ ESPS( 50),STRS( 50),TESPS( 50),TSTRS( 50,2)
      COMMON/BOND  / EPSB(100,2),STRB(100,2),TEPSB(100,2),TSTRB(100,2)
      DATA DVKV/1.0D+06/ , ZERO/0.0D+00/
      DIMENSION UB(1300),KA(10)
```

```
      EQUIVALENCE (KA(1),K1),(KA(2),K2),(KA(3),K3),(KA(4),K4)
      EQUIVALENCE (KA(5),K5),(KA(6),K6),(KA(7),K7),(KA(8),K8)
      EQUIVALENCE (KA(9),K9),(KA(10),K10)
C
C     READ NUMBER OF PROBLEMS
C
      READ(5,8000) NPROB
 8000 FORMAT(I5)
      KPROB = 0
      IF(NPROB.GT.0) GOTO 100
      WRITE(6,9000) NPROB
 9000 FORMAT(1H1,1X,'NUMBER OF PROBLEMS MUST BE GREATER THAN ZERO',/,1X,
     1'NUMBER OF PROBLEMS SPECIFIED = ',I5)
      CALL EXIT
C
C     EACH NEW PROBLEM STARTS AT STATEMENT NO. 100
C     INITIALIZE MAIN PROBLEM PARAMETERS
C
  100 CONTINUE
      KPROB = KPROB + 1
C
      CALL INITL
C
      ICHECK = 0
      CALL READIN(ICHECK)
C
      CALL OUTPUT
C
C     EACH NEW LOAD INCREMENT STARTS AT STATEMENT NO. 200
C
  200 CONTINUE
      IT = 0
      DO 210 L = 1,NUMTOF
  210 U(L) = ZERO
C
C     EACH ITERATION CYCLE STARTS AT STATEMENT NO. 300
C
  300 CONTINUE
      IT     = IT + 1
      I      = 0
      IS     = 0
      NUMBLK = 0
      KSHIFT = 0
      ISWICH = 1
      IL     = 1
      DO 305 IA = 1,NH
      B(IA) = ZERO
      DO 305 JA = 1,MA
  305 A(IA,JA) = ZERO
      DO 310 KU = 1,NUMTOF
  310 UB(KU) = TU(KU) + U(KU)
C
      REWIND 1
      REWIND 2
C
C     SETUP BLOCKS OF THE MAIN STIFFNESS MATRIX
C     EACH BLOCK SETUP STARTS AT STATEMENT NO. 400
C
  400 CONTINUE
      NUMBLK = NUMBLK + 1
C
C     PROCESS ONE ELEMENT AT THE TIME
C
      MTYP = 1
  410 I  = I + 1
      J = 4*I - 3
      J1 = JTOP(I,1)
      J2 = JTOP(I,2)
      J3 = JTOP(I,3)
      J4 = JTOP(I,4)
      J5 = J1 + NODV
C
      K2 = 2*JCNL(J1,2)
      K1 = K2 - 1
      IF((K1 - KSHIFT) - NN) 430,430,420
  420 I  = I - 1
      GOTO 490
  430 K4 = 2*JCNL(J2,2)
      K3 = K4 - 1
      K6 = 2*JCNL(J3,2)
      K5 = K6 - 1
      K8 = 2*JCNL(J4,2)
      K7 = K8 - 1
      K10 = 2*JCNL(J5,2)
      K9 = K10 - 1
C
      CALL STIFF(1,MTYP,J,I,J1,J2,J5,1,2,5)
      J = J + 1
      CALL STIFF(2,MTYP,J,I,J2,J3,J5,2,3,5)
      J = J + 1
      CALL STIFF(2,MTYP,J,I,J5,J3,J4,5,3,4)
      J = J + 1
      CALL STIFF(2,MTYP,J,I,J1,J5,J4,1,5,4)
C
C     CHOSE PROPER VKV - VALUE
C
  470 IF(NREINF.EQ.0) GOTO 475
C
C     ASSEMBLY OF RECTANGULAR ELEMENT STIFFNESS MATRIX
C     FORM BLOCKS OF MAIN STIFFNESS MATRIX
C     SIZE OF WORKING AREA IS MA*NH
C     ONE BLOCK IS HALF OF THE WORKING AREA
C
  475 DO 480 KI = 1,10
      IINA = KA(KI) - KSHIFT
      KINA = KA(KI) - K1 + 1
      DO 480 KJ = 1,10
      JS  = KA(KJ) - K1 + 1
      IF(JS.LT.KINA) GOTO 480
      JINA = KA(KJ) - KA(KI) + 1
      A(IINA,JINA) = A(IINA,JINA) + S(KI,KJ)
  480 CONTINUE
      IF((VKV.EQ.0.0).AND.(I.EQ.1)) VKV = DVKV*A(1,1)
C
C     END OF LOOP FOR ELEMENT PROCESSING WITHIN ONE BLOCK
C     CHECK IF THE ELEMENT BELONGS TO THE CURRENT BLOCK
C
      IF(I.LT.NREL) GOTO 410
```

```
  490 CONTINUE
      IF ( NREINF .EQ. 0 ) GO TO 560
C
C     ASSEMBLY OF LINEAR BAR STIFFNESS MATRICES
C
      DO 500 IR = 1,10
      DO 500 JR = 1,10
  500 S(IR,JR) = 0.0
      DO 550 IS = 1,NUMBAR
      MTYP = 2
      ISWICH = 1
      ILINK = 0
      IS1 = ITOP(IS,3)
      IS2 = ITOP(IS,4)
      K2 = 2*ITOP(IS,5)
      K1  = K2 - 1
      K4 = 2*ITOP(IS,6)
      K3 = K4 - 1
      IF ( IS2 .LT. ( IS1 + NOD2 ) ) ISWICH = 2
C
C     CHECK IF BAR BELONGS TO CURRENT BLOCK
C
      IF((K1.LE.KSHIFT).OR.(K1.GT.(KSHIFT+NN))) GOTO 541
      CALL STIFF(ISWICH,MTYP,IS,IS,IS1,IS2,IS2,1,1,1)
C
C     ASSEMBLE
C
  535 DO 540 KI = 1,4
      IINA = KA(KI) - KSHIFT
      KINA = KA(KI) - K1 + 1
      DO 540 KJ = 1,4
      JS  = KA(KJ) - K1 + 1
      IF(JS.LT.KINA) GOTO 540
      JINA = KA(KJ) - KA(KI) + 1
      A(IINA,JINA) = A(IINA,JINA) + S(KI,KJ)
  540 CONTINUE
  541 ILINK = ILINK + 1
      GOTO (545,548,550) , ILINK
C
C     CHECK IF LINK BELONGS TO CURRENT BLOCK
C     IF IT DOES NOT, SKIP BOTH LINKS
C
  545 K11 = 2*JCNL(IS1,2) - 1
      IF((K11.LE.KSHIFT).OR.(K11.GT.(KSHIFT+NN))) GOTO 548
C
C     BOND LINK FOR START NODE OF STEEL BAR
C
  546 K44 = K4
      K33 = K3
      K4 = K2
      K3 = K1
      K1 = K11
      K2 = K11 + 1
      IV = IS
      GOTO 549
C
C     BOND LINK FOR END NODE OF STEEL BAR
C
  548 IF ( IS .NE. NUMBAR ) GO TO 550
      K2 = 2*JCNL(IS2,2)
      K1 = K2 - 1
      IF(K1.LE.KSHIFT.OR.K1.GT.(KSHIFT+NN)) GOTO 550
      K4 = K44
      K3 = K33
      IV = IS + 1
  549 CONTINUE
      II = 2 * IS - ISWICH + 1
      EP1 = TEPSB(II,1)
      MTYP = 3
      CALL STIFF ( ISWICH, MTYP, IV, IS, IS, IS, IS, 1, 1, 1)
      GOTO 535
  550 CONTINUE
C
C     PROCESS LOADS AND BOUNDARY CONDITIONS FOR EACH BLOCK
C     PUT CONCENTRATED LOADS FROM ARRAY XLOAD INTO B
C
  560 DO 580 IL = 1,NLOAD
      JL = 2*JCNL(ILOAD(IL),2) - KSHIFT
      IF(((JL-1).GT.NN).OR.((JL-1).LE.0)) GOTO 570
      B(JL-1) = XLOAD(IL,1)
  570 IF((JL.GT.NN).OR.(JL.LE.0)) GOTO 580
      B(JL)   = XLOAD(IL,2)
  580 CONTINUE
C
C     PROCESS BOUNDARY CONDITIONS
C     MODIFY EQUATIONS FOR SPECIFIED DISPLACEMENTS AT BOUNDARY
C     MODIFICATIONS FOR STEEL BOUNDARY CONDITIONS INCLUDED
C
      DO 600 IB = 1,NBCS
      IF(IBOUND(IB).LE.0) GOTO 590
      JB = 2*JCNL(IBOUND(IB),2) - KSHIFT
      IF(((JB-1).GT.NH).OR.((JB-1).LE.0)) GOTO 600
      CALL MODIFY((JB-1),XBOUND(IB,1))
      IF(JCNL(IBOUND(IB),1).EQ.0) GOTO 600
      IF(((JB+1).GT.NH).OR.((JB+1).LE.0)) GOTO 600
      CALL MODIFY((JB+1),XBOUND(IB,1))
      GOTO 600
  590 JBN = -IBOUND(IB)
      JB = 2*JCNL(JBN,2) - KSHIFT
      IF((JB.GT.NH).OR.(JB.LE.0)) GOTO 600
      CALL MODIFY(JB,XBOUND(IB,2))
      IF(JCNL(JBN,1).EQ.0) GOTO 600
      IF(((JB+2).GT.NH).OR.((JB+2).LE.0)) GOTO 600
      CALL MODIFY((JB+2),XBOUND(IB,2))
  600 CONTINUE
C
C     WRITE BLOCK ON TAPE 2 AND SHIFT LOWER PART INTO UPPER PART
C
      WRITE(2) (B(N),(A(N,M),M=1,MBAND),N=1,NN)
      DO 610 N = 1,NN
      K = N + NN
      B(N) = B(K)
      B(K) = 0.0
      DO 610 M = 1,MBAND
      A(N,M) = A(K,M)
  610 A(K,M) = 0.0
      KSHIFT = KSHIFT + NN
C
```

```
C     CHECK FOR LAST BLOCK
C     COMPLETE LOADS AND BOUNDARY CONDITIONS IN LAST BLOCK
C     IF I = NREL , INCREASE NUMBLK AND I BY 1
C
      IF(I - NREL) 400,620,630
  620 IF((2*JCNL(JTOP(I,3),2)).LE.KSHIFT) GOTO 630
      NUMBLK = NUMBLK + 1
      I      = I       + 1
      GOTO 560
C
C     SOLVE SYSTEM OF EQUATIONS BLOCKWISE BY GAUSS ELIMINATION
C
  630 CONTINUE
C
      CALL BANSOL
C
      KU  = 0
      NIN = NN + 1
      DO 640 NBU = 1,NUMBLK
      DO 640 NMU = NIN,NH
      KU  = KU + 1
  640 U(KU) = A(NMU,NBU)
C
C     CALCULATE STRAINS, PRINCIPAL STRAINS, STRESSES AND PRINCIPAL STR.
C
      DO 670 I = 1,NREL
      J = 4*I - 3
      J1 = JTOP(I,1)
      J2 = JTOP(I,2)
      J3 = JTOP(I,3)
      J4 = JTOP(I,4)
      J5 = J1 + NODV
      MTYP = 1
      CALL STRESS(I,J,J1,J2,J5,MTYP)
      J = J + 1
      CALL STRESS(I,J,J2,J3,J5,MTYP)
      J = J + 1
      CALL STRESS(I,J,J5,J3,J4,MTYP)
      J = J + 1
  670 CALL STRESS(I,J,J1,J5,J4,MTYP)
      IF(NREINF.EQ.0) GOTO 690
      MTYP = 2
      DO 680 J = 1,NUMBAR
      JS1 = ITOP(J,3)
      JS2 = ITOP(J,4)
  680 CALL STRESS(J,J,JS1,JS2,JS2,MTYP)
C
C     DETERMINE ITERATION PROCEDURE
C     SEARCH FOR LARGEST ERROR IN DISPLACEMENTS
C
  690 IF((NINCR.EQ.0).OR.(KINCR.EQ.1)) GOTO 720
      IF(IT.EQ.1) GOTO 300
      DIFF = ZERO
      LTOL = 1
      DO 700 L = 1,NUMTDF
      DIFF1 = DABS((TU(L) + U(L)) - UB(L))
      IF(DIFF1.LE.DIFF) GOTO 700
      DIFF = DIFF1
      LTOL = L
  700 CONTINUE
      DTOL = DABS(TOL*(TU(LTOL) + U(LTOL)))
      WRITE(6,9020) DTOL,LTOL
 9020 FORMAT(/,1X,'TOLERANCE = ',1PD12.4,'  AT DISPLACEMENT',I5)
      IF(DIFF.LE.DTOL) GOTO 720
  710 IF(IT.LT.NIT) GOTO 300
      WRITE(6,9100) NIT,DIFF,DTOL
 9100 FORMAT(/,1X,'ERROR IN ITERATION NO ',I3,' IS BIGGER THAN TOLERANCE
     1',/,1X,'ERROR = ',1PD12.5,3X,'TOLERANCE = ',1PD12.5,/,1X,'NUMBER O
     2F ITERATIONS INCREASED BY 5',/)
      NIT = NIT + 5
      IF(NIT.LT.15) GOTO 300
      WRITE(6,9200)
 9200 FORMAT(/////,1X,'SOLUTION STOPPED BECAUSE NIT EXEEDS 15')
      CALL EXIT
C
C     CALCULATE TOTAL STRAINS, STRESSES AND DISPLACEMENTS
C
  720 DO 800 KU = 1,NUMTDF
  800 TU(KU) = TU(KU) + U(KU)
      DO 810 KE = 1,NTEL
      DO 810 JE = 1,3
      TEPS(KE,JE) = TEPS(KE,JE) + EPS(KE,JE)
  810 TSTR(KE,JE) = TSTR(KE,JE) + STR(KE,JE)
      IF(NREINF.EQ.0) GOTO 8020
      DO 815 KS = 1,NUMBAR
      TESPS(KS) = TESPS(KS) + ESPS(KS)
  815 TSTRS(KS,1) = TSTRS(KS,1) + STRS(KS)
      NB11 = 2 * ( NUMBAR + 1 )
      DO 816 JS = 1,2
      DO 816 KS = 1, NB11
      TEPSB(KS,JS) = TEPSB(KS,JS) + EPSB(KS,JS)
  816 TSTRB(KS,JS) = TSTRB(KS,JS) + STRB(KS,JS)
 8020 CALL RESOUT(ICHECK)
C
      IF(NINCR.EQ.0) GOTO 850
      IF(NINCR.GT.1) GOTO 830
      IF(ISCALE) 830,820,830
C
C     SCALE LOADS AND PRESCRIBED DISPLACEMENTS
C
  820 ISCALE = 1
      CALL SCALE
      GOTO 200
C
C     PROCESS NEW LOAD INCREMENT
C
  830 IF(NINCR - KINCR) 460,850,840
  840 CALL SCALE2
      GOTO 200
  850 WRITE(6,9010) KPROB
 9010 FORMAT(//,1X,'END OF PROBLEM',I5)
      IF(KPROB.LT.NPROB) GOTO 100
  460 CALL EXIT
      END
      SUBROUTINE INITL
C
C   ...................................................................
C   .                                                                 .
```

```
C  .  THIS SUBROUTINE INITIALIZES ALL ARRAYS USED IN THE PROGRAM        .
C  .  DO LOOPS ARE MORE EFFICIENT THAN DATA STATEMENTS                  .
C  .                                                                    .
C  .  PARAMETERS USED:                                                  .
C  .                                                                    .
C  .  MUMNOD  =  NUMNOD                                                 .
C  .  MREL    =  NREL                                                   .
C  .  MTEL    =  NTEL                                                   .
C  .  MUMBAR  =  NUMBAR                                                 .
C  .  MLOAD   =  NLOADR                                                 .
C  .  MBCS    =  NBCS                                                   .
C  .                                                                    .
C  ......................................................................
C
      IMPLICIT  REAL*8 (A-H,O-Z)
      COMMON S(10,10),VKH,VKV
      COMMON DXX,DYY,E1,E2,CNU,CNU12,CNU21,EP1,EP2,SNU,PI
      COMMON NN,NH,MA,NODV,NODH,NUMNOD,NREL,NTEL,NUMBAR,MAXNDF,MBAND
      COMMON NINCR,KINCR,NCURVC,NCURVS,NZC,NZS,ISCALE,NDF,IAUTO
      COMMON NUMDF,NUMTDF,NIT,IT,NCURVB,NZB,NOD2
C
      COMMON/ELEM  / X(650),Y(650),THETA(1152,2),U(1300),TU(1300)
      COMMON/ELEM1 / JTOP(288,4),THICK(288)
      COMMON/POOL  / A(108,54),B(108),NUMBLK
      COMMON/TYP   / ITYPE(1152),ISTYPE(50),IBTYPE(100),KOIR
      COMMON/MODULC/ SEC(10,2),EC(9),XNUC(9),TOL
      COMMON/MODULS/ SES(20,2),ES(19),XNUS(19)
      COMMON/MODULB/ SEB(10,2),EB(9)
      COMMON/LOADS / XLOAD( 20,2),XBOUND( 20,2)
      COMMON/LOADS1/ ALOAD( 20,2),ABOUND( 20,2)
      COMMON/LOADS2/ ILOAD( 20),IBOUND( 20),NLOAD,NBCS
      COMMON/REINF / SAREA( 50),JCNL(650,2),ITOP( 50,6),NREINF,IS1,IS2
      COMMON/CUNSTR/ STR(1152,3),STRP(1152,2),EPS(1152,3),EPR(1152,2)
      COMMON/CONST1/ TSTR(1152,3),TEPS(1152,3)
      COMMON/STLSTR/ ESPS( 50),STRS( 50),TESPS( 50),TSTRS( 50,2)
      COMMON/BOND  / EPSB(100,2),STRB(100,2),TEPSB(100,2),TSTRB(100,2)
C
      PI      = 3.1415926535898
      NDF     =    2
      MUMTDN  =  650
      MUMNOD  =  650
      MREL    =  288
      MTEL    = 4 * MREL
      MUMBAR  =   50
      MLOAD   =   20
      MBCS    =   20
C
      KINCR   =    1
      ISCALE  =    0
      IAUTO   =    0
      ITYPES  =    1
      KTYP    =    1
      ITYP    =    1
C
      MA      =   54
      NN      =   54
      NH      =   NN + NN
      NUMTDF  =   MUMTDN*NDF
C
      DO 10 I = 1,MUMNOD
         JCNL(I,1)       =  0
         JCNL(I,2)       =  I
         X(I)            =  0.0
   10    Y(I)            =  0.0
      DO 20 I = 1,MREL
         THICK(I)        =  1.0
      DO 20 J = 1,4
   20    JTOP(I,J)       =  0
      DO 30 I = 1,NUMTDF
         TU(I)           =  0.0
   30    U(I)            =  0.0
      DO 50 I = 1,MTEL
         ITYPE(I)        =  1
      DO 40 J = 1,3
         EPS(I,J)        =  0.0
         TEPS(I,J)       =  0.0
         TSTR(I,J)       =  0.0
   40    STR(I,J)        =  0.0
      DO 50 J = 1,2
         THETA(I,J)      =  0.0
         EPR(I,J)        =  0.0
   50    STRP(I,J)       =  0.0
      DO 60 I = 1,NH
         B(I)            =  0.0
      DO 60 J = 1,MA
   60    A(I,J)          =  0.0
      DO 80 I = 1,MUMBAR
         ISTYPE(I)       =  1
      IBTYPE(I)          =  1
      IBTYPE(I+MUMBAR)   =  1
         SAREA(I)        =  0.0
         ESPS(I)         =  0.0
         STRS(I)         =  0.0
         TESPS(I)        =  0.0
         TSTRS(I,1)      =  0.0
         TSTRS(I,2)      =  0.0
      DO 80 J = 1,6
   80    ITOP(I,J)       =  0
      NUMBD = 2*MUMBAR
      DO 70 I = 1,NUMBD
      DO 70 J = 1,2
         EPSB(I,J)       =  0.0
         STRB(I,J)       =  0.0
         TEPSB(I,J)      =  0.0
   70    TSTRB(I,J)      =  0.0
      DO 90 I = 1,MLOAD
         ILOAD(I)        =  0
      DO 90 J = 1,2
         ALOAD(I,J)      =  0.0
   90    XLOAD(I,J)      =  0.0
      DO 100 I = 1,MBCS
         IBOUND(I)       =  0
      DO 100 J = 1,2
         XBOUND(I,J)     =  0.0
  100    ABOUND(I,J)     =  0.0
      DO 110 I = 1,9
         EC(I)           =  0.0
  110    XNUC(I)         =  0.0
```

```
      DO 120 I = 1,10
      DO 120 J = 1,2
      SEB(I,J)           =  0.0
  120    SEC(I,J)           =  0.0
      DO 130 I = 1,19
         ES(I)           =  0.0
  130    XNUS(I)         =  0.0
      DO 140 I = 1,20
      DO 140 J = 1,2
  140    SES(I,J)        =  0.0
C
      RETURN
      END
      SUBROUTINE READIN(ICHECK)
C
C     ..................................................................
C     .                                                                .
C     . SUBROUTINE READIN READS ALL INPUT INFORMATION AND DOES ALL     .
C     . NECESSARY AUTOMATIC NUMBERING PROCESSES                        .
C     .                                                                .
C     ..................................................................
C
      IMPLICIT   REAL*8(A-H,O-Z)
      COMMON S(10,10),VKH,VKV
      COMMON DXX,DYY,E1,E2,CNU,CNU12,CNU21,EP1,EP2,SNU,PI
      COMMON NN,NH,MA,NODV,NODH,NUMNOD,NREL,NTEL,NUMBAR,MAXNDF,MBAND
      COMMON NINCR,KINCR,NCURVC,NCURVS,NZC,NZS,ISCALE,NDF,IAUTO
      COMMON NUMDF,NUMTDF,NIT,IT,NCURVB,NZB,NOD2
C
      COMMON/ELEM  / X(650),Y(650),THETA(1152,2),U(1300),TU(1300)
      COMMON/ELEM1 / JTOP(288,4),THICK(288)
      COMMON/MODULC/ SEC(10,2),EC(9),XNUC(9),TOL
      COMMON/MODULS/ SES(20,2),ES(19),XNUS(19)
      COMMON/MODULB/ SEB(10,2),EB(9)
      COMMON/LOADS / XLOAD( 20,2),XBOUND( 20,2)
      COMMON/LOADS1/ ALOAD( 20,2),ABOUND( 20,2)
      COMMON/LOADS2/ ILOAD( 20),IBOUND( 20),NLOAD,NBCS
      COMMON/REINF / SAREA( 50),JCNL(650,2),ITOP( 50,6),NREINF,IS1,IS2
      DATA D20/1.0D-20/ , ZERO/0.0D 00/ , LABEL1/'X'/ , LABEL2/'Y'/
      DATA IEQUI/'EQUI'/
      DIMENSION NAME(18)
C
 8000 FORMAT(18A4)
 8010 FORMAT(4I5)
 8020 FORMAT(5I5)
 8030 FORMAT(I5,2D12.4)
 8040 FORMAT(I5,D12.4)
 8050 FORMAT(I5,D12.4,D12.6)
 8060 FORMAT(I5,D12.6)
 8070 FORMAT(I5,2D12.4)
 8080 FORMAT(I5,4X,A1,D12.4)
 8090 FORMAT(2I5,D12.4,2I5)
 8100 FORMAT(5X,D12.5)
 8110 FORMAT(A4)
 9000 FORMAT(1H1,//,1X,79(1H*),/,2H *,77X,1H*,/,1X,1H*,3X,18A4,2X,1H*,/,
     12H *,77X,1H*,/,2H *,78(1H*),//)
 9010 FORMAT(//,1X,'* FINITE ELEMENT ANALYSIS OF'/' * PLANE STRESS REINF
     1ORCED CONCRETE STRUCTURES'/)
 9020 FORMAT(1X,'* PROGRAM CHECKS EQUILIBRIUM'/)
```

```
 9030 FORMAT(//)
C
      READ(5,8000) (NAME(I),I=1,18)
      WRITE(6,9000) (NAME(I),I=1,18)
C
C     READ NODAL INFORMATION
C
      READ(5,8010) NNV,NNH,KTOP,NREINF
      IF(NREINF.LT.0) CALL PRER1(6)
      NODV = IABS(NNV)
      NODH = IABS(NNH)
      NELV = NODV - 1
      NELH = NODH - 1
      IF(KTOP.EQ.1) GOTO 30
      NREL = NELV*NELH
      NTEL = 4*NREL
      NUMNOD = NODV*NODH + NREL
      NOD2 = 2*NODV - 1
C
C     INDIVIDUEL INPUT OF TOPOLOGY
C
      DO 10 N = 1,NREL
   10 READ(5,8020) IP,(JTOP(IP,J),J=1,4)
      DO 20 N = 1,NUMNOD
   20 READ(5,8030) K,X(K),Y(K)
      READ(5,8010) MBAND
      GOTO 220
C
C     AUTOMATIC PROCESSING OF NODAL ARRANGEMENTS -
C
   30 READ(5,8040) JJ,DTEMP
      IF(NNV) 40,2000,60
C
C     EQUAL SPACING
C
   40 DD = DTEMP/NELV
      DDD = DD/2.0
      DO 50 N = 1,NELV
      Y(N) = (N - 1)*DD
   50 Y(NODV + N) = Y(N) + DDD
      Y(NODV) = NELV*DD
      GOTO 90
C
C     UNEQUAL SPACING
C
   60 L = 1
      D = 0.0
      DO 80 I = 1,NELV
      READ(5,8040) J,DYTEMP
      K  = J - JJ
      DD = (DYTEMP - DTEMP)/K
      DO 70 LK = 1,K
      Y(L) = D
      L = L + 1
   70 D = D + DD
      DTEMP = DYTEMP
   80 JJ = J
      Y(L) = D
      NODV = JJ
```

```
      NELV = NODV - 1
      NOD2 = 2*NODV - 1
      DO 85 LV = 1,NELV
   85 Y(LV + NODV) = 0.5*(Y(LV+1) - Y(LV)) + Y(LV)
   90 READ(5,8040) MM, DTEMP
      D = 0.
      IF(NNH) 100,2010,130
C
C     EQUAL SPACING
C
  100 D = DTEMP/NELH
      DDDD = D/2.0
      NI = 1
      DO 120 N = 1,NELH
      X(NI) = (N - 1)*D
C
C     UNEQUAL SPACING
      DO 110 NV = 1,NELV
      X(NI + NV) = X(NI)
  110 X(NI + NV + NELV) = X(NI) + DDDD
  120 NI = NI + 2*NODV - 1
      X(NI) = X(NI-1) + DDDD
      DO 125 NV = 1,NELV
  125 X(NI + NV) = X(NI)
      GOTO 170
C
  130 KSTART = 1
      KEND = NODV
      DO 160 I = 1,NELH
      READ(5,8040) M,DXTEMP
      K  = M - MM
      DD = (DXTEMP - DTEMP)/K
      DC = DD/2.0
      DO 150 LK = 1,K
      DO 140 L = KSTART,KEND
  140 X(L) = D
      IF(LK.EQ.1) GOTO 145
      DO 142 KS = 1,NELV
  142 X(KSTART - KS) = X(KSTART) - DC
  145 KSTART = KSTART + NOD2
      KEND = KSTART + NELV
  150 D  = D + DD
      NOD4 = NOD2 - 1
      KD = KEND - NOD2
      DO 155 KS = 1,NOD4
  155 X(KEND - KS) = X(KD - KS) + DD
      DTEMP = DXTEMP
  160 MM = M
      X(KEND) = X(KEND - 1)
      NODH = MM
      NELH = MM - 1
  170 NREL = NELV*NELH
      NTEL = 4*NREL
      NUMNOD = NODV*NODH + NREL
      NOD2 = 2*NODV - 1
      NUMDF  = NUMNOD*NDF
      NSTART = 0
      I = 1
  180 DO 190 K = 1,NELV
      NK  = NSTART + K
      NKK = NK + NOD2
      JTOP(I,1) = NK
      JTOP(I,2) = NK + 1
      JTOP(I,3) = NKK + 1
      JTOP(I,4) = NKK
  190 I = I + 1
      IF(I.GT.NREL) GOTO 200
      NSTART = NSTART + NOD2
      GOTO 180
C     CHECK NUMBERING
 2000 IPRER = 2
  999 CALL PRER1(IPRER)
 2010 IPRER = 1
      GOTO 999
 2020 IPRER = 8
      GOTO 999
  200 CONTINUE
      NOD3 = NOD2 + 1
      DO 210 KNS = NOD3,NUMNOD
  210 Y(KNS) = Y(KNS - NOD2)
  220 CONTINUE
C
C     READ STRESS - STRAIN LAWS
C
      READ(5,8010) NTH,NCURVC,NCURVS,NCURVB
      IF(NTH.EQ.0) GOTO 240
      READ(5,8040) NT,THIC
      IF(NT.GT.0) GOTO 211
      DO 21 N = 1,NREL
   21 THICK(N) = THIC
      GOTO 240
  211 IF(NT.EQ.1) GOTO 212
      DO 230 N = 2,NTH
  230 READ(5,8040) NT,THICK(NT)
      GOTO 240
  212 THICK(1) = THIC
  240 CONTINUE
C
C     CONCRETE
C
      DO 250 N = 1,NCURVC
      READ(5,8050) K,SEC(K,1),SEC(K,2)
  250 IF((SEC(N,1).EQ.0.0).AND.(SEC(N,2).EQ.0.0)) NZC = N
      NNC = NCURVC - 1
      DO 260 N = 1,NNC
      EC(N) = (SEC(N+1,1) - SEC(N,1))/(SEC(N+1,2) - SEC(N,2))
  260 READ(5,8060) K,XNUC(K)
      IF(NREINF.EQ.0) GOTO 490
C
C     STEEL
C
      DO 270 N = 1,NCURVS
      READ(5,8050) K,SES(K,1),SES(K,2)
  270 IF((SES(N,1).EQ.0.0).AND.(SES(N,2).EQ.0.0)) NZS = N
      NNS = NCURVS - 1
      DO 280 N = 1,NNS
      ES(N) = (SES(N+1,1) - SES(N,1))/(SES(N+1,2) - SES(N,2))
  280 READ(5,8060) K,XNUS(K)
```

```
C
C      BOND
C
       DO 290 N = 1,NCURVB
       READ(5,8050) K,SEB(K,1),SEB(K,2)
   290 IF((SEB(N,1).EQ.0.0).AND.(SEB(N,2).EQ.0.0)) NZB = N
       NNB = NCURVB - 1
       DO 300 N = 1,NNB
   300 EB(N) = (SEB(N+1,1) - SEB(N,1))/(SEB(N+1,2) - SEB(N,2))
       READ(5,8100) VKV
C
C      READ INFORMATION ABOUT REINFORCEMENT
C
       NUMBAR = 0
       DO 350 NJC = 1,NUMNOD
   350 JCNL(NJC,2) = 0
       DO 450 NR = 1,NREINF
       READ(5,8090) NBAR,NO,RAREA,NBEG,NEND
       IF(NEND - NBEG) 370,360,380
   360 KPRER1 = 3
       CALL PRER1(KPRER1)
   370 NNBEG = NBEG
       NBEG = NEND
       NEND = NNBEG
       KPRER2 = 5
       CALL PRER2(KPRER2)
   380 IF(DABS(RAREA).LT.D20) RAREA = ZERO
       IF(RAREA) 390,400,410
   390 KPRER1 = 4
       CALL PRER1(KPRER1)
   400 RAREA = NBAR*(0.25*PI*(NO*0.125)*(NO*0.125))
C      CHECK WHETHER HORIZONTAL OR VERTICAL BAR
   410 KEND = NBEG + NOD2
       IF(NEND.GE.KEND) GOTO 430
C      VERTICAL BAR
   420 NUMBAR = NUMBAR + 1
       ITOP(NUMBAR,1) = NBAR
       ITOP(NUMBAR,2) = NO
       ITOP(NUMBAR,3) = NBEG
       ITOP(NUMBAR,4) = NBEG + 1
       JCNL(NBEG,1) = 1
       JCNL(NBEG + 1,1) = 1
       SAREA(NUMBAR) = RAREA
       NBEG = NBEG + 1
       IF((NBEG+1) - NEND) 420,420,450
C      HORIZONTAL BAR
   430 NUMBAR = NUMBAR + 1
       ITOP(NUMBAR,1) = NBAR
       ITOP(NUMBAR,2) = NO
       ITOP(NUMBAR,3) = NBEG
       ITOP(NUMBAR,4) = KEND
       JCNL(NBEG,2) = 1
       JCNL(KEND,2) = 1
       SAREA(NUMBAR) = RAREA
       NBEG = KEND
       KEND = NBEG + NOD2
       IF(NEND.GE.KEND) GOTO 430
       IF(((NEND+NOD2) - KEND).EQ.0) GOTO 450
       KPRER1 = 5
       CALL PRER1(KPRER1)
       GOTO 430
   450 CONTINUE
       DO 460 N = 1,NUMNOD
   460 JCNL(N,1) = JCNL(N,1) + JCNL(N,2)
C
C      GENERATE CUMULATIVE NODE LIST
C
       JCNL(1,2) = 1
       DO 470 N = 2,NUMNOD
   470 JCNL(N,2) = JCNL(N-1,2) + JCNL(N-1,1) + 1
       DO 480 IS = 1,NUMBAR
       IBOND = 1
       JBOND = 1
       IS1 = ITOP(IS,3)
       IS2 = ITOP(IS,4)
       KS1 = JCNL(IS1,2)
       KS2 = JCNL(IS2,2)
       IF((JCNL(IS1,1).EQ.2).AND.(ITOP(IS,4).LT.(ITOP(IS,3) + NOD2)))
      1IBOND = 2
       IF((JCNL(IS2,1).EQ.2).AND.(ITOP(IS,4).LT.(ITOP(IS,3) + NOD2)))
      1JBOND = 2
       ITOP(IS,5) = KS1 + IBOND
   480 ITOP(IS,6) = KS2 + JBOND
C
   490 CONTINUE
C
C      READ TOLERANCE
C
       READ(5,8100) TOL
C
C      READ LOADING INFORMATION AND BOUNDARY CONDITIONS
C
       READ(5,8010) NLOAD,NBCS,NINCR,NIT
       DO 310 N = 1,NLOAD
       READ(5,8070) ILOAD(N),XLOAD(N,1),XLOAD(N,2)
       ALOAD(N,1) = XLOAD(N,1)
   310 ALOAD(N,2) = XLOAD(N,2)
       DO 320 N = 1,NBCS
       READ(5,8080) NIBND,IBTYP,VALUE
       IF(IBTYP.NE.LABEL1) GOTO 315
       IBOUND(N) = NIBND
       XBOUND(N,1) = VALUE
       GOTO 318
   315 IF(IBTYP.NE.LABEL2) CALL PRER1(9)
       IBOUND(N) = -NIBND
       XBOUND(N,2) = VALUE
   318 ABOUND(N,1) = XBOUND(N,1)
   320 ABOUND(N,2) = XBOUND(N,2)
C
C      DETERMINE BANDWIDTH
C
       MBAND = 0
       NK = 0
       DO 510 N = 1,NELH
       DO 500 K = 1,NELV
       NK = NK + 1
       JD = JCNL(JTOP(NK,3),2) - JCNL(JTOP(NK,1),2)
   500 IF(JD.GT.MBAND) MBAND = JD
```

```
      JD = JCNL(JTOP(NK,4),2) - JCNL(JTOP(NK,2),2)
  510 IF(JD.GT.MBAND) MBAND = JD
      MBAND = 2*MBAND + 2
      IF(MBAND.GT.MA) GOTO 2020
      NUMTDF = 2*JCNL(JTOP(NREL,3),2)
      READ(5,8110) ICHECK
      WRITE(6,9010)
      IF(ICHECK.EQ.IEQUI) WRITE(6,9020)
      WRITE(6,9030)
      RETURN
      END
      SUBROUTINE OUTPUT
C
C     ...............................................................
C     . THIS SUBROUTINE PRINTS ALL INITIAL INFORMATION IN TABULAR FORM    .
C     .                                                                   .
C     ...............................................................
      IMPLICIT  REAL*8(A-H,O-Z)
      COMMON S(10,10),VKH,VKV
      COMMON DXX,DYY,E1,E2,CNU,CNU12,CNU21,EP1,EP2,SNU,PI
      COMMON NN,NH,MA,NODV,NODH,NUMNOD,NREL,NTEL,NUMBAR,MAXNDF,MBAND
      COMMON NINCR,KINCR,NCURVC,NCURVS,NZC,NZS,ISCALE,NDF,IAUTO
      COMMON NUMDF,NUMTDF,NIT,IT,NCURVB,NZB,NOO2
C
      COMMON/ELEM  / X(650),Y(650),THETA(1152,2),U(1300),TU(1300)
      COMMON/ELEM1 / JTOP(288,4),THICK(288)
      COMMON/MODULC/ SEC(10,2),EC(9),XNUC(9),TOL
      COMMON/MODULS/ SES(20,2),ES(19),XNUS(19)
      COMMON/MODULB/ SEB(10,2),EB(9)
      COMMON/LOADS / XLOAD( 20,2),XBOUND( 20,2)
      COMMON/LOADS1/ ALOAD( 20,2),ABOUND( 20,2)
      COMMON/LOADS2/ ILOAD( 20),IBOUND( 20),NLOAD,NBCS
      COMMON/REINF / SAREA( 50),JCNL(650,2),ITOP( 50,6),NREINF,IS1,IS2
C
 9000 FORMAT(/1X,42(1H*)/43H * COORDINATES OF NODES                    *,
     1/,1X,42(1H*),/)
 9010 FORMAT(' NODE NO      X-COORDINATE       Y-COORDINATE',/)
 9020 FORMAT(1X,I5,6X,1PD12.4,5X,1PD12.4)
 9025 FORMAT(1X,I5,6X,1PD12.4,5X,1PD12.4,5X,'STEEL')
 9030 FORMAT(1H1,42(1H*)/43H * TOPOLOGICAL PROPERTIES OF ELEMENTS      *,
     1/,1X,42(1H*),/)
 9040 FORMAT(' ELEMENT NO   NODE A    NODE B    NODE C    NODE D    NODE M
     1  THICKNESS',/)
 9050 FORMAT(1H1,//)
 9060 FORMAT(I8,4X,I5,4(4X,I5),5X,1PD12.4)
 9070 FORMAT(///,' NO REINFORCEMENT IN THIS PROBLEM')
 9080 FORMAT(1H1,42(1H*)/43H * ARRANGEMENT OF REINFORCEMENT            *,
     1/,1X,42(1H*),/)
 9090 FORMAT(' BAR NUMBER    NO OF BARS    TYPE OF BARS     TOTAL AREA
     1 FROM JOINT TO JOINT',/)
 9100 FORMAT(1X,I6,11X,I3,11X,'NO',I3,6X,1PD12.4,5X,I4,5X,I4)
 9110 FORMAT(///,1X,'CALCULATED BANDWIDTH:   MBAND = ',I4)
 9120 FORMAT(1H1,42(1H*)/43H * STRESS - STRAIN LAWS                    *,
     1/,1X,42(1H*),/)
 9130 FORMAT(/,' CONCRETE:',I5,'  POINTS GIVEN',/)
 9140 FORMAT(' POINT      SIGMA IN PSI        EPSILON IN IN/INN      E-MODUL
     1       NU-VALUE       BETWEEN POINTS',/)
 9150 FORMAT(1X,I3,5X,1PD12.5,8X,1PD12.5,/,50X,1PD12.5,2X,0PD12.5,5X,I2,
     1' AND ', I2)
 9160 FORMAT(1X,I3,5X,1PD12.5,8X,1PD12.5,/)
 9170 FORMAT(/,' STEEL:',3X,I5,'  POINTS GIVEN',/)
 9180 FORMAT(/,' BOND:',4X,I5,'  POINTS GIVEN',/)
 9190 FORMAT(///,' NUMBER OF ITERATIONS =',I4,5X,'TOLERANCE = ',1PD12.5)
 9200 FORMAT(1H1,42(1H*)/43H * LOADING INFORMATION                     *,
     1/,1X,42(1H*),/)
 9210 FORMAT(1X,'LOADS FOR INCREMENT   NO',I5,/)
 9220 FORMAT (' NODE NO        X-LOAD',10X,'Y-LOAD          IN LBS' / )
 9230 FORMAT(1H1,42(1H*)/43H * BOUNDARY CONDITIONS                     *,
     1/,1X,42(1H*),/)
 9240 FORMAT(1X,'PRESCRIBED DISPLACEMENTS FOR INCREMENT    NO',I5,/)
 9250 FORMAT(' NODE NO        X-DISPL',7X,'  Y-DISPL     IN INCHES',/)
 9255 FORMAT(1X,I5,8X,1PD12.5,4X,1PD12.5)
 9260 FORMAT(1X,I5,8X,1PD12.5)
 9265 FORMAT(1X,I5,24X,1PD12.5)
 9270 FORMAT(///,1X,'ELASTIC SOLUTION ONLY')
 9280 FORMAT(///,1X,'AUTOMATIC SCALING',/,1X,'GIVEN LOADS AND DISPL. ARE
     1 ASSUMED AS TOTAL VALUES')
 9290 FORMAT(///,1X,'INDIVIDUAL LOAD AND DISPL. INPUT',/,1X,'PROGRAM REA
     1DS',I5,'   INCREMENTS')
C
C     PRINT NODAL INFORMATION
C
      WRITE(6,9000)
      WRITE(6,9010)
      KP = 0
      IPAGE = 36
      DO 10 NC = 1,NUMNOD
      IF(KP.NE.IPAGE) GOTO 5
      KP = 0
      IPAGE = 55
      WRITE(6,9050)
      WRITE(6,9010)
    5 KP = KP + 1
      WRITE(6,9020) JCNL(NC,2),X(NC),Y(NC)
      IF(JCNL(NC,1).EQ.0) GOTO 10
      KC = JCNL(NC,2) + 1
      WRITE(6,9025) KC,X(NC),Y(NC)
      KP = KP + 1
      IF(JCNL(NC,1).EQ.1) GOTO 10
      KC = KC + 1
      WRITE(6,9025) KC,X(NC),Y(NC)
      KP = KP + 1
   10 CONTINUE
C
C     PRINT TOPOLOGICAL QUANTITIES
C
      WRITE(6,9030)
      WRITE(6,9040)
      KP = 0
      IPAGE = 45
      DO 30 IP = 1,NREL
      IF(KP.NE.IPAGE) GOTO 20
      KP = 0
      IPAGE = 55
      WRITE(6,9050)
      WRITE(6,9040)
   20 KP = KP + 1
   30 WRITE(6,9060) IP,(JCNL(JTOP(IP,J),2),J=1,4),JCNL((JTOP(IP,1)+NODV)
```

```
     1,2),THICK(IP)
C
C     PRINT REINFORCEMENT INFORMATION
C
      IF(NREINF.NE.0) GOTO 40
      WRITE(6,9070)
      GOTO 60
   40 WRITE(6,9080)
      WRITE(6,9090)
      DO 50 IR = 1,NUMBAR
   50 WRITE(6,9100) IR,ITOP(IR,1),ITOP(IR,2),SAREA(IR),
     1ITOP(IR,5),ITOP(IR,6)
   60 CONTINUE
      WRITE(6,9110) MBAND
C
C     PRINT CONSTITUTIVE LAWS FOR CONCRETE AND STEEL
C
      WRITE(6,9120)
      WRITE(6,9130) NCURVC
      WRITE(6,9140)
      NCURV = NCURVC - 1
      DO 70 NC = 1,NCURV
      NE = NC + 1
   70 WRITE(6,9150) NC,SEC(NC,1),SEC(NC,2),EC(NC),XNUC(NC),NC,NE
      WRITE(6,9160) NE,SEC(NE,1),SEC(NE,2)
      IF(NREINF.EQ.0) GOTO 90
      WRITE(6,9120)
      WRITE(6,9170) NCURVS
      WRITE(6,9140)
      NCURV = NCURVS - 1
      DO 80 NS = 1,NCURV
      NE = NS + 1
   80 WRITE(6,9150) NS,SES(NS,1),SES(NS,2),ES(NS),XNUS(NS),NS,NE
      WRITE(6,9160) NE,SES(NE,1),SES(NE,2)
      WRITE(6,9120)
      WRITE(6,9180) NCURVB
      WRITE(6,9140)
      NCURV = NCURVB - 1
      DO 85 NB = 1,NCURV
      NE = NB + 1
   85 WRITE(6,9145) NB,SEB(NB,1),SEB(NB,2),EB(NB),NB,NE
      WRITE(6,9160) NE,SEB(NE,1),SEB(NE,2)
 9145 FORMAT(1X,I3,5X,1PD12.5,8X,1PD12.5,/,48X,'BOND STIFFNESS ',1PD12.5
     1,5X,I2,' AND ',I2)
      IF(VKV.EQ.0.0) GOTO 86
      WRITE(6,9155) VKV
 9155 FORMAT(//,1X,'SPECIFIED DOWEL ACTION STIFFNESS KV = ',1PD12.5)
      GOTO 90
   86 WRITE(6,9165)
 9165 FORMAT(//,1X,'DOWEL ACTION STIFFNESS CHOSEN AS (10**10)*S(1,1)')
C
C     PRINT ITERATION AND TOLERANCE VALUES
C
   90 WRITE(6,9190) NIT,TOL
      IF(NINCR - 1) 91,92,93
   91 WRITE(6,9270)
      GOTO 95
   92 WRITE(6,9280)
      GOTO 95
```

```
   93 WRITE(6,9290) NINCR
C
C     PRINT LOADING INFORMATION
C
   95 WRITE(6,9200)
      WRITE(6,9210) KINCR
      WRITE(6,9220)
      DO 100 IL = 1,NLOAD
  100 WRITE(6,9255) JCNL(ILOAD(IL),2) , XLOAD(IL,1) , XLOAD(IL,2)
C
C     PRINT BOUNDARY CONDITIONS
C
      WRITE(6,9230)
      WRITE(6,9240) KINCR
      WRITE(6,9250)
      DO 110 IB = 1,NBCS
      NIBND = IBOUND(IB)
      IF(NIBND.LE.0) GOTO 105
      WRITE(6,9260) JCNL(NIBND,2) , XBOUND(IB,1)
      GOTO 110
  105 NIBND = -NIBND
      WRITE(6,9265) JCNL(NIBND,2) , XBOUND(IB,2)
  110 CONTINUE
      RETURN
      END
      SUBROUTINE STIFF(ISWICH,MTYP,J,I,JJ1,JJ2,JJ3,K1,K2,K3)
C
C   ****************************************************************
C   *                                                              *
C   * THIS SUBROUTINE CALCULATES ALL ISOTROPIC AND ANISOTROPIC     *
C   * STIFFNESS MATRICES FOR THE CONCRETE PANELS, STEEL BARS AND   *
C   * AND BOND LINKS.                                              *
C   *                                                              *
C   *    PARAMETERS AND ARRAYS:                                    *
C   * S(8,8)    = STIFFNESS MATRIX                                 *
C   * SS(8,8)   = AUXILIARY STIFFNESS MATRIX                       *
C   * ST(8,8)   = AUXILIARY STIFFNESS MATRIX                       *
C   * ROT(8,8)  = ROTATIONAL TRANSFORMATION MATRIX                 *
C   *                                                              *
C   ****************************************************************
C
      IMPLICIT  REAL*8(A-H,O-Z)
      REAL*8 NUA,NUB,NUAB,NUAB2,DSQRT,DCOS,DSIN
      COMMON S(10,10),VKH,VKV
      COMMON DXX,DYY,E1,E2,CNU,CNU12,CNU21,EP1,EP2,SNU,PI
      COMMON NN,NH,MA,NODV,NODH,NUMNOD,NREL,NTEL,NUMBAR,MAXNDF,MBAND
      COMMON NINCR,KINCR,NCURVC,NCURVS,NZC,NZS,ISCALE,NDF,IAUTO
      COMMON NUMDF,NUMTDF,NIT,IT,NCURVB,NZB,NOD2
C
      COMMON/ELEM  / X(650),Y(650),THETA(1152,2),U(1300),TU(1300)
      COMMON/ELEM1 / JTOP(288,4),THICK(288)
      COMMON/TYP   / ITYPE(1152),ISTYPE(50),IBTYPE(100),KDIR
      COMMON/MODULC/ SEC(10,2),EC(9),XNUC(9),TOL
      COMMON/MODULS/ SES(20,2),ES(19),XNUS(19)
      COMMON/MODULB/ SEB(10,2),EB(9)
      COMMON/REINF / SAREA( 50),JCNL(650,2),ITOP( 50,6),NREINF,IS1,IS2
      DIMENSION ROT(10,10) , SS(10,10) , ST(10,10)
      DIMENSION H(10,3),BB(3,10),DD(3,3)
      DATA ZERO/0.0D+00/ , ROT/100*0.0D+00/ , D180/180.0D+00/
```

```
      DO 8 L = 1,3
      DO 9 M = 1,3
    9 DD(L,M) = ZERO
      DO 8 M = 1,10
    8 BB(L,M) = ZERO
      IF(NINCR.EQ.1) GOTO 19
      CALL TYPE(J,MTYP)
   19 CONTINUE
      GOTO (10,200,230) , MTYP
   10 IF(ISWICH.NE.1) GOTO 30
      DO 20 L = 1,10
      DO 20 M = 1,10
   20 S(L,M) = 0.0
   30 ITYP = ITYPE(J)
      IF(ITYP.GT.1) GOTO 35
      X32 = X(JJ3) - X(JJ2)
      X31 = X(JJ3) - X(JJ1)
      X21 = X(JJ2) - X(JJ1)
      Y32 = Y(JJ3) - Y(JJ2)
      Y31 = Y(JJ3) - Y(JJ1)
      Y21 = Y(JJ2) - Y(JJ1)
      GOTO 38
   35 OMEGA = THETA(J,1)*PI/D180
      OMEGA1 = DCOS(OMEGA)
      OMEGA2 = DSIN(OMEGA)
      CHI1 =  OMEGA1*X(JJ1) + OMEGA2*Y(JJ1)
      ETA1 = -OMEGA2*X(JJ1) + OMEGA1*Y(JJ1)
      CHI2 =  OMEGA1*X(JJ2) + OMEGA2*Y(JJ2)
      ETA2 = -OMEGA2*X(JJ2) + OMEGA1*Y(JJ2)
      CHI3 =  OMEGA1*X(JJ3) + OMEGA2*Y(JJ3)
      ETA3 = -OMEGA2*X(JJ3) + OMEGA1*Y(JJ3)
      X32 = CHI3 - CHI2
      Y32 = ETA3 - ETA2
      X31 = CHI3 - CHI1
      Y31 = ETA3 - ETA1
      X21 = CHI2 - CHI1
      Y21 = ETA2 - ETA1
   38 KJ2 = 2*K1
      KJ1 = KJ2 - 1
      KJ4 = 2*K2
      KJ3 = KJ4 - 1
      KJ6 = 2*K3
      KJ5 = KJ6 - 1
      A1 = 0.5*(X32*Y21 - X21*Y32)
      AR = 1.0/(2.0*A1)
      BB(1,KJ1) =  AR*Y32
      BB(1,KJ3) = -AR*Y31
      BB(1,KJ5) =  AR*Y21
      BB(2,KJ2) = -AR*X32
      BB(2,KJ4) =  AR*X31
      BB(2,KJ6) = -AR*X21
      BB(3,KJ1) =  BB(2,KJ2)
      BB(3,KJ2) =  BB(1,KJ1)
      BB(3,KJ3) =  BB(2,KJ4)
      BB(3,KJ4) =  BB(1,KJ3)
      BB(3,KJ5) =  BB(2,KJ6)
      BB(3,KJ6) =  BB(1,KJ5)
C
      IF(ITYP.GT.1) GOTO 100
```

```
C
C     ELASTIC STIFFNESS
C
      CN = E1*THICK(I)/(1.0 - CNU*CNU)
      DD(1,1) = CN*A1
      DD(1,2) = CN*CNU*A1
      DD(2,1) = DD(1,2)
      DD(2,2) = DD(1,1)
      DD(3,3) = CN*0.5*(1.0 - CNU)*A1
C
C     BBT*DD*BB
C
   50 DO 2 L = 1,10
      DO 2 M = 1,3
      H(L,M) = 0.0
      DO 2 K = 1,3
    2 H(L,M) = H(L,M) + BB(K,L)*DD(K,M)
      IF(ITYP.GT.1) GOTO 4
      DO 3 L = 1,10
      DO 3 M = 1,10
      DO 3 K = 1,3
    3 S(L,M) = S(L,M) + H(L,K)*BB(K,M)
      RETURN
    4 DO 5 L = 1,10
      DO 5 M = 1,10
      SS(L,M) = ZERO
      DO 5 K = 1,3
    5 SS(L,M) = SS(L,M) + H(L,K)*BB(K,M)
C
C     ROTATE ANISOTROPIC OR CRACKED STIFFNESS MATRIX
C
      DO 150 L = 1,10,2
      ROT(L,L)     = OMEGA1
      ROT(L,L+1)   = OMEGA2
      ROT(L+1,L)   =-OMEGA2
  150 ROT(L+1,L+1) = OMEGA1
      DO 170 L = 1,10
      DO 170 M = 1,10
      ST(L,M) = 0.0
      DO 160 LM = 1,10
  160 ST(L,M) = ST(L,M) + ROT(LM,L)*SS(LM,M)
  170 CONTINUE
      DO 190 L = 1,10
      DO 190 M = 1,10
      STEMP = 0.0
      DO 180 LM = 1,10
  180 STEMP = STEMP + ST(L,LM)*ROT(LM,M)
  190 S(L,M) = S(L,M) + STEMP
      RETURN
C
C     CRACKED ELEMENT
C
  100 CONTINUE
C
C     ANISOTROPIC OR CRACKED ELEMENT STIFFNESS MATRIX
C
      CA = 1.0/(1.0 - CNU12*CNU21)*THICK(I)
      DD(1,1) = E1*CA
      DD(1,2) = E1*CA*CNU12
```

```
      DD(2,1) = DD(1,2)
      DD(2,2) = E2*CA
      DD(3,3)= DSQRT((E1/(1.0+CNU12))*(E2/(1.0+CNU21))/4.0)*THICK(I)
      IF(ITYPE(J).EQ.4) DD(3,3) = ZERO
      GOTO 50
C
C     THIS PART OF STIFF CALCULATES THE MEMBER STIFFNESS MATRIX
C     FOR THE REINFORCEMENT BARS
C
  200 XX = X(JJ2) - X(JJ1)
      YY = Y(JJ2) - Y(JJ1)
      XLS =DSQRT(XX*XX + YY*YY)
      CX = XX/XLS
      CY = YY/XLS
      C = E1*SAREA(J)/XLS
      CX2 = CX*CX
      CY2 = CY*CY
      CXY = CX*CY
C
      S(1,1) = C*CX2
      S(1,2) = C*CXY
      S(1,3) = -S(1,1)
      S(1,4) = -S(1,2)
      S(2,2) = C*CY2
      S(2,3) = S(1,4)
      S(2,4) = -S(2,2)
      S(3,3) = S(1,1)
      S(3,4) = S(1,2)
      S(4,4) = S(2,2)
  210 DO 220 IR = 1,4
      DO 220 JR = 1,4
  220 S(JR,IR) = S(IR,JR)
      RETURN
C
C     STIFFNESS OF BOND LINKS
C     ISWICH = 1  HORIZONTAL
C     ISWICH = 2  VERTICAL
C
  230 DO 240 IB = 1,4
      DO 240 JB = 1,4
  240 S(IB,JB) = 0.0
      IF(ISWICH.EQ.1) GOTO 250
C     VERTICAL STEEL BAR
      CO = 0.0
      SI = 1.0
      GOTO 260
C     HORIZONTAL STEEL BAR
  250 CO = 1.0
      SI = 0.0
  260 C2 = CO*CO
      S2 = SI*SI
      SC = SI*CO
      S(1,1) =  VKH*C2 + VKV*S2
      S(1,2) =  VKH*SC - VKV*SC
      S(1,3) = -S(1,1)
      S(1,4) = -S(1,2)
      S(2,2) =  VKH*S2 + VKV*C2
      S(2,3) =  S(1,4)
      S(2,4) = -S(2,2)
      S(3,3) =  S(1,1)
      S(3,4) =  S(1,2)
      S(4,4) =  S(2,2)
      GOTO 210
C
      END
      SUBROUTINE TYPE(J,MTYP)
C
C     ..................................................................
C     .                                                                .
C     . THIS SUBROUTINE CALCULATES THE PROPER MODULI OF ELASTICITY     .
C     . AND ASSIGNS THE PARAMETER ITYPE TO THE ELEMENTS                .
C     . ITYPE = 1  ISOTROPIC, ELASTIC                                  .
C     . ITYPE = 2  ANISOTROPIC, ELASTIC                                .
C     . ITYPE = 3  ANISOTROPIC, PLASTIC                                .
C     .                                                                .
C     ..................................................................
C
C
      IMPLICIT  REAL*8(A-H,O-Z)
      COMMON S(10,10),VKH,VKV
      COMMON DXX,DYY,E1,E2,CNU,CNU12,CNU21,EP1,EP2,SNU,PI
      COMMON NN,NH,MA,NODV,NODH,NUMNOD,NREL,NTEL,NUMBAR,MAXNDF,MBAND
      COMMON NINCR,KINCR,NCURVC,NCURVS,NZC,NZS,ISCALE,NDF,IAUTO
      COMMON NUMDF,NUMTDF,NIT,IT,NCURVB,NZB,NOD2
      COMMON/TYP    / ITYPE(1152),ISTYPE(50),IBTYPE(100),KDIR
      COMMON/MODULC/ SEC(10,2),EC(9),XNUC(9),TOL
      COMMON/MODULS/ SES(20,2),ES(19),XNUS(19)
      COMMON/MODULB/ SEB(10,2),EB(9)
      COMMON/CONSTR/ STR(1152,3),STRP(1152,2),EPS(1152,3),EPR(1152,2)
      COMMON/STLSTR/ ESPS( 50),STRS( 50),TESPS( 50),TSTRS( 50,2)
      COMMON/BOND  / EPSB(100,2),STRB(100,2),TEPSB(100,2),TSTRB(100,2)
      DATA D10/1.0D-10/ , D20/1.0D-20/ , ZERO/0.0D+00/
C
 9000 FORMAT(//,1X,'CONCRETE ELEMENT NO.',I6)
 9010 FORMAT(//,1X,'STEEL ELEMENT NO.',I6)
C
C     NINCR = 1 CALLS FOR AUTOMATIC SCALING
C     TYPE MUST REMAIN UNCHANGED (ITYPE(J) = 1) FOR PROPER SCALING
C
      KPRER = 0
      IF(MTYP - 2) 10,120,210
C
C     DETERMINE TYPE OF CONCRETE ELEMENTS
C
   10 K   = 1
      I1  = NZC
      I2  = NZC
      EP1 = EPR(J,1)
      EP2 = EPR(J,2)
      GOTO 30
   20 K   = 2
      EP1 = EP2
      I1  = I2
   30 IF ( EP1 .GE. 0.0000 ) GO TO 70
      I2 = NZC - 1
   50 IF ( EP1 .GE. SEC(I2,2) ) GO TO 60
      I2 = I2 - 1
      IF(I2.GE.1) GOTO 50
```

```
      KPRER = 1
      I2    = 1
   60 IF(K-1) 65,20,100
   65 WRITE(6,2000) J, MTYP
 2000 FORMAT ( / ' STOP AT J =', I5, '  MTYP =', I3 )
      STOP
   70 I2 = NZC
      IF(EP1.LT.SEC(I2+1,2)) GOTO 90
C
C     CRACKED ELEMENT
C
      ITYPE(J) = 4
      E1  = ZERO
      CNU12 = ZERO
      I2 = NZC
       CNU21 = ZERO
      IF(EP2.LT.0.0D00) GOTO 75
      IF(EP2.LT.SEC(NCURVC,2)) GOTO 78
      WRITE(6,7310)
 7310 FORMAT(' DOUBLY CRACKED ELEMENT, PROGRAM STQPS')
      CALL EXIT
C     EP2 NEGATIVE
   75 I2 = NZC - 1
   76 IF(EP2.GE.SEC(I2,2)) GOTO 78
      I2 = I2 - 1
      IF(I2.GE.1) GOTO 76
      KPRER = 1
      I2 = 1
   78 E2 = EC(I2)
      GOTO 110
   90 IF(K.EQ.1) GOTO 20
  100 E2    = EC(I2)
      E1 = EC(I1)
      IF(E1.LT.E2) GOTO 101
C     ASSIGN NU12
      CNU12 = XNUC(I1)
      CNU21 = E2/E1*CNU12
      GOTO 106
C     ASSIGN NU21
  101 CNU21=XNUC(I2)
      CNU12 = E1/E2*CNU21
  106 ITYPE(J) = 3
      IF((I1.LT.(NZC-1)).OR.(I1.GE.(NZC+1))) GOTO 110
      IF((I2.LT.(NZC-1)).OR.(I2.GE.(NZC+1))) GOTO 110
      ITYPE(J) = 1
      IF((I1.EQ.NZC).AND.(I2.EQ.I1.OR.I2.EQ.NZC-1)) GO TO 110
      IF((I1.EQ.NZC-1).AND.(I2.EQ.I1.OR.I2.EQ.NZC)) GO TO 110
      ITYPE(J) = 2
  110 CONTINUE
      CNU        = CNU12
      IF(KPRER.EQ.0) RETURN
      WRITE(6,9000) J
      CALL PRER2(KPRER)
      RETURN
C
C     DETERMINE TYPE OF STEEL ELEMENTS
C
  120 I1    = NZS
      EP1   = TESPS(J)
      IF ( EP1 .GE. 0.0D00 ) GO TO 160
      I1 = NZS - 1
  140 IF ( EP1 .GE. SES(I1,2) ) GO TO 190
      I1 = I1 - 1
      IF(I1.GE.1) GOTO 140
      KPRER = 1
      I1    = 1
      GO TO 190
  160 I1    = NZS
  170 IF ( EP1 .LE. SES(I1+1,2) ) GO TO 190
      I1 = I1 + 1
      IF(I1.LE.(NCURVS-1)) GOTO 170
      KPRER = 2
      I1    = NCURVS - 1
  190 E1    = ES(I1)
      SNU   = XNUS(I1)
      ISTYPE(J) = 2
      IF ( ( I1 .EQ. NZS ) .OR. ( I1 .EQ. ( NZS-1) ) ) ISTYPE(J) = 1
      IF ( KPRER .EQ. 0 ) RETURN
      WRITE(6,9010) J
      CALL PRER2(KPRER)
      RETURN
C
C     DETERMINE TYPE OF BOND ELEMENTS
C
  210 I1 = NZB
      IF ( EP1 .GE. 0.0D00 ) GO TO 250
      I1 = NZB - 1
  230 IF ( EP1 .GE. SEB(I1,2) ) GO TO 280
      I1 = I1 - 1
      IF(I1 - 1) 270,230,230
  250 I1 = NZB
  260 IF ( EP1 .LE. SEB(I1+1,2) ) GO TO 280
      I1 = I1 + 1
      IF(I1.LE.(NCURVB-1)) GOTO 260
C
C     BOND LINK BREAKES DOWN, ASSIGN ZERO STIFFNESS
C     IN DIRECTION OF FAILURE
C
  270 VKH = ZERO
      IBTYPE(J) = 3
      RETURN
C
C     ACTING BOND LINK VKH = STIFFNESS PARALLEL TO BAR
C                      VKV = STIFFNESS PERPENDICULAR TO BAR,
C                            ASSIGNED FOR ALL LINKS IN MAIN
C
  280 VKH = EB(I1)
      IBTYPE(J) = 2
      IF((I1.EQ.(NZB-1)).OR.(I1.EQ.NZB)) IBTYPE(J) = 1
      RETURN
      END
      SUBROUTINE BANSOL
C
C     SUBROUTINE BANSOL SOLVES BANDED MATRICES BY THE GAUSS ELIMINATION
C     PROCEDURE. THE BANDMATRIX IS STORED DIRECTLY IN A RECTANGULAR
C     ARRAY AND TREATED BLOCKWISE.
C
      IMPLICIT  REAL*8(A-H,O-Z)
```

```
      COMMON S(10,10),VKH,VKV
      COMMON DXX,DYY,E1,E2,CNU,CNU12,CNU21,EP1,EP2,SNU,PI
      COMMON NN,NH,MA,NODV,NODH,NUMNOD,NREL,NTEL,NUMBAR,MAXNDF,MBAND
      COMMON NINCR,KINCR,NCURVC,NCURVS,NZC,NZS,ISCALE,NDF,IAUTO
      COMMON NUMDF,NUMTDF,NIT,IT,NCURVB,NZB,NOD2
      COMMON/POOL  / A(108,54),B(108),NUMBLK
      NB = 0
      NL = NN + 1
C
      REWIND 1
      REWIND 2
C
      GOTO 2
C
C     SHIFT LOWER PART OF B AND A INTO UPPER PART
C     CLEAR LOWER PART
C
    1 NB = NB + 1
      DO 3 N = 1,NN
      NM = NN + N
      B(N) = B(NM)
      B(NM) = 0.0
      DO 3 M = 1,MBAND
      A(N,M) = A(NM,M)
    3 A(NM,M) = 0.0
      IF(NUMBLK - NB) 2,4,2
C
C     READ LOADS AND STIFFNESS INTO LOWER PART OF A AND B
C
    2 READ(2) ( B(N),(A(N,M), M=1,MBAND), N=NL,NH)
      IF(NB) 4,1,4
C
C     REDUCE UPPER PART
C
    4 CONTINUE
      DO 5 N = 1,NN
      IF(A(N,1)) 6,5,6
    6 B(N) = B(N)/A(N,1)
      DO 7 L = 2,MBAND
      IF(A(N,L)) 8,7,8
    8 Q = A(N,L)/A(N,1)
      I = N + L - 1
      J = 0
      DO 9 K = L,MBAND
      J = J + 1
    9 A(I,J) = A(I,J) - Q*A(N,K)
      B(I)   = B(I) - A(N,L)*B(N)
      A(N,L) = Q
    7 CONTINUE
    5 CONTINUE
      IF(NUMBLK - NB) 10,11,10
C
C     WRITE LOADS AND STIFFNESS ON TAPE NUMBER 1
C
   10 WRITE(1) (B(N),(A(N,M), M=2,MBAND), N=1,NN)
      GOTO 1
C
C     BACKSUBSTITUTION STARTS WITH LAST BLOCK STILL IN CURE
C
   11 CONTINUE
      DO 12 M = 1,NN
      N = NN + 1 - M
      DO 13 K = 2,MBAND
      L = N + K - 1
   13 B(N) = B(N) - A(N,K)*B(L)
      NM = N + NN
      B(NM) = B(N)
   12 A(NM,NB) = B(N)
      NB = NB - 1
      IF(NB) 14,15,14
C
C     BACKSPACE TAPE ONE RECORD AND READ NEW BLOCK
C
   14 BACKSPACE 1
      READ(1)  (B(N),(A(N,M), M=2,MBAND), N=1,NN)
      BACKSPACE 1
      GOTO 11
   15 RETURN
      END
      SUBROUTINE MODIFY(I,USTAR)
C
C     THIS SUBROUTINE MODIFIES THE MAIN STIFFNESS MATRIX
C     FOR PRESCRIBED DISPLACEMENTS U(I) = USTAR
C     THE ELEMENTS OF THE I-TH ROW AND COLUMN ARE SET EQUAL TO ZERO
C     THE ELEMENT IN THE DIAGONAL IS SET EQUAL TO 1.0
C
C     PARAMETERS:  I       =  POSITION OF DISPLACEMENT RELATIVE TO BLOCKBEG
C                  USTAR   =  VALUE OF THE DISPLACEMENT
C
      IMPLICIT  REAL*8(A-H,O-Z)
      COMMON S(10,10),VKH,VKV
      COMMON DXX,DYY,E1,E2,CNU,CNU12,CNU21,EP1,EP2,SNU,PI
      COMMON NN,NH,MA,NODV,NODH,NUMNOD,NREL,NTEL,NUMBAR,MAXNDF,MBAND
      COMMON NINCR,KINCR,NCURVC,NCURVS,NZC,NZS,ISCALE,NDF,IAUTO
      COMMON NUMDF,NUMTDF,NIT,IT,NCURVB,NZB,NOD2
      COMMON/POOL  / A(108,54),B(108),NUMBLK
C
      DO 30 J = 1,MBAND
      K = I - J + 1
      IF(K.LE.0) GOTO 20
      B(K) = B(K) - A(K,J)*USTAR
      A(K,J) = 0.0
   20 K = I + J - 1
      IF(K.GT.NH) GOTO 30
      B(K) = B(K) - A(I,J)*USTAR
      A(I,J) = 0.0
   30 CONTINUE
      A(I,1) = 1.0
      B(I)   = USTAR
      RETURN
      END
      SUBROUTINE SCALE
C
C     SUBROUTINE SCALE ADJUSTS THE LOAD VECTOR ALOAD AND THE
C     PRESCRIBED DISPLACEMENT VECTOR AFTER THE FIRST SOLUTION STEP
C     FOR NINCR = 1 THE LOAD INCREMENTS ARE AUTOMATICALLY CALCULATED
C     AND PLACED IN XLOAD, ALOAD CONTAINS THE CURRENT LOAD APPLIED
C     IF THE STRAINS ARE LESS THAN THE ELASTIC LIMIT, THE PROGRAM
```

```
C       STOPS AFTER THE FIRST RUN
C       FOR NINCR GT 1 THE PROGRAM READS THE INDIVIDUAL LOAD INCREMENTS
C       AND ADDS THEM TO ALOAD
C
        IMPLICIT  REAL*8(A-H,O-Z)
        COMMON S(10,10),VKH,VKV
        COMMON DXX,DYY,E1,E2,CNU,CNU12,CNU21,EP1,EP2,SNU,PI
        COMMON NN,NH,MA,NODV,NODH,NUMNOD,NREL,NTEL,NUMBAR,MAXNDF,MBAND
        COMMON NINCR,KINCR,NCURVC,NCURVS,NZC,NZS,ISCALE,NDF,IAUTO
        COMMON NUMDF,NUMTDF,NIT,IT,NCURVB,NZB,NOD2
        COMMON/MODULC/ SEC(10,2),EC(9),XNUC(9),TOL
        COMMON/MODULS/ SES(20,2),ES(19),XNUS(19)
        COMMON/MODULB/ SEB(10,2),EB(9)
        COMMON/LOADS / XLOAD( 20,2),XBOUND( 20,2)
        COMMON/LOADS1/ ALOAD( 20,2),ABOUND( 20,2)
        COMMON/LOADS2/ ILOAD( 20),IBOUND( 20),NLOAD,NBCS
        COMMON/REINF / SAREA( 50),JCNL(650,2),ITOP( 50,6),NREINF,IS1,IS2
        COMMON/CONSTR/ STR(1152,3),STRP(1152,2),EPS(1152,3),EPR(1152,2)
        COMMON/CONST1/ TSTR(1152,3),TEPS(1152,3)
        COMMON/STLSTR/ ESPS( 50),STRS( 50),TESPS( 50),TSTRS( 50,2)
C
C       SEARCH FOR LARGEST STRAIN AT THE END OF THE FIRST SOLUTION
C
        ECMAX = 0.0
        ETMAX = 0.0
        DO 1 KSCALE = 1,NTEL
        IF(STRP(KSCALE,1) - ETMAX) 2,2,20
   20   ETMAX = STRP(KSCALE,1)
        IELMAX = KSCALE
    2   IF(STRP(KSCALE,2) - ECMAX) 100,1,1
  100   ECMAX = STRP(KSCALE,2)
        JELMAX = KSCALE
    1   CONTINUE
C       FIND THE PROPER SCALING FACTORS
        XSCAL1 = SEC(NZC+1,1)/ETMAX
        XSCAL2 = SEC(NZC-1,1)/ECMAX
        WRITE(6,2000)  IELMAX,STRP(IELMAX,1),XSCAL1,JELMAX,STRP(JELMAX,2),
       1XSCAL2
 2000   FORMAT(///,' SCALING INFORMATION',/,' ELEMENT  NO',I5,'  PRINCIPAL
       1 STRESS:  SP1 = ',1PD12.5,'  SCALE FACTOR = ',1PD10.3,/,' ELEMENT
       2 NO',I5,'  PRINCIPAL STRESS:  SP2 = ',1PD12.5,'  SCALE FACTOR = ',
       31PD10.3)
        IF((XSCAL1.GT.1.0).OR.(XSCAL2.GT.1.0)) GOTO 3
C
C       LOAD YIELDS STRAINS LARGER THAN ELASTIC LIMIT
C       XLOAD CONTAINS INCREMENTAL LOADS
C       ALOAD CONTAINS LOAD AT LIMIT AND SUBSEQUENT TOTAL LOADS
C
        XSCAL = XSCAL2
        NINCR = 2*XSCAL
        IF(XSCAL1.LT.XSCAL2) XSCAL = XSCAL1
C       DECREASE LOADS, CALCULATE INCREMENTS
        DO 5 IL = 1,NLOAD
        ALOAD(IL,1) = XLOAD(IL,1)*XSCAL
        ALOAD(IL,2) = XLOAD(IL,2)*XSCAL
        XLOAD(IL,1) = (XLOAD(IL,1) - ALOAD(IL,1))/NINCR
    5   XLOAD(IL,2) = (XLOAD(IL,2) - ALOAD(IL,2))/NINCR
        DO 6 IB = 1,NBCS
        ABOUND(IB,1) = XBOUND(IB,1)*XSCAL
        ABOUND(IB,2) = XBOUND(IB,2)*XSCAL
        XBOUND(IB,1) = (XBOUND(IB,1) - ABOUND(IB,1))/NINCR
    6   XBOUND(IB,2) = (XBOUND(IB,2) - ABOUND(IB,2))/NINCR
        IAUTO = 1
        GOTO 14
C       LARGEST STRAIN IS LESS THAN ELASTIC LIMIT
    3   IF(NINCR - 1) 8,9,10
C       WRONG NINCR, CHANGED TO 1
    8   NINCR = 1
        KPRER2 = 3
        CALL PRER2(KPRER2)
C       TOTAL LOAD APPLIED DOES NOT GIVE INELASTIC BEHAVIOR
C       ELASTIC SOLUTION ONLY
    9   KPRER2 = 4
        CALL PRER2(KPRER2)
   11   CONTINUE
        STOP
C
C       FIRST INCREMENT IS TOO SMALL TO YIELD INELASTIC STRAINS
C       READ NEXT INCREMENT
C
   10   CONTINUE
C
C
        ENTRY SCALE2
        IF(KINCR.EQ.NINCR) GOTO 11
        DO 12 IL = 1,NLOAD
        IF(IAUTO.EQ.1) GOTO 122
        READ(5,1000) KLOAD,XLOAD(IL,1),XLOAD(IL,2)
        IF(KLOAD.EQ.ILOAD(IL)) GOTO 122
        WRITE(6,1100) IL
 1100   FORMAT(' LOAD ERROR AT LOAD',I5)
        CALL EXIT
 1000   FORMAT(I5,2F12.4)
  122   ALOAD(IL,1) = ALOAD(IL,1) + XLOAD(IL,1)
   12   ALOAD(IL,2) = ALOAD(IL,2) + XLOAD(IL,2)
        READ(5,1919) KBCS
 1919   FORMAT(I5)
        IF(KBCS.EQ.0) GOTO 14
        DO 13 IB = 1,NBCS
        IF(IAUTO.EQ.1) GOTO 133
        READ(5,1000) IBOUND(IB),XBOUND(IB,1),XBOUND(IB,2)
  133   ABOUND(IB,1) = ABOUND(IB,1) + XBOUND(IB,1)
   13   ABOUND(IB,2) = ABOUND(IB,2) + XBOUND(IB,2)
   14   CONTINUE
        KINCR = KINCR + 1
        WRITE(6,5020)
 5020   FORMAT(1H1,42(1H*)/43H * LOADING INFORMATION                  *,
       1/,1X,42(1H*),/)
        WRITE(6,5220) NINCR,KINCR
 5220   FORMAT(1X,'TOTAL NUMBER OF INCREMENTS',I5///1X,'LOADS FOR INCREMEN
       1T    NO.',I5,/)
        WRITE(6,5021)
 5021   FORMAT (' NODE NO        X-LOAD',10X,'Y-LOAD          IN LBS' / )
        DO 62 IL = 1,NLOAD
   62   WRITE(6,5032) JCNL(ILOAD(IL),2),XLOAD(IL,1),XLOAD(IL,2)
        WRITE(6,9010) KINCR
 9010   FORMAT(///,1X,'TOTAL LOADS FOR INCREMENT',I5,/)
        WRITE(6,5021)
```

```
      DO 602 IL = 1,NLOAD
  602 WRITE(6,5032) JCNL(ILOAD(IL),2),ALOAD(IL,1),ALOAD(IL,2)
      IF(KBCS.EQ.0) RETURN
      WRITE(6,5030)
 5030 FORMAT(1H1,42(1H*)/43H * BOUNDARY CONDITIONS                    *,
     1/,1X,42(1H*),/)
      WRITE(6,5331) NINCR,KINCR
 5331 FORMAT(1X,'TOTAL NUMBER OF INCREMENTS',I5,///,1X,'DISP. FOR INCREM
     1ENT    NO.',I5,/)
      WRITE(6,5031)
 5031 FORMAT(' NODE NO        X-DISPL',7X,'  Y-DISPL    IN INCHES',/)
      DO 61 IB = 1,NBCS
   61 WRITE(6,5032) IBOUND(IB),XBOUND(IB,1),XBOUND(IB,2)
 5032 FORMAT(1X,I5,8X,1PD12.5,4X,1PD12.5)
      WRITE(6,9020) KINCR
 9020 FORMAT(///,1X,'TOTAL DISPL. FOR INCREMENT',I5,/)
      WRITE(6,5031)
      DO 612 IB = 1,NBCS
  612 WRITE(6,5032) IBOUND(IB),ABOUND(IB,1),ABOUND(IB,2)
      RETURN
      END
      SUBROUTINE STRESS(I,J,JJ1,JJ2,JJ3,MTYP)
C    ...............................................................
C    .                                                             .
C    . I       = NUMBER OF RECTANGULAR ELEMENT PROCESSED           .
C    . J       = NUMBER OF TRIANGULAR ELEMENT PROCESSED            .
C    . JJ1     = NUMBER OF NODE 1                                  .
C    . JJ2     = NUMBER OF NODE 2                                  .
C    . JJ3     = NUMBER OF NODE 3                                  .
C    . MTYP    = MATERIAL TYPE      = 1 FOR CONCRETE               .
C    .                              = 2 FOR STEEL                  .
C    ...............................................................
C
      IMPLICIT  REAL*8(A-H,O-Z)
      COMMON S(10,10),VKH,VKV
      COMMON DXX,DYY,E1,E2,CNU,CNU12,CNU21,EP1,EP2,SNU,PI
      COMMON NN,NH,MA,NODV,NODH,NUMNOD,NREL,NTEL,NUMBAR,MAXNDF,MBAND
      COMMON NINCR,KINCR,NCURVC,NCURVS,NZC,NZS,ISCALE,NDF,IAUTO
      COMMON NUMDF,NUMTDF,NIT,IT,NCURVB,NZB,NOD2
C
      COMMON/ELEM  / X(650),Y(650),THETA(1152,2),U(1300),TU(1300)
      COMMON/ELEM1 / JTOP(288,4),THICK(288)
      COMMON/TYP   / ITYPE(1152),ISTYPE(50),IBTYPE(100),KDIR
      COMMON/MODULC/ SEC(10,2),EC(9),XNUC(9),TOL
      COMMON/MODULS/ SES(20,2),ES(19),XNUS(19)
      COMMON/MODULB/ SEB(10,2),EB(9)
      COMMON/REINF / SAREA( 50),JCNL(650,2),ITOP( 50,6),NREINF,IS1,IS2
      COMMON/CONSTR/ STR(1152,3),STRP(1152,2),EPS(1152,3),EPR(1152,2)
      COMMON/CONST1/ TSTR(1152,3),TEPS(1152,3)
      COMMON/STLSTR/ ESPS( 50),STRS( 50),TESPS( 50),TSTRS( 50,2)
      COMMON/BOND  / EPSB(100,2),STRB(100,2),TEPSB(100,2),TSTRB(100,2)
      DATA D20/1.0D-20/ , ZERO/0.0D 00/ , D10/1.0D-10/
      DIMENSION H(3,6) , UU(6) , D(3,3) , JC(3)
      CALL TYPE(J,MTYP)
      IF(MTYP.NE.1) GOTO 110
      DO 10 LH = 1,3
      DO 10 MH = 1,6
   10 H(LH,MH) = 0.0
      DO 20 LD = 1,3
      DO 20 MD = 1,3
   20 D(LD,MD) = 0.0
      ALPHA = 0.0
      BETA  = 0.0
      JC(1) = JJ1
      JC(2) = JJ2
      JC(3) = JJ3
      CB = 1.0/((X(JJ3)-X(JJ2))*(Y(JJ2)-Y(JJ1)) -
     1          (X(JJ2)-X(JJ1))*(Y(JJ3)-Y(JJ2)))
C
C     SIGMA = D*H*U
C
      DO 25 K = 1,3
      JI = 2*JCNL(JC(K),2)
      UU(2*K - 1) = U(JI - 1)
   25 UU(2*K) = U(JI)
C
C     SET UP H - MATRIX
C
      H(1,1) =  CB*(Y(JJ3) - Y(JJ2))
      H(1,3) = -CB*(Y(JJ3) - Y(JJ1))
      H(1,5) =  CB*(Y(JJ2) - Y(JJ1))
      H(2,2) = -CB*(X(JJ3) - X(JJ2))
      H(2,4) =  CB*(X(JJ3) - X(JJ1))
      H(2,6) = -CB*(X(JJ2) - X(JJ1))
      H(3,1) = H(2,2)
      H(3,2) = H(1,1)
      H(3,3) = H(2,4)
      H(3,4) = H(1,3)
      H(3,5) = H(2,6)
      H(3,6) = H(1,5)
C
      DO 40 L = 1,3
      ETEMP = 0.0
      DO 30 M = 1,6
   30 ETEMP = ETEMP + H(L,M)*UU(M)
   40 EPS(J,L) = ETEMP
C
C     CALCULATE PRINCIPAL STRAINS
C
      EPSX = TEPS(J,1) + EPS(J,1)
      EPSY = TEPS(J,2) + EPS(J,2)
      EPSXY = TEPS(J,3) + EPS(J,3)
      RAD = ((EPSX - EPSY)/2.0)**2 + (0.5*EPSXY)**2
      EMAX = DSQRT(RAD)
      EAVR = (EPSX + EPSY)/2.0
      EPR(J,1) = EAVR + EMAX
      EPR(J,2) = EAVR - EMAX
      IF(DABS(EPSX - EPSY).LT.D20) GOTO 50
      TAN2A = EPSXY/(EPSX - EPSY)
      ALPHA = DATAN(TAN2A)*90.0/PI
      THETA(J,1) = ALPHA
C
C     UPDATE TYPE
C
   50 CALL TYPE(J,MTYP)
      IF(ITYPE(J).NE.1) GOTO 80
C
C     ELASTIC , ISOTROPIC
```

```
C
      CD = E1/(1.0 - CNU*CNU)
      D(1,1) = CD
      D(1,2) = CD*CNU
      D(2,1) = D(1,2)
      D(2,2) = CD
      D(3,3) = CD*0.5*(1.0 - CNU)
C
      DO 70 L = 1,3
      STEMP = 0.0
      DO 60 M = 1,3
   60 STEMP = STEMP + D(L,M)*EPS(J,M)
   70 STR(J,L) = STEMP
C
C     PRINCIPAL STRESSES
C
      SIGX  = TSTR(J,1) + STR(J,1)
      SIGY  = TSTR(J,2) + STR(J,2)
      SIGXY = TSTR(J,3) + STR(J,3)
      RAD = ((SIGX - SIGY)/2.0)**2 + SIGXY**2
      SMAX = DSQRT(RAD)
      SAVR = (SIGX + SIGY)/2.0
      STRP(J,1) = SAVR + SMAX
      STRP(J,2) = SAVR - SMAX
      IF(DABS(SIGX - SIGY).LT.D20) RETURN
      TAN2B = (SIGXY*2.0)/(SIGX - SIGY)
      BETA = DATAN(TAN2B)*90./PI
      THETA(J,2) = BETA
      RETURN
C
C     ANISOTROPIC
C
   80 CDA = 1.0/(1.0 - CNU12*CNU21)
      D(1,1) = E1*CDA
      D(1,2) = E1*CDA*CNU12
      D(2,1) = D(1,2)
      D(2,2) = E2*CDA
      DO 100 L = 1,2
      STEMP = 0.0
      DO  90 M = 1,2
   90 STEMP = STEMP + D(L,M)*EPR(J,M)
  100 STRP(J,L) = STEMP
      RETURN
C
C     STEEL BAR STRAINS AND STRESSES
C     BOND LINK STRAINS AND STRESSES
C
  110 XJI = X(JJ2) - X(JJ1)
      YJI = Y(JJ2) - Y(JJ1)
      SL2 = XJI*XJI + YJI*YJI
      KS1 = 2*ITOP(J,5)
      KS2 = 2*ITOP(J,6)
      ESPS(J) = (XJI*(U(KS2-1) - U(KS1-1)) + YJI*(U(KS2) - U(KS1)))/SL2
      IF(E1.LT.D20) CALL EXIT
C     IF STEEL STRESS-STRAIN CURVE HORIZONTAL PROGRAM STOPS
      STRS(J) = E1*ESPS(J)
C
C     DETERMINE TOTAL STEEL STRESS FROM STRESS-STRAIN CURVE
C
      NCURV = NCURVS - 1
      IE = NZS
      DO 120 K = 1,NCURV
      DE = 1.0D 10
      IF(DABS(E1 - ES(K)).LT.D10) DE = ZERO
      IF(DE.EQ.ZERO) IE = K
  120 CONTINUE
      ED = TESPS(J) + ESPS(J) -SES(IE,2)
      TSTRS(J,2) = SES(IE,1) + ED*(SES(IE+1,1) - SES(IE,1))/
     1(SES(IE+1,2) - SES(IE,2))
C
C     BOND STRAINS AND STRESSES
C
      MTYP = 3
      IF(JJ2.GE.(JJ1 + NOD2)) GOTO 125
C
C     VERTICAL BAR
C
      CO = 0.0D+00
      SI = 1.0D+00
      KDIR = 2
      GOTO 130
C
C     HORIZONTAL BAR
C
  125 CO = 1.0D+00
      SI = 0.0D+00
      KDIR = 1
  130 CONTINUE
      KLAST = 0
      KB1 = 2 * JCNL(ITOP(J,3),2)
      KB2 = 2 * ITOP(J,5)
      L   = 2 * J - KDIR + 1
  140 EPSB(L,1) = - CO * U(KB1-1) - SI * U(KB1) + CO * U(KB2-1) +
     1                SI * U(KB2)
      EPSB(L,2)     =  SI*U(KB1-1) - CO*U(KB1) - SI*U(KB2-1) + CO*U(KB2)
      EP1 = TEPSB(L,1) + EPSB(L,1)
      CALL TYPE ( L, MTYP )
      STRB(L,1) = VKH * EPSB(L,1)
      STRB(L,2) = VKV * EPSB(L,2)
C
C     IF LAST BAR ENCOUNTERED, PROCESS END LINK STRESSES
C
      IF ( J .NE. NUMBAR ) GO TO 170
      IF ( KLAST .EQ. 1 ) GO TO 170
      L   = 2 * ( J + 1 ) - KDIR + 1
      KB1 = 2 * JCNL(ITOP(J,4),2)
      KB2 = 2 * ITOP(J,6)
      KLAST = 1
      GO TO 140
  170 RETURN
      END
      SUBROUTINE RESOUT(ICHECK)
C
C     ..................................................................
C     . SUBROUTINE RESOUT PRINTS ALL RESULTS AFTER EACH LOAD INCREMENT .
C     . RESULTS APPEAR IN TABULAR FORM                                 .
C     ..................................................................
C
```

```
      IMPLICIT  REAL*8(A-H,O-Z)
      COMMON S(10,10),VKH,VKV
      COMMON DXX,DYY,E1,E2,CNU,CNU12,CNU21,EP1,EP2,SNU,PI
      COMMON NN,NH,MA,NODV,NODH,NUMNOD,NREL,NTEL,NUMBAR,MAXNDF,MBAND
      COMMON NINCR,KINCR,NCURVC,NCURVS,NZC,NZS,ISCALE,NDF,IAUTO
      COMMON NUMDF,NUMTDF,NIT,IT,NCURVB,NZB,NOD2
C
      COMMON/ELEM  / X(650),Y(650),THETA(1152,2),U(1300),TU(1300)
      COMMON/ELEM1 / JTOP(288,4),THICK(288)
      COMMON/TYP   / ITYPE(1152),ISTYPE(50),IBTYPE(100),KDIR
      COMMON/REINF / SAREA( 50),JCNL(650,2),ITOP( 50,6),NREINF,IS1,IS2
      COMMON/CONSTR/ STR(1152,3),STRP(1152,2),EPS(1152,3),EPR(1152,2)
      COMMON/CONST1/ TSTR(1152,3),TEPS(1152,3)
      COMMON/STLSTR/ ESPS( 50),STRS( 50),TESPS( 50),TSTRS( 50,2)
      COMMON/BOND  / EPSB(100,2),STRB(100,2),TEPSB(100,2),TSTRB(100,2)
      DIMENSION SUM(3),ROWSUM(3)
      DATA ZERO/0.0D00/ , D4/4.0D00/ , IEQUI/'EQUI'/
C
C     PRINT NODAL DISPLACEMENTS
C
      ILINE = 0
      WRITE(6,1000)
 1000 FORMAT(1H1,42(1H*)/43H * NODAL DISPLACEMENTS                   *,
     1/,1X,42(1H*),/)
      WRITE(6,1001)
 1001 FORMAT(' NODE NO.     X-DISPLACEMENT      Y-DISPLACEMENT',/)
      K = 0
      DO 10 NM = 1,NUMTDF,2
      ILINE = ILINE + 1
      K = K + 1
      WRITE(6,1002) K,TU(NM),TU(NM+1)
 1002 FORMAT(1X,I5,8X,1PD12.5,8X,1PD12.5)
      IF(ILINE.NE.55) GOTO 10
      WRITE(6,1003)
 1003 FORMAT(1H1,//)
      WRITE(6,1001)
      ILINE = 0
   10 CONTINUE
C
C     PRINT ITERATION INFORMATION
C
      WRITE(6,7200) IT
 7200 FORMAT(/,1X,'ERROR LESS THAN TOLERANCE IN ITERATION NO ',I3,/,1X,'
     1SOLUTION CONTINUES',/)
C
C     PRINT STRESSES IN CONCRETE
C
      WRITE(6,1004)
 1004 FORMAT(1H1,42(1H*)/43H * STRESSES AND STRAINS IN CONCRETE       *,
     1/,1X,42(1H*),/)
      WRITE(6,1005)
 1005 FORMAT(' ELEM NO.   SIGMA - X      SIGMA - Y       TAU - XY      SI
     1GMA - 1      SIGMA - 2       THETA(1)      TYPE',/,12X,'EPSILON-X
     2 EPSILON-Y        GAMMA-XY          EPSILON-1       EPSILON-2       THETA(
     32)',/)
      ILINE = 0
      DO 20 NE = 1,NTEL
      ILINE = ILINE + 1
      IF(ITYPE(NE).EQ.1) GOTO 15
      WRITE(6,1060) NE,STRP(NE,1),STRP(NE,2),THETA(NE,1),ITYPE(NE),
     1EPR(NE,1),EPR(NE,2),THETA(NE,1)
 1060 FORMAT(1X,I5,48X,2(1PD12.5,2X),1X,1PD12.5,2X,I3,/,54X,2(1PD12.5,2X
     1),1X,1PD12.5)
      IF(ITYPE(NE).EQ.4) WRITE(6,1007)
 1007 FORMAT(1H+,105X,'CRACKED')
      GOTO 16
   15 WRITE(6,1006) NE,TSTR(NE,1),TSTR(NE,2),TSTR(NE,3),STRP(NE,1),STRP(
     1NE,2),THETA(NE,2),ITYPE(NE),TEPS(NE,1),TEPS(NE,2),TEPS(NE,3),EPR(N
     2E,1),EPR(NE,2),THETA(NE,1)
 1006 FORMAT(1X,I5,4X,3(1PD12.5,2X),2X,2(1PD12.5,2X),1X,1PD12.5,2X,I3,/,
     11X,5X,4X,3(1PD12.5,2X),2X,2(1PD12.5,2X),1X,1PD12.5)
   16 IF(ILINE.NE.28) GOTO 20
      WRITE(6,1003)
      WRITE(6,1005)
      ILINE = 0
   20 CONTINUE
      IF(NREINF.EQ.0) GOTO 50
C
C     PRINT STEEL AND BOND STRAINS AND STRESSES
C
      WRITE(6,9070)
 9070 FORMAT(1H1,42(1H*),/43H * STRESSES & STRAINS IN REINFORCEMENTS  *
     1,/,1X,42(1H*),/)
      WRITE(6,9080)
 9080 FORMAT(' BAR NO.     EPSILON               SIGMA  CALC.      SIGMA  DEP
     1ICTED',5X,'TYPE'/)
      DO 30 IB = 1,NUMBAR
   30 WRITE(6,9090) IB,TESPS(IB),TSTRS(IB,1),TSTRS(IB,2),ISTYPE(IB)
 9090 FORMAT(I5,5X,1PD12.5,6X,1PD12.5,5X,1PD12.5,10X,I3)
      WRITE(6,9100)
 9100 FORMAT(1H1,42(1H*),/43H * BOND STRESSES                          *
     1,/,43H * IN DIRECTION OF GLOBAL AXES              *,/,1X,42(1H*),/)
      WRITE(6,9110)
      DO 40 L = 1, NUMBAR
      K = 2 * L
      WRITE(6,9120) ITOP(L,5), TEPSB(K,1), TEPSB(K,2), TSTRB(K,1),
     1                  TSTRB(K,2),IBTYPE(K)
      IF ( IBTYPE(L) .EQ. 3 ) WRITE(6,9030)
   40 CONTINUE
      K = 2 * ( NUMBAR + 1 )
      WRITE(6,9120) ITOP(NUMBAR,6), TEPSB(K,1), TEPSB(K,2), TSTRB(K,1),
     1                  TSTRB(K,2),IBTYPE(K)
      IF ( IBTYPE(NUMBAR+1) .EQ. 3 ) WRITE(6,9030)
 9110 FORMAT(' LINK NO.  AT NODE   REL. X-DISP.    REL. Y-DISP.      X -
     1FORCE      Y - FORCE',3X,'TYPE'/)
 9120 FORMAT ( 11X, I5, 5X, 4 ( 1PD12.5, 3X ) ,I2)
 9030 FORMAT(1H+,99X,'FAILED')
   50 IF(ICHECK.NE.IEQUI) RETURN
C
C     STATIC EQUILIBRIUM CHECK
C
      WRITE(6,9130)
 9130 FORMAT(1H1,42(1H*),/43H * EQUILIBRIUM CHECK FOR EACH COLUMN      *
     1,/,1X,42(1H*),/)
      WRITE(6,9135)
 9135 FORMAT(1X,' RECT. ELEM.      AV. X-FORCE    AV. Y-FORCE    AV.SHEAR FO
     1RCE'/)
      KP = 0
```

```
      NHOR = NODH - 1
      NVER = NODV - 1
      IPAGE = 48/(NVER+3)
 9160 FORMAT(1H1,1X,' RECT. ELEM.     AV. X-FORCE    AV. Y-FORCE    AV.SHEA
     1R FORCE'/)
      IEL = 0
      DO 80 K = 1,NHOR
      DO 55 KK = 1,3
   55 ROWSUM(KK) = ZERO
      DO 70 KROW = 1,NVER
      DO 60 KK = 1,3
   60 SUM(KK) = ZERO
      IEL = IEL + 1
      ITEL = 4*IEL - 3
      DO 65 L = 1,4
      DO 66 LL = 1,3
   66 SUM(LL) = SUM(LL) + TSTR(ITEL,LL)
   65 ITEL = ITEL + 1
      DO 67 LL = 1,3
      SUM(LL) = SUM(LL)*(Y(KROW+1) - Y(KROW))*THICK(IEL)/D4
   67 ROWSUM(LL) = ROWSUM(LL) + SUM(LL)
   70 WRITE(6,9140) IEL,(SUM(LL),LL=1,3)
      WRITE(6,9150) K,(ROWSUM(LL),LL=1,3)
      IF(KP.LT.IPAGE) GOTO 80
      KP = 0
      IPAGE = 54/(NVER+3)
      WRITE(6,9160)
   80 KP = KP + 1
 9140 FORMAT(I6,9X,3(2X,1PD12.5))
 9150 FORMAT(/'    ROW',I5,4X,3(2X,1PD12.5),'  RESIDUAL FORCES'/)
      RETURN
      END
      SUBROUTINE PRER1(J)
C
C     THIS SOUBROUTINE DECLARES ALL FATAL ERRORS
C     PROGRAM STOPS AFTER ERROR MESSAGE IS PRINTED
C
C
      IMPLICIT  REAL*8(A-H,O-Z)
      COMMON S(10,10),VKH,VKV
      COMMON DXX,DYY,E1,E2,CNU,CNU12,CNU21,EP1,EP2,SNU,PI
      COMMON NN,NH,MA,NODV,NODH,NUMNOD,NREL,NTEL,NUMBAR,MAXNDF,MBAND
      COMMON NINCR,KINCR,NCURVC,NCURVS,NZC,NZS,ISCALE,NDF,IAUTO
      COMMON NUMDF,NUMTDF,NIT,IT,NCURVB,NZB,NOD2
C
      GOTO(1,2,3,4,5,6,7,8,9) , J
    1 PRINT 11
   11 FORMAT(///,1X,11HFATAL ERROR,//,1X,35HNUMBER OF NODES HORIZONTAL
     1IS ZERO)
      GOTO 99
    2 PRINT 12
   12 FORMAT(///,1X,11HFATAL ERROR,//1X,33HNUMBER OF NODES VERTICAL   IS
     1ZERO)
      GOTC 99
    3 PRINT 13
   13 FORMAT(///,1X,11HFATAL ERROR,//1X,'NBEG IS EQUAL TO NEND')
      GOTO 99
    4 PRINT 14
   14 FORMAT(///,1X,11HFATAL ERROR,//1X,'STEEL AREA NEGATIVE')
      GOTO 99
    5 PRINT 15
   15 FORMAT(///,1X,11HFATAL ERROR,//1X,'INCORRECT ENDNODE')
      GOTO 99
    6 PRINT 16
   16 FORMAT(///,1X,11HFATAL ERROR,//1X,'NUMBER OF REINFORCEMENTS SPECIF
     1IED IS NEGATIVE')
      GOTO 99
    7 PRINT 17
   17 FORMAT(///,1X,11HFATAL ERROR,//1X,'STATEMENT 1 BEFORE STATEMENT 40
     14 IS FALSE , CALL ON PROGRAMMER')
      GOTO 99
    8 PRINT 18 , MBAND
   18 FORMAT(///,1X,11HFATAL ERROR,//1X,'BANDWIDTH EXEEDED MBAND =',I4,'
     1     CALL PROGRAMMER')
      GOTO 99
    9 PRINT 19
   19 FORMAT(///,1X,11HFATAL ERROR,//,1X,'BOUNDARY CONDITIONS ARE INCORR
     1ECTLY LABELED')
   99 CALL EXIT
      RETURN
      END
      SUBROUTINE PRER2(J)
C
C     THIS SUBROUTINE PRINTS NONFATAL ERROR MESSAGES
C     THE PROGRAM CONTINUES AFTER ASSIGNING A VALUE TO THE VARIABLE
C     WHICH IS OUT OF RANGE
C
      IMPLICIT  REAL*8(A-H,O-Z)
      COMMON S(10,10),VKH,VKV
      COMMON DXX,DYY,E1,E2,CNU,CNU12,CNU21,EP1,EP2,SNU,PI
      COMMON NN,NH,MA,NODV,NODH,NUMNOD,NREL,NTEL,NUMBAR,MAXNDF,MBAND
      COMMON NINCR,KINCR,NCURVC,NCURVS,NZC,NZS,ISCALE,NDF,IAUTO
      COMMON NUMDF,NUMTDF,NIT,IT,NCURVB,NZB,NOD2
      COMMON/MODULC/ SEC(10,2),EC(9),XNUC(9),TOL
      COMMON/MODULS/ SES(20,2),ES(19),XNUS(19)
      COMMON/MODULB/ SEB(10,2),EB(9)
      PRINT 99
   99 FORMAT(/)
      PRINT 100
  100 FORMAT(1X,'*** SUBROUTINE PRER2                           ',14X,'***',/,
     11X,'*** NONFATAL ERROR, SOLUTION CONTINUES                        ***')
      GOTO(1,2,3,4,5) , J
    1 PRINT 101,E1,E2
  101 FORMAT(/,1X,'*** EMODUL OUT OF RANGE, LAST VALUES',19X,'***',T40,'
     1E1 =',1PD12.4,/,T40,'E2 =',1PD12.4)
      GOTO 999
    2 PRINT 101,E1,E2
      GOTO 999
    3 PRINT 103
  103 FORMAT(//,1X,'*** NUMBER OF INCREMENTS SPECIFIED IS LESS    ***',/,
     11X,'*** THAN ONE, ROGRAM ASSUMES NINCR = 1.                   ***')
      GOTO 999
    4 PRINT 104
  104 FORMAT(/,1X,'*** TOTAL LOAD APPLIED DOES NOT GIVE RISE',9X,'***'/,
     11X,'*** TO INELASTIC BEHAVIOR, SOLUTION ELASTIC ONLY         ***')
      GOTO 999
    5 PRINT 105
  105 FORMAT(//,1X,'*** NUMBER OF START NODE GREATER THAN NUMBER ***',/,
```

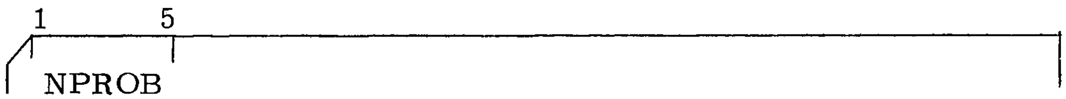
```
    11X,'*** OF END NODE. PROGRAM SETS NBEG=NENE AND NEND=NBEG  ***')
999 CONTINUE
    RETURN
    END
```

APPENDIX C

INPUT SEQUENCE FOR NARCOS-2

1. Number of Problems:

one card: (I5)



NPROB = number of problems

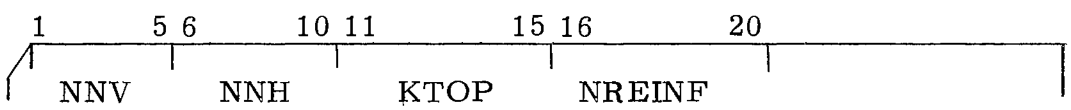
2. Problem Identification Card:

one card: (18A4)

NAME = problem title

3. First Control Card:

one card: (4I5)



NNV = number of nodes vertically

NNV > 0: unequal spacing

NNV < 0: equal spacing

NNH = number of nodes horizontally

NNH > 0: unequal spacing

NNH < 0: equal spacing

KTOP = input mode parameter

KTOP = 1: automatic mesh generation

KTOP = 2: individual input

NREINF = number of reinforcements

4. Coordinate Data Cards:

min. two cards: (I5, D12.4)

1 – 5	6 – 17	
J	DTEMP	

J = node number

DTEMP = x- or y-coordinate of nodes

for NNV < 0 or NNH < 0: one card, respectively

for NNV > 0: NNV cards

for NNH > 0: NNH cards

5. Second Control Card:

one card: (4I5)

1 – 5	6 – 10	11 – 15	16 – 20	
NTH	NCURVC	NCURVS	NCURVB	

NTH = number of elements with irregular thickness

NCURVC = number of points on concrete curve

NCURVS = number of points on steel curve

NCURVB = number of points on bond curve

for NTH = 0 - all elements are of unit thickness

6. Thickness (Optional):

NTH cards: (I5, D12.4)

1 – 5	6 – 17	
NT	THICK	

NT = element number (rectangle)

NT = 0: all elements are changed to new thickness

NT > 0: supply NTH cards

7. Stress-Strain Data:

NCURVB cards: (I5, D12.4, D12.6)

1	5 6	17 18	29
I	STRESS	STRAIN	

I = number of point on curve

STRESS = stress at point I

STRAIN = strain at point I

8. Poisson's ratios:

(NCURVB - 1) cards: (I5, D12.6)

1	5 6	17
I	NUB	

I = interval (between two points given under 7)

NUB = Poisson's ratio

Blocks 7 and 8 are repeated for steel and bond if the parameter NREINF is greater than zero.

9. Bond Stiffness (Optional):

one card: (5X, D12.5)

1	5	17
	VKV	

VKV = bond stiffness in direction perpendicular to reinforcement

This card must be omitted if NREINF = 0. If a blank card is supplied, VKV is chosen as mentioned in Chapter IV.

10. Reinforcement (Optional):

NREINF cards: (2I5, D12.4, 2I5)

1–5	6–10	11–22	23–27	28–32
NBAR	NØ	RAREA	NBEG	NEND

NBAR = number of bars

NØ = bar number (ACI Code)

RAREA = total cross-sectional area

NBEG = start node

NEND = end node

When standard bars (ACI Code) are selected, the total cross-sectional area is computed automatically. In this case, the parameter RAREA must be omitted. If RAREA is specified, the parameters NBAR and NØ may be omitted.

11. Tolerance:

one card: (5X, D12.5)

1–5	6–17	
	TOL	

TOL = tolerance

12. Third Control Card:

one card: (4I5)

1–5	6–10	11–15	16–20	
NLOAD	NBCS	NINCR	NIT	

NLOAD = number of loads

NBCS = number of boundary conditions

NINCR = number of increments

NINCR = 0: elastic solution

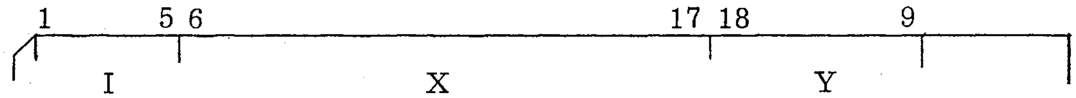
NINCR = 1: automatic scaling

NINCR > 1: specified increments

NIT = number of iterations

13. Loading Data:

NLOAD cards: (I5, 2D12.4)



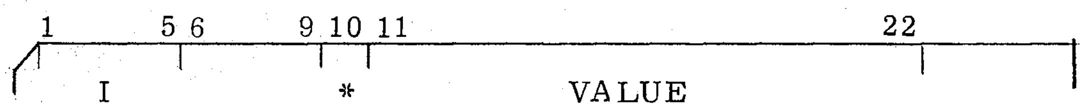
I = joint number

X = X-component of load at I

Y = Y-component of load at I

14. Boundary Conditions:

NBCS cards: (I5, 4X, A1, D12.4)



I = restrained node

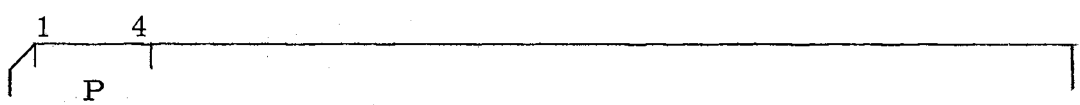
* = X: X-restraint

* = Y: Y-restraint

VALUE = value of prescribed displacement

15. Equilibrium Check:

one card: (A4)



P = alphanumeric parameter

P = EQUI: initializes checking procedure

A blank card is required if no checking is requested.

16. Remarks:

a. For NINCR > 1 group 13 is repeated NINCR-1 times.

b. If during the increments 2, 3, . . . , etc., the boundary conditions remain the same, one blank card may be supplied instead of the whole block 14.

c. If the EQUI card is inserted, the checks are done for all increments.

APPENDIX D

SAMPLE INPUT AND OUTPUT

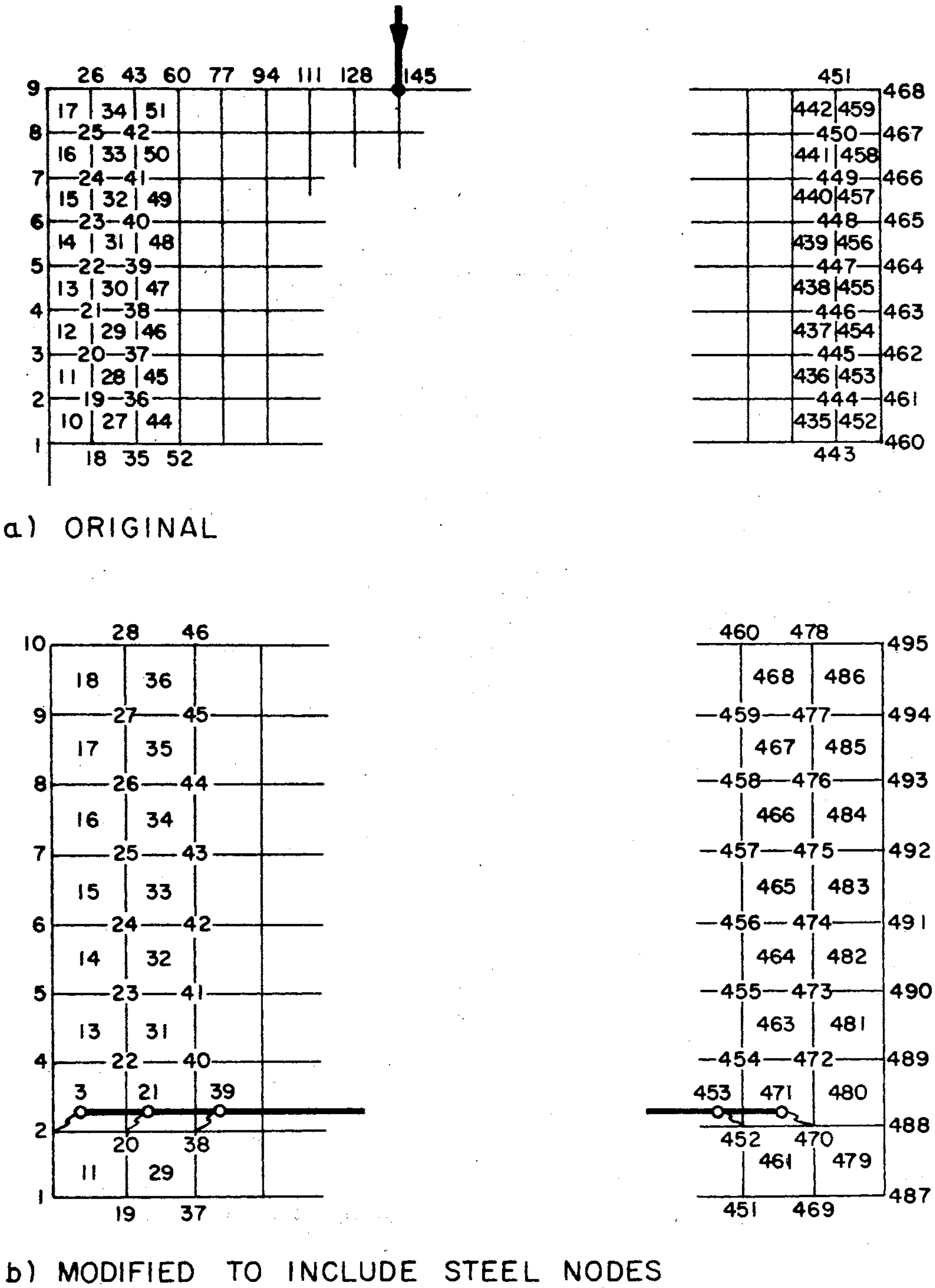


Figure 32. Nodal Arrangement of Scordelis' Beam A-1

```
SIMPLE BEAM, SCORDELIS CHECK PROBLEM9 1 REINF., 468 NODES, 864 TRIANG.
   3  -28     1    1
   1
   3       5.0
   9      22.0
 468      81.0
   1     4     5    3
          12.0
   1  -4000.0       -0.002
   2  -3000.0       -0.001
   3      0.0        0.0
   4    300.0        0.0001
   1    0.15
   2    0.15
   3    0.15
   1 -40000.0       -0.002
   2 -30000.0       -0.001
   3      0.0        0.0
   4  30000.0        0.001
   5  40000.0        0.002
   1    0.3
   2    0.3
   3    0.3
   4    0.3
   1  -2200.0       -0.001
   2
   3   2200.0        0.001
    1.00000D.12
   2     9                    2  444
            0.05
   1    10    10    3
 145               -7000.0
   1     X
   2     X
   3     X
   4     X
   5     X
   6     X
   7     X
   8     X
   9     X
 409     Y
EQUILIBRIUM CHECK

 145                -200.0

 145                -200.0

 145                -200.0

 145                -200.0

 145                -200.0

 145                -200.0

 145                -200.0
```

6	-8.33333
7	-4.16667
1	-16.66667
2	-33.33333
3	-33.33333
4	-33.33333
5	-33.33333
6	-33.33333
7	-16.66667
1	-83.33333
2	-166.66667
3	-166.66667
4	-166.66667
5	-166.66667
6	-166.66667
7	-83.33333
1	-83.33333
2	-166.66667
3	-166.66667
4	-166.66667
5	-166.66667
6	-166.66667
7	-83.33333

```
     1
SIMPLE BEAM APPROXIMATION ,637 NODES WITH STEEL, PMAX = 3000 LBS
    -7  -46     1    1
     7         12.0
    46         90.0
     1    7     5    3
     0      8.0
     1  -3000.0        -.002
     2  -2980.0        -.00175
     3  -2760.0        -.00125
     4  -2475.0        -.001
     5  -1500.0        -.0005
     6      0.0        0.0
     7    300.0        0.0001
     1      .15
     2      .15
     3      .15
     4      .15
     5      .15
     6      .15
     1 -50000.0        -.005
     2 -47000.0        -.00162
     3      0.0        0.0
     4  47000.0        0.00162
     5  50000.0        0.005
     1       .333333
     2       .333333
     3       .333333
     4       .333333
     1  -2200.0        -.001
     2      0.0        0.0
     3   2200.0        0.001
      1.00000D.12
     1    0            1.0     2   574
           0.05
     1    8     5    3
     1                  -62.5
     2                 -125.
     3                 -125.
     4                 -125.
     5                 -125.
     6                 -125.
     7                  -62.5
     1    X
     2    X
     3    X
     4    X
     5    X
     6    X
     7    X
   573    Y
EQUI
     1                   -4.16667
     2                   -8.33333
     3                   -8.33333
     4                   -8.33333
     5                   -8.33333
```

145	-200.0
145	-200.0

ELEM NO.	SIGMA - X EPSILON-X	SIGMA - Y EPSILON-Y	TAU - XY GAMMA-XY	SIGMA - 1 EPSILON-1	SIGMA - 2 EPSILON-2	THETA(1) THETA(2)	TYPE	
253	-4.03533D 02	-8.92756D 00	-3.03160D 01	-6.61209D 00	-4.05849D 02	4.36765D 00	1	
	-1.34065D-04	1.72008D-05	-2.32423D-05	1.80884D-05	-1.34952D-04	4.36765D 00		
254	-4.64396D 02	-8.69907D 01	-8.12503D 01	-7.02419D 01	-4.81145D 02	1.16477D 01	1	
	-1.50449D-04	-5.77709D-06	-6.22919D-05	6.43287D-07	-1.56870D-04	1.16477D 01		
255	-3.51229D 02	-2.14609D 02	-8.27862D 01	-1.75589D 02	-3.90249D 02	2.52363D 01	1	
	-1.06346D-04	-5.39749D-05	-6.34694D-05	-3.90171D-05	-1.21304D-04	2.52363D 01		
256	-2.90366D 02	-1.36546D 02	-3.18519D 01	-1.30211D 02	-2.96701D 02	1.12483D 01	1	
	-8.99614D-05	-3.09970D-05	-2.44198D-05	-2.85687D-05	-9.23897D-05	1.12483D 01		
257	2.98416D 02	9.14862D-02	1.50205D 01	2.99170D 02	-6.62883D-01	2.87513D 00	1	
	9.94673D-05	-1.48903D-05	1.15157D-05	9.97565D-05	-1.51795D-05	2.87513D 00		
258	2.62184D 02	1.07839D 01	1.12352D 00	2.62189D 02	1.07789D 01	2.56050D-01	1	
	8.68555D-05	-9.51456D-06	8.61363D-07	8.68574D-05	-9.51648D-06	2.56050D-01		
259				0.0	-4.54758D 01	-2.75066D 00	4	
				1.00842D-04	-1.51586D-05	-2.75066D 00		CRACKED
260				0.0	-6.08027D 01	-1.00132D-01	4	
				1.13187D-04	-2.02676D-05	-1.00132D-01		CRACKED
261	2.20041D 02	-4.30254D 00	3.19375D 01	2.24499D 02	-8.76056D 00	7.94635D 00	1	
	7.35621D-05	-1.24362D-05	2.44854D-05	7.52710D-05	-1.41451D-05	7.94635D 00		
262	1.81855D 02	7.89360D 00	1.75543D 01	1.83609D 02	6.13989D 00	5.70504D 00	1	
	6.02238D-05	-6.46157D-06	1.34583D-05	6.08960D-05	-7.13382D-06	5.70504D 00		
263	2.19961D 02	-3.93749D 00	3.10479D 00	2.20004D 02	-3.98053D 00	7.94314D-01	1	
	7.35172D-05	-1.23105D-05	2.38034D-06	7.35337D-05	-1.23270D-05	7.94314D-01		
264	2.58146D 02	-1.61336D 01	1.74879D 01	2.59257D 02	-1.72442D 01	3.63353D 00	1	
	8.68555D-05	-1.82852D-05	1.34074D-05	8.72812D-05	-1.87109D-05	3.63353D 00		
265	1.35147D 02	-1.10296D 01	3.81943D 01	1.44525D 02	-2.04077D 01	1.37953D 01	1	
	4.56004D-05	-1.04339D-05	2.92823D-05	4.91954D-05	-1.40288D-05	1.37953D 01		
266	9.27810D 01	1.45720D 00	2.55655D 01	9.94507D 01	-5.21258D 00	1.46220D 01	1	
	3.08541D-05	-4.15331D-06	1.96002D-05	3.34109D-05	-6.71006D-06	1.46220D 01		
267	1.34915D 02	-1.01174D 01	1.27176D 01	1.36022D 02	-1.12241D 01	4.97359D 00	1	
	4.54775D-05	-1.01182D-05	9.75017D-06	4.59017D-05	-1.05424D-05	4.97359D 00		
268	1.77281D 02	-2.26041D 01	2.53464D 01	1.80445D 02	-2.57681D 01	7.11538D 00	1	
	6.02238D-05	-1.63987D-05	1.94323D-05	6.14366D-05	-1.76116D-05	7.11538D 00		
269	4.69157D 01	-2.10163D 01	4.26122D 01	6.74427D 01	-4.15433D 01	2.57209D 01	1	
	1.66894D-05	-9.35121D-06	3.26694D-05	2.45581D-05	-1.72199D-05	2.57209D 01		
270	5.36715D 00	-8.03671D 00	3.04776D 01	2.98710D 01	-3.25406D 01	3.87991D 01	1	
	2.19089D-06	-2.94726D-06	2.33662D-05	1.15840D-05	-1.23404D-05	3.87991D 01		
271	4.62859D 01	-1.85397D 01	1.77483D 01	5.08270D 01	-2.30808D 01	1.43519D 01	1	
	1.63556D-05	-8.49421D-06	1.36070D-05	1.80964D-05	-1.02350D-05	1.43519D 01		
272	8.78345D 01	-3.15193D 01	2.98829D 01	9.48983D 01	-3.85831D 01	1.32996D 01	1	
	3.08541D-05	-1.48982D-05	2.29102D-05	3.35619D-05	-1.76059D-05	1.32996D 01		
273	-4.01684D 01	-3.48224D 01	4.47490D 01	7.33331D 00	-8.23241D 01	-4.32908D 01	1	
	-1.16484D-05	-9.59904D-06	3.43075D-05	6.56064D-06	-2.78080D-05	-4.32908D 01		
274	-8.18849D 01	-2.01995D 01	3.31088D 01	-5.79326D 00	-9.62911D 01	-2.35147D 01	1	
	-2.62850D-05	-2.63891D-06	2.53834D-05	2.88347D-06	-3.18074D-05	-2.35147D 01		
275	-4.16731D 01	-2.89056D 01	2.00476D 01	-1.42499D 01	-5.63287D 01	-3.61685D 01	1	
	-1.24457D-05	-7.55156D-06	1.53698D-05	-1.93355D-06	-1.80638D-05	-3.61685D 01		
276	4.33772D-02	-4.35286D 01	3.16877D 01	1.67118D 01	-6.01970D 01	2.77453D 01	1	
	2.19089D-06	-1.45117D-05	2.42939D-05	8.58046D-06	-2.09013D-05	2.77453D 01		
277	-1.27635D 02	-5.62569D 01	4.65236D 01	-3.33101D 01	-1.50581D 02	-2.62539D 01	1	
	-3.97320D-05	-1.23706D-05	3.56681D-05	-3.57429D-06	-4.85283D-05	-2.62539D 01		
278	-1.71368D 02	-3.65157D 01	3.58275D 01	-2.75881D 01	-1.80295D 02	-1.39922D 01	1	
	-5.52967D-05	-3.60354D-06	2.74678D-05	-1.81278D-07	-5.87190D-05	-1.39922D 01		
279	-1.31631D 02	-4.05427D 01	2.13573D 01	-3.57837D 01	-1.36390D 02	-1.25619D 01	1	
	-4.18497D-05	-6.93271D-06	1.63740D-05	-5.10843D-06	-4.36740D-05	-1.25619D 01		
280	-8.78975D 01	-6.02839D 01	3.20534D 01	-3.91901D 01	-1.08991D 02	-3.33482D 01	1	
	-2.62850D-05	-1.56997D-05	2.45743D-05	-7.61380D-06	-3.43709D-05	-3.33482D 01		

VITA

Alexander Jean Lassker

Candidate for the Degree of

Doctor of Philosophy

Thesis: NONLINEAR BEHAVIOR OF REINFORCED CONCRETE BEAMS BY THE FINITE ELEMENT METHOD

Major Field: Civil Engineering

Biographical:

Personal Data: Born in Arbon, Switzerland, May 11, 1940, son of Mr. and Mrs. Jean Paul Lassker.

Education: Graduated from Kantonsschule, Frauenfeld, Switzerland, in 1959; received the Diploma in Civil Engineering from the Swiss Federal Institute of Technology, December, 1964; enrolled at Oklahoma State University in September, 1967, and completed requirements for the Doctor of Philosophy degree at Oklahoma State University in May, 1972.

Professional Experience: Research assistant, Swiss Federal Institute of Technology, 1965-1967; graduate teaching assistant, Oklahoma State University, 1967-1968; Consulting Engineer at the Computing Center of Fides, Inc., Zurich, Switzerland, summer, 1968; teaching assistant, Oklahoma State University, 1968-1971; Senior Engineer with R. R. Nicolet and Associates, Consulting Engineers, Montreal, Canada, 1971.