ANALYSIS OF REINFORCED AND PRESTRESSED CONCRETE SLABS BY FINITE ELEMENT METHOD

ANALYSIS OF REINFORCED AND PRESTRESSED CONCRETE SLABS BY FINITE ELEMENT METHOD

by

JIANPING JIANG, B.Sc.

(Nanjing Institute of Technology, China)

A THESIS

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Engineering

McMaster University

May 1985

MASTER OF ENGINEERING (1985) McMASTER UNIVERSITY
(Civil Engineering and Engineering Mechanics) Hamilton, Ontario
Canada

TITLE: ANALYSIS OF REINFORCED AND PRESTRESSED CONCRETE SLABS BY FINITE ELEMENT METHOD

AUTHOR: Jianping Jiang

SUPERVISOR: Dr. F.A. Mirza and Dr. R.G. Drysdale

NUMBER OF PAGES: x, 151.

ABSTRACT

The finite element model, which includes the bond slip, dowel action and the tension stiffening effects in reinforced and prestressed concrete slabs, has been developed. Based on the previous investigations in this area, the emphasis of the present study is placed on investigating the influences of the three factors above on the predicted response of the concrete slabs.

The bond slip and dowel action effects are modelled by interface elements which join the steel elements to the concrete elements and are able to transfer internal stresses from concrete to steel bar or vice versa. Modelling of the tension stiffening effect is based on a fracture mechanics approach. The concept of this model is that an opposite and equal magnitude of existing tensile stress which is equal to or greater than the tensile strength of concrete is applied to a newly cracked surface in order to eliminate the stress which was left on the new crack surface due to the smeared cracking model and to include the tension stiffening effect.

Two numerical examples of a simply supported reinforced concrete slab and a post-tensioned concrete deck are presented. The results of the examples are compared with the test data and the analytical results obtained by other investigator and were found to be in fairly good agreement.

ACKNOWLEDGEMENTS

The author wishes to express his deepest appreciation to the following individuals and organizations:

- his supervisors, Drs. F.A. Mirza and R.G. Drysdale for bringing up the present topic and for valuable advice during the research and preparation of this thesis;

- his fellow graduate students, especially Mr. W. Chow for allowing his program to be modified and to be used in the present investigation and Mr. E. Moll for providing some test data in the simulations;

- the financial supports from McMaster University, the Nanjing Institute of Technology of P.R.C., NSERC and Ministry of Transport and Communications, are gratefully acknowledged;

- the Burke Sciences Word Processing Center for typing the manuscript of this thesis;

- finally, a special thanks to his wife, Xiaolu, for her patience and encouragement during the preparation of this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 General Remarks

The behavior of reinforced concrete structures has received the attention of many researchers over the past several decades. A vast amount of experimental work has been reported in the literature on scaled and full-scale models of reinforced concrete structural members and connections subjected to simulated loadings. Furthermore, the empirical formulae, based on the test data, have been widely used in structural design and in the development of the design codes.

Concurrent with the more recent development of the advanced finite element techniques, a large amount of effort has also been put into numerical modelling of the reinforced concrete structures. The objectives of analysis have been made wider in scope from early studies concentrated on the behavior of isolated elements such as beams, columns, beam-column joints, etc., to the total structural systems including slabs and beams, shear walls, shells, prestressed concrete atomic reactor vessels and atomic reactor buildings. Despite the applications mentioned above, the behavior of concrete is still a very complex one and extremely difficult to model numerically.

A realistic analysis of a reinforced concrete structure should include the following complexities:

(1) the nonlinearities in the behavior of concrete and steel;

(2) the influence of concrete cracking and crack propagation;

(3) the effect of bond-slip between steel and concrete;

(4) the shear transfer through dowel action, aggregate interlock, etc.;

(5) the influence of creep and shrinkage of concrete;

(6) the dynamic behavior such as causing load reversals.

Not all of the factors mentioned above have been incorporated in the finite element modelling of the concrete slabs to be presented in this thesis. The selection of a particular modelling approach can be dependent on the subjective judgement involved in an investigation in terms of how concrete would behave and which factors above realistically effect the structural response. Therefore, it is necessary to develop an analytical model for basic properties of reinforced concrete which is a rational one and to compare this model with the experimental results in order to verify it.

1.2 Purpose and Scope

The present study is a continuation of the finite element modelling of reinforced concrete slabs which involves the addition of bond slip, dowel action as well as tension stiffening effect.

1.2.1 Brief Review of Previous Study

The previous study by W. Chow [10] concentrated on the nonlinear behavior of concrete slabs due to concrete cracking and development of a finite element computer programme which included the following items:

(1) concrete was modelled by using twenty degrees of freedom, rectangular plate element with both in-plane and out-of-plane actions, the element stiffness matrix was computed via numerical integration based on a 3×3×5 integration point scheme (five across the depth);

(2) steel reinforcement was accommodated as discrete elements and not through an equivalent concrete area, and which was capable of modelling both ordinary and prestressed reinforcing steel;

(3) only linear constitutive properties of concrete and steel were considered;

(4) smeared cracking model and maximum stress criterion were employed for concrete cracking and was checked at each integration point;

(5) Newton-Raphson method for nonlinear analysis, under incremental loads, was adopted.

1.2.2 Scope of the Present Study

Although the results for the reinforced and the prestressed concrete slabs or decks from the previously developed model were reasonably fair, the finite element model itself was limited in explaining some of the discrepancies observed. The linear constitutive equations are, once again, employed in the present study and further extensions are incorporated such as the bond-slip behavior, the dowel action and the tension stiffening effect.

In Chapter Two, the phenomenon of bond slip between steel and concrete interface is described first. The available bond stress-slip relationships are briefly reviewed. Subsequently, the bond interface element is introduced and its stiffness matrix is derived. The transformation for the stiffness matrix and the load vector for the interface element is also presented. Finally, the effect of bond slip on the overall behavior of concrete slabs is examined by employing different values for the bond stiffness.

Chapter Three describes the phenomenon and the modelling of dowel action in concrete slabs. The element stiffness matrix, load vector and the transformation matrix are derived using similar concepts and method as used for the bond interface element. The influence of the dowel action on the response of concrete slabs and the interaction of bond slip and dowel action are also discussed.

In Chapter Four, a fracture mechanics approach is used to consider the effect of tension stiffening on the response of concrete slabs. This is based on the equality of loss of potential energy, due to a crack extension, to the work done in creating the new stress free surface. This is accomplished by eliminating stresses on the new crack surface just as the cracking takes place instead of reducing the stiffness gradually. This procedure is then incorporated into the computer program and some numerical examples are presented to study the influence of tension stiffening and also for comparison purposes.

Some comparisons of the results from the finite element model for the normally reinforced and the prestressed concrete slabs with the experimental results are presented in Chapter Five. The former was reported in the literature and the latter was carried out recently in the Applied Dynamics Laboratory at McMaster University.

In Chapter Six, the conclusions based on the present study and the recommendations for future investigation are reported.

CHAPTER 2

MODELLING OF BOND SLIP BEHAVIOR

2.1 Introduction

The bond between the reinforcing steel and concrete is considered to be one of the important factors in determining the load carrying capacity of a reinforced concrete member. For a realistic finite element modelling of concrete slabs, it is necessary to consider the influence of bond slip on the response of a slab and to develop a quantitative formulation of concrete-steel interface behavior. In this chapter, the emphasis is placed on the modelling of bond slip behavior of concrete slabs.

The interface element, which can be arranged along the entire steel-concrete interface such that the bond slip relationships derived from experimental testing can be taken into account, is employed to model the bond slip behavior of concrete slabs in the present study.

2.2 Description of the Bond Slip Phenomenon

The bond between the steel reinforcement and the surrounding concrete generally depends upon the following three components:

(a) chemical adhesion;

(b) friction;

(c) mechanical interaction between steel and concrete.

The bond behavior of plain bars and deformed bars is quite different. Since deformed bars are extensively used in concrete structures, the behavior of such bars should be fully understood.

Lutz and Gergely [1] have indicated that the bonding in plain bars depends primarily on chemical adhesion and friction, while for deformed bars, it depends mainly on the mechanical properties. For a deformed bar embedded in concrete, initially, chemical adhesion combined with mechanical interaction prevents slip. After the adhesion has been destroyed, the slip tries to occur and the ribs of the bar restrain slip by bearing against the concrete between the ribs. Friction does not occur in this case because of presence of the ribs. They summarized that slip of a deformed bar can occur in two ways: (1) the ribs can push the concrete away from the bar (wedging action), and (2) the ribs can crush concrete. Mirza and Houde [2] found, from experimental testing, that there was no crushing due to rib pressure or polishing of the surface due to sliding of the bar. The slip was explained only by the internal cracking of the first layer of the concrete surrounding the bar and by the bending and/or cracking of the small concrete teeth near the bar ribs.

In order that the nature of bond slip phenomenon can be well understood, the associated cracking pattern in bond slip problems must be taken into account. A simple example by Bresler (3) is shown in Figure 2.1. In this example, primary cracks form at random critical sections where the uniform tensile stress exceeds the concrete tensile strength. Slip occurs between the concrete and the reinforcing bar at the primary crack section. Concrete surfaces at the crack sections are free of stress and the force in the steel equals the external load.

Tensile stresses in concrete are present across the sections between the primary cracks, causing the tension stiffening. These are due to bending action that takes place as the concrete tries to deform with the reinforcing bar. The distribution and the magnitude of the bond stress along the concrete and steel interface controls the distribution of stresses in concrete and steel between the primary crack sections. A new crack forms as the external load increases and the uniform concrete stress exceeds the concrete tensile strength. Cracking will continue to take place between the existing cracks until the concrete stress does not exceed the concrete strength. At this stage, the remaining bond between the concrete and steel becomes minimal compared with the initial bond of the uncracked concrete.

From the example above, it can be recognized that perfect bond does not exist after the cracking has occurred and the bond stiffness between the concrete and steel approaches a very low value (nearly zero) as the load increases and the spacing of cracks decrease. In the finite element analysis of concrete structures, the analytical deflection

tion, based on the assumption of a perfect bond, showed a tendency to be smaller than those determined experimentally. This is more remarkable in the case of concentrated arrangements of steel such as in beams, columns, and prestressed concrete slabs. Therefore, it becomes necessary to include the bond slip behavior in the modelling of such concrete structures.

2.3 Bond Stress-Slip Relationship

The well-known formula for the bond stress, reported in standard concrete design textbooks, derived under the assumption that concrete resists no tension is given by

$$u = \frac{V}{\Sigma \text{ ojd}} \qquad (2.1)$$

where u is the bond stress, V is the tensile force in steel, Σ ojd is the surface area of steel surrounded by concrete. However, this equation does not always give the true bond stress.

Bresler[3] has derived a more general formula for bond stress-slip relationship. In his work, considering a section of a reinforced concrete prism of length Δx shown in Figure 2.2 in which the steel reinforcement and concrete are subjected to varying tensile stress, the following relationship has been derived from the equilibrium condition:

$$(\Delta f_s)A_s = - (\Delta f_c)A_c = u\pi D(\Delta x). \qquad (2.2)$$

In the equation above, Δf_s and Δf_c are the incremental forces in concrete and steel, respectively; D is the diameter of the round steel bar; u is the bond stress uniformly distributed over the length Δx. In the limit, as Δx becomes small, the local bond stress u(x) is then given by

$$u(x) = \frac{df_s}{dx} \frac{A_s}{\pi D} = \frac{df_s}{dx} \frac{D}{4} \qquad (2.3)$$

or

$$u(x) = \frac{df_c}{dx} \frac{A_c}{\pi D} = \frac{df_c}{dx} \frac{D}{4\overline{\rho}} \qquad (2.4)$$

where $\overline{\rho} = A_s/A_c$ is the reinforcement ratio based on concrete prism area A_c which is subjected to the uniformly distributed stress f_c. Equation (2.3) can be used to calculate the bond stress u(x) at the interface if the gradient (df_s/dx) of the tensile stress in steel is known.

Assuming that the origin 0 of the reference axes corresponds to the point of no relative displacement between steel and concrete, then the relative displacement g between steel and concrete can be computed at any point distant x by

$$g = \int_0^x \varepsilon_s \, dx - \int_0^x \varepsilon_c \, dx \qquad (2.5)$$

where ε_s and ε_c are the strains in steel and concrete, respectively. The differentiation of Equation (2.5) with respect to x and assuming elastic behavior for both steel and concrete:

$$\frac{dg}{dx} = \varepsilon_s - \varepsilon_c = \frac{f_s}{E_s} - \frac{f_c}{E_c} = \frac{1}{E_s}(f_s - nf_c) \qquad (2.6)$$

where E_s, E_c are the elastic modulii of steel and concrete, respectively; and n is the modular ratio E_s/E_c. Differentiating Equation (2.6) with respect to x again

$$\frac{d^2g}{dx^2} = \frac{1}{E_s}\left(\frac{df_s}{dx} - n\frac{df_c}{dx}\right) \qquad (2.7)$$

and substitution of Equations (2.3) and (2.4) yields

$$\frac{d^2g}{dx^2} = \frac{1}{E_s}\frac{4}{D}\ u(x)\ (1 + n\overline{\rho}). \qquad (2.8)$$

Equation (2.8) has been derived without considering any bond characteristic of the steel-concrete interface which may be defined by the bond stress-slip relationship. If such a relationship is defined in terms of slip only, then a function u(x) = F(g) defines a bond slip law.

The bond characteristic of the steel-concrete interface is too complicated to formulate from a purely theoretical point of view. The experimental work is required for realization of the actual behavior of the interface. Thus far, various empirical or semi-empirical formulations have been proposed. Nilson [4,5] suggested the following formulation in the form of a third-degree polynominal:

$$u = 3606(10^3d) - 5356(10^3d)^2 + 1986(10^3d)^3 \qquad (2.9)$$

where u is the bond stress in psi and d is the slip in inches. Mirza and Houde [2] proposed a fourth-degree polynomial relationship of the following form:

$$u = 1.95\times10^{6}d - 2.35\times10^{9}d^{2} + 1.39\times10^{12}d^{3} - 0.33\times10^{15}d^{4} \qquad (2.10)$$

and u and d are as those defined for Equation (2.9). From another test series [6,7], Nilson obtained a simplified bond stress-slip relationship of the following type:

$$u = 3100\ (1.43C + 1.5)d \times \sqrt{f'_c} \qquad (2.11)$$

where $u < (1.43C + 1.5)\sqrt{f'_c}$, C is the end distance in inches and f'_c is the compressive concrete strength of concrete in psi.

Although the bond-slip relationships above were derived based on the experimental data, it is still questionable if such equations (based on the tension specimen tests) would be appropriate for all concrete structures, e.g. concrete slabs under more complex state of stresses. An investigation of applicability of these relationships is beyond the scope of the present study, and will not be dealt with here.

2.4 Bond Interface Element (Joint Element)

One of the earliest finite element models for bond slip phenomena was developed by Ngo and Scordelis [8]. In their study, the bond link element with a discrete crack model was used to represent the bond behavior of the steel-concrete interface shown in Figure 2.3. It was composed of two orthogonal discrete springs which connect and transmit shear and normal forces between the two faces shown. The bond slip between the steel reinforcement and concrete and the corresponding bond stress transfer are expressed by the deformation characteristics of the springs.

As mentioned before for the deformed bar, the concrete around the bar follows the displacement of the bar to a tolerable degree through making contact with the ribs of the steel bar. Consequently, the internal cracks occur at the bar ribs and propagate into the surrounding concrete. Therefore, it appears to be difficult to model the behavior of concrete around the deformed bar using the bond link element. Another drawback to the use of the link element is that an artificially discrete crack pattern must be pre-selected. This leads to loss of the random nature of cracking.

The bond interface element shown in Figure 2.4 is a satisfactory element which avoids the artificial discretness of the bond link element and has the ability of modelling the entire interface. It must be noted that the stiffness matrix of the bond interface element is formulated in terms of the relative displacement between the top and the bottom surfaces. (See Goodman et al. [9].)

In the case of normally reinforced concrete or prestressed concrete slabs, the reinforcing bars are usually arranged with short spacings along the longitudinal and/or transverse directions. It will be costly computationally to use explicit nodes to connect the steel and the concrete elements. A more economical and reliable approach to deal with this problem is the use of implicit nodes at the steel level for joining steel and concrete elements and keeping the steel element within a concrete element. This has been adopted in the present study.

2.5 Stiffness Matrix of the Interface Element

The geometric layout of a steel bar within a rectangular concrete element and the deformed shape, used by Chow [10], are also employed in this study. While Chow's model for the steel-concrete interface employed only the corner nodes of the rectangular concrete element, the extension of his model is presented here which allows the use of separate nodes for the interface elements and thus also for the reinforcing steel elements. Figure 2.5 illustrates a steel element embedded within the concrete and Figure 2.6 shows the kinematic constraints for the steel element after deformation. Assuming that the normals to the mid-surface of the slab remain normal after deformation, it can be shown that

$$u^c = u^s - e\,\frac{\partial w^s}{\partial x} \tag{2.12}$$

in which u^c is the in-plane displacement of concrete along the x direction at mid-surface, u^s is the axial displacement of steel along the centroidal axis of the steel bar, and $w^s$is the vertical displacement of steel at the centroidal axis of the steel bar. The displacements u^s, w^s and u^c, for the finite element formulation, are given by the following equations and expressed in terms of the nodal degrees of freedom for the corner nodes of the concrete element at the mid-surface as indicated in Figure 2.7. The Derivation of the shape functions of u^s and w^s is presented in Appendix A.

$$u^s = \sum_{i=1}^{6} \bar{N}_i \; \delta_{si} \tag{2.13}$$

where $\bar{N}_1 = \frac{1 - \zeta}{2}$

$$\bar{N}_2 = \frac{6e}{\ell} \left(\frac{1 + \zeta}{2}\right) \left(\frac{\zeta - 1}{2}\right)$$

$$\bar{N}_3 = e[1 - 4\left(\frac{1 + \zeta}{2}\right) + 3 \left(\frac{1 + \zeta}{2}\right)^2]$$

$$\bar{N}_4 = \frac{1 + \zeta}{2}$$

$$\bar{N}_5 = \frac{6e}{\ell} \left(\frac{1 + \zeta}{2}\right) \left(\frac{1 - \zeta}{2}\right)$$

$$\bar{N}_6 = e \; [-2\left(\frac{1 + \zeta}{2}\right) + 3\left(\frac{1 + \zeta}{2}\right)^2]$$

$$\{\delta_s\}^T = \langle u_1{}^s, \; w_1{}^s, \; \Theta_1{}^s, \; u_2{}^s, \; w_2^s, \; \Theta_2{}^s \rangle$$

$$w^s = \sum_{i=1}^{4} N_i \; \delta'_{si} \tag{2.14}$$

where $$N_1 = 1 - (2 - \zeta)\left(\frac{1 + \zeta}{2}\right)^2$$

$$N_2 = \ell\left(\frac{1 - \zeta}{2}\right)^2\left(\frac{1 + \zeta}{2}\right)$$

$$N_3 = \left(\frac{1 + \zeta}{2}\right)^2(2 - \zeta)$$

$$N_4 = -\ell\left(\frac{1 - \zeta}{2}\right)\left(\frac{1 + \zeta}{2}\right)^2$$

$$\{\delta'_s\}^T = \langle w_1^{\,s},\ \Theta_1^{\,s},\ w_2^{\,s},\ \Theta_2^{\,s}\rangle$$

and $$u^c = \sum_{i=1}^{4} \tilde{N}_i\ \delta_{ci} \qquad (2.15)$$

where $$\tilde{N}_1 = \frac{1}{4}(1 - s)(1 - t)$$

$$\tilde{N}_2 = \frac{1}{4}(1 + s)(1 - t)$$

$$\tilde{N}_3 = \frac{1}{4}(1 + s)(1 + t)$$

$$\tilde{N}_4 = \frac{1}{4}(1 - s)(1 + t)$$

$$\{\delta_c\}^T = \langle u_1,\ v_1,\ w_1,\ w_{1,x},\ w_{1,y},\ u_2,\ v_2,\ w_2,\ w_{2,x},\ w_{2,y},$$
$$u_3,\ v_3,\ w_3,\ w_{3,x},\ w_{3,y},\ u_4,\ v_4,\ w_4,\ w_{4,x},\ w_{4,y}\rangle.$$

It should be noted that u_i, v_i, w_i, $w_{i,x}$, $w_{i,y}$ are the corner nodal degrees of freedom of the i^{th} node. It is pointed out here that the two nodes indicated in Figure 2.6 overlap and for perfect bond and no dowel action, the two sets of degrees of freedom are kept the same. In the case of bond slip and/or dowel action, the appropriate degrees of freedom at the two nodes of the ends of steel element are forced to be independent as the case may be. Rewriting Equation (2.12) in the following form

$$u^s = u^c + e \frac{\partial w^s}{\partial x} \tag{2.16}$$

can be used to evaluate the in-plane displacements of concrete u_c^s at the steel level along the x direction. In order to get the expression for u_c^s in terms of $\{\delta_c\}$, it is required to derive a transformation matrix to relate $\{\delta_s'\}$ to $\{\delta_c\}$ through the following equation;

$$\underset{4\times1}{\{\delta_s'\}} = \underset{4\times20}{[T_c]}\ \underset{20\times1}{\{\delta_c\}} \tag{2.17}$$

where $[T_c]$ can be obtained from the shape functions used for the retangular, nonconforming plate bending element in the following manner.

$$\underset{4\times20}{[T_c]} = \left[\begin{array}{cccccccccc} 0 & 0 & N_1^c & N_2^c & N_3^c & 0 & 0 & N_4^c & N_5^c & N_6^c \\ 0 & 0 & N_{1,x}^c & N_{2,x}^c & N_{3,x}^c & 0 & 0 & N_{4,x}^c & N_{5,x}^c & N_{6,x} \\ 0 & 0 & N_1^c & N_2^c & N_3^c & 0 & 0 & N_4^c & N_5^c & N_6^c \\ 0 & 0 & N_{1,x}^c & N_{2,x}^c & N_{3,x}^c & 0 & 0 & N_{4,x}^c & N_{5,x}^c & N_{6,x}^c \end{array}\right.$$

$$\left.\begin{array}{cccccccccc} 0 & 0 & N_7^c & N_8^c & N_9^c & 0 & 0 & N_{10}^c & N_{11}^c & N_{12}^c \\ 0 & 0 & N_{7,x}^c & N_{8,x}^c & N_{9,x}^c & 0 & 0 & N_{10,x}^c & N_{11,x}^c & N_{12,x}^c \\ 0 & 0 & N_7^c & N_8^c & N_9^c & 0 & 0 & N_{10}^c & N_{11}^c & N_{12}^c \\ 0 & 0 & N_{7,x}^c & N_{8,x}^c & N_{9,x}^c & 0 & 0 & N_{10,x}^c & N_{11,x}^c & N_{12,x}^c \end{array}\right]$$

The shape functions N_1^c to N_{12}^c are given in Appendix A; $N_{i,x}^c$ denotes the derivative of N_i^c with respect to x and $\{\delta_s'\}$ and $\{\delta_c\}$ are as defined before. Substituting Equations (2.14) and (2.15) into Equation (2.16) yields

$$u_c^s = u^c + e \frac{\partial w^s}{\partial x}$$

or

$$u_c^s = \sum_{i=1}^{4} \tilde{N}_i \delta_{ci} + e \sum_{i=1}^{4} N_{i,x} \delta'_{si}$$

$$= [\tilde{N}]\{\delta_c\} + e\,[N_{i,x}][T_c]\{\delta_c\}$$

$$= ([\tilde{N}] + e < N_{i,x} > [T_c])\{\delta_c\}$$

$$= [N_c{}^s]\ \{\delta_c\}. \qquad (2.18)$$

Therefore, the required matrix of the shape functions for the interface element is given by

$$\underset{1\times 20}{[N_c{}^s]} = [\tilde{N}] + e < N_{i,x}> [T_c].$$

To derive the interface element stiffness matrix, Equation (2.18) is rewritten so as to include the degrees of freedom allowed for the steel element. That is,

$$u_c^s = [\bar{N}_c^s]\{\delta_b\} \tag{2.19}$$

where

$$\underset{1\times 22}{[\bar{N}_c{}^s]} = \langle \underset{1\times 20}{[N_c{}^s]} \quad 0 \quad 0\rangle$$

and $\underset{1\times 22}{\{\delta_b\}} = \langle u_1,\ v_1,\ \ldots\ldots\ w_{4,y},\ u_1{}^s,\ u_2{}^s\rangle.$

Note that the first twenty degrees of freedom are those employed for the rectangular concrete element and u_1^s and u_2^s are the longitudial degrees of freedom for the steel element.

Equation (2.13) is employed to calculate the axial displacement of steel. It is also required to be transformed from $\{\delta_b\}$ to $\{\delta_s''\}$, as follows

$$\underset{6\times 1}{\{\delta_s''\}} = \underset{6\times 22}{[T_s]}\ \underset{22\times 1}{\{\delta_b\}} \tag{2.20}$$

where

$$\underset{6\times 22}{[T_s]} = \left[\begin{matrix} 0 & 0 & N_1^c & N_2^c & N_3^c & 0 & 0 & N_4^c & N_5^c & N_6^c & 0 \\ 0 & 0 & N_{1,x}^c & N_{2,x}^c & N_{3,x}^c & 0 & 0 & N_{4,x}^c & N_{5,x}^c & N_{6,x}^c & 0 \\ 0 & 0 & N_1^c & N_2^c & N_3^c & 0 & 0 & N_4^c & N_5^c & N_6^c & 0 \\ 0 & 0 & N_{1,x}^c & N_{2,x}^c & N_{3,x}^c & 0 & 0 & N_{4,x}^c & N_{5,x}^c & N_{6,x}^c & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}\right.$$

$$\left.\begin{matrix} 0 & N_7^c & N_8^c & N_9^c & 0 & 0 & N_{10}^c & N_{11}^c & N_{12}^c & 0 & 0 \\ 0 & N_{7,x}^c & N_{8,x}^c & N_{9,x}^c & 0 & 0 & N_{10,x}^c & N_{11,x}^c & N_{12,x}^c & 0 & 0 \\ 0 & N_7^c & N_8^c & N_9^c & 0 & 0 & N_{10}^c & N_{11}^c & N_{12}^c & 0 & 0 \\ 0 & N_{7,x}^c & N_{8,x}^c & N_{9,x}^c & 0 & 0 & N_{10,x}^c & N_{11,x}^c & N_{12,x}^c & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{matrix}\right]$$

and $\{\delta_s''\} = \langle w_1^s, \theta_1^s, w_2^s, \theta_2^s, u_1^s, u_2^s\rangle$ are the displacement degrees of freedom of the steel element at the local level, as shown in Figure 2.6. [Note that the first two rows of $[T_c]$ and $[T_s]$ are evaluated at $(-1, t_0)$ and the second two rows at $(+1, t_0)$, and t_0 is the dimensionless coordinate of the nodes of steel element.]

Once the axial displacement of steel u^s and the in-plane displacement of concrete at the steel level u_c^s are known, the relative displacement at the interface can then be obtained from

$$u_r = u_c^s - u^s = [\overline{N}_c^s]\{\delta_b\} - [N_s^s]\{\delta_b\}$$

$$= ([\bar{N}_c^s] - [N_s^s])\ \{\delta_b\} = [N_b]\{\delta_b\} \qquad (2.21)$$

where

$$[N_b] = [\bar{N}_c^s] - [N_s^s]$$

$$[N_s^s] = [\bar{N}'][T_s]$$

$$[\bar{N}'] = \langle \bar{N}_2, \bar{N}_3, \bar{N}_5, \bar{N}_6, \bar{N}_1, \bar{N}_4 \rangle$$

and $\bar{N}_i$ above are as those presented in Equation (2.13)

The strain energy of the interface element due to the relative displacement u_r is given by

$$U = \frac{1}{2} \int_0^L k_b (u_r)^2\ dx = \frac{1}{2} \int_0^L \{\delta_b\}^T [N_b]^T k_b [N_b]\{\delta_b\} dx$$

where k_b represents the distributed stiffness of the interface due to bonding. The equation above can be further manipulated to yield

$$U = \frac{1}{2}\ \{\delta_b\}^T (\int_0^L\ [N_b]^T k_b [N_b] dx)\{\delta_b\}$$

$$= \frac{1}{2}\ \{\delta_b\}^T [K_b]\{\delta_b\}. \qquad (2.22)$$

The stiffness matrix of the interface element is therefore computed by

$$[K_b] = \int_0^L [N_b]^T k_b [N_b] dx \qquad (2.23)$$

in terms of the degrees of freedom $\{\delta_b\}$.

The Gaussion quadrature has been used to perform the integration of Equation (2.23) numerically. Also note that the bond stiffness k_b is kept constant during an analysis and the integration follows the steps below;

$$[K_b] = \frac{k_b L}{2} \int_{-1}^{+1} [N_b]^T [N_b] d\zeta$$

$$= \frac{k_b L}{2} \sum_{i=1}^{3} H_i [N_b(\zeta_i)]^T [N_b(\zeta_i)] \qquad (2.24)$$

where ζ is the dimensionless coordinate from -1 to +1, H_i are the weighting factors and ζ_i are the integration points.

The following assumptions have been made in deriving Equation (2.24) above: (1) steel and concrete have the same vertical displacement and rotations along the interface element except for the horizontal displacements; (2) mechanical characteristics of the interface element depends on the value of k_b; and (3) dowel action is neglected.

2.6 Load Vector for Interface Element

Evaluation of the load vector for forces transferred by the interface element, from steel to concrete and vice versa, is now considered. In the present study, a simplified formulation has been used to obtain the load vector for the interface element using the linear elastic theory.

The load vector due to the bond stresses is obtained as

$$\{P^n\} = [K_b]\{\delta_b\} \tag{2.25}$$

where $\{P^n\}$ = shear force transferred by the bond stresses along the interface

$[K_b]$ = interface element stiffness matrix

$\{\delta_b\}$ = relative displacement vector for concrete and steel.

Goodman et al. [9] can be consulted for more details. Equation (2.25) has been used to calculate the load vectors for the interior interface elements. If the elements are at the boundary, at least one node of the steel element is at the boundary and the displacements of this node must be recalculated to satisfy the compatibility of deformations at the boundary.

The corresponding transformation matrix for recalculation of the load vectors of the elements at the boundary is derived in the next section. It should be mentioned here that the load vectors of steel elements at the boundary is also required for such recalculation.

2.7 Transformation Matrix for Interface Element at Boundary

Anand and Shaw [11] reported the following equations for a transformed set of degrees of freedom

$$\{f'\} = [T]^T\{f\} \tag{2.26}$$

$$[K'] = [T]^T[K]\ [T] \tag{2.27}$$

$$\{\delta'\} = [T]\ \{\delta\} \tag{2.28}$$

where $\{f\}$, $[K]$ and $\{\delta\}$ are the non-transformed load vector, stiffness matrix and the degrees of freedom, respectively;

$\{f'\}$, $[K']$ and $\{\delta'\}$ are the transformed load vector, stiffness matrix and the degrees of freedom, respectively; and

$[T]$ is the transformation matrix.

Following this method, the transformation matrix, the new stiffness matrix and the new load vector are as follows:

$$\underset{22\times1}{\{\delta_b\}} = \underset{22\times21}{[T_b]}\ \underset{21\times1}{\{\delta_b'\}} \tag{2.29}$$

$$\underset{21\times21}{[K_b']} = \underset{21\times22}{[T_b]^T}\ \underset{22\times22}{[K_b]}\ \underset{22\times21}{[T_b]} \tag{2.30}$$

$$\underset{21\times1}{\{\overline{P}^n\}} = \underset{21\times22}{[T_b]^T} \underset{22\times1}{\{P^n\}} \tag{2.31}$$

Where $\{\delta_b\}$ is as defined in Equation (2.19)

$\{\delta_b'\}$ is the vector of reduced degrees of freedom, for example, if node 1 at boundary (see Figure 2.8), $\underset{1\times21}{\{\delta_b'\}} = \langle u_1, v_1,$ $\ldots\ldots w_{4,y}, u_2\rangle$

$[K_b]$ is as defined in Equation (2.23)

$[K_b']$ is the new stiffness matrix after transformation

$\{P^n\}$ is as defined in Equation (2.25)

$\{\overline{P}^n\}$ is the new load vector after transformation

$[T_b]$ is the transformation matrix given in Appendix B.

2.8 Effect of Bond Stiffness k_b

As described in Sections 2.2 and 2.3, the bond slip behavior of the steel-concrete interface is very complicated and numerical formulation is extremely difficult. Even though a vast amount of experimental investigation has been done and several empirical equations have been proposed, a simple and reliable analytical expression for a convenient adaptation of value of k_b in the stiffness matrix for the general case is not yet available.

For an investigation of the effect of k_b, a sensitivity analysis has been carried out for the value of k_b on the predicted response of concrete slabs. Therefore, the effect of k_b has been examined for different constant values of k_b instead of using some empirical equations.

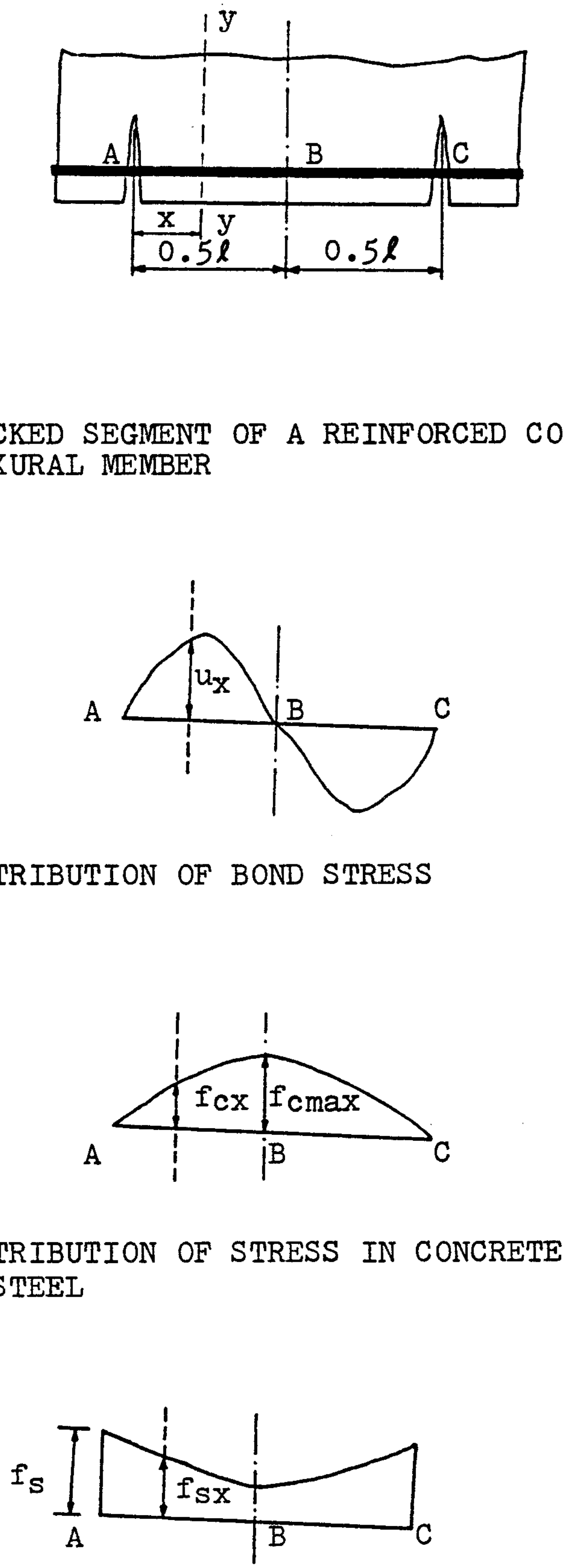
The simply supported normally reinforced concrete slab tested by Taylor et al. [12] is used to perform the simulations. The results obtained from analyses for different values of k_b are shown in Figures 2.9 to 2.11. From these analytical results, the following conclusions can be ascertained:

(1) the value of k_b has no effect on the predicted response of a reinforced concrete slab before the appearance of a crack;

(2) after cracking of concrete, there is very little difference among the responses for different values of k_b.

The first conclusion is obviously correct because steel and concrete have the same deformation before concrete cracks because of the perfect bond existing between the steel bar and concrete interface. The second one can be explained due to the smeared cracking model used. The cracking pattern consists of many finely spaced (or smeared) cracks perpendicular to the principal stress direction (see Figure 2.12). Once the applied load reaches the cracking load there are innumerable cracks in an element so that the bond stiffness and the bond stress are reduced and the shear resistance of the interface element deteriorates very fast. On the other hand, the crack spacing and widths are important factors for determining the bond stiffness of the interface element. When the smeared cracking model is assumed, the crack spacings and widths are difficult to evaluate, and hence k_b does not include these factors in the modelling. Both of the conclusions above are as expected.

As much as the bond slip has very little effect on the behaviour of the reinforced concrete slabs, it does effect the stiffness of a reinforced concrete beam. The model presented in this chapter is equally applicable to the reinforced concrete beams.



(a) CRACKED SEGMENT OF A REINFORCED CONCRETE FLEXURAL MEMBER

(b) DISTRIBUTION OF BOND STRESS

(c) DISTRIBUTION OF STRESS IN CONCRETE ADJACENT TO STEEL

(d) DISTRIBUTION OF STRESS IN STEEL

FIG. 2.1 STRESS DISTRIBUTION IN CRACKED SEGMENT

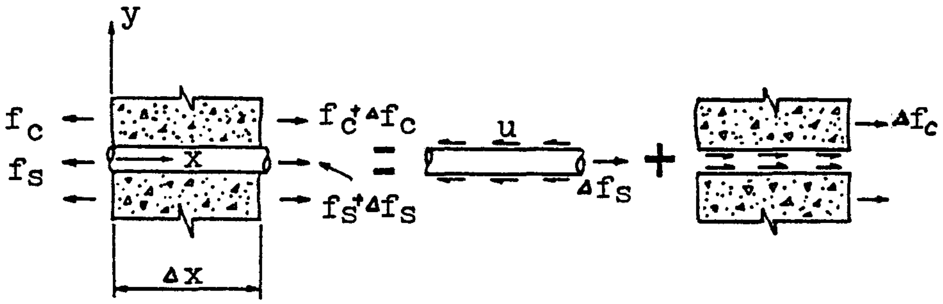
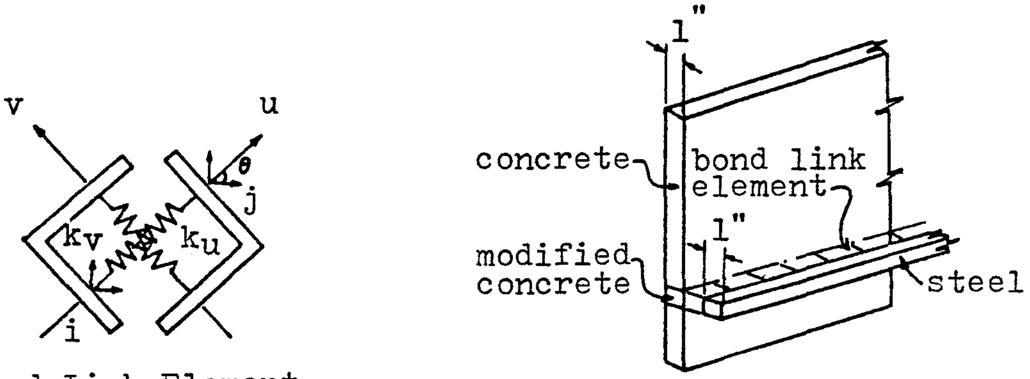


FIG. 2.2 STRESSES ON ISOLATED STEEL AND CONCRETE ELEMENTS



Bond Link Element

FIG. 2.3 BOND LINK MODEL

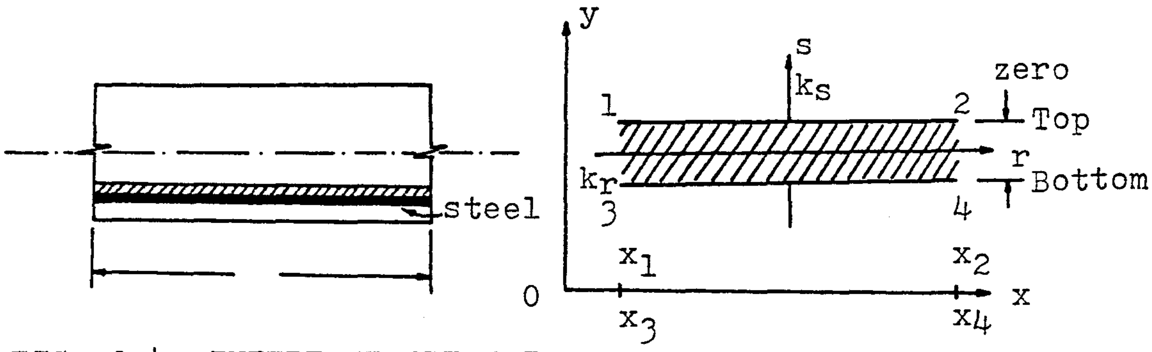


FIG. 2.4 INTERFACE ELEMENT

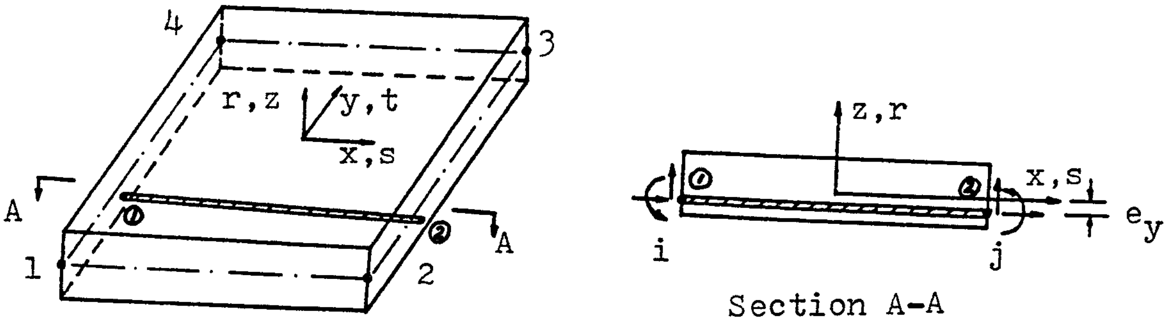


FIG. 2.5 DEGREE OF FREEDOM OF STEEL BAR ELEMENT PARALLEL TO X AXIS

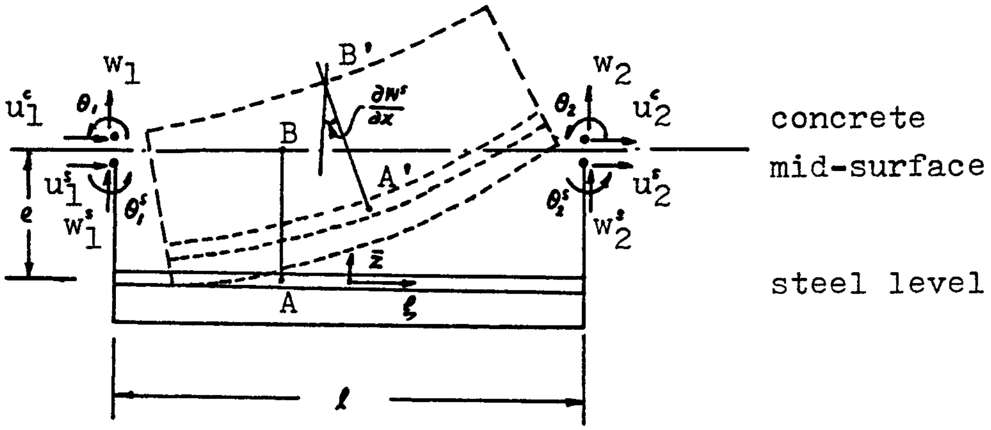


FIG. 2.6 DEFORMATION OF A REINFORCED CONCRETE SLAB ELEMENT

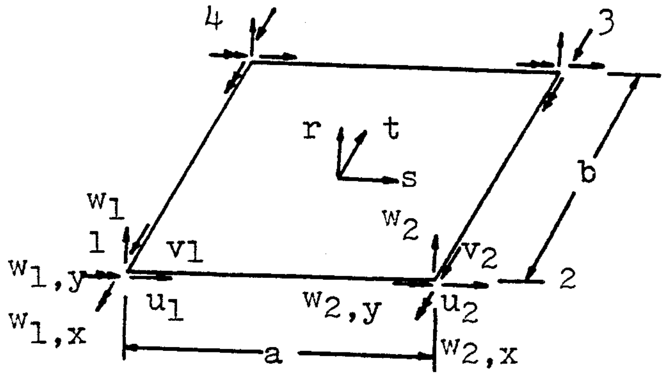


FIG. 2.7 NODAL DEGREES OF FREEDOM OF RECTAGULAR CONCRETE ELEMENT

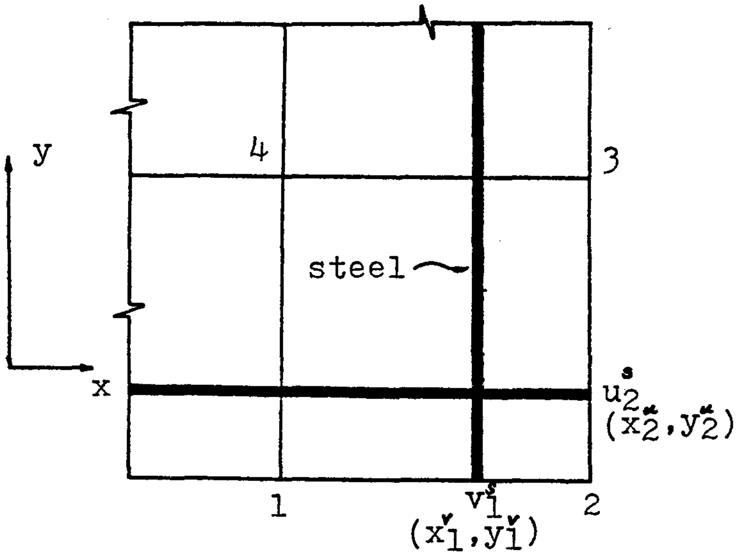


FIG. 2.8 STEEL ELEMENT AT THE BOUNDARY

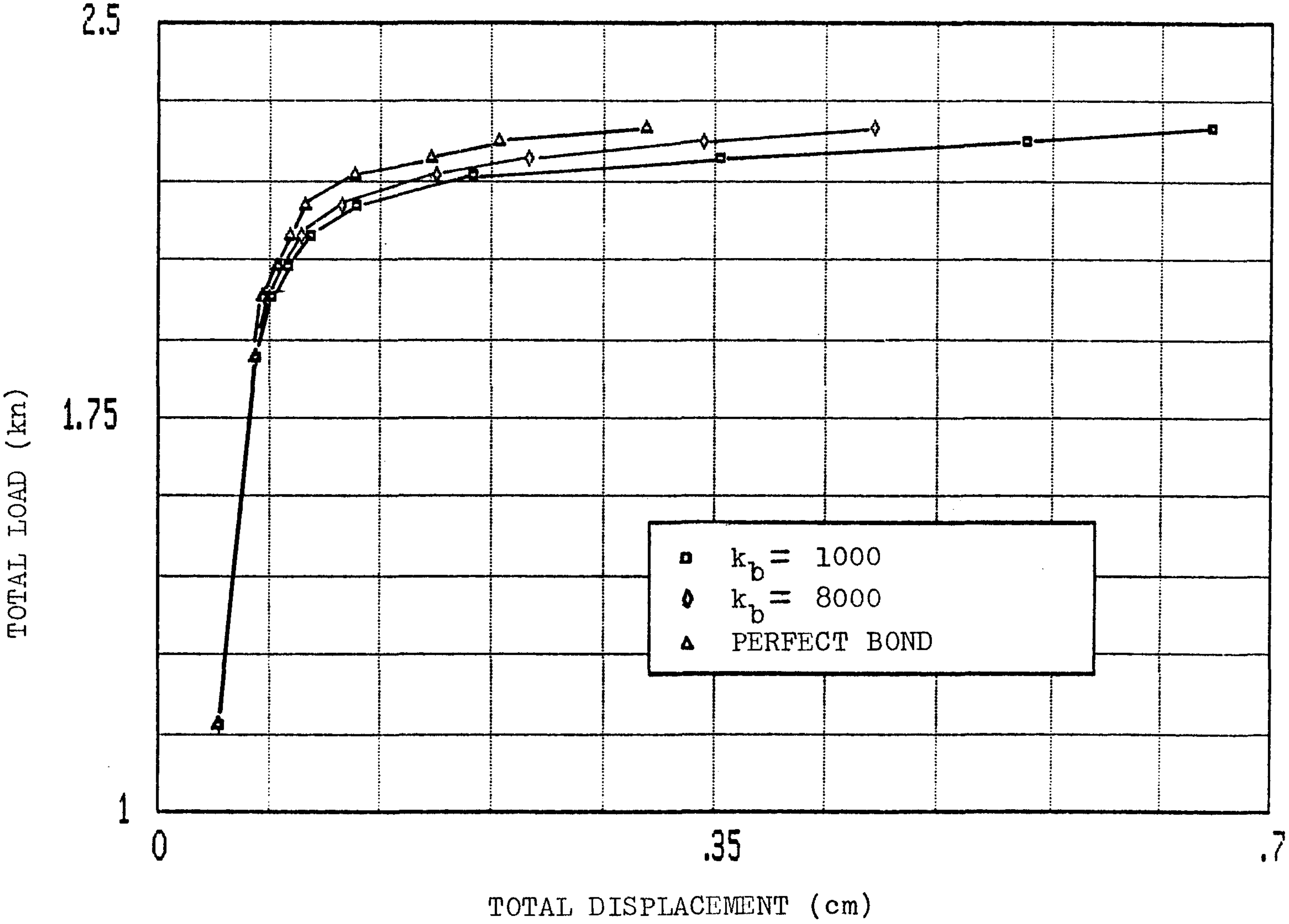


FIG. 2.9 EFFECT OF BOND STIFFNESS k_b ON REINFORCED CONCRETE SLAB

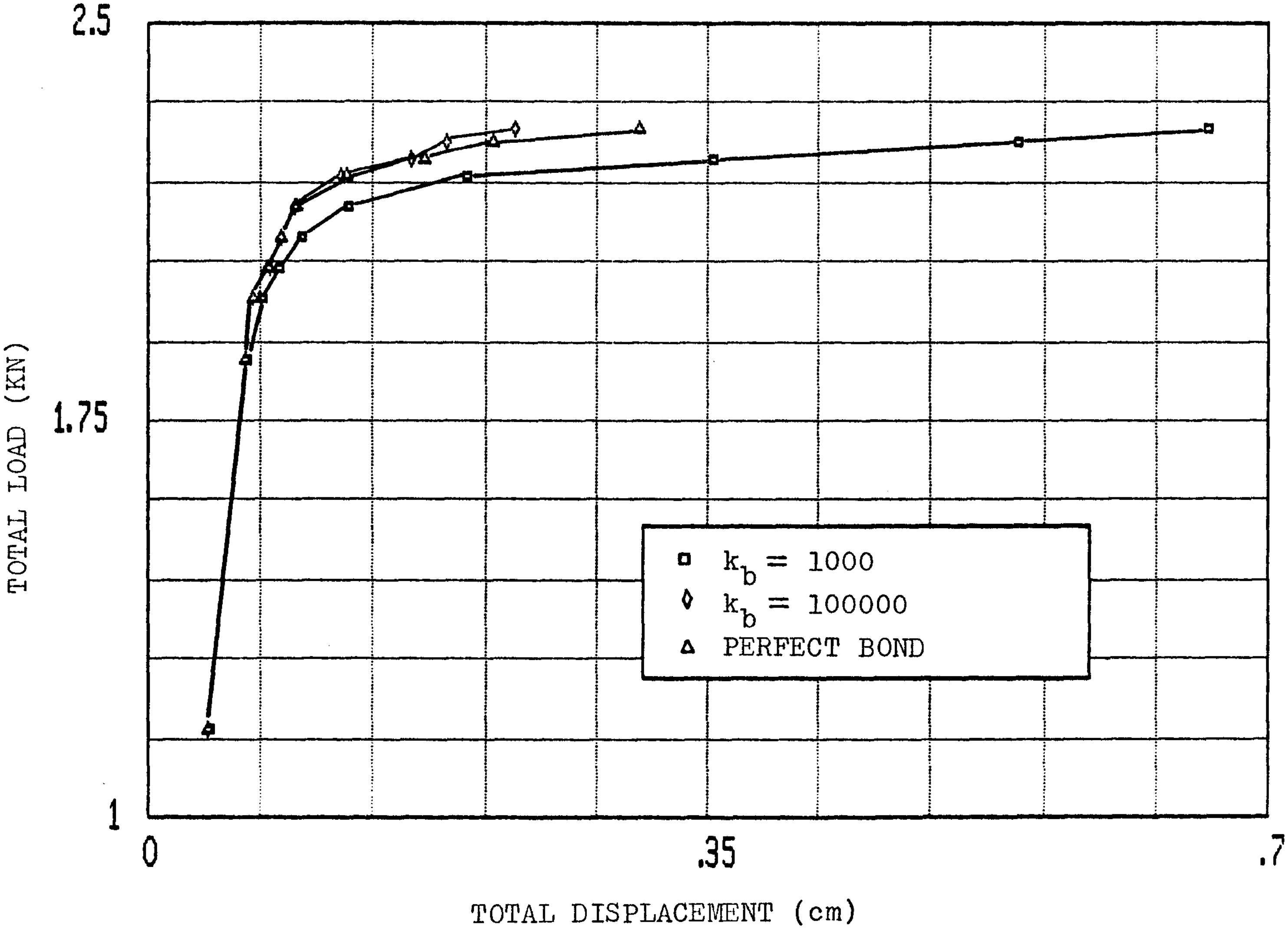


FIG. 2.10 EFFECT OF BOND STIFFNESS k_b ON REINFORCED CONCRETE SLAB

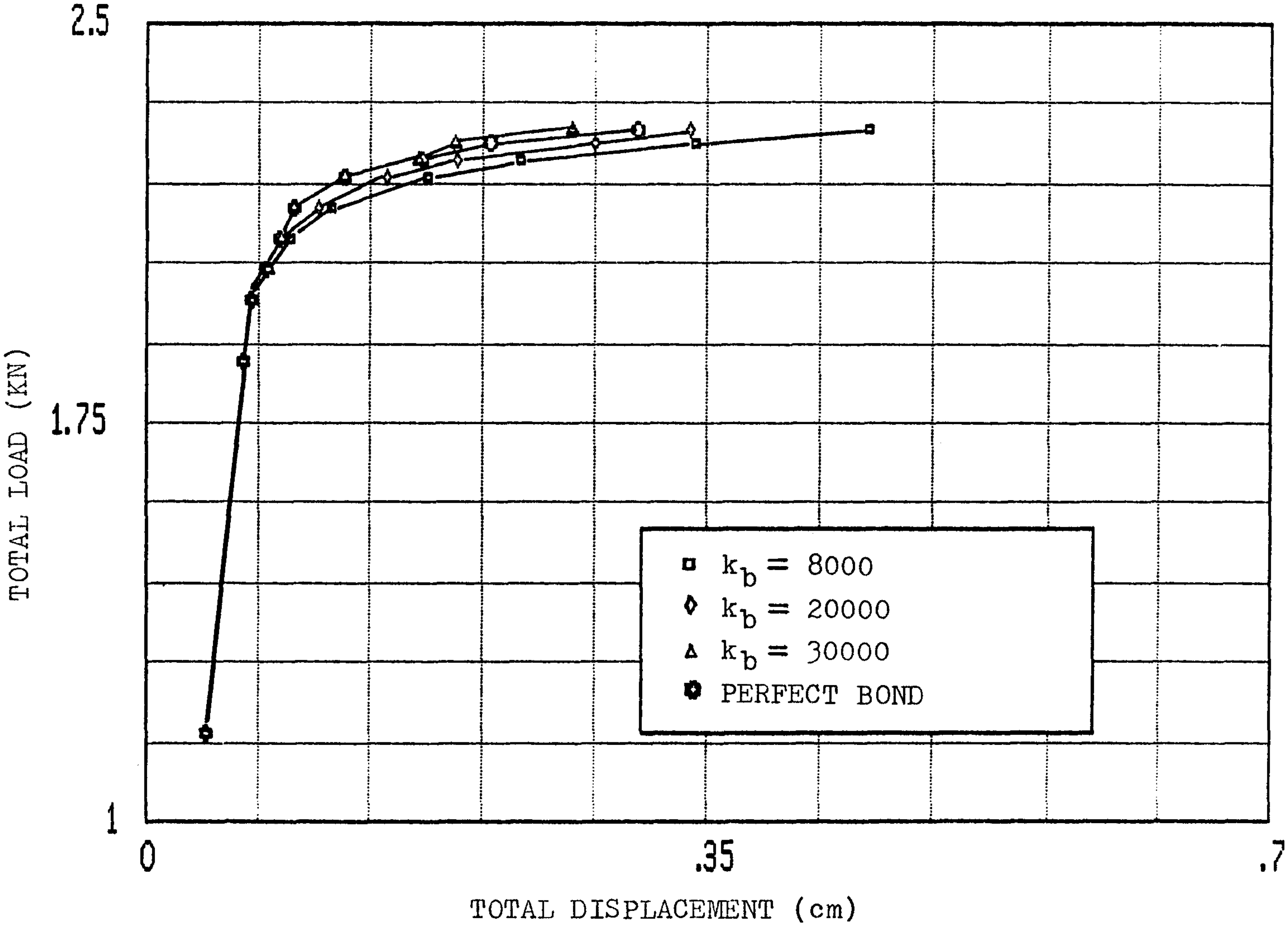


FIG. 2.11 EFFECT OF BOND STIFFNESS k_b ON REINFORCED CONCRETE SLAB

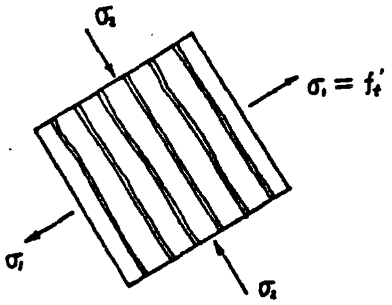


FIG. 2.12 IDEALIZATION OF A SINGLE CRACK FOR SMEARED CRACKING MODEL

CHAPTER 3

MODELLING OF DOWEL ACTION BEHAVIOR

3.1 Introduction

The reinforced concrete slabs supported along their edges by beams or walls seldom produce problems in shear. However, the strength of slabs supported directly by columns or bridge decks subjected to concentrated loads are very often governed by shear. If shear failure occurs, it is likely to be by punching shear as shown in Figure 3.1. A diagonal crack front forms following the shape of a truncated cone or a pyramid around a column (or a concentrated load), extending from the bottom (or the top) of the slab (or deck) diagonally upward (or downward) to the top (or the bottom) surface. At such a section, due to reinforcing bars crossing the crack, shearing displacements along the crack are resisted in part due to the dowel action in the steel bars. Thus, the shear resistance is mostly offered by the dowel action due to local bending of the reinforcing bars as shown in Figure 3.2.

Although the dowel action in the reinforced concrete beams has been studied by a number of investigators, only limited attention has focused on modelling of the dowel action in concrete slabs. Due to the high complexity of the actual dowel action, many previous studies only focused on the determination of some empirical dowel stiffness relationships and the ultimate load capacity from dowel tests. Therefore, reliable results are available only for a few particular cases.

The finite element modelling of dowel action in concrete beams was first introduced by Ngo et al. [8]. In their study, a linear strain triangular element was used to model the local flexural behavior of the reinforcing bar and linkage elements to transfer the dowel forces as shown in Figure 3.2. It was assumed that the effective dowel length was dependent upon the spring stiffness of the linkage element. Following this study, some other investigations have also been reported in the literature.

In the sections to follow the interface element, previously introduced in Section 2.4, is also used to model the dowel action. The stiffness matrix and the load vector as well as the transformation matrix presented follow the method described previously. The effect of dowel stiffness k_d and the interaction of dowel action and bond slip are also discussed.

3.2 Dowel stiffness k_d

For finite element modelling of the dowel action, the dowel stiffness of the steel reinforcement k_d is an important parameter in the relationship between the dowel force and the shear displacement. It has been observed from many experimental results that the dowel stiffness of a reinforcing bar crossing a crack is dependent on the bar size, embedded length, deterioration of bond, spliting of the concrete, etc. An extensive review of work on the dowel stiffness has been reported by Jiminez-Perez et al.[13]

Among the several alternatives that have been proposed, Fardis and Buyukozturk[14] have found that the following expression for the dowel stiffness of a bar of diameter d and Modulus of elasticity of steel E_s is in a reasonable agreement with the experimental results:

$$k_d = \frac{d}{8} [\pi E_s \times k_f^3 (\frac{f'_c}{4,000})^6]^{1/4} \qquad (3.1)$$

where f'_c is the concrete strength in psi; k_f is the foundation modulus which depends mainly on the tensile stress and is approximately equal to 3,000ksi/in (8.2X10 MN/m^3) for no direct tension in the reinforcement and decreases almost linearly with tensile stress to a value of about 1,500ksi/in (4.1X10 MN/m^3) at 50ksi(350 MN/m^2). It must be noted that Equation (3.1) has been developed by considering the steel bar as a beam on an elastic foundation and its validity depends on the assumption that no local concrete crushing occurs under the dowel bearing stresses and in the absence of tensile stresses normally observed for the concrete slabs. Therefore, its applicability to concrete slabs, failing in punching shear, is questionable and has not been used in this study.

However, the proposed model does allow the use of an expression for k_d which can effectively represent the dowel action.

3.3 Stiffness Matrix for Modelling of Dowel Action

In Section 2.4, the interface element was introduced to model the bond slip behavior and is now modified to develop the interface element to model dowel action.

Through the strain energy expression, the stiffness matrix for modelling of the dowel action can now be derived. Equations (2.14) and (2.17) can also be employed to obtain the vertical displacement of steel w^s as

$$w^s = [N]\{\delta_s'\} \qquad (3.2)$$

where $[N]$ and $\{\delta_s'\}$ are the shape functions and the degrees of freedom of the steel element as defined in Equation (2.14).

The vertical displacement of concrete at the steel level is given by

$$w_c^s = [N][T_c]\{\delta_c\} = [N_c{}^c]\{\delta_c\} \qquad (3.3)$$

where $[N_c{}^c] = [N][T_c]$, $[N]$ is the same as Equation (3.2), $[T_c]$ is as defined in Equation (2.17) and $\{\delta_c\}$ as given in Equation (2.15). The relative vertical displacement is now calculated as

$$w_r = w_c^s - w^s = \underset{1X26}{[N_d]} \; \underset{26X1}{\{\delta_d\}} \qquad (3.4)$$

where $[N_d] = < N_{c1}^c, N_{c2}^c, N_{c3}^c, \ldots\ldots N_{c20}^c, 0, N_1, N_2, 0, N_3, N_4 >$

and $\{\delta_d\} = < u_1, v_1, w_1, \ldots\ldots w_{4,y}, u_1{}^s, w_1{}^s, \Theta_1{}^s, u_2{}^s, w_2{}^s, \Theta_2{}^s >.$

The strain energy in the steel bar due to dowel action can be written as

$$U = \frac{1}{2} \int_0^L k_d \, (w_r)^2 \, dx = \frac{1}{2} \int_0^L \{\delta_d\}^T \, [N_d]^T \, k_d [N_d] \, \{\delta_d\}^T \, dx$$

$$= \frac{1}{2} \, \{\delta_d\}^T \int_0^L ([N_d]^T \, k_d \, [N_d] \, dx) \, \{\delta_d\}$$

$$= \frac{1}{2} \{\delta_d\}^T [K_d] \{\delta_d\} \qquad (3.5)$$

Therefore, the stiffness matrix for the interface element due to dowel action is

$$[K_d] = \int_0^L [N_d]^T \, k_d \, [N_d] \, dx \qquad (3.6)$$

and can be evaluated using the numerical integration as was done for Equation (2.24). Although, the procedure above allows k_d to vary along the steel bar length L, a constant value for k_d has been used in this study. This can be thought of as the distributed stiffness along the interface length.

It should be noted that if both the dowel action and the bond slip are considered, then the total stiffness of the interface element is the sum of two components $[K_d]$ and $[K_b]$ in Equations (2.24) and (3.6).

3.4 Load Vector and Transformation Matrix

The contribution of the internal stresses to the load carrying capacity of the concrete slabs via the load vector due to the dowel action must be included in a finite element analysis. Prior to onset of punching shear, the shear force due to the external load is shared by both the concrete and steel. As soon as the punching shear occurs, the shear force is carried only by the reinforcing bar through the dowel action.

Referring to Equation (2.25) and following the method described in Section 2.6, the load vector due to the dowel action is given by

$$\{P^n{}_d\} = [K_d]\{\delta_d\} \qquad (3.7)$$

where $\{P^n{}_d\}$ = shear force transferred by the dowel action

$[K_d]$ = stiffness matrix for the dowel action

$\{\delta_d\}$ = degrees of freedom defined in Equation (3.4).

When the interface element for the dowel action is located at the boundary, not only the load vector has to be transformed to the corresponding load vector for the rectangular concrete element, but the stiffness matrix must also be transformed in the same manner as was done for the bond interface element in the previous chapter. Based on the approach used in Section 2.7, the transformation matrix, the transformed stiffness matrix and the load vector due to dowel action can be derived in the following manner.

$$\underset{26X1}{\{\delta_d\}} = \underset{26X23}{[T_d]} \; \underset{23}{\{\bar{\delta}_d\}} \qquad (3.8)$$

$$\underset{23X3}{[\bar{K}_d]} = \underset{23X26}{[T_d]^T} \; \underset{26X26}{[K_d]} \; \underset{26X23}{[T_d]} \qquad (3.9)$$

$$\underset{23X1}{\{\bar{p}^n{}_d\}} = \underset{23X26}{[T_d]} \; \underset{26X1}{\{p_d{}^n\}} \qquad (3.10)$$

where $\{\delta_d\}$ is as defined in Equation (3.4)

$\{\bar{\delta}_d\}$ is the reduced degrees of freedom, for instance, assuming node 1 at boundary, $\{\bar{\delta}_d\} = \langle u_1, v_1, \ldots\ldots w_{4,y}, u_2{}^s, w_2{}^s, \Theta_2{}^s \rangle$;

$[K_d]$ is as defined in Equation (3.6);

$[\bar{K}_d]$ is the transformed stiffness matrix;

$\{p_d^n\}$ is as defined in Equation (3.7);

$\{\bar{p}^n\}$ is the transformed load vector;

and $[T_d]$ is the transformation matrix, and its explicit form is presented in Appendix C.

It should be noted that the difference between the bond interface element and the dowel interface element is that the former only considers the axial displacement of steel to be different from that of concrete, and the latter separates the degrees of freedom related to vertical displacements and rotations of steel from those of the concrete element at the steel level. For modelling of the interaction of bond slip and dowel action in concrete slabs, the two are combined by separating all of the degrees of freedom of the steel element from those of the concrete element and adding the individual stiffness matrices $[K_b]$ and $[K_d]$ as mentioned before. From a theoretical point of view, the combined model is expected to predict the behaviour of reinforced concrete slabs much better than the bond slip model or the dowel action model alone.

3.5 Effect of Dowel Stiffness k_d

To study the steel-concrete interface behavior of the reinforced slabs, the combined effects of several parameters such as bond slip, dowel action, aggregate interlock, etc., should be considered. At the present time only the dowel action is considered as a major parameter. Although, the combined model described in Section 3.4 can be employed to investigate the interaction of bond and dowel effects if required.

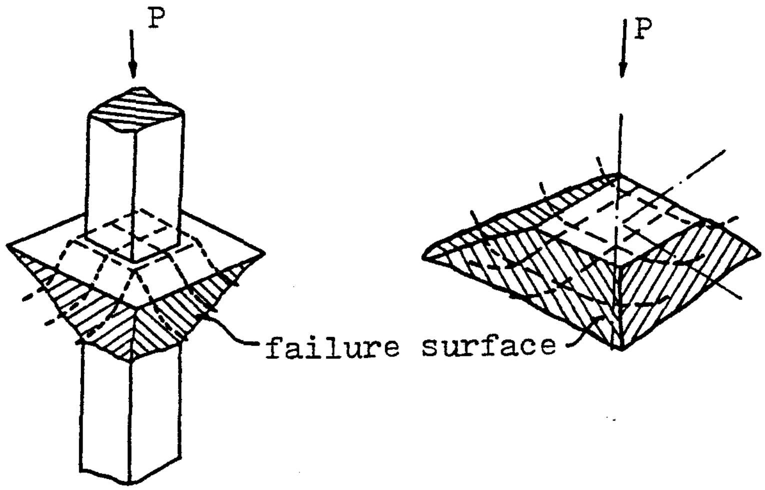
The effect of the dowel action alone on the response of the concrete slabs is now considered. This is accomplished by setting a very large value for the bond stiffness k_d such that a perfect bond exists at the interface. The value of k_d is then reduced to zero to show the sensitivity of the predicted response.

A post-tensioned concrete deck tested recently in the Applied Dynamics Laboratory at McMaster University has been selected as a numerical example for the finite element modelling. A summary of the geometry, the material properties, the boundary and the loading conditions, and the prestressing forces in tendons are given in Figures 5.5 and 5.6. The detailed information about the finite element analysis such as the grid layout, the treatment of boundary conditions etc., is described in Chapter Five.

The results from the finite element analysis for the two different values of k_d and experimental load deflection curve are shown in Figure 3.3. From the analytical results obtained, the following two conclusions can be drawn.

(1) The dowel stiffness k_d has a significant effect on the response of the post-tensioned concrete decks subjected to a point load. In this example, the lower value of k_d tends to give a better response when compared with the test data. There is an indication that the dowel stiffness decreases quickly under high tension in the steel. This confirms the observation by Jimenez-Perez et al. [13] during their dowel tests.

(2) From Chapter 2 it was shown that the bond stiffness k_b did not affect the response of the reinforced concrete slabs. Therefore, it can be concluded that for the post-tensioned concrete slabs again varying k_b would not effect the response for different values of k_d as shown in Figure 3.4.



(a) COlumn

(b) Slab

FIG. 3.1 PUNCHING SHEAR FAILURES

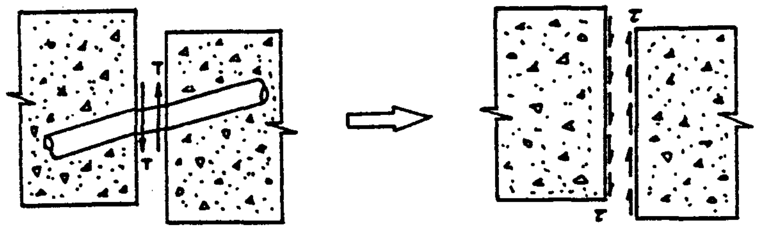


FIG. 3.2 SHEAR FORCE TRANSFERED BY DOWEL ACTION

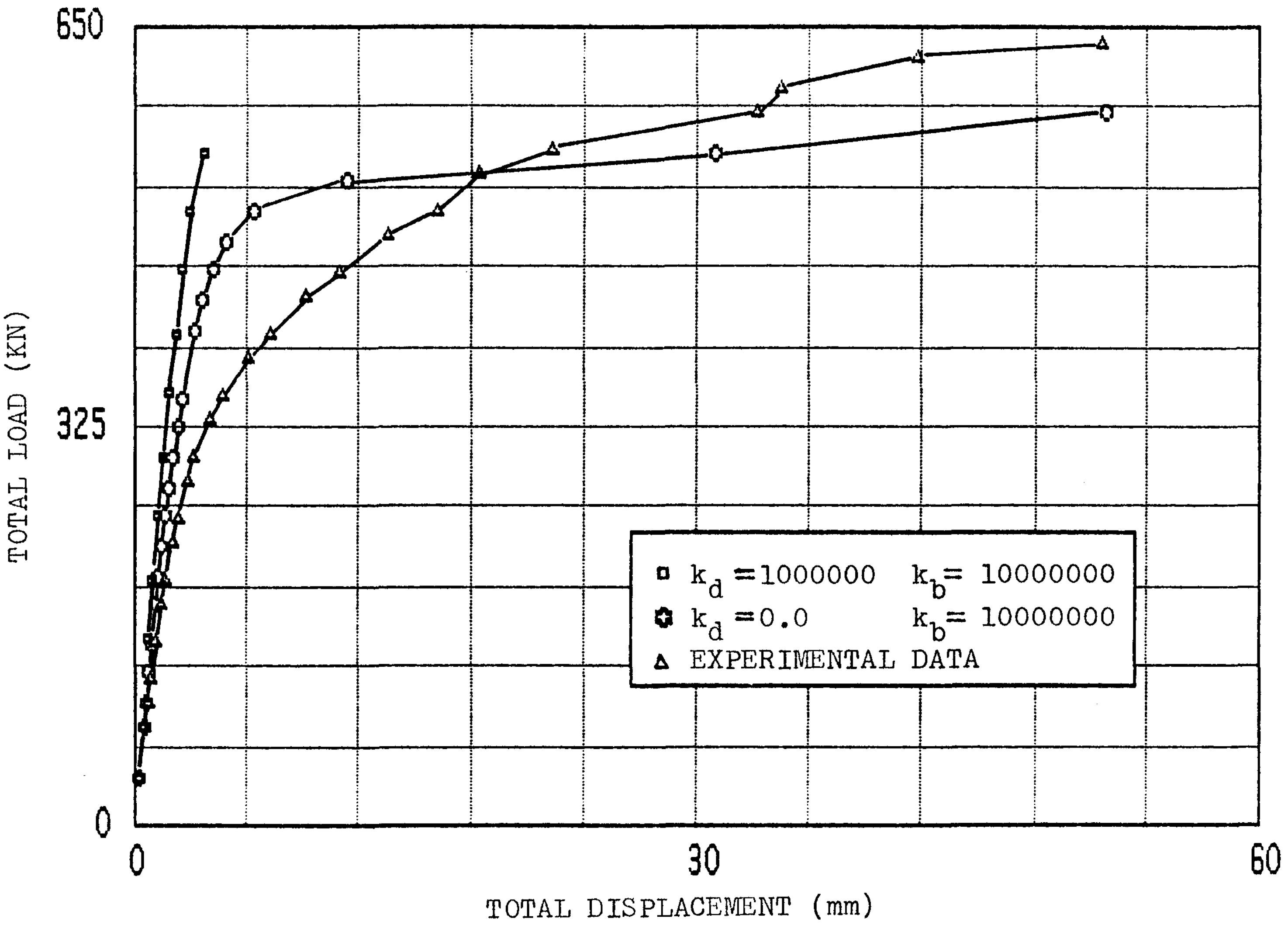


FIG. 3.3 EFFECT OF DOWEL STIFFNESS k_d ON RESPONSE OF POST-TENSIONED CONCRETE SLAB

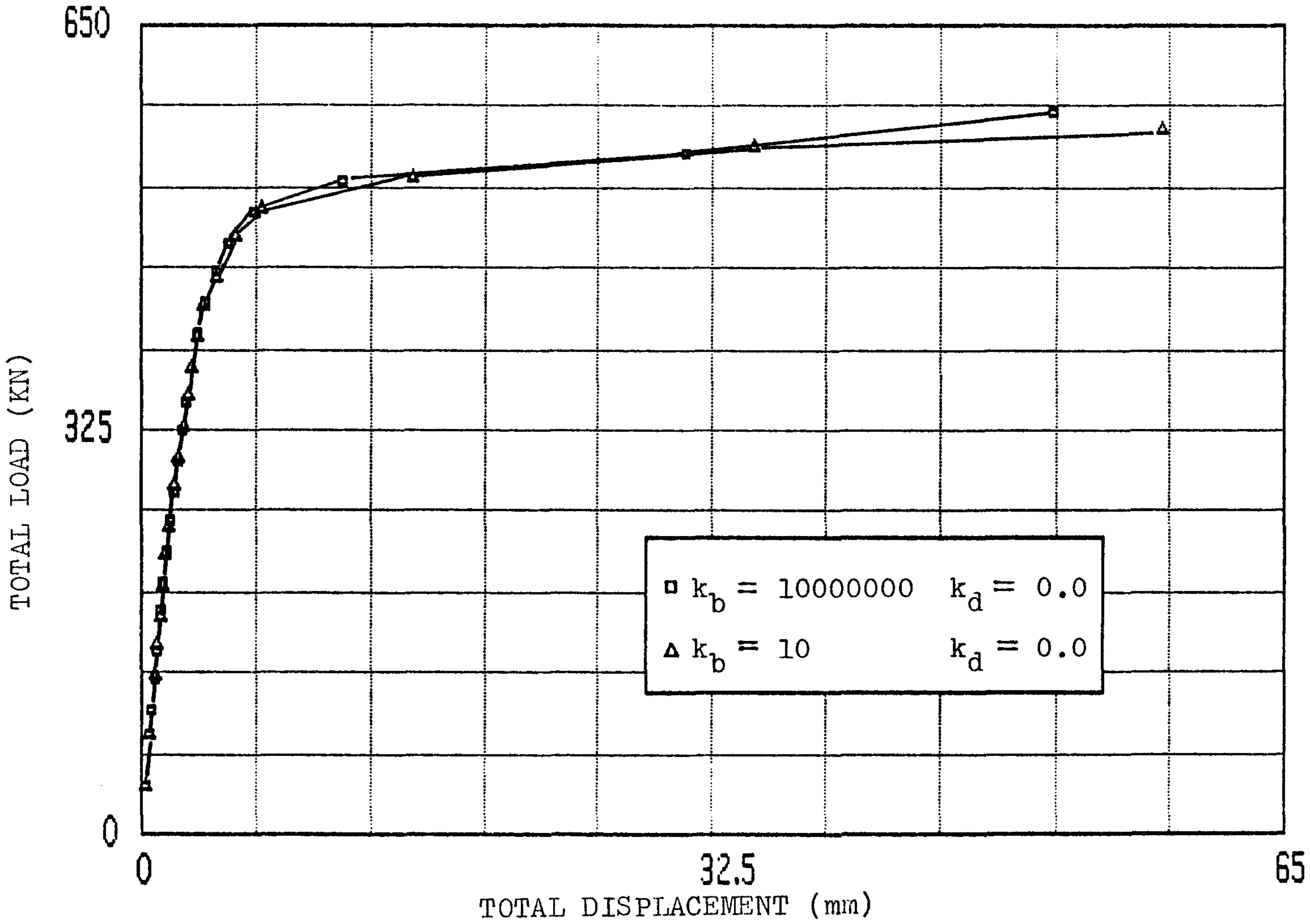


FIG. 3.4 EFFECT OF BOND STIFFNESS k_b ON RESPONSE OF POST-TENSIONED CONCRETE SLAB

CHAPTER 4

MODELLING OF TENSION STIFFENING EFFECT

4.1 Introduction

It was mentioned in Section 2.2 that the concrete in a cracked region does not lose all of its resisting capacity in tension and can still carry some tensile stress. This is the so-called tension stiffening effect. The tension stiffening effect can become significant in the post-cracking behavior and depends mainly on the size and placement of the reinforcing bars. The paper by Taylor et al. [12] contains details.

At the present time, there are two general approaches followed for modelling of the tension stiffening effect. The first approach, introduced by Scanlon [16], is to model the tension stiffening effect by considering a descending branch for the tension-tension stress-strain curve for concrete as shown in Figure 4.1. This process can be represented by a step reduction in the tensile stress, Lin and Scordelis [17], for gradual unloading indicated in Figure 4.2. The second approach is to increase the steel stiffness through modification of the stress-strain curve for steel as illustrated in Figure 4.3. The total internal tensile force carried by the concrete in the cracked region is represented by the additional stress in the steel. [See Gilbert and Warner [18] for details.] Besides these two methods, other approaches for the tension stiffening effect have also been reported in the literature [19, 20] and are not considered or used in this study.

It should be noted that all these approaches for modelling of the tension stiffening effect are based on the modification of the stress-strain diagram for the concrete or steel. However, no real justification has been provided for the use of some artificial criteria such as the ultimate tensile strain of concrete or the increased tangent modulii of the steel. This is because the actual behavior of the concrete or steel during crack propagation is a very difficult phenomenon to explain through experiments. In most of the previous modelling techniques for tension stiffening, these criteria have been based on curve fitting of the experimental results [21]. Through this procedure the analytical model obtained is not a rational one and is only valid for particular cases. Hence a search for a rational model, applicable to modelling of the tension stiffening effect in general, is essential and necessary.

In this chapter, a model based on the fracture mechanics theory to model the tension stiffening effect is presented. The emphasis is placed on establishing as rational a model as possible to avoid drawbacks encountered in the previous studies. This model is then included in the computer program and applied to some numerical examples.

4.2 Energy Variations in Elastic Crack Problems

Before introducing the fracture mechanics model for the tension stiffening effect, it is helpful to outline the theory of linear elastic fracture mechanics regarding crack extensions. [See Rice [22,23] for details of the theoretical background.]

For the purpose of this study two identical bodies of linear elastic material, each containing a notch or void and subjected to the same surface tractions T_i^0 on the boundary are considered S_T. If u_i^0, ε_{ij}^0, and σ_{ij}^0 denote the displacements, strains, and stresses in the initial state (Figure 4.4a), while u_i, ε_{ij}, and σ_{ij} denote the new values of displacements, strains, and stresses in the deformed state (Figure 4.4b), in which the notch increases by ΔV in volume with a new traction free surface ΔS.

The total potential energy corresponding to the initial state is given by

$$\pi^0 = \frac{1}{2} \int_V \sigma_{ij}^0 \, \varepsilon_{ij}^0 \, dV - \int_{S_T} T_i^0 \, u_i^0 \, ds. \qquad (4.1)$$

While the potential energy for the deformed state can be found from

$$\pi = \pi^0 + \Delta\pi$$

$$= \frac{1}{2} \int_{V-\Delta V} \sigma_{ij} \, \varepsilon_{ij} \, dV - \int_{S_T} T_i^0 \, u_i \, ds. \qquad (4.2)$$

Thus, the change in the potential energy is given by

$$-\Delta\pi = \frac{1}{2}\int_{\Delta V} \sigma_{ij}^{0}\ \varepsilon_{ij}^{0}\ dV - \frac{1}{2}\int_{V-\Delta V} (\sigma_{ij}\ \varepsilon_{ij} - \sigma_{ij}^{0}\ \varepsilon_{ij}^{0})\ dV$$

$$+ \int_{S_T} T_i^0\ (u_i - u_i^0\ )\ ds. \qquad (4.3)$$

For an elastic material and a crack extension along the plane of the crack, $\Delta\pi$ is path-independent and T_i^0 remains unchanged. If it is supposed that the body has passed through an intermediate state during a transition between the two states, at this intermediate state the volume, ΔV, of the material is removed from the body, but the surface tractions are applied to the newly formed surface, ΔS, in order to preserve the initial state of deformation within the remaining volume of the body. The second step is to release tractions T_i^0acting on ΔS, which then leads to the final state.

The equivalence of the internal virtual strain energy and the external virtual work within the volume V-ΔV during the second step of the transition therefore yields

$$\frac{1}{2}\int_{V-\Delta V} (\sigma_{ij}\ \varepsilon_{ij} - \sigma_{ij}^{0}\ \varepsilon_{ij}^{0})\ dV$$

$$= \int_{S_T} T_i^0\ (u_i - u_i^0)\ ds + \frac{1}{2}\int_{\Delta S} T_i^0\ (u_i - u_i^0)\ ds. \qquad (4.4)$$

Substituting Equation (4.4) into Equation (4.3) yields

$$-\Delta\pi = \frac{1}{2} \int_{\Delta V} \sigma_{ij}^{0} \; \varepsilon_{ij}^{0} \; dV - \frac{1}{2} \int_{\Delta S} T_i^{0} \; (u_i - u_i^{0}) \; ds. \qquad (4.5)$$

For $\Delta u_i = u_i - u_i^{0}$, Equation (4.5) can be rewritten as

$$- \Delta\pi = \frac{1}{2} \int_{\Delta V} \sigma_{ij}^{0} \; \varepsilon_{ij}^{0} \; dv - \frac{1}{2} \int_{\Delta S} T_i^{0} \; \Delta u_i \; ds. \qquad (4.6)$$

The calculation of $\Delta\pi$ in Equation (4.6) involves both the volume and the surface integrals and involves the quantities defined in the vicinity of the crack tip. An equivalent procedure of $\Delta\pi$ in Equation (4.6) is employed in the fracture mechanics approach to modelling the tension stiffening effect.

4.3 Fracture Mechanics Model

Concrete is a brittle and heterogeneous material. Under external loads, a concrete structure reveals its brittle fracture behavior in the earlier stages of loading and then the post-fracture behavior that dominates the response until a failure state has been reached. The behavior of concrete is very complex (with a nonlinear stress-strain relationship) because of the inherent heterogeneity and the randomness of microcracks. The effective behavior can be represented by an isotropic, elastic brittle material until macrocracks have occurred. In this light, the concrete may be treated as an elastic brittle material which is limited to the uncracked state in tension. Then, with the aid of the elastic brittle fracture mechanics described in the previous section one can establish an analytical model for the tension stiffening effect.

Figure 4.5 illustrates the tension stiffening effect for the response of reinforced concrete slabs. The curve OABC and the curve OADE denote the experimental response (actual) and an analytical response, respectively. If, assuming, the difference is mainly due to the tension stiffening effect, then the hatched area ABD at the load level P_n represents the incorrect loss of potential energy due to occurrence of cracks and their extensions. According to Griffith's theory [24], the potential energy decrease rate (per unit area) due to a crack advance should be equal to the energy expended in the newly created cracked surfaces and also a cohesive zone ahead of the cracks exists due to the cohesive theory [25]. It is known that the atomic or molecular attractions on the separating surfaces of crack should vanish with the crack propagation and the potential energy decrease rate should be exactly the same as the loss of potential energy due to concrete cracking for plain concrete slabs. However, for a reinforced concrete slab, the presence of the steel reinforcement will reduce the loss of potential energy due to cracking. Also, the tensile stress of concrete in a cracking zone should release gradually. From an energy dissipation point of view, an opposite tensile stress may be employed to a newly created surface of a crack in order to eliminate the restraining stress and to release it gradually so as to bring curve ADE close to curve ABC. This, indirectly, models the tension stiffening effect in the finite element analysis of reinforced concrete slabs and establishes the equivalence of $\Delta\pi$ in Equation (4.6).

In accordance with this concept, a fracture mechanics model for representing the tension stiffening effect has been accommodated in the computer program. The essential steps are summarized below.

(1) Evaluate principal stresses at each integration point for each incremental load and check whether the principal tensile stress is greater than the maximum tensile strength of the concrete.

(2) Apply an opposite of the tensile stress of the same magnitude as the existing stress to each newly cracked integration point in a direction coincident with the principal tensile stress immediately after the concrete cracks.

(3) Use this opposite tensile stress to calculate an equivalent load vector $\{\Delta P\}$ and maintain the structure stiffness matrix [K] as prior to cracking.

(4) Solve the system of equations

$$\{\Delta P\} = [K] \{\Delta u\}$$

to determine the change in displacements $\{\Delta u\}$.

(5) Deduct the displacements $\{\Delta u\}$ from the total displacements $\{u\}$ to obtain the modified displacements $\{u'\}$.

(6) Use the modified displacements $\{u'\}$ to recalculate stresses and strains

(7) Continue with the next iteration.

It should be noted that the elimination of the existing tensile principal stresses is accomplished only prior to the first iteration for each load increment. The change in potential energy $\Delta\pi$ in Equation

(4.6) is expected to be equivalent to $\{\Delta P\}^T\{\Delta u\}/2$ computed from the steps described above and expected to represent the shaded area ABD in Figure 4.5.

4.4 Numerical Study

In order to study the fracture mechanics model of the previous section for predicting the tension stiffening effect, a numerical example using the computer program discussed in Section 4.3 has been carried out. In the current investigation, the simply supported reinforced concrete slab tested by Taylor et al. [12] is taken as the test numerical example. The geometry and the material properties are given in Figure 5.1 and more information on the boundary and loading conditions are also presented in Chapter Five.

The numerical results obtained from the finite element fracture mechanics model are compared with both the corresponding results for the case of no tension stiffening and the experimental results, Figure 4.6. It is evident that the use of the fracture mechanics model for the tension stiffening effect for the reinforced concrete slab improves the response of the slab. Table 4.1 shows the principal stresses around the first cracking point before and after elimination of the restraining tensile stress. It can be observed that the proposed model enables a gradual release of the concrete tensile stress in a cracking zone. In Table 4.1, point A is the first cracking point; points B and C are the uncracked points which are located above and to the right of point A respectively, as shown in Figure 4.7.

The potential energy release rates for the plain concrete and the reinforced concrete plates under pure tension have also been attempted to find the amount of opposite tensile stress that ought to be applied. Because of the smeared cracking model, it is difficult to identify exact direction of the crack extension and evaluation of the crack width. As a result, the two rates and their differences are still unknown. At present, the opposite of the existing restraining stress on the crack surfaces has been used and yielded good results. It should be mentioned here that the use of the tension stiffening model presented also reduces the poor grid sensitivity as reported by Chow [10]. This is fairly well demonstrated in Figure 4.8 for a two by two grid and a four by four grid.

POINT	TYPE OF MODEL	AT CRACKING		AFTER CRACKING		CHANGES	
		σ_1	σ_2	σ_1	σ_2	$\Delta\sigma_1$	$\Delta\sigma_2$
A	$-\sigma_t^c$ IS APPLIED	474.00	524.22	474.10	0.00	0.10	-524.22
	NO $-\sigma_t^c$ IS APPL.	474.00	0.00	497.16	0.00	23.07	0.00
B	$-\sigma_t^c$ IS APPLIED	309.05	279.05	308.91	278.86	-0.15	-0.19
	NO $-\sigma_t^c$ IS APPL.	309.05	279.05	326.96	297.05	17.91	18.00
C	$-\sigma_t^c$ IS APPLIED	471.04	440.64	471.10	440.70	0.06	0.05
	NO $-\sigma_t^c$ IS APPL.	471.04	440.64	486.39	452.04	15.35	11.40

TABLE 4.1 STRESS DISTRIBUTION AROUND THE FIRST CRACKED POINT (UNIT:psi)

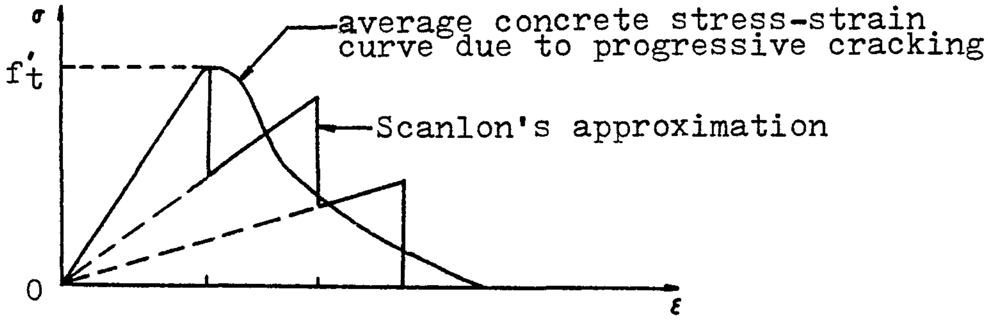


FIG. 4.1 SCANLON'S STEPPED MODEL

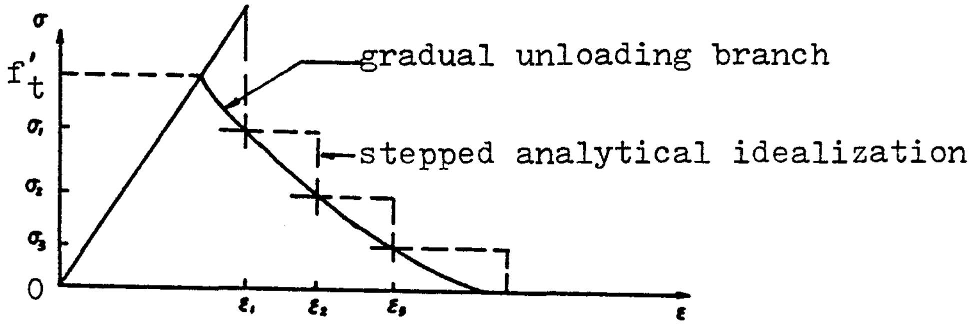


FIG. 4.2 LIN'S GRADUALLY UNLOADING MODEL

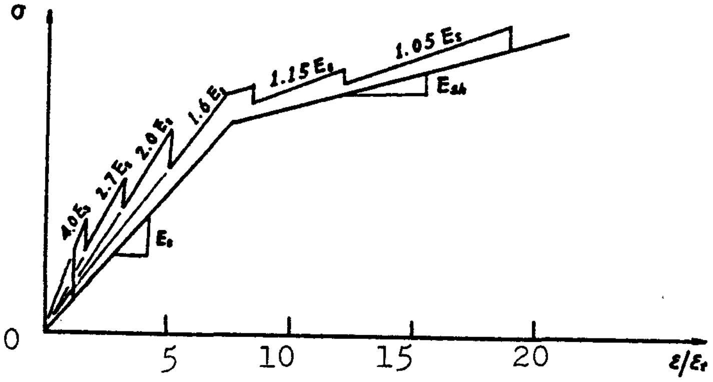


FIG. 4.3 MODIFIED STRESS-STRAIN DIAGRAM FOR REINFORCING STEEL

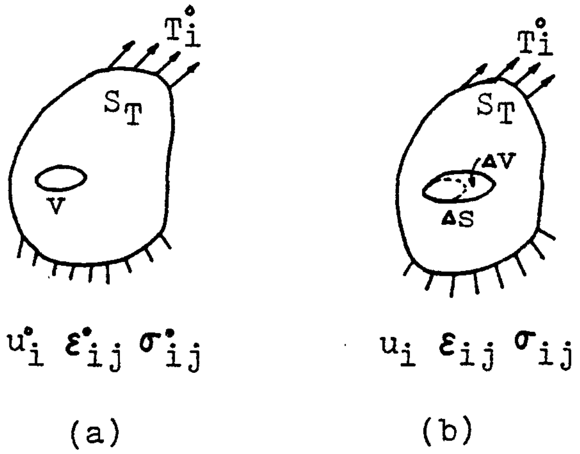

T_i°
S_T
V
u_i° ε_{ij}° σ_{ij}°
(a)
T_i°
S_T
ΔV
ΔS
u_i ε_{ij} σ_{ij}
(b)

FIG. 4.4 (a) NOTCH IN ELASTIC BODY; (b) ADVANCE

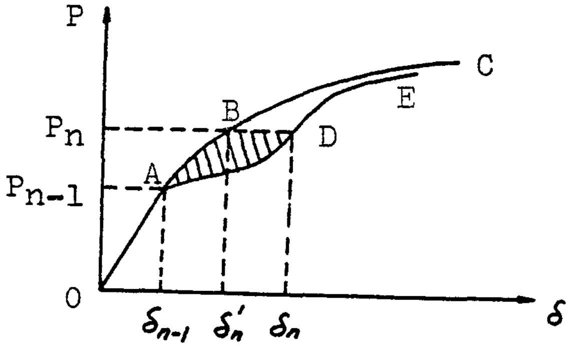


FIG. 4.5 P-δ CURVE FOR REINFORCED CONCRETE SLAB

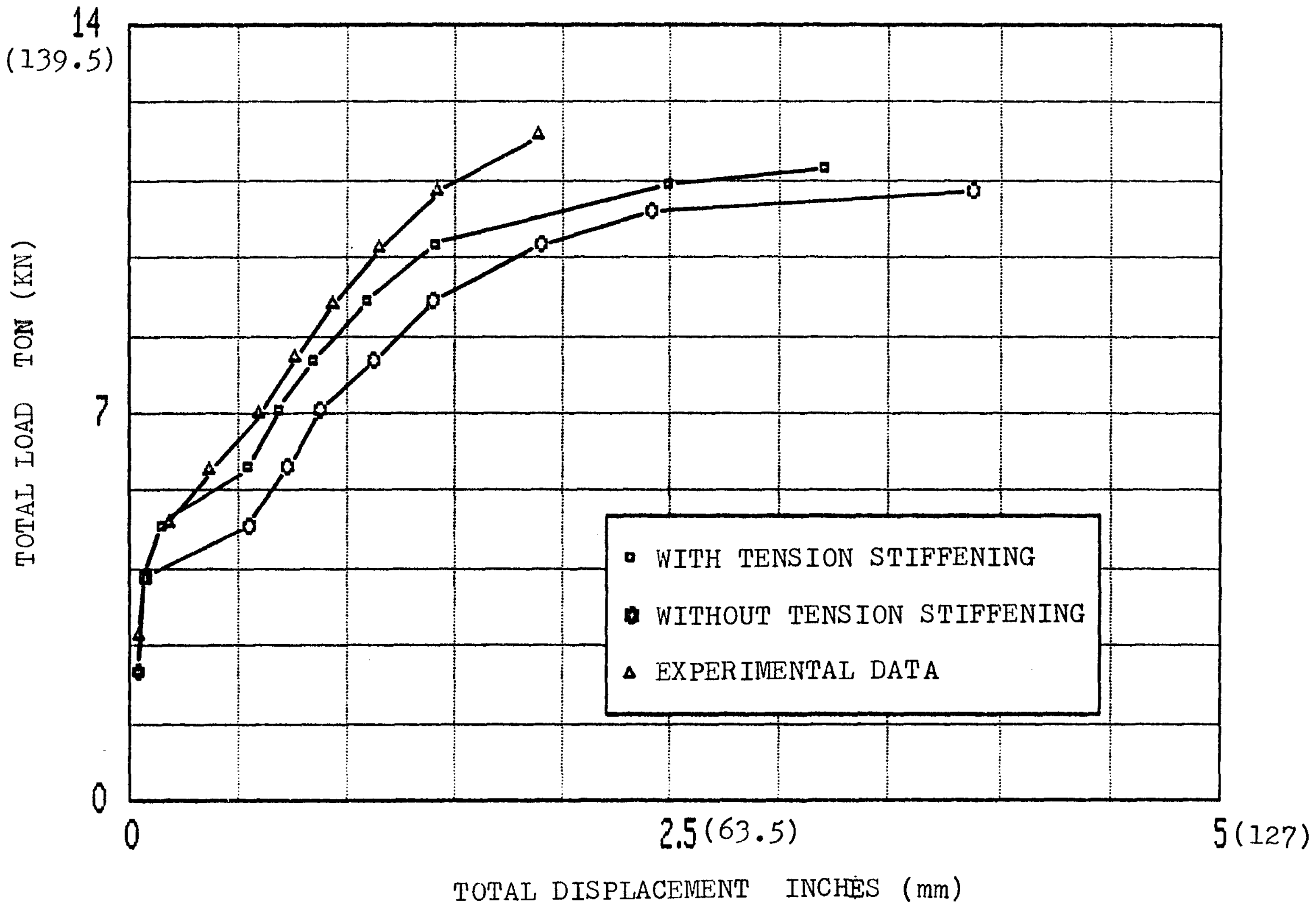
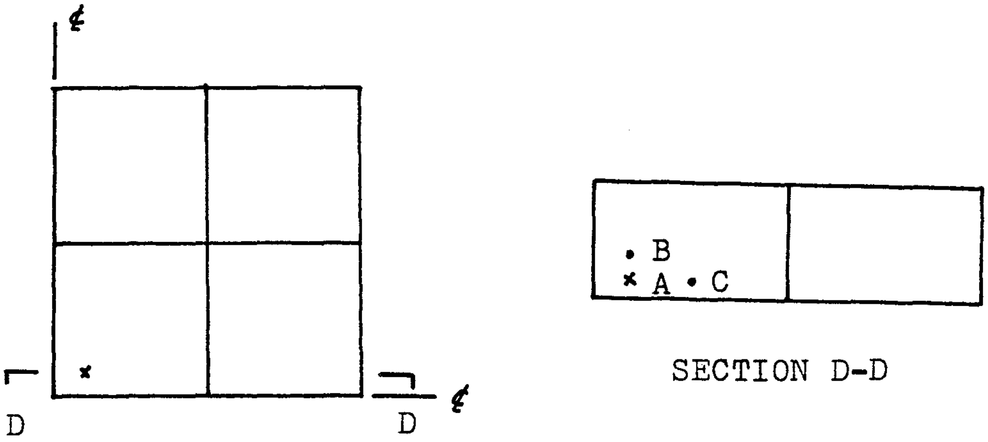


FIG. 4.6 TENSION STIFFENING EFFECT ON RESPONSE OF REINFORCED CONCRETE SLAB (TWO BY TWO GRID)



SECTION D-D

FIG. 4.7 POSITION OF THE FIRST CRACKING POINT

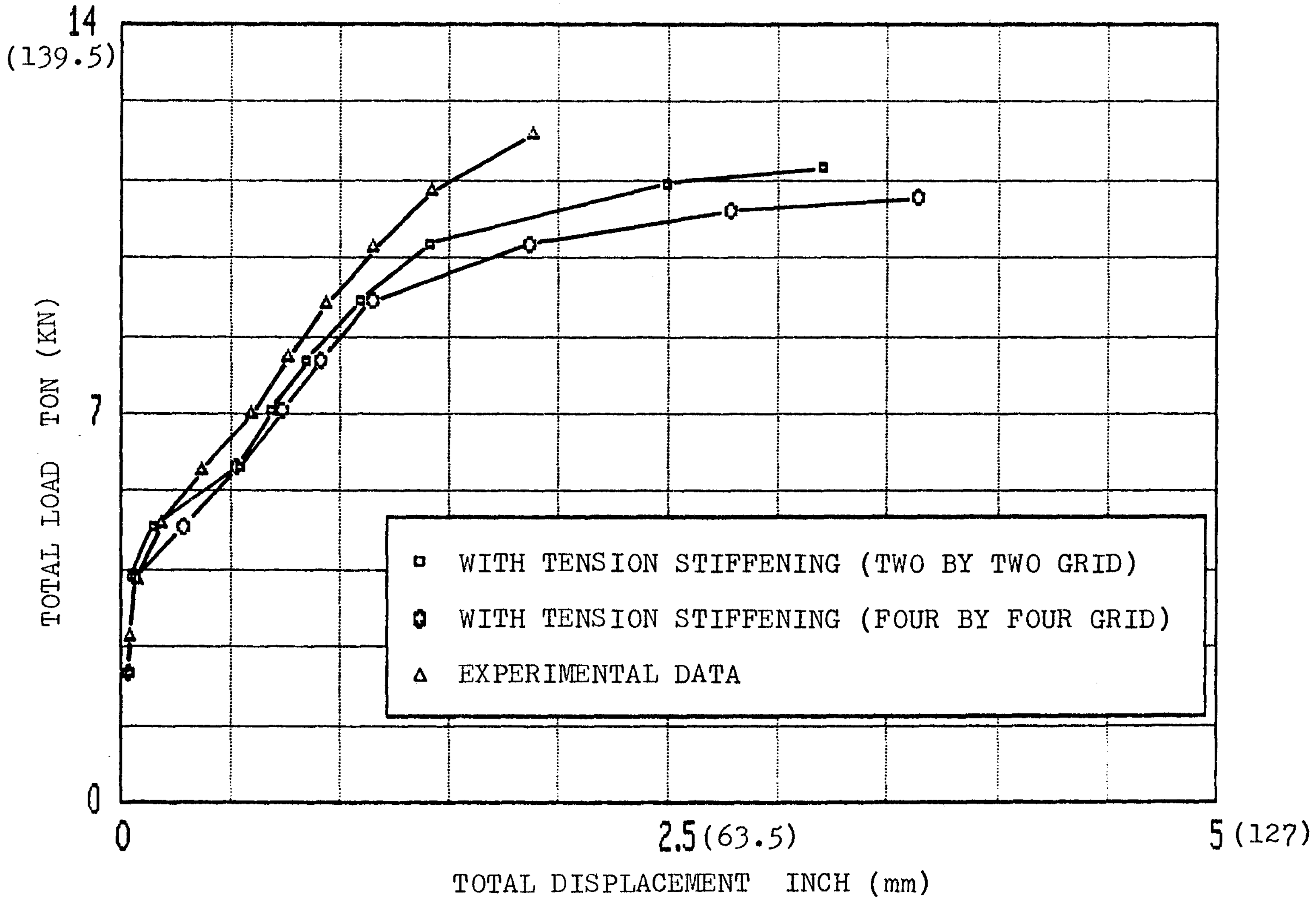


FIG. 4.8 EFFECT OF F.E. GRID SIZE ON RESPONSE OF REINFORCED CONCRETE SLAB

CHAPTER 5

NUMERICAL EXAMPLES AND COMPARISON

5.1 Introduction

In order to verify the finite element model for the reinforced concrete slabs presented in the previous chapters, the following two numerical examples have been simulated.

(a) A simply supported normally reinforced concrete slab tested by Taylor et al. [12].

(b) A post-tensioned concrete bridge deck recently tested by Moll [28] in the Applied Dynamics Laboratory at McMaster University.

The numerical results obtained from the finite element model developed are then compared with the experimental results reported.

It is pointed out here that the computer program used for numerical modelling is an extension of the program using the incremental load, nonlinear analysis of concrete slabs originally developed by Chow [10]. Extensive modifications have been made to incorporate the bond-slip, dowel action and the tension stiffening effect. Also note that the computer programs are coded in FORTRAN V language and run on the VAX 11-785 computer in the Faculty of Engineering at McMaster University.

5.2 Simply Supported Reinforced Concrete Slab

The geometric details and the material properties of the reinforced concrete slab tested by Taylor et al. [12] are shown in Figure 5.1. Because of the double symmetry of the slab and the applied loading, only one quarter of the slab (the finite element meshes shown in Figure 5.2) is analyzed using the proposed model. For the square slab, simply supported on all four edges and subjected to uniformly distributed load, no punching shear failure was observed by Taylor et al. This implies that contribution of the dowel action to the load carrying capacity of the slab is negligible. Hence only the bond-slip and the tension stiffening effect were incorporated.

Figure 5.3 shows the comparison of the load-deflection response obtained by the present model, using a two by two mesh, with the experimental curve and the numerical results obtained by Chow [10]. It can be seen that when the applied load is below the load of initial cracking, very good agreement with the test data is obtained. Beyond this load, the analytical curves from the present model and Chow's solution indicate greater flexibility. However, the numerical results are substantially improved when the tension stiffening effect is taken into account and the present model gives more than 50% reduction of the relative difference between the two curves (Chow's curve and the test curve). This is due to properly accounting for the change in potential energy of Equation (4.6).

The observed deviation in the response from the present model is probably caused by neglecting the material non-linearities and the possible errors associated with the idealization of the supports as a simply supported plate may have problems in its enforcement near the corners. It is well recognized that the stress-strain relationship for concrete is nonlinear and steel can be represented by a bilinear curve. It is also known that in a compression-compression region there is about a 20% increase in the strength of concrete which has been neglected in the assumed linear elastic behaviour for concrete and can tend to yield a stiffer response for the overall structure. It is also evident from Figure 5.3 that when the external load, which was applied in increments, was close to the ultimate stages the load-displacement curve by Chow becomes stiffer. The larger stiffness was mainly due to ignoring yielding of steel reinforcement near the ultimate stages. It should be noted that in the previous program developed by Chow [10], the stress of steel was not evaluated for each load increment and was roughly checked through hand calculations afterwards. The present program calculated the stress in steel during each iteration. It was found that the first steel bar yielded at a load of about 6.02 Tons. Most of the steel bars had yielded near the ultimate loading stage.

A more refined grid (4×4 for quarter of the plate) was then employed to further check the grid sensitivity and objectivity. This was not obtainable in the previous study reported by Chow [10]. The load-deflection curves (using the four by four grid) for both, with tension stiffening and without tension stiffening are presented in Figure 5.4. Again, the trend was similar to what was observed for the two by two grid. The improvement is also similar to the results presented in Figure 4.8 in the previous chapter. However, once again the results without the tension stiffening tend to be overly stiff beyond a displacement of about 3 mm.

5.3 Post-tensioned Concrete Bridge Deck

The post-tensioned concrete bridge deck tested by Moll [28] was also modelled numerically and the response compared with the experimental results. The geometry and the cross sectional properties of the deck are shown in Figure 5.5. The finite element idealization used to model the deck appears in Figure 5.6. Due to symmetry of the deck and the applied load, only one half of the deck was analyzed.

The prestressing tendons of the deck, consisting of seven wire (NO. 13) strands (with a nominal tensile yield stress of 1860 MPa and the elastic modulus of 190,000 MPa) were placed in the mid-surface of the deck along the shorter span direction and were stressed to 80% of the yield stress causing a prestressing force of about 138 KN. The non-prestressed reinforcement (with 400 MPa yield stress) was placed underneath the prestressing tendons along the longer span direction. The material properties of the non-prestressed steel bars and the concrete are given in Table 5.1.

In the test setup, the deck was supported underneath by the girders and tie bars along the longitudinal directions. The tie bars, with diameter of 2.0 inches, were placed vertically outside the girders in an attempt to model certain continuity that might prevail in the actual structure, (see Figure 5.5). In the finite element idealization, the tie bars were modelled by equivalent springs and the girders as simple supports. Figure 5.6 shows the details.

Figure 5.7 shows the load-displacement response of the concrete deck (displacement at the loading point) obtained from the present model, Chow's model and the test results. The ultimate load obtained from the present model (576KN) was found to be closer to the experimental value (637KN) than that obtained from the previous model (464KN) by Chow [10]. The cracking pattern around the loading point at P=512KN indicated a punching shear failure which was similar to the failure mode observed during the experiments, and shown in Figures 5.8 and 5.9. In these figures, the cracking at the integration points has been indicated by the numbers 1 to 5. The numbering starts from the bottom-most layer indicated as 1 to the top-most layer indicated as 5.

The present analytical model was then used to study the effect of full and partial (half) prestressing while maintaining the full concrete strength. In order to study the effect of reduced tensile concrete strength, the concrete deck was also analysed with full prestressing and half the tensile concrete strength. All other parameters were kept at the values as reported in Table 5.1 and the geometric layout as was indicated in Figure 5.5. The load and displacement values for the three cases (fully prestressed, half prestressed and fully prestressed with tensile strength of concrete reduced to half) are tabulated in Table 5.2 and load-deflections plots are shown in Figures 5.10 and 5.11.

It can be observed that the predicted load-displacement response is stiffer than the experimental results in the range of serviceability loads. This is probably caused by neglecting loss of prestress in the prestressing tendons due to shortening of the concrete around the tendons, relaxation of the stress within the tendons and external factors which reduce the prestressing forces in the tendons. In fact, it was found that the load-displacement behaviour of the deck is very sensitive to the initial prestressing force, as can be observed from Figure 5.10.

The effect of tensile strength of concrete on the predicted load-displacement response was also studied. Figure 5.11 shows the sensitivity of the response of the deck to the concrete tensile strength. It is recognized that the magnitude of the prestressing force and the initial cracking load depend on the tensile strength of concrete. This, of course, will effect the predicted load-displacement response and probably explains the observed predicted stiffer behavior in the servicability range.

In summary, it is clear that the tension stiffening effect is a very important factor, more important than the bond slip and the dowel action, when predicting response of a normally reinforced concrete slab.

While for the post-tensioned concrete deck subjected to a concentrated load, the overall response has been greatly improved through incorporation of the dowel action into the present model and is very sensitive to the varying of the prestressing force in the tendons and the tensile concrete strength.

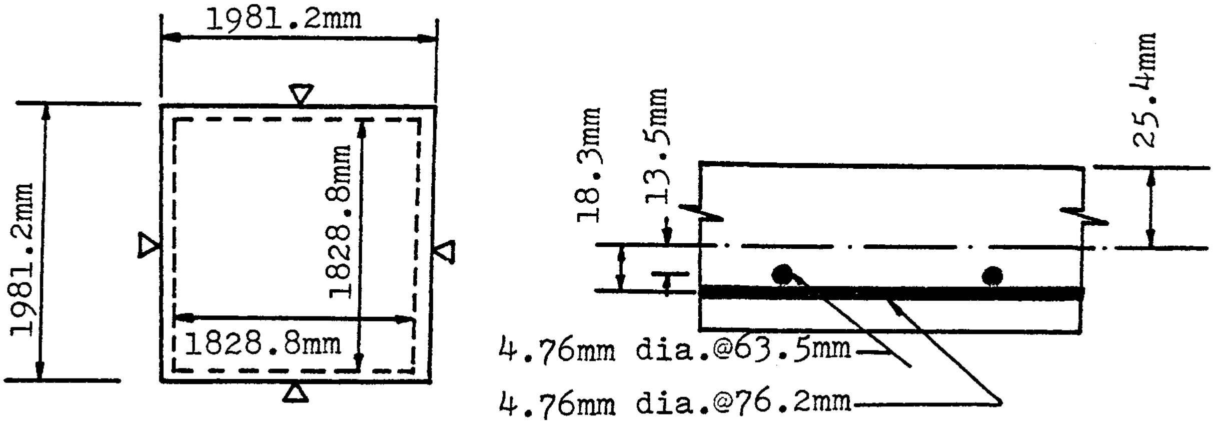
MATERIAL PROPERTIES

Concrete properties		Steel properties	
Compressive strength	43.8 MPa	Elastic modulus	2x10^5 MPa
Tensile strength	5.07 MPa	Tangent modulus	5x10^3 MPa
Initial modulus	33380.0 MPa	Yield stress	4x10^2 MPa
Poisson's ratio	0.2	Eccentricity of x	12.0 mm
Shear retention factor	0.0	Eccentricity of y	0.0 mm
Strain at peak stress	-0.003	Diameter	11.3 mm
Thickness of deck	175.0 mm	Area	100.0 mm^2

TABLE 5.1 MATERIAL PROPERTIES OF POST-TENSIONED CONCRETE DECK (STEEL IS NON-PRESTRESSED BAR)

LOAD		DISPLACEMENT AT LOADING POINT (mm)		
LOAD INCREMENT	TOTAL LOAD (KN)	FULLY PRESTR. FORCE	HALF PRESTR. FORCE	HALF TENSILE STREN.
1	40	0.1988	0.1988	0.1988
2	80	0.4473	0.4473	0.4445
3	120	0.6919	0.7036	0.7088
7	152	0.8943	0.9031	0.9307
10	176	1.028	1.053	1.109
13	200	1.189	1.229	1.323
16	224	1.346	1.420	1.535
19	248	1.536	1.649	1.796
22	272	1.771	1.908	2.025
26	304	2.050	2.381	2.472
29	328	2.325	2.832	2.946
32	352	2.554	3.423	3.608
35	376	2.837	4.911	6.927
38	400	3.139	8.104	17.28
41	424	3.467	16.02	33.32
44	448	4.159	33.87	43.13
47	472	4.914	56.06	66.39
51	504	6.854	-	-
54	528	15.30	-	-
57	552	34.87	-	-
60	576	69.42	-	-

TABLE 5.2 RESULTS OF POST-TENSIONED CONCRETE DECK FOR DIFFERENT PRESTRESSED FORCE AND TENSILE STRENGTH



(a) Plan of Slab

(b) Cross Section

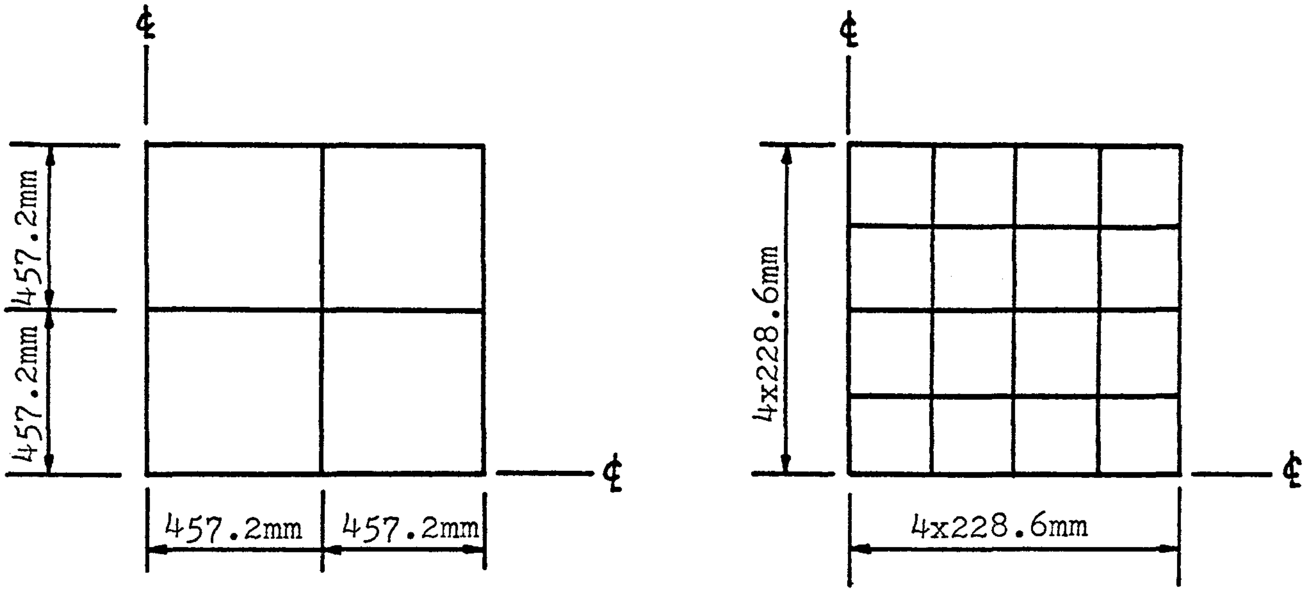
Concrete properties

Compressive strength	34.5 MPa
Tensile strength	3.6 MPa
Initial modulus	3×10^4 MPa
Poisson's ratio	0.2
Shear retention factor	0.0
Strain at peak stress	-0.003
Thickness of slab	50.8 mm

Steel Properties

Steel modulus	2.06×10^5MPa
Yield strength	372 MPa
Steel eccentricity in	
X direction	18.3 mm
Y direction	13.5 mm
Diameter	4.76 mm
Area	17.8 mm^2

FIG. 5.1 GEOMETRY AND MATERIAL PROPERTIES FOR SIMPLY SUPPORTED SLAB BY TAYLOR et al.



(a) 2x2 Grid

(b) 4x4 Grid

FIG. 5.2 GRID LAYOUT FOR SIMPLY SUPPORTED SLAB

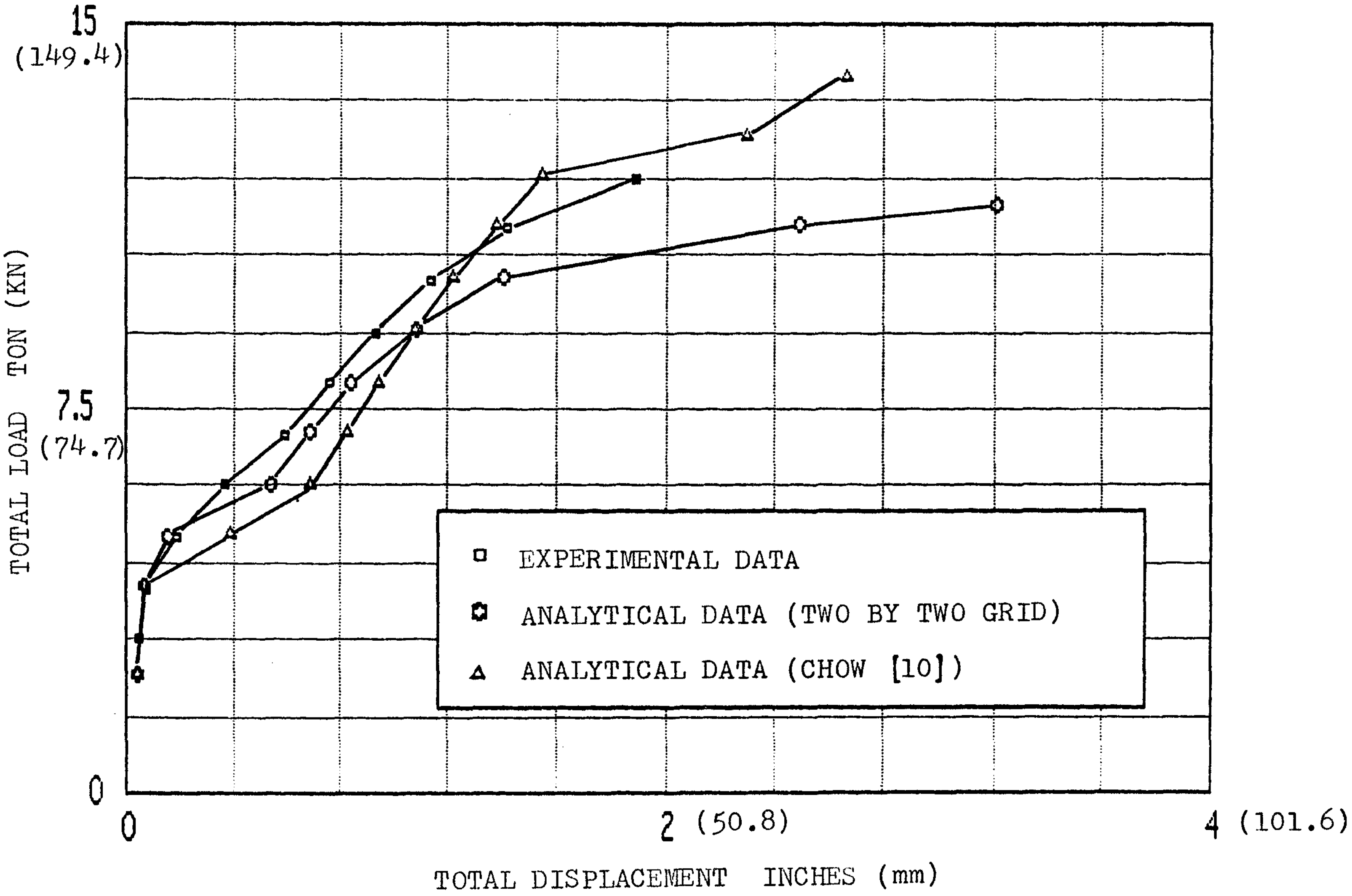


FIG. 5.3 TOTAL LOAD-DISPLACEMENT DIAGRAM FOR 2x2 GRID LAYOUT

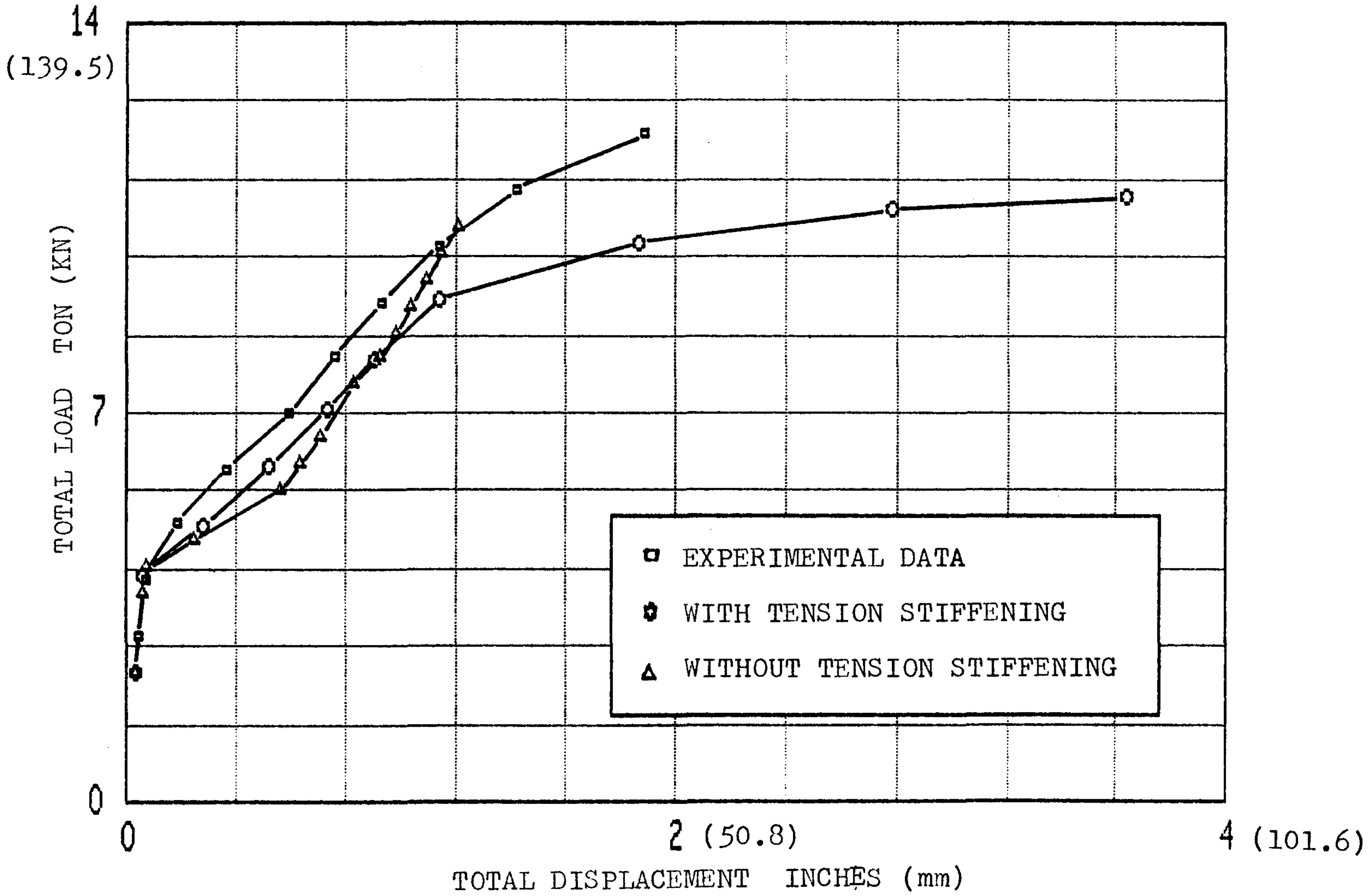
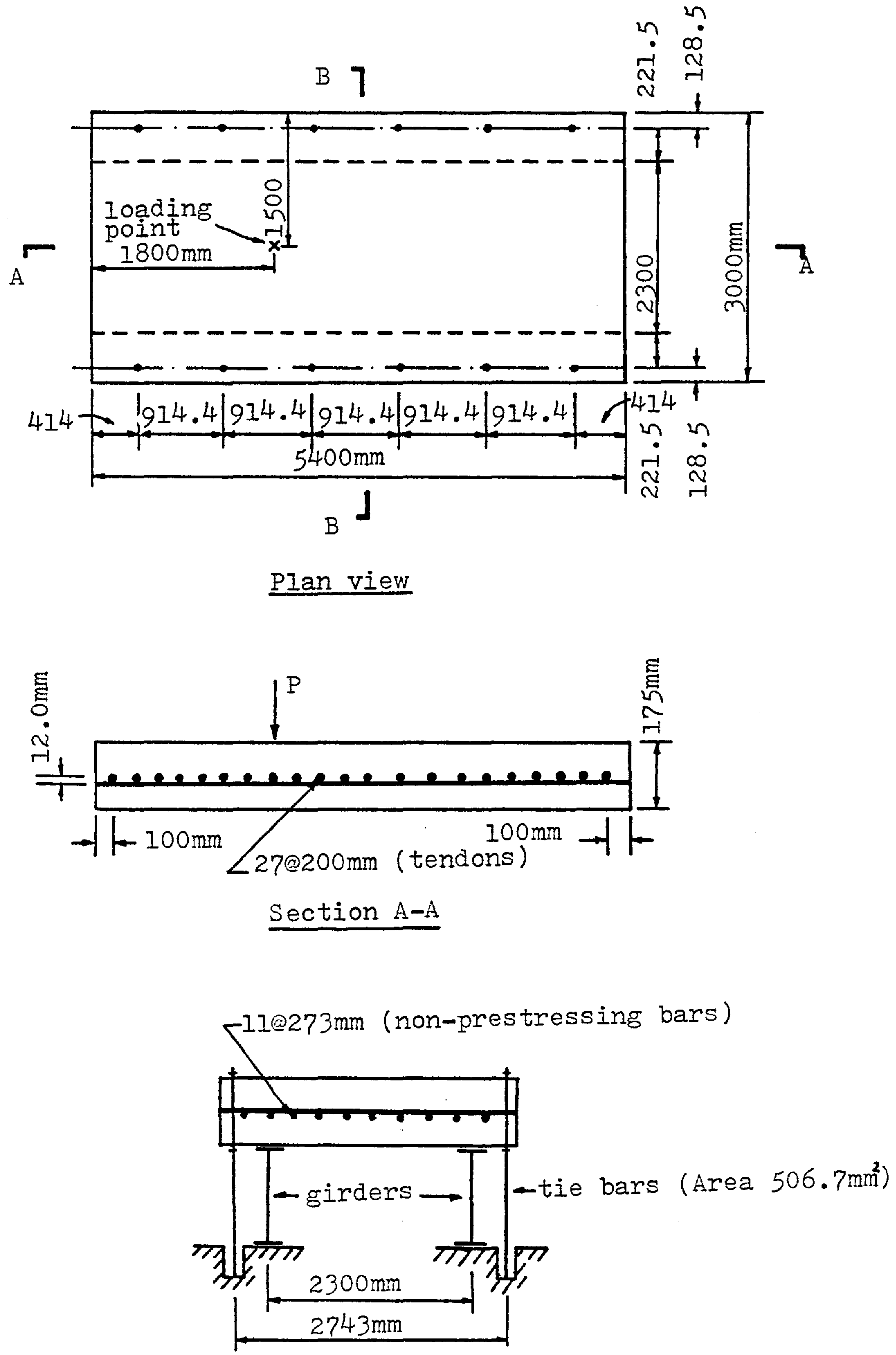


FIG. 5.4 TOTAL LOAD-DISPLACEMENT DIAGRAM FOR 4x4 GRID LAYOUT

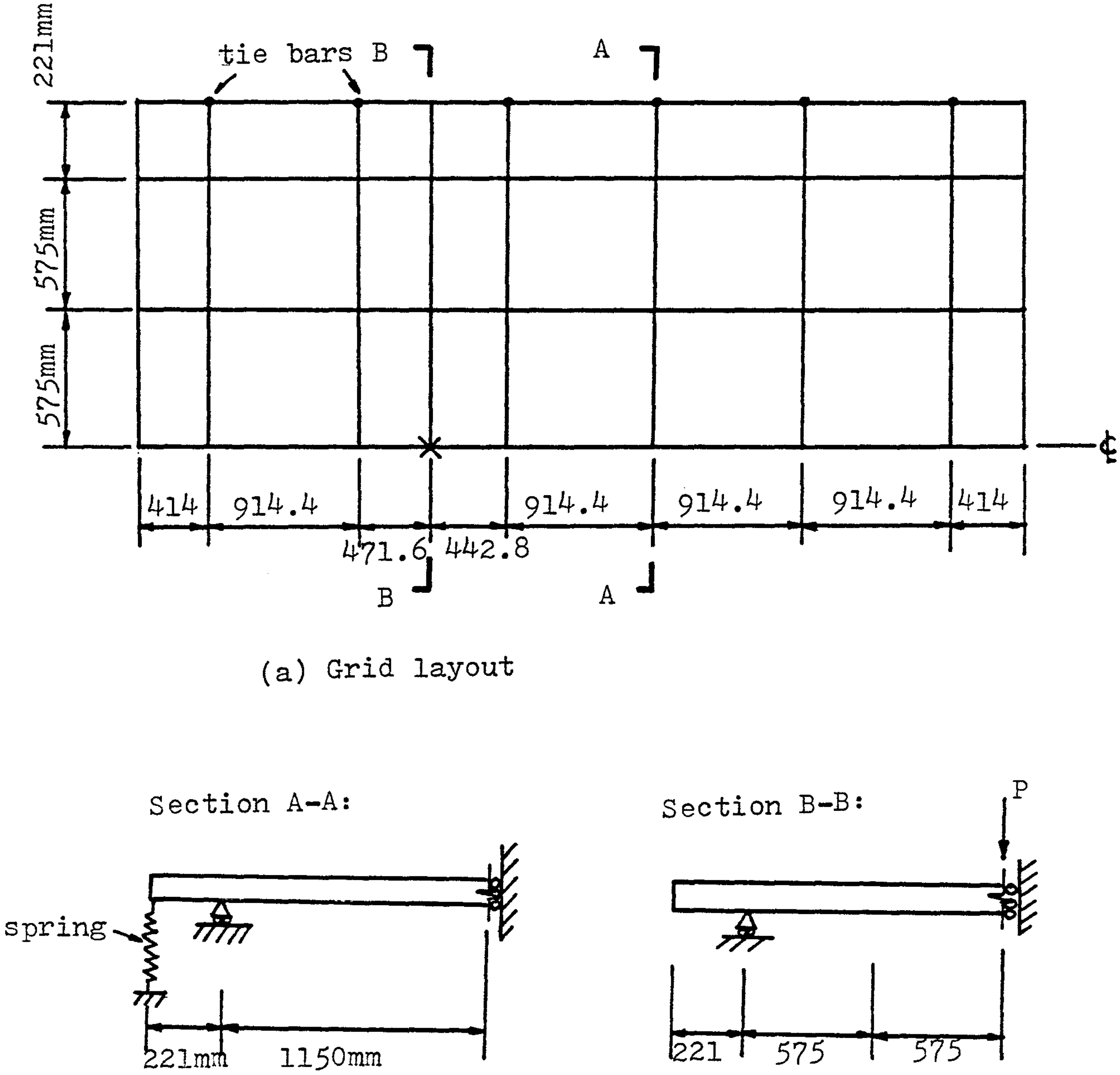


Plan view

Section A-A

Section B-B

FIG. 5.5 GEOMETRY AND CROSS SECTIONAL PROPERTIES FOR POST-TENSIONED CONCRETE DECK



(a) Grid layout

(b) Boundary conditions

FIG. 5.6 GRID LAYOUT AND BOUNDARY CONDITIONS FOR POST-TENSIONED CONCRETE DECK

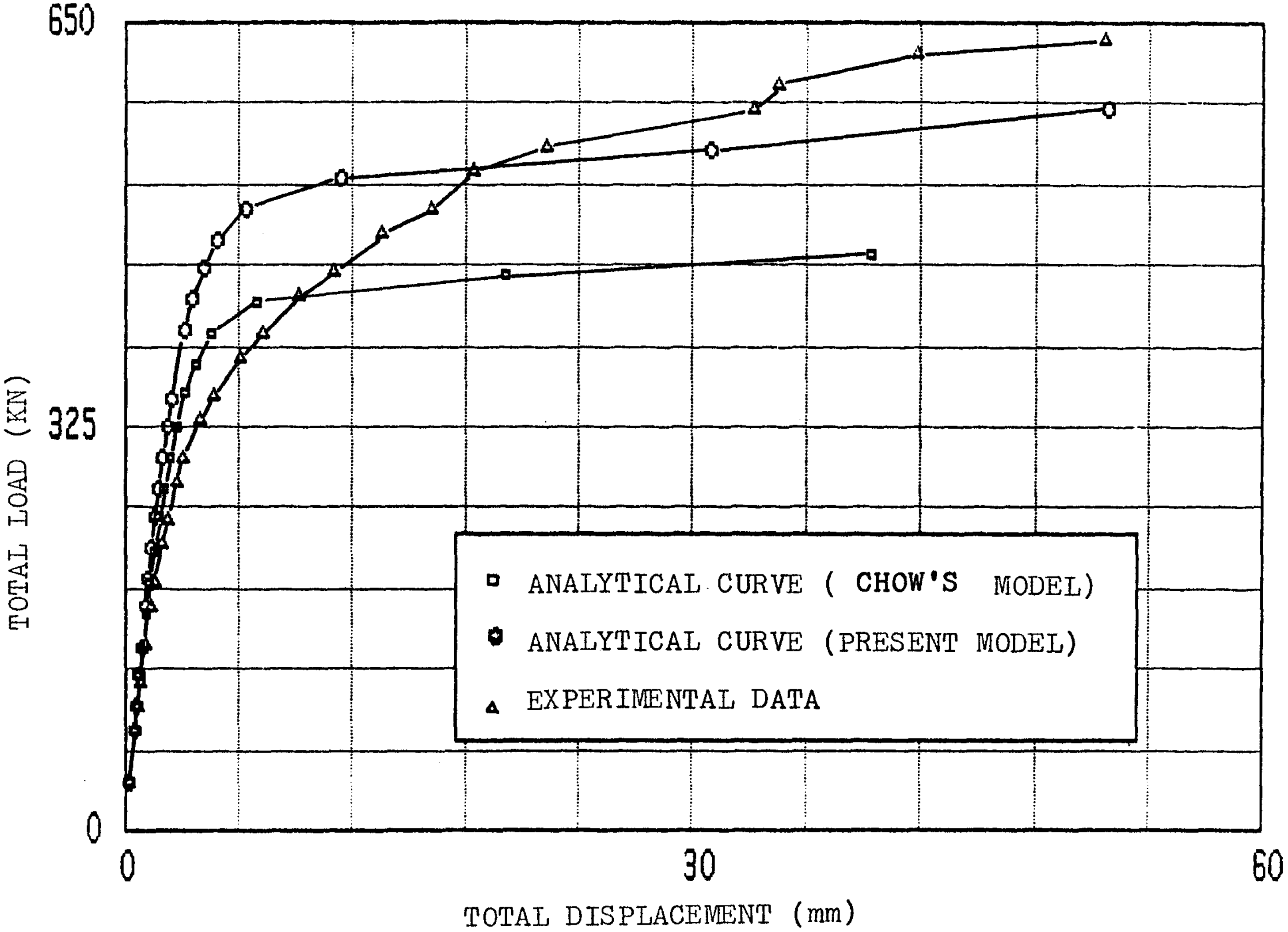
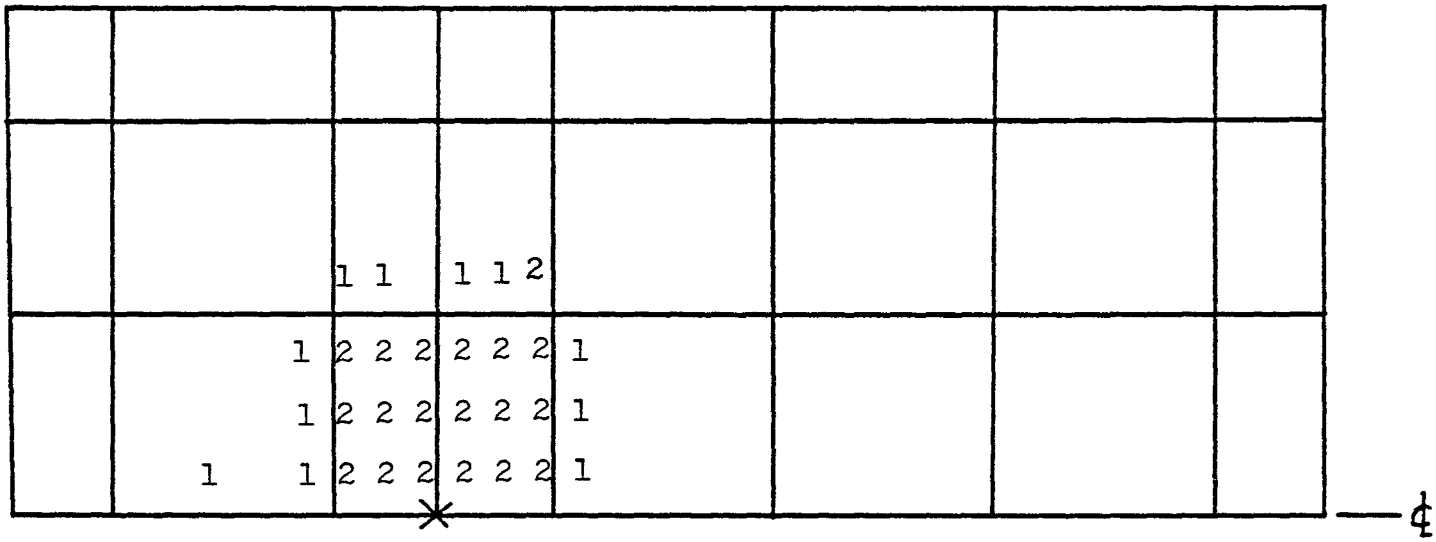


FIG. 5.7 TOTAL LOAD-DISPLACEMENT DIAGRAM AT THE LOADING POINT FOR POST-TENSIONED CONCRETE DECK

INCR=4 P=128KN NO CRACKS

INCR=30 P=336KN



INCR=45 P=456KN

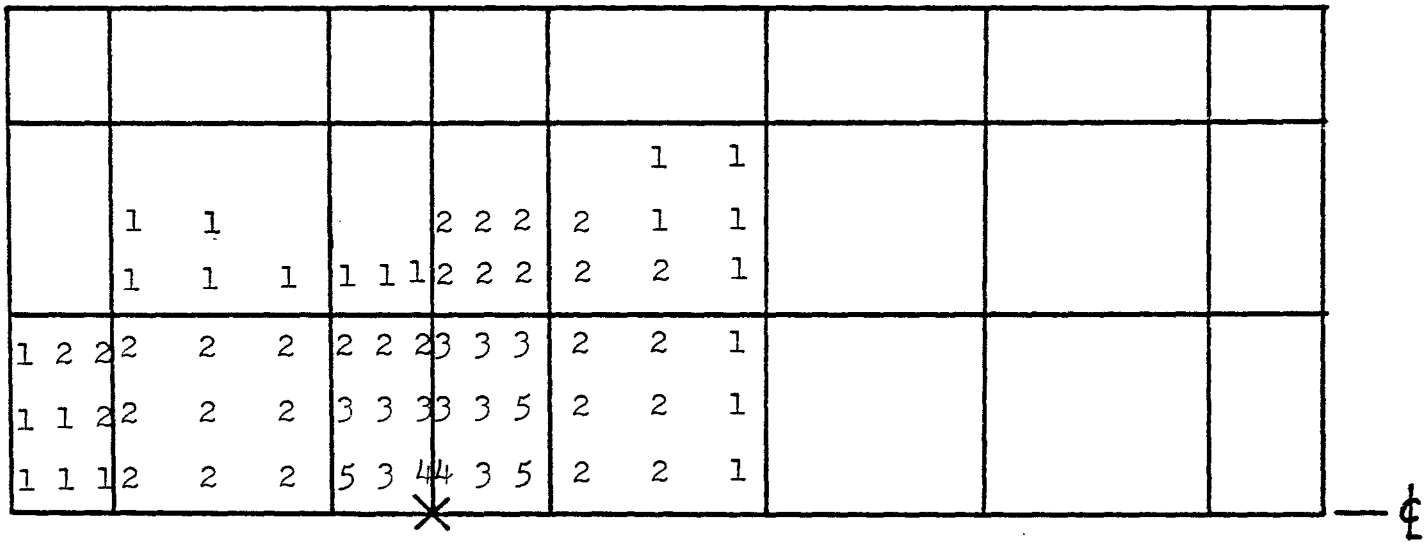
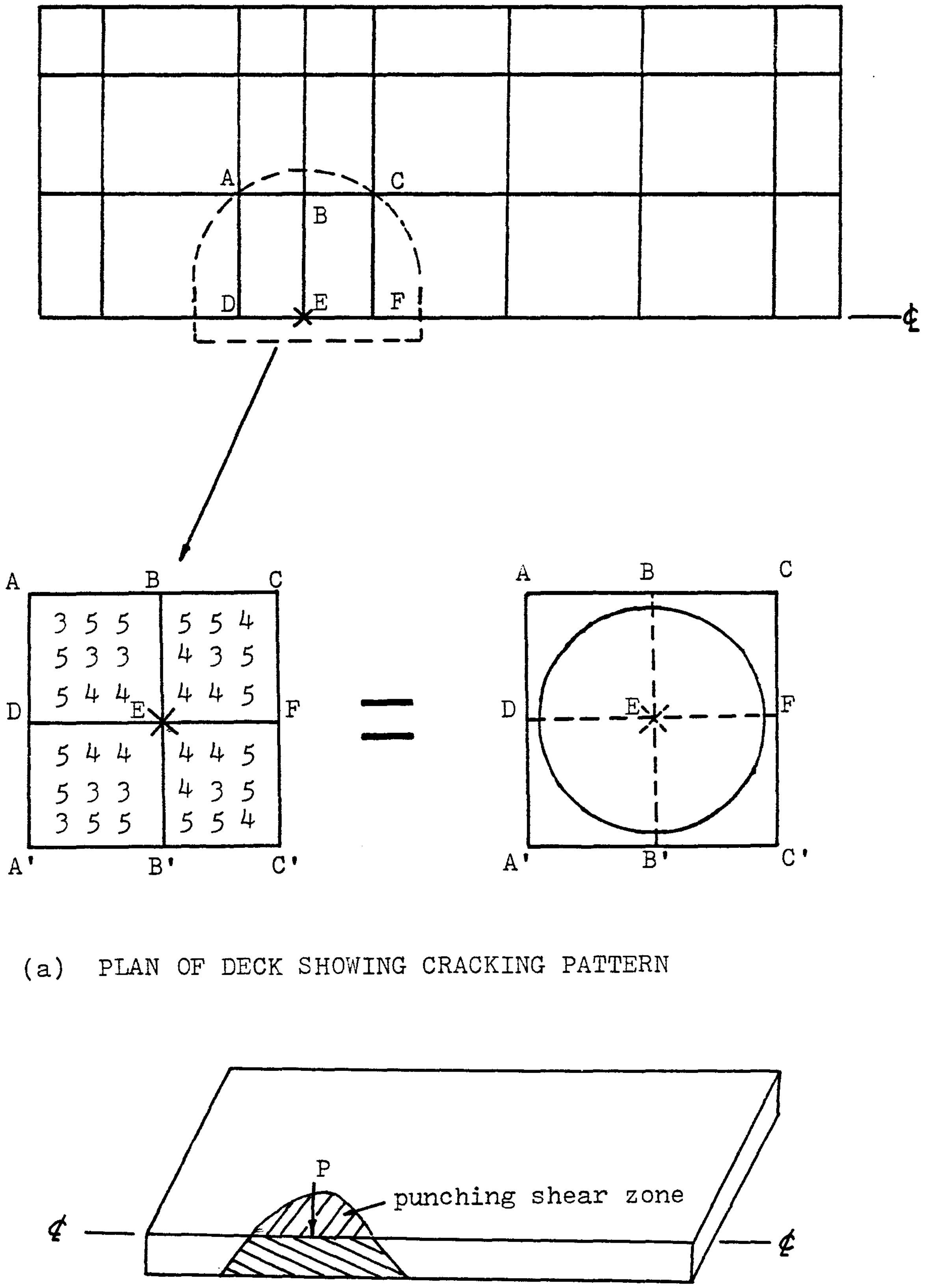


FIG. 5.8 PROGRESSION OF CRACKING AT DIFFERENT LOAD LEVEL FOR POST-TENSIONED CONCRETE DECK (1,2,3,4,5 SIGNIFY NUMBER OF CRACKED INTEGRATION POINTS)



(a) PLAN OF DECK SHOWING CRACKING PATTERN

(b) PUNCHING SHEAR FAILURE MODE

FIG. 5.9 CRACKING PATTERN AROUND LOADING POINT

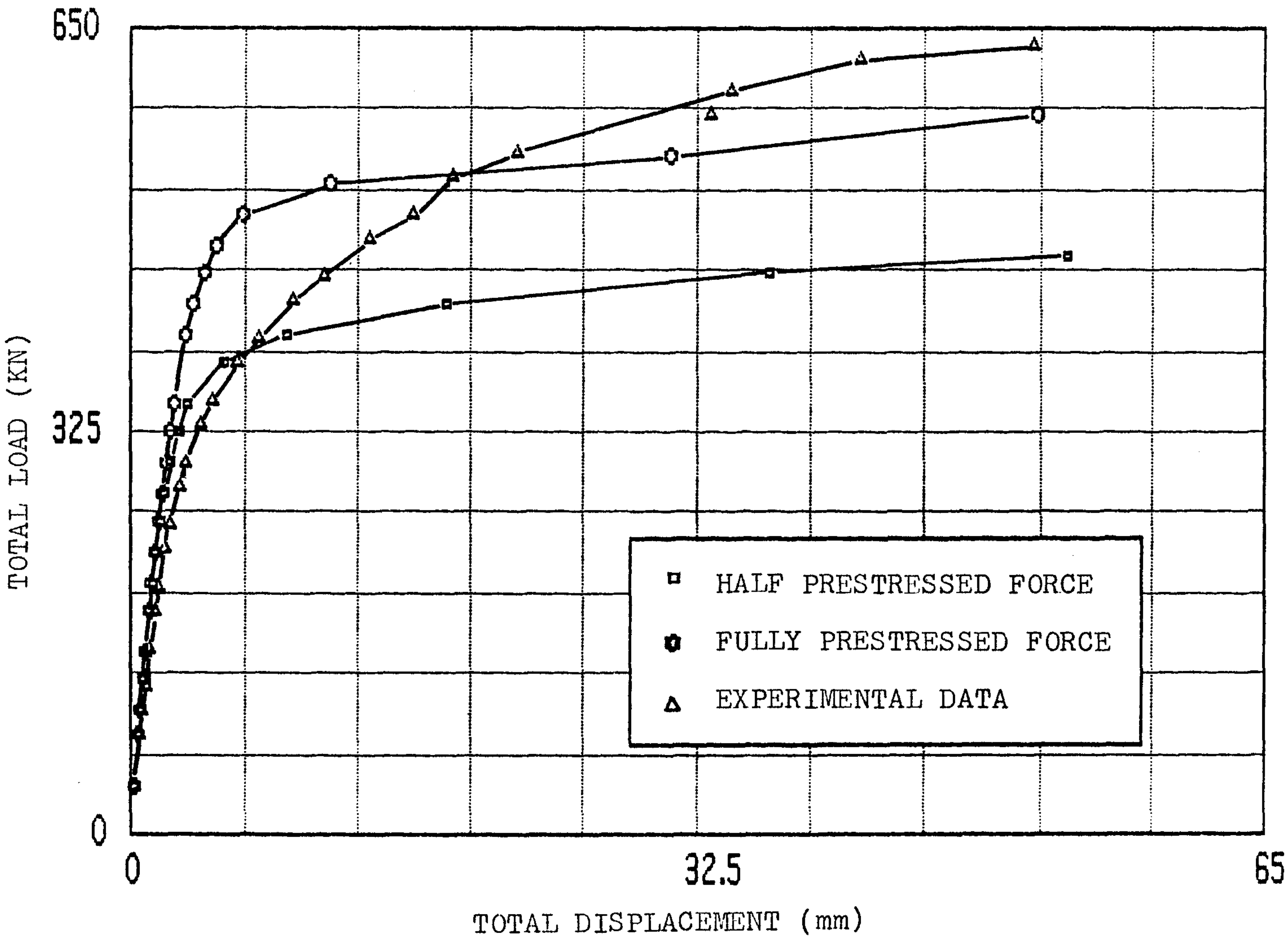


FIG. 5.10 TOTAL LOAD-DISPLACEMENT DIAGRAM FOR DIFFERENT PRESTRESSED FORCES

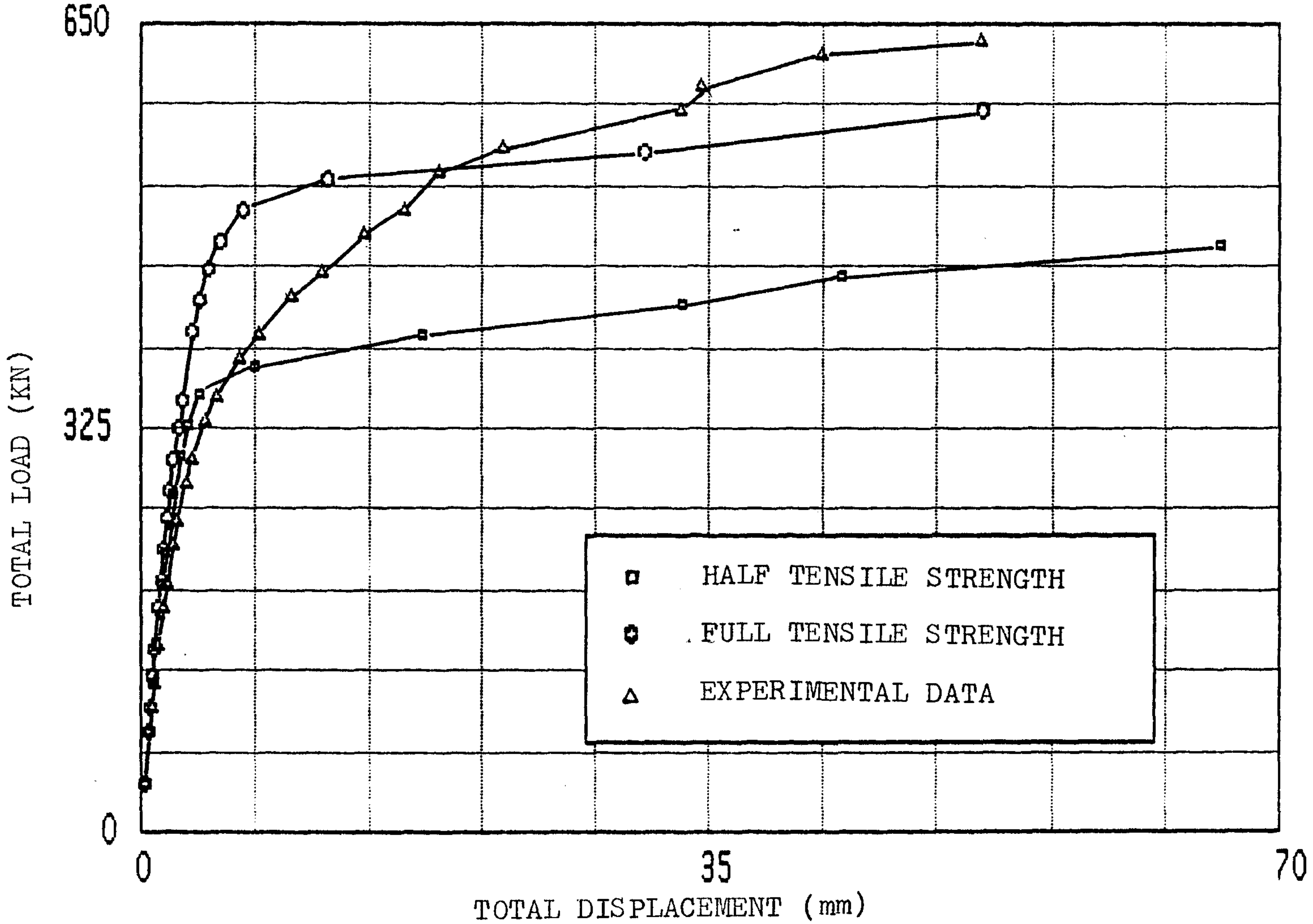


FIG. 5.11 TOTAL LOAD-DISPLACEMENT DIAGRAM FOR DIFFERENT TENSILE STRENGTH

CHAPTER 6

SUMMARY AND CONCLUSIONS

6.1 Summary and Conclusions

The main objective of this investigation was to develop a finite element model to incorporate the bond slip, dowel action and the tension stiffening effects for analysis of the reinforced concrete slabs and the prestressed concrete slabs. As an extension of the previous investigation by Chow [10], the emphasis in the present study was placed on investigating the influences of the three factors mentioned above on the predicted response that had been neglected by Chow.

The bond slip and the dowel action reflect the interface behavior of the reinforced concrete structures. In the present study, the bond slip and dowel action effects were modelled through interface elements which join steel elements to the concrete elements and are capable of transferring interface shear stresses from concrete to steel and vice versa. Both individual influences and the combined effect of the two factors above have been examined using various values of bond stiffness k_b and dowel stiffness k_d.

In the modelling of bond slip behavior, it was found that the response was not sensitive to the bond stress distribution for both the normally reinforced concrete slabs and the prestressed concrete slabs. In the present study, the effect of bond stiffness k_b has been examined for different constant values of k_b instead of using some empirical formulations.

The dowel action behavior was taken into account in the modelling of the post-tensioned concrete deck subjected to a concentrated load. The results showed that the contribution of the dowel action mechanism to the ultimate strength of the deck is very important and the proposed model predicts a punching shear failure mode similar to that observed in the experiment. It was found that the predicted behaviour of the deck is very sensitive to the dowel interface stiffness k_d and a lower value of k_d (nearly zero) tends to give a better analytical result. To examine the combined effect of bond slip and dowel action, the two effects were included in the finite element computer program. Again, the effect of bond-slip was found to be negligible.

The modelling of tension stiffening effect differs from the available approaches reported in the literature. The drawbacks (regarding the stress-strain relationships used for strain-softening) of the previous models were discussed. To avoid these drawbacks, a more rational model to include the tension stiffening effect was developed. This model was based on a fracture mechanics approach. It was observed that the elimination of the existing tensile stress perpendicular to the new crack surface not only yielded much more improved response, but also achieved a gradual release of tensile stresses in concrete in the neighbourhood of a new crack extension.

The results of the two numerical examples showed good agreement with the test data. The overall load-displacement curve for the simply supported reinforced concrete slab from the finite element model was still below the experimental curve and might have been due to the assumptions made in the analysis such as, idealization of the boundary conditions, smear cracking model, etc. For the post-tensioned concrete deck, the predicted load-displacement curve in the serviceability loading range was stiffer than the experimental curve. This can be explained in terms of neglecting certain factors in the model such as strain softening of concrete in a higher compression-compression regions, and loss of prestress before application of the external load and shrinkage. Furthermore, there were some uncertainties about the measured compressive and tensile strengths and the modulus of elasticity for concrete, yield stress and modulus of elasticity for steel as well as the tangent modulus used after yielding of steel.

6.2 Recommendations for Future Research

In order to know more about the behavior of prestressed concrete slabs, additional experimental and analytical research is required. The following recommendations can be drawn from the present study.

(1) Experimental investigations of the behavior of prestressed concrete slabs with different supports, (e.g., simple or fixed supports in one way or two ways) and subjected to different loading (e.g., uniformly distributed load or concentrated load). This will provide the necessary test data for verification of the proposed model.

(2) Analytical equations for bond slip and dowel action effects, based on more rational theories, are needed for finite element modelling.

(3) More experimental data on the crack spacing and width at different load levels is required. This will allow the potential energy release rate to be evaluated as a crack propogates. This will help develop a more accurate model to represent the tension stiffening effect.

(4) The nonlinear constitutive equations for the concrete must be taken into account and be incorporated into the finite element model. For this purpose, development of an objective set of nonlinear constitutive equations for concrete is needed.

(5) Many other aspects of concrete such as aggregate interlock, creep and shrinkage, cyclic loading and large deformations must also be investigated for a better understanding of the overall behavior near failure.

If the above mentioned recommendations are carried out in the future, a complete mathematical model, which reflects the actual behavioral characteristics of the reinforced concrete slabs can be obtained. Of course, one attempts to accomplish as much as possible but thinking of a complete model is still wishful thinking at the present time.

APPENDIX A

Shape functions for plate bending element and steel element

$$N_1^c = 1 - (\frac{1+s}{2})(\frac{1+t}{2}) - (2-s)(\frac{1+s}{2})^2(\frac{1-t}{2}) - (\frac{1-s}{2})(2-t)(\frac{1+t}{2})^2$$

$$N_2^c = a(\frac{1-s}{2})^2(\frac{1+s}{2})(\frac{1-t}{2})$$

$$N_3^c = 6(\frac{1-s}{2})(\frac{1+t}{2})(\frac{1-t}{2})^2$$

$$N_4^c = (\frac{1+s}{2})^2(\frac{1-t}{2})(2-s) - (\frac{1+s}{2})(\frac{1+t}{2})(\frac{1-t}{2})t$$

$$N_5^c = -a(\frac{1-s}{2})(\frac{1+s}{2})^2(\frac{1-t}{2})$$

$$N_6^c = b(\frac{1+s}{2})(\frac{1+t}{2})(\frac{1-t}{2})^2$$

$$N_7^c = (2-s)(\frac{1+s}{2})(\frac{1+t}{2}) + (\frac{1+s}{2})(\frac{1+t}{2})(\frac{1-t}{2})$$

$$N_8^c = -a(\frac{1-s}{2})(\frac{1+s}{2})^2(\frac{1+t}{2})$$

$$N_9^c = -b(\frac{1+s}{2})(\frac{1-t}{2})(\frac{1+t}{2})^2$$

$$N_{10}^c = (\frac{1-s}{2})(\frac{1+t}{2})^2(2-t) - (\frac{1+s}{2})(\frac{1-s}{2})(\frac{1+t}{2})s$$

$$N_{11}^c = a(\frac{1+s}{2})(\frac{1-s}{2})^2(\frac{1+t}{2})$$

$$N_{12}^c = -b(\frac{1-s}{2})(\frac{1-t}{2})(\frac{1+t}{2})^2$$

Note that a,b are the dimensions of rectangular element and s,t the non-dimensionalized coordinates shown in Figure 2.7.

The displacement field along the centroidal axis of the steel element used is given by polynomials of the form:

$$w^s = a_1 + a_2x + a_3x + a_4x^3 \tag{A.1}$$

$$u^s = a_5 + a_6 + a_7x^2 \tag{A.2}$$

Substituting Equations (A.1) and (A.2) into Equation (2.12) yields

$$u^c = a_s + a_6x + a_7x^2 - e\ (a_2 + 2a_3x + 3a_4x^2)$$

$$= (a_5 - ea_2) + (a_6 - 2ea_5)x + (a_7 - 3ea_4)x^2. \tag{A.3}$$

See Figure 2.6 for the kinemitic constraints.

Since the in-place displacements for the concrete slab are linear along the x and y directions, u^c must be linear as well, then from Equation (A.3)

$$a_7 - 3ea_4 = 0 \tag{A.4}$$

$$a_7 = 3ea_4 \tag{A.5}$$

Substituting the equations above back into Equation (A.2), the displacement fields in Equations (A.1), (A.2) and (A.3) become:

$$w^s = a_1 + a_2x + a_3x^2 + a_4x^3 \tag{A.5}$$

$$u^s = a_5 + a_6x + 3ea_4x^2 \tag{A.6}$$

$$u^c = (a_5 - ea_2) + (a_6 - 2ea_3)x \tag{A.7}$$

These equations define the displacement field of the steel element at the steel level (u^s, w^s) and at the concrete mid-surface (u^c). Using the degrees of freedom shown in Figure 2.6, the following end conditions result:

$$u_1^s = u^c(0),\ w_1^s = w^s(0),\ \Theta_1^s = \frac{\partial w^s}{\partial x}(0)$$

$$u_2^s = u^c(\ell),\ w_2^s = w^s(\ell),\ \Theta_2^s = \frac{\partial w^s}{\partial x}(\ell) \qquad \text{(A.8)}$$

Substituting the nodal coordinate of the steel element into Equations (A.5) and (A.7) and using Equation (A.8), the following transformation is obtained

$$\{\delta_s\} = [T]\ \{A\}$$

Where

$$[T] = \begin{bmatrix} 0 & -e & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -e & -2e\ell & 0 & 1 & \ell \\ 1 & \ell & \ell^2 & \ell^3 & 0 & 0 \\ 0 & 1 & 2\ell & 3\ell^2 & 0 & 0 \end{bmatrix} \qquad \text{(A.9)}$$

$$\{\delta_s\}^T = < u_1^s,\ w_1^s,\ \Theta_1^s,\ u_2^s,\ w_2^s,\ \Theta_2^s >$$

and $\{A\}^T = < a_1, a_2, a_3, a_4, a_5, a_6 >$

The unknown coefficients {A} can be expressed in terms of the nodal degrees of freedom through the following inverse of [T]

$$\{A\} = [T]^{-1}\ \{\delta_s\} \qquad \text{(A.10)}$$

Where

$$[T]^{-1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -\frac{3}{\ell^2} & -\frac{2}{\ell} & 0 & \frac{3}{\ell^2} & -\frac{1}{\ell} \\ 0 & \frac{2}{\ell^2} & \frac{1}{\ell^2} & 0 & -\frac{2}{\ell^3} & \frac{1}{\ell^2} \\ 1 & 0 & e & 0 & 0 & 0 \\ -\frac{1}{\ell^2} & -\frac{6e}{\ell^2} & -\frac{4e}{\ell} & \frac{1}{\ell} & \frac{6e}{\ell^2} & -\frac{2e}{\ell} \end{bmatrix} \qquad (A.11)$$

The coefficients {A} from Equation (A.10) can be substituted into Equations (A.5) and (A.7) to obtain the shape functions for the steel element. By using the non-dimensional coordinate in Figure 2.6 (ζ, $\bar{z}$), the finite element approximation for w^s and u^s take the following form,

$$w^s = \sum_{i=1}^{4} N_i \delta'_{s_i} \qquad (A.12)$$

where

$$N_1 = 1- (2-\zeta)(\frac{1+\zeta}{2})^2$$

$$N_2 = \ell(\frac{1-\zeta}{2})^2(\frac{1+\zeta}{2})$$

$$N_3 = (\frac{1+\zeta}{2})^2(2-\zeta)$$

$$N_4 = -\ell(\frac{1-\zeta}{2})(\frac{1+\zeta}{2})^2$$

$$\{\delta'_s\}^T = < w_1^s, \Theta_1^s, w_2^s, \Theta_2^s >$$

and

$$u^s = \sum_{i=1}^{6} \bar{N}_i \delta_{s_i}$$

where $\bar{N}_1 = \frac{1-\zeta}{2}$

$$\bar{N}_2 = \frac{6e}{\ell}(\frac{1+\zeta}{2})(\frac{\zeta-1}{2})$$

$$\bar{N}_3 = e[1-4(\frac{1+\zeta}{2}) + 3(\frac{1+\zeta}{2})^2]$$

$$\bar{N}_4 = \frac{1+\zeta}{2}$$

$$\bar{N}_5 = \frac{6e}{e}(\frac{1+\zeta}{2})(\frac{1-\zeta}{2})$$

$$\bar{N}_6 = e[-2(\frac{1+\zeta}{2}) + 3(\frac{1+\zeta}{2})^2]$$

$$\{\delta_s\}^T = < u_1^s, w_1^s, \Theta_1^s, u_2^s, w_2^s, \Theta_2^s >.$$

APPENDIX B

Transformation matrix for bond-slip element

$$\underset{22X21}{[T_b]} = \begin{bmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
a_1&b_1&0&0&0&0&b_2&0&0&0&0&0&0&0&0&a_2&0&0&0&0&c_1\\
0&0&0&0&0&a_3&0&0&0&0&a_4&b_4&0&0&0&0&b_3&0&0&0&c_2
\end{bmatrix}$$

(1) ${u_1}^s$ at the boundary

$$a_1 = d - \frac{y_s-y_1}{y_4-y_1}, \qquad a_2 = \frac{y_s-y_1}{y_4-y_1}, \qquad c_2 = 1$$

$a_3=a_d=b_1=b_2=b_3=b_4=c_1=0$

(2) ${u_2}^s$ at the boundary

$$a_3 = s - \frac{y_s-y_2}{y_3-y_2}, \qquad a_4 = \frac{y_s-y_2}{y_3-y_2}, \qquad c_1 = 1$$

$a_1=a_2=b_1=b_2=b_3=b_4=c_2=0$

(3) ${v_1}^s$ at the boundary

$$b_1 = 1 - \frac{x_s-x_1}{x_2-x_1}, \qquad b_2 = \frac{x_s-x_1}{x_2-x_1}, \qquad c_2 = 1$$

$a_1=a_2=a_3=a_4=b_3=b_4=c_1=0$

(4) ${v_2}^s$ at the boundary

$$b_3 = 1 - \frac{x_s-x_4}{x_3-x_4}, \qquad b_4 = \frac{x_s-x_4}{x_3-x_4}, \qquad c_1 = 1$$

$a_1=a_2=a_3=a_4=b_1=b_2=c_2=0$

${u_1}^s$, ${u_2}^s$, ${v_1}^s$, ${v_2}^s$, x_1, x_2, x_3, x_4, x_s, y_1, y_2, y_3, y_4 and y_s are shown in Figure 2.12.

APPENDIX C

Transformation matrix for dowel-action element

$$
\begin{bmatrix} u_1 \\ v_1 \\ w_1 \\ w_{1,x} \\ w_{1,y} \\ u_2 \\ v_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ w_{4,y} \\ u_1^s \\ w_1^s \\ \theta_1^s \\ u_2^s \\ w_2^s \\ \theta_2^s \end{bmatrix}
=
\left[\begin{array}{ccccccccccccccccccccccc}
 & & & & & & & & & [\ I\ ] & & & & & & & & & & & [\ 0\ ] & & \\
 & & & & & & & & & 20\times20 & & & & & & & & & & & 20\times3 & & \\
 & & & & & & & & [\ 0\ ] & & & & & & & & & & & & 1 & 0 & 0 \\
 & & & & & & & & 3\times20 & & & & & & & & & & & & 0 & 1 & 0 \\
 & & & & & & & & & & & & & & & & & & & & 0 & 0 & 1 \\
N_1^c & 0 & 0 & 0 & 0 & N_2^c & 0 & 0 & 0 & 0 & N_3^c & 0 & 0 & 0 & 0 & N_4^c & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & N_1^c & N_2^c & N_3^c & 0 & 0 & N_4^c & N_5^c & N_6^c & 0 & 0 & N_7^c & N_8^c & N_9^c & 0 & 0 & N_{10}^c & N_{11}^c & N_{12}^c & 0 & 0 & 0 \\
0 & 0 & N_{1,}^c & N_{2,x}^c & N_{3,x}^c & 0 & 0 & N_{4,x}^c & N_{5,x}^c & N_{6,x}^c & 0 & 0 & N_{7,x}^c & N_{8,x}^c & N_{9,x}^c & 0 & 0 & N_{10,x}^c & N_{11,x}^c & N_{12,x}^c & 0 & 0 & 0
\end{array}\right]
\begin{bmatrix} u_1 \\ v_1 \\ w_1 \\ w_{1,x} \\ w_{1,y} \\ u_2 \\ v_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ u_1^s \\ w_1^s \\ \theta_1^s \end{bmatrix}
$$

Note that $N_1 - N_4$, $N_1^c - N_{12}^c$, and $N_{1,x}^c - N_{12,x}^c$ are calculated at $(+1,t_0)$ for node 2 $(u_2^s,\ w_2^s,\ {}_2^s)$ at boundary; [I] is an identity matrix and [0] ia zero matrix.

APPENDIX D

INPUT DATA REQUIREMENTS

(1) TITLE

(2) NEL, NELS, NNOD, NNODS, NVAR, NNODEL, NNODSEL
NEL - number of concrete elements
NELS - number of steel elements
NNOD - number of nodes of concrete elements
NNODS - number of nodes of steel elements
NVAR - number of variables per node for concrete elements
NNODEL - number of nodes per concrete element
NNODSEL - number of nodes of steel within a concrete element

(3) NINCR, MITER, NPRINT
NINCR - number of load increments
MITER - maximum number of iterations allowed
NPRINT - counter for outputting stresses and strains

(4) (FAC(I), I=1, NINCR)
FAC(I) - factor controlling magnitude of load increment expressed as a fraction of the initial load increment

(5) E, E1, E2, BETA
E - Young's modulus of concrete
E1, E2 - tangent modulus of concrete for non-linear analysis
BETA - factor controlling magnitude of shear transfer in cracked region

(6) ET, EP, YL, YLP, TOLER, ECX, ECY

ET - Young's modulus of non-prestressed steel
EP - Young's modulus of prestressed steel
YL - yield stress of non-prestressed steel
YLP - yield stress of prestressed steel
TOLER - tolerance for convergence
ECX - distance of steel centroid from concrete mid-surface in x direction
ECY - distance of steel centroid from concrete mid-surface in y direction

(7) SKUIN, SKWIN, SLIPM
SKUIN - initial value of bond stiffness coefficient
SKWIN - initial value of dowel stiffness coefficient
SLIPM - allowed maximum value of slip

(8) ((X(I), Y(I), (IX(J), J=I1, I2)), I=1, NNOD)
X(I) - x coordinate of node I of concrete element
Y(I) - y coordinate of node I of concrete element
IX(J) - degree of freedom at node I of concrete element
= 0 degree of freedom constrained
= 1 degree of freedom unknown

(9) ((SX(I), SY(I), (IX(J), J=J1, J2), IBP(I)), I=1, NNODS)
SX(I) - x coordinate of node I of steel element
SY(I) - y coordinate of node I of steel element
IX(J) - degree of freedom dt node J of steel element
= 0 degree of freedom constrained
= 1 degree of freedom unknown
IBP(I) - counter for indicating position of nodes of steel
= 0 internal node
= 1 boundary node

(10) (((ICO(I,J), J=1, NNN), THICK(I)), I=1, NEL)
ICO(I,J)- concrete element node numbering and IS, IB

IS - counter for generation of new element stiffness matrix
= 0 stiffness matrix same as previous
= 1 need to generate new element stiffness matrix
IB - counter for loading condition
= 0 no loads on element
= 1 uniformly distributed load
= 2 concentrated load

(11) (((ISO(I,J), I=1,6) AREA(J)), J=1, NELS)
ISO(I,J)- steel element node numbering and IS, IB, IC and ID
IS, IB - definition same as concrete element
IC - number of concrete element within which steel element locates
ID - counter for indicating direction of steel element
= 1 in horizontal direction
= 2 in vertical direction
AREA(J) - area of each steel element J

APPENDIX E

SUBROUTINE DEFINITION

ADD Adds two vectors or matrixes together

ADDSTIF Adds the concrete, steel and joint element stiffnesses together to form a composite element stiffness matrix

BOLT Brings the contribution of stiffnesses of tie bars along the boundary to global stiffness matrix

BOND Calculates the bond stiffness coefficient k_b

BTRANST Transformation matrix ralates degrees of freedom of a node of steel element which is at the boundary to corresponding degrees of freedom of a concrete element

CLOAD Builds the applied load vector for concrete

COHESST Applies an opposite and equal tensile stress to a newly cracking point

COLHT Calculates column height of the global stiffness matrix for skyline storage

COLSOL Solves the algebratic equations using Gaussian elimination and skyline storage

CONCRT Builds concrete stiffness matrix and residual load vector

CONVGE Checks convergence of the iterations and exits iteration loop when convergence is reached

CURRENT	Updates the current element properties
DCONC	Builds the non-linear stress-strain matrix
DIAADD	Calculates the address of the diagonal term
DMATX	Checks stresses in concrete and calculates the concrete stress-strain matrix modified for cracking
DSTEEL	Forms the steel stress-strain matrix
ELASTIC	Finds the stresses and strains for first load increment
INCLOAD	Calculates the load vector for each increment
INPUT	Reads all input data except initial loading and prestressing forces
JOINT	Builds joint stiffness matrix
JTRANST	Transformation matrix ralates degrees of freedom of joint element which is at the boundary to corresponding degrees of freedom of concrete element
JUSHAPE	Builds the $[N_b]$ matrix for joint element
JWSHAPE	Builds the $[N_d]$ matrix for joint element
LAYOUT	Reads the element data and sets up the element
MATRIX	Builds the [B] matrix for concrete
MULT	Multiplies an MxN matrix with an NxL matrix
MULT2	Multiplies three matrices C=BT*A*B
OUTPUT	Outputs the results for each iteration
OUTSIG	Outputs the stresses and strains for each element
PSET	Zeros one dimensional array
PRESET	Zeros two dimensional array
ROTAT	Transformation matrix to bring principle stresses to global direction

SBMATX Builds the [B] matrix for steel

SETUP Assembles the global stiffness matrix and load vector

SHAPE Contains the shape functions for each concrete element

SSTIFF Builds the element stiffness matrix for steel

SSTRESS Checks to see if steel element has yielded and modifies the stress-strain matrix accordingly

STEEL Loops over number of steel element and enters steel and joint element contribution to the composite stiffness matrix

STORE Extracts or stores [B] or [D] matrix on tape for the 45 integration points

STRESS Checks to see if concrete element has cracked and modifies the stress-strain matrix accordingly

TAPE Rewinds and switch tape numbers

TENSTIF Considers tension stiffening effect using the suggested model

TRANPS Finds the transpose of a matrix

TRANST Relates steel degrees of freedom to concrete element

TRNDISP Finds the displacements for the node of steel element at the boundary

TRNLOAD Transforms the steel internal load vector for which exists at the boundary to the corresponding concrete internal load vector

TRNSTIF Transforms the stiffness matrices of steel and joint elements at the boundary to the corresponding concrete stiffness matrix

TRNPRST Transforms the prestressing forces for which are applied at the boundary to the corresponding concrete load vector

UCSHAPE Calculates the shape functions corresponding to u^c

UPDATE Updates the displacements

USSHAPE Calculates the shape functions corresponding to u^s

WSSHAPE Calculates the shape functions corresponding to w^s

ZERO Initializes arrays

ZEROSIG Zeros out stresses in major principle direction after concrete cracking

APPENDIX F

PRESTRESSED CONCRETE SLAB COMPUTER PROGRAM

```
      PROGRAM PLATE
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION TSTIF(40000),TSTIF2(40000),
     +          RLOAD(400),RLOAD2(400),TLOAD(400),DELOAD(400),
     +          TDISP(400),DEDISP(400),MDIA(400),TP(400),
     +          MHT(400),IX(1000),JX(1000),IBP(300),TPR(400),
     +          X(100),Y(100),TSTLOAD(68),TLOADST(500),P2(68),
     +          THICK(45,36),ICO(6,36),IPTCR(46,36),IPTCR2(46,36),
     +          ICR(46,36),THETA(45,36),TLOAD2(400),XK(10),ND(10),
     +          SIGS(200),DSIGS(200),EPSS(200),DEPSS(200),
     +          AREA(200),ISO(6,200),TOTST(26,200),DSTL(3,200),
     +          SX(300),SY(300),IPTYL(3,200),BSTEL(6,3,200),
     +          FAC(200),BSTORE(3,20,45),DSTORE(3,3,45)
      COMMON/CSTIF/LJ(68),P(68),STIF(68,68),DISP(20,1)
      COMMON/CSIGS/SIG(3,45),EPS(3,45),BMATX(3,20),D(3,3),BSTORE,DSTORE
      COMMON/CSTSP/STSIG(3),STEPS(3),STIFS(6,6),SPSTIF(26,26),
     +            STLOAD(6,1),SPLOAD(26,1),TRLOAD(23,1),TRSTIF(26,26),
     +            SUSH(26),SWSH(26),UCSH(4),USSH(6),WSSH(4)
      COMMON/CCONCR/BDBC(20,20),DBC(3,20),BCT(20,3),TEMP(3,1),BSIG(20,1)
      COMMON/CSTRES/DSIG(3,1),DEPS(3,1),TEPS(3,45),TSIG(3,45)
      COMMON/CELAST/TEPS1(3,1),TSIG1(3,1),BTEMP1(1,6),DTDISP(26,1),
     +              STDISP(6,1)
      COMMON/CTENST/PP(20),REDISP(400),TPR1(400),TSTIF1(40000),
     +              RSIG(3,1),BRSIG(20,1),BST(20,3)
      COMMON/CTRAN/TRAN(10,20),TRAN1(6,23),TRANJ(26,23),TRANT(20,10),
     +             TRAN1T(23,6),C(11)
      COMMON/CJWSH/TRN(4,20),TEMPWS(1,4),TEMPWC(1,20)
      COMMON/CCOHZ/R(3,3),TEMPFT(3,1)
      COMMON/CSSTIF/BDB(6,6),DB(1,6)
      COMMON/CTRNSF/TST(23,23),ST(6,23),PT(26,23)
      COMMON/CTRNLD/BTEMP(6,1),SPDISP(26,1),TEMPLD(20)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      COMMON/CTAPE/MTAPE,NTAPE,LTAPE,KTAPE
      COMMON/CONST/E,E1,E2,ANU,BETA,FCU,FTU,ET,EP,YL,YLP,EPSU,ECX,ECY
      OPEN (UNIT=1,FORM='UNFORMATTED',STATUS='SCRATCH')
      OPEN (UNIT=2,FORM='UNFORMATTED',STATUS='SCRATCH')
      OPEN (UNIT=3,FORM='UNFORMATTED',STATUS='SCRATCH')
      OPEN (UNIT=5,FILE='DATA',STATUS='OLD')
      OPEN (UNIT=6,FILE='OUTPUT',STATUS='NEW')
      OPEN (UNIT=7,FORM='UNFORMATTED',STATUS='SCRATCH')
      OPEN (UNIT=8,FORM='UNFORMATTED',STATUS='SCRATCH')
      CALL INPUT(NINCR,MITER,FAC,TOLER,NPRINT,SKUIN,SLIPM,SKWIN,NSP)
      CALL LAYOUT(X,Y,ICO,IX,JX,MHT,LJ,MDIA,NVA,THICK,NNET1,AREA,SX,
     +            SY,ISO,IBP,XK,ND,NSP)
      CALL ZERO(TDISP,TLOAD,DELOAD,P,DET,IEXP,RLOAD,IPTCR,DSIGS,SIGS,
     +          DEPSS,EPSS,THICK,DSTL,IPTYL,BSTEL)
```

```
      DO INCR=1,NINCR
      CALL INCLOAD(INCR,MITER,NITER,DELOAD,FAC,TLOAD,RLOAD,JX,TLOADST)
        DO ITER=1,NITER
        IF(INCR.GT.1)THEN
         CALL COLSOL(TSTIF,RLOAD,DEDISP,MDIA,NNETT,NVA,1,TSTIF2,
     +                RLOAD2)
         CALL STRESS(IPTCR,DEDISP,ICO,JX,X,Y,INCR,ITER,THICK,THETA,
     +                ICR,TOTST,IBP,DSTL,IPTYL,BSTEL,ISO,SX,SY)
         IF(ITER.EQ.1) CALL TENSTIF(TSTIF,DEDISP,TPR,THETA,THICK,
     +                              ICR,MDIA,ICO,JX,X,Y,NVA,NCR)
        ENDIF
        CALL REWIND(MTAPE,NTAPE,LTAPE,KTAPE)
        CALL PSET(TSTIF,NVA)
        CALL PSET(TSTIF2,NVA)
        CALL PSET(TLOAD2,NNETT)
        CALL PSET(TP,NNETT)
          DO IEL=1,NEL
          CALL CURRENT(IEL,NNODEL,ICO,JX,AA,BB,NVAR,LJ,X,Y,IS,IB,NEL)
          CALL CONCRT(INCR,ITER,THICK,AA,BB,NPRINT,IEL,IS,THETA,ICR)
          CALL STEEL(SX,SY,AREA,JX,DEDISP,X,Y,ISO,ICO,DSIGS,INCR,ITER,
     +             TOTST,IEL,IBP,DSTL,BSTEL,SKUIN,SLIPM,SKWIN,TSTLOAD)
          IF(INCR.EQ.1)THEN
            CALL CLOAD(P,AA,BB,IEL,IB,NVELT)
            CALL ADD(P,TSTLOAD,P2,1,1,NVELT)
          ELSE
            DO 2 I=1,NVEL
    2       P2(I)=P(I)
          ENDIF
          CALL SETUP(TSTIF,TP,STIF,P2,MDIA,LJ,NVELT)
          CALL SETUP(TSTIF2,TLOAD2,STIF,P,MDIA,LJ,NVELT)
          ENDDO
        CALL BOLT(NVAR,JX,MDIA,TSTIF,XK,ND,NSP)
        CALL BOLT(NVAR,JX,MDIA,TSTIF2,XK,ND,NSP)
        IF(INCR.GT.1 .AND. NCR.EQ.1) CALL ADD(TP,TPR,TP,1,1,NNET)
        CALL ELASTIC(ICO,JX,X,Y,DEDISP,NPRINT,TSTIF2,RLOAD2,THICK,
     +               RLOAD,INCR,ITER,TSTIF,TP,MDIA,NVA,NNET1,THETA,
     +               ISO,SX,SY,TOTST,IBP,DSTL,BSTEL)
        CALL CONVGE(TSTIF2,MDIA,DET,IEXP,TOLER,INCR,NCONV,ITER)
        CALL UPDATE(FAC,INCR,TLOAD,RLOAD,DEDISP,TDISP,ITER,TP,JX,
     +              NCONV,MITER,TLOADST,TLOAD2)
        IF(NCONV.EQ.1) GO TO 1
        ENDDO
    1 CONTINUE
      ENDDO
      END
C
C
C
      SUBROUTINE DCONC(D,ALPHA,EES1,EES2,C1,C2,N)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION D(3,3)
      COMMON/CONST/E,E1,E2,ANU,BETA,FCU,FTU,ET,EP,YL,YLP,EPSU,ECX,ECY
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      IF(N.EQ.0)THEN                                ! ELASTIC
        E1B=E
        E2B=E
      ELSEIF(N.EQ.1)THEN                            ! T-C
        E1B=E
        E2B=E2
      ELSEIF(N.EQ.2)THEN                            ! C-C
        CC1=1.D0/(1.D0-ANU/ALPHA)
        CC2=1.D0/(1.D0-ANU*ALPHA)
        E1B=E*(1.D0-C1**2)/(1.D0+(CC1*EES1-2.D0)*C1+C1**2)**2
        E2B=E*(1.D0-C2**2)/(1.D0+(CC2*EES2-2.D0)*C2+C2**2)**2
      ELSEIF(N.EQ.3)THEN                            ! T-T
        E1B=0.5*E
        E2B=0.5*E
      ENDIF
      CALL PRESET(D,3,3)
      XLAMDA=E1B/((E1B/E2B)-ANU*ANU)
      D(1,1)=XLAMDA*E1B/E2B
      D(1,2)=XLAMDA*ANU
      D(2,1)=D(1,2)
      D(2,2)=XLAMDA
      D(3,3)=E1B*E2B/(E1B+E2B+2.D0*E2B*ANU)
      RETURN
      END
C
C
C
      SUBROUTINE DMATX(EPS,SIG,IPT,IEL,ICRACK,ICR,D,THICK,THETA,IPTCR)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION EPS(3,45),SIG(3,45),ICR(46,1),D(3,3),THICK(45,1),
     +          THETA(45,1),IPTCR(46,1)
      COMMON/CONST/E,E1,E2,ANU,BETA,FCU,FTU,ET,EP,YL,YLP,EPSU,ECX,ECY
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      IF(IPTCR(IPT,IEL).EQ.0)THEN
        IF(ABS(SIG(1,IPT)).LT.1.D-3 .AND. ABS(SIG(2,IPT)).LT.1.D-3
     +     .AND. ABS(SIG(3,IPT)).LT.1.D-3)THEN
          CALL DCONC(D,ALPHA,EES1,EES2,C1,C2,0)
          RETURN
        ENDIF
      ENDIF
      EMEAN=(EPS(1,IPT)+EPS(2,IPT))/2.D0
      EROOT=DSQRT(((EPS(1,IPT)-EPS(2,IPT))/2.D0)**2 + EPS(3,IPT)**2)
      SMEAN=(SIG(1,IPT)+SIG(2,IPT))/2.D0
      SROOT=DSQRT(((SIG(1,IPT)-SIG(2,IPT))/2.D0)**2 + SIG(3,IPT)**2)
      IF(IPTCR(IPT,IEL).EQ.0)THEN
        SIG1=SMEAN + SROOT
        SIG2=SMEAN - SROOT
        EPS1=EMEAN + EROOT
        EPS2=EMEAN - EROOT
      ELSE
        SIG1=SMEAN - SROOT
        SIG2=SMEAN + SROOT
        EPS1=EMEAN - EROOT
        EPS2=EMEAN + EROOT
      ENDIF
      IF(SIG2.EQ.0.0)THEN
        ALPHA=SIG1/0.001
      ELSE
        ALPHA=SIG1/SIG2
      ENDIF
      IF(ABS(SIG(1,IPT)-SIG(2,IPT)).LT.1.D-5)THEN
        THE=0.7853982234
      ELSE
        THE=0.5*DATAN(2.DO*SIG(3,IPT)/(SIG(1,IPT)-SIG(2,IPT)))
      ENDIF
      SIGX=(SIG(1,IPT)+SIG(2,IPT))/2.DO + (SIG(1,IPT)-SIG(2,IPT))*DCOS(
     +     2.DO*THE)/2.DO + SIG(3,IPT)*DSIN(2.DO*THE)
      IF(ABS(SIGX-SIG2).LT.1.D-5) THE=THE-1.570796345
      IF(THE.LT.0.DO) THE=THE+6.283185307
      THETA(IPT,IEL)=THE
      IF(IPTCR(IPT,IEL).EQ.0)THEN                      ! 1ST TIME
        IF(SIG1.LT.0.5*FCU .OR. SIG2.LT.0.5*FCU)THEN
          CALL DCONC(D,ALPHA,EES1,EES2,C1,C2,3)
        ELSE
          CALL DCONC(D,ALPHA,EES1,EES2,C1,C2,1)
        ENDIF
        IF(ALPHA.LT.0.0)THEN                                     ! T-C
          IF(SIG1.GE.FTU .OR. SIG2.LE.FCU)THEN
            IF(ALPHA.GT.-0.17)GO TO 2                  ! CRUSHED
            IF(ALPHA.LE.-0.17)GO TO 1                  ! CRACKED
          ENDIF
        ELSE
          IF(SIG1.LT.0.DO)THEN                                   ! C-C
            IF(SIG1.LE.FCU .OR. SIG2.LE.FCU)THEN
    2         ICR(IPT,IEL)=2
              DO 3 I=1,3
              DO 3 J=1,3
    3         D(I,J)=D(I,J)/10000.DO
            ELSEIF(SIG1.LT.0.5*FCU .OR. SIG2.LT.0.5*FCU)THEN
              DO 23 I=1,3
              DO 23 J=1,3
   23         D(I,J)=D(I,J)/2.0
            ENDIF
          ELSE                                                   ! T-T
            IF(SIG1.GE.FTU .OR. SIG2.GE.FTU)THEN        ! CRACKED
    1         ICR(IPT,IEL)=3
              CALL PRESET(D,3,3)
              D(3,3)=BETA*0.5*E/(1.DO+ANU)
              D(2,2)=E
              CALL ROTAT(D,THE)
            ENDIF
          ENDIF
        ENDIF
      ELSE                                                   ! 2ND TIME
        IF(ALPHA.LT.0)THEN
          IF(SIG1.GE.FTU .OR. SIG2.LE.FCU)THEN
            IPTCR(IPT,IEL)=5
            IF(ALPHA.GT.-0.17)GO TO 4                  ! CRUSHED
            IF(ALPHA.LE.-0.17)GO TO 5                  ! CRACKED
          ENDIF
        ELSE
          IF(SIG1.LT.0)THEN
    4       IF(SIG1.LE.FCU .OR. SIG2.LE.FCU) IPTCR(IPT,IEL)=5
          ELSE
    5       IF(SIG1.GE.FTU .OR. SIG2.GE.FTU) IPTCR(IPT,IEL)=5
          ENDIF
        ENDIF
      ENDIF
      IF(ICR(IPT,IEL).GT.0)ICRACK=ICRACK+1
      RETURN
      END
C
C
C
      SUBROUTINE ROTAT(D,THETA)
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
      DIMENSION R(3,3),D(3,3),DR(3,3),RDR(3,3)
      C2=(DCOS(THETA))**2
      S2=(DSIN(THETA))**2
      SC=DSIN(THETA)*DCOS(THETA)
      R(1,1)= C2
      R(2,1)= S2
      R(3,1)=-2.D0*SC
      R(1,2)= S2
      R(2,2)= C2
      R(3,2)= 2.D0*SC
      R(1,3)= SC
      R(2,3)=-SC
      R(3,3)= C2-S2
      CALL MULT2 (D,R,RDR,3,3,DR)
      DO 1 I=1,3
      DO 1 J=1,3
    1 D(I,J)=RDR(I,J)
      RETURN
      END
C
C
C
      SUBROUTINE CONCRT(INCR,ITER,THICK,AA,BB,NPRINT,IEL,IS,THETA,ICR)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION BSTORE(3,20,45),DSTORE(3,3,45)
      DIMENSION THICK(45,1),THETA(45,1),XI(3),W(3),XJ(5),V(5),
     +          ICR(46,1),IPRINT(37)
      COMMON/CSTIF/LJ(68),P(68),STIF(68,68),DISP(20,1)
      COMMON/CSIGS/SIG(3,45),EPS(3,45),BMATX(3,20),D(3,3),BSTORE,DSTORE
      COMMON/CCONCR/BDBC(20,20),DBC(3,20),BCT(20,3),TEMP(3,1),BSIG(20,1)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      COMMON/CTAPE/MTAPE,NTAPE,LTAPE,KTAPE
      COMMON/CONST/E,E1,E2,ANU,BETA,FCU,FTU,ET,EP,YL,YLP,EPSU,ECX,ECY
      DATA XI/-0.774596669241483,0.000000000000000,0.774596669241483/
      DATA  W/ 0.555555555555556,0.888888888888889,0.555555555555556/
      DATA XJ/-0.906179845938664,-0.538469310105683,0.000000000000000,
     +         0.538469310105683, 0.906179845938664/
      DATA  V/ 0.236926885056189,0.478628670499366,0.568888888888889,
     +         0.478628670499366,0.236926885056189/
      DATA IPRINT/ 1, 2, 4, 6, 8,10,14,18,22,26,30,34,38,42,46,50,54,
     +            66,70,74,78,82,86,90,94,98,102,106,110,114,118,122,
     +            126,130,134,138,142/
      IF(IS.EQ.0)RETURN
      IF(INCR.GT.1)THEN
        READ(MTAPE) ((SIG(L,IPT),L=1,3),IPT=1,45),
     +              ((EPS(L,IPT),L=1,3),IPT=1,45),
     +              (((DSTORE(M,L,IPT),M=1,3),L=1,3),IPT=1,45)
        READ(3)(((BSTORE(M,L,IPT),M=1,3),L=1,20),IPT=1,45)
        IF(NPRINT.EQ.2) THEN
          IF(IEL.EQ.4 .OR. IEL.EQ.7 .OR. IEL.EQ.10 .OR. IEL.EQ.13
     +       .OR. IEL.EQ.3 .OR.IEL.EQ.6 .OR. IEL.EQ.9 .OR.
     +       IEL.EQ.12)THEN
            DO IM=1,37
            IF(INCR.EQ.IPRINT(IM))
     +                  CALL OUTSIG(SIG,EPS,IEL,INCR,ITER,THICK,THETA)
            ENDDO
          ENDIF
        ENDIF
        CALL PSET(P,NVELT)         !CHECK DIMENSION
      ENDIF
      CALL PRESET(STIF,NVELT,NVELT)      !CHECK DIMENSION
      IPT=0
      CONST=AA*BB/8.DO
      IF(INCR.EQ.1)CALL DCONC(D,ALPHA,EES1,EES2,C1,C2,0)
      DO 2 I=1,3
      DO 2 J=1,3
      DO 2 K=1,5
      S=XI(I)
      T=XI(J)
      R=XJ(K)
      IPT=IPT+1
      IF(INCR.EQ.1)THEN
        CALL MATRIX(S,T,R,BMATX,AA,BB,THICK,IEL,IPT,NEL)
        CALL STORE(BMATX,BSTORE,3,20,IPT,1)
      ELSEIF(INCR.GT.1)THEN
        CALL STORE(BMATX,BSTORE,3,20,IPT,0)
        CALL STORE (D,DSTORE,3,3,IPT,0)
        DO 6 L=1,3
    6   TEMP(L,1)=SIG(L,IPT)
        CALL TRANPS(BMATX,BCT,3,20)
        CALL MULT (BCT,TEMP,BSIG,20,3,1)
        DO 8 L=1,NVEL
    8   P(L)=P(L) + W(I)*W(J)*V(K)*BSIG(L,1)*CONST*THICK(IPT,IEL)
      ENDIF
      CALL MULT2(D,BMATX,BDBC,3,20,DBC)
      DO 5 M=1,NVEL
      DO 5 L=1,NVEL
    5 STIF(M,L)=STIF(M,L)+W(I)*W(J)*V(K)*BDBC(M,L)*CONST*THICK(IPT,IEL)
    2 CONTINUE
      IF(INCR.EQ.1)WRITE(3)(((BSTORE(M,L,IPT),M=1,3),L=1,20),IPT=1,45)
      RETURN
      END
C
C
C
      SUBROUTINE STRESS(IPTCR,DEDISP,ICO,JX,X,Y,INCR,ITER,THICK,THETA,
     +                  ICR,TOTST,IBP,DSTL,IPTYL,BSTEL,ISO,SX,SY)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION BSTORE(3,20,45),DSTORE(3,3,45)
      DIMENSION IPTCR(46,1),DEDISP(1),Y(1),ICO(6,1),JX(1),ISO(6,1),
     +          THICK(45,1),THETA(45,1),X(1),ICR(46,1),SX(1),SY(1),
     +          TOTST(26,1),DSTL(3,1),IPTYL(3,1),IBP(1),BSTEL(6,3,200)
      COMMON/CSTIF/LJ(68),P(68),STIF(68,68),DISP(20,1)
      COMMON/CSIGS/SIG(3,45),EPS(3,45),BMATX(3,20),D(3,3),BSTORE,DSTORE
      COMMON/CSTRES/DSIG(3,1),DEPS(3,1),TEPS(3,45),TSIG(3,45)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      COMMON/CTAPE/MTAPE,NTAPE,LTAPE,KTAPE
      COMMON/CONST/E,E1,E2,ANU,BETA,FCU,FTU,ET,EP,YL,YLP,EPSU,ECX,ECY
      WRITE(6,102) INCR,ITER
      DO 10 IEL=1,NEL
      DO 10 IPT=1,46
   10 ICR(IPT,IEL)=0
      DO 1 IEL=1,NEL
        CALL CURRENT(IEL,NNODEL,ICO,JX,AA,BB,NVAR,LJ,X,Y,IS,IB,NEL)
        READ(NTAPE) ((SIG(L,IPT),L=1,3),IPT=1,45),
     +              ((EPS(L,IPT),L=1,3),IPT=1,45),
     +              (((DSTORE(M,L,IPT),M=1,3),L=1,3),IPT=1,45)
        READ(3)     (((BSTORE(M,L,IPT),M=1,3),L=1,20),IPT=1,45)
        DO 2 I=1,NVEL
        IF(LJ(I).GT.0) DISP(I,1)=DEDISP(LJ(I))
    2   IF(LJ(I).EQ.0) DISP(I,1)=0.D0
        ICRACK=0
        DO 3 IPT=1,45
        CALL STORE(BMATX,BSTORE,3,20,IPT,0)
        CALL MULT (BMATX,DISP,DEPS,3,20,1)
        CALL STORE(D,DSTORE,3,3,IPT,0)
        CALL MULT (D,DEPS,DSIG,3,3,1)
        DO 5 L=1,3
        EPS(L,IPT)=EPS(L,IPT)+DEPS(L,1)
    5   SIG(L,IPT)=SIG(L,IPT)+DSIG(L,1)
        IF(ITER.EQ.1)THEN
          IF(IPTCR(IPT,IEL).LT.5) CALL DMATX(EPS,SIG,IPT,IEL,ICRACK,
     +                                       ICR,D,THICK,THETA,IPTCR)
          IF(IPTCR(IPT,IEL).EQ.5)THEN
            CALL PRESET(D,3,3)
            DO 7 L=1,3
    7      SIG(L,IPT)=0.D0
          ENDIF
          IF(IPTCR(IPT,IEL).EQ.0) IPTCR(IPT,IEL)=ICR(IPT,IEL)
          IF(ICR(IPT,IEL).GT.0) CALL ZEROSIG(SIG,IPT,IEL,THETA)
        ENDIF
        CALL STORE(D,DSTORE,3,3,IPT,1)
    3   CONTINUE
        IF(ICRACK.GT.0) IPTCR(46,IEL)=1
!        IF(IEL.EQ.1. AND .ITER.EQ.2) WRITE(6,101)
!        IF(ITER.EQ.2) WRITE(6,100) IEL,(ICR(L,IEL),L=1,45)
        WRITE(MTAPE) ((SIG(L,IPT),L=1,3),IPT=1,45),
     +               ((EPS(L,IPT),L=1,3),IPT=1,45),
     +               (((DSTORE(M,L,IPT),M=1,3),L=1,3),IPT=1,45)
    1 CONTINUE
      IF(ITER.EQ.2) THEN
        DO 6 IEL=1,NEL
        IF(IEL.EQ.1) WRITE(6,101)
    6   WRITE(6,100) IEL,(IPTCR(L,IEL),L=1,45)
      ENDIF
      CALL SSTRESS(DSTL,IPTYL,BSTEL,JX,ISO,SX,SY,ICO,X,Y,DEDISP,
     +             ITER,INCR,TOTST,IBP)
  100 FORMAT(1X,I2,4X,45I2)
  101 FORMAT(//,'  STRESSES AT INTEGRATION POINTS  ',//,
     +       '      0=UNCRACKED       1=TEN-COM      2=COM-COM',
     +       '     3=TEN-TEN     4=SHEAR  ',/,'  IEL',2X,' 1 2 3 4 ',
     +      '5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 ',
     +      '1 2 3 4 5 6 7 8 9 0 1 2 3 4 5',/)
  102 FORMAT(///,1X,'***** INSIDE STRESS *****     JUST SOLVED AX=B',/,
     +       1X,'INCR= ',I4,5X,'ITER=',I4)
      RETURN
      END
C
C
C
      SUBROUTINE ZEROSIG(SIG,IPT,IEL,THETA)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION SIG(3,45),THETA(45,1),TSIG(3,1)
      COMMON/CCOHZ/R(3,3),TEMPFT(3,1)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      SMEAN=(SIG(1,IPT)+SIG(2,IPT))/2.D0
      SROOT=DSQRT(((SIG(1,IPT)-SIG(2,IPT))/2.D0)**2 + SIG(3,IPT)**2)
      SIG1=SMEAN + SROOT
      SIG2=SMEAN - SROOT
      C2=DCOS(THETA(IPT,IEL))*DCOS(THETA(IPT,IEL))
      S2=DSIN(THETA(IPT,IEL))*DSIN(THETA(IPT,IEL))
      SC=DSIN(THETA(IPT,IEL))*DCOS(THETA(IPT,IEL))
      R(1,1)= C2
      R(2,1)= S2
      R(3,1)= SC
      R(1,2)= S2
      R(2,2)= C2
      R(3,2)=-SC
      R(1,3)=-2.D0*SC
      R(2,3)= 2.D0*SC
      R(3,3)= C2-S2
      TEMPFT(1,1)= 0.D0
      TEMPFT(2,1)= SIG2
      TEMPFT(3,1)= 0.D0
      CALL MULT(R,TEMPFT,TSIG,3,3,1)
      DO 1 I=1,3
    1 SIG(I,IPT)=TSIG(I,1)
      RETURN
      END
C
C
C
      SUBROUTINE INCLOAD(INCR,MITER,NITER,DELOAD,FAC,TLOAD,RLOAD,JX,
     +                   TLOADST)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      DIMENSION FAC(1),TLOAD(1),DELOAD(1),RLOAD(1),JX(1),TLOADST(1)
      IF(INCR.EQ.1)THEN
        NITER=1
      ELSEIF(INCR.GT.1)THEN
        IF(INCR.EQ.2)THEN
          DO 2 I=1,NNET
    2     DELOAD(I)=FAC(INCR)*TLOAD(I)
        ELSE
          DO 3 I=1,NNET
    3     DELOAD(I)=FAC(INCR)/FAC(INCR-1)*DELOAD(I)
        ENDIF
        NITER=MITER
        DO 1 I=1,NNET
        TLOAD(I)=TLOAD(I)+DELOAD(I)
        TLOADST(I)=TLOADST(I)+DELOAD(I)
    1   RLOAD(I)=RLOAD(I)+DELOAD(I)
!        WRITE(6,500)INCR
!        CALL OUTPUT(JX,NMAT,NMATT,NNOD,NVAR,RLOAD,TLOAD,0)
      ENDIF
  500 FORMAT(1X,//,100('*'),//
     +       ,' NEW LOAD INCREMENT ',15X,' INCREMENT NO.',I3,
     +       /,' LOAD INCREMENT AND TOTAL LOAD     (RLOAD,TLOAD)',/)
  501 FORMAT(5(3X,E10.4))
      RETURN
      END
C
C
C
      SUBROUTINE UPDATE(FAC,INCR,TLOAD,RLOAD,DEDISP,TDISP,ITER,TP,JX,
     +                  NCONV,MITER,TLOADST,TLOAD2)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION FAC(1),RLOAD(1),TLOAD(1),DEDISP(1),TDISP(1),TP(1),
     +          JX(1),TLOADST(1),TLOAD2(1)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      COMMON/CTAPE/MTAPE,NTAPE,LTAPE,KTAPE
      IF(INCR.EQ.1)THEN
        DO 1 I=1,NNETT
        TDISP(I)=TDISP(I)+DEDISP(I)
        TLOAD(I)=TLOAD2(I)
    1   TLOADST(I)=TP(I)
        WRITE(6,104)                     ! ELASTIC SOLUTION
        CALL OUTPUT(JX,NMAT,NMATT,NNOD,NVAR,TLOAD,TDISP,0)
      ELSE
        CALL ADD(TP,TLOADST,RLOAD,-1,1,NNET)
!        IF(ITER.EQ.1) THEN
!        WRITE(6,102) INCR,ITER         ! RESIDUAL LOAD AND DISPLACEMENT
!        CALL OUTPUT(JX,NMAT,NMATT,NNOD,NVAR,RLOAD,DEDISP,0)
!        ENDIF
        DO 2 I=1,NNETT
    2   TDISP(I)=TDISP(I)+DEDISP(I)
        IF(ITER.EQ.MITER .OR. NCONV.EQ.1)THEN
          WRITE(6,103) INCR,ITER        ! TOTAL LOAD AND DISPLACEMENTS
          CALL OUTPUT(JX,NMAT,NMATT,NNOD,NVAR,TLOAD,TDISP,0)
        ENDIF
      ENDIF
      CALL TAPE(MTAPE,NTAPE,LTAPE,KTAPE)
  102 FORMAT(///,1X,'***** INSIDE UPDATE *****     RESIDUAL LOAD AND',
     +       ' DISPLACEMENTS ',/,5X,'FOR INCREMENT NO.',I3,10X,
     +       'ITERATION NO.',I3,'   (RLOAD,DEDISP) ',//)
  103 FORMAT(1X,/, ' TOTAL LOAD AND TOTAL DISPLACEMENTS ',/,
     +       5X,'FOR INCREMENT NO.',I3,10X,'ITERATION NO.',I3,
     +       5X,' (TLOAD,TDISP)',/)
  104 FORMAT(//,5X,'***** INSIDE UPDATE *****     ELASTIC SOLUTION',/
     +       ,5X,' (TLOAD,TDISP) ',/)
      RETURN
      END
C
```

```
C
C
      SUBROUTINE ADD(A,B,C,NA,NB,M)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION A(M),B(M),C(M)
      DO 1 I=1,M
    1 C(I)=NA*A(I)+NB*B(I)
      RETURN
      END
C
C
C
      SUBROUTINE CLOAD(P,AA,BB,IEL,IB,NVELT)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION P(1),FNODE(12)
      IF(IB.EQ.4)RETURN
      CALL PSET(P,NVELT)
      IF(IB.EQ.0)RETURN
      WRITE(6,154)
      IF(IB.EQ.1)THEN
        READ (5,*) Q
        WRITE(6,150) IEL,Q
        CONST=AA*BB*Q
        P( 3)= CONST/4.D0
        P( 4)= AA*CONST/24.D0
        P( 5)= BB*CONST/24.D0
        P( 8)= P(3)
        P( 9)=-P(4)
        P(10)= P(5)
        P(13)= P(3)
        P(14)=-P(4)
        P(15)=-P(5)
        P(18)= P(3)
        P(19)= P(4)
        P(20)=-P(5)
      RETURN
      ELSEIF(IB.EQ.2)THEN
        READ (5,*) (FNODE(I),I=1,12)
        WRITE(6,151) IEL,(FNODE(I),I=1,12)
        K=1
        DO 1 I=3,20,5
        P(I)  =P(I)  +FNODE(K)
        P(I+1)=P(I+1)+FNODE(K+1)
        P(I+2)=P(I+2)+FNODE(K+2)
    1   K=K+3
      RETURN
      ELSEIF(IB.EQ.3)THEN
        READ (5,*) TX1,TY1,TZ1,TM1,TX2,TY2,TZ2,TM2,TX3,TY3,TZ3,TM3,
     +                 TX4,TY4,TZ4,TM4
        WRITE(6,152) IEL
        WRITE(6,153) TX1,TY1,TZ1,TM1,TX2,TY2,TZ2,TM2,TX3,TY3,TZ3,TM3,
     +               TX4,TY4,TZ4,TM4
        P( 1)= AA*TX1/2.D0+BB*TX4/2.D0
        P( 2)= AA*TY1/2.D0+BB*TY4/2.D0
        P( 3)= AA*TZ1/2.D0+BB*TZ4/2.D0
        P( 4)= (BB*BB/12.D0)*(TM2/AA+TM4/AA+TZ4)
        P( 5)=-AA*AA*TZ1/12.D0+BB*TM4/2.D0
        P( 6)= AA*TX1/2.D0+BB*TX2/2.D0
        P( 7)= AA*TY1/2.D0+BB*TY2/2.D0
        P( 8)= AA*TZ1/2.D0+BB*TZ2/2.D0
        P( 9)= (BB*BB/12.D0)*(TZ2-TM2/AA-TM4/AA)
        P(10)= AA*AA*TZ1/12.D0+BB*TM2/2.D0
        P(11)= BB*TX2/2.D0+AA*TX3/2.D0
        P(12)= BB*TY2/2.D0+AA*TY3/2.D0
        P(13)= BB*TZ2/2.D0+AA*TZ3/2.D0
        P(14)= (BB*BB/12.D0)*(-TZ2+TM2/AA+TM4/AA)
        P(15)= AA*AA*TZ3/12.D0+BB*TM2/2.D0
        P(16)= AA*TX3/2.D0+BB*TX4/2.D0
        P(17)= AA*TY3/2.D0+BB*TY4/2.D0
        P(18)= AA*TZ3/2.D0+BB*TZ4/2.D0
        P(19)= (BB*BB/12.D0)*(-TZ4-TM2/AA-TM4/AA)
        P(20)=-AA*AA*TZ3/12.D0+BB*TM4/2.D0
      RETURN
      ENDIF
  150 FORMAT(/,1X,' ELEMENT NO.=',I5,'    GRAVITY LOAD =',F10.1,//)
  151 FORMAT(/,1X,' ELEMENT NO.=',I5,'    NODAL FORCES =',/,12F10.1,//)
  152 FORMAT(/,1X,' ELEMENT NO.=',I5,//)
  153 FORMAT(/,1X,' TX1,TY1,TZ1,TM1 ',4F10.1,//)
  154 FORMAT(//,1X,' LOADING CONDITIONS ')
      RETURN
      END
C
C
C
      SUBROUTINE COLHT(NVELT,LJ,MHT)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION LJ(NVELT),MHT(1)
      LS=10000000
      DO 14 M=1,NVELT
      IF(LJ(M))17,14,17
   17 IF(LJ(M)-LS)18,14,14
   18 LS=LJ(M)
   14 CONTINUE
      DO 15 N=1,NVELT
      II=LJ(N)
      IF(II.EQ.0)GO TO 15
      ME=II-LS
      IF(ME.GT.MHT(II)) MHT(II)=ME
   15 CONTINUE
      RETURN
      END
C
```

```
C
C
      SUBROUTINE COLSOL(AA,VV,DEDISP,MAXA,NN,NWK,KKK,A,V)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION A(1),V(1),MAXA(1),AA(1),VV(1),DEDISP(1)
      DO 1 I=1,NWK
    1 A(I)=AA(I)
      DO 2 I=1,NN
    2 V(I)=VV(I)
      NNM=NN+1
C     PERFORM L*D*L(T) FACTORIZATION OF STIFFNESS MATRIX
      IF(KKK-2)40,150,150
   40 DO 140 N=1,NN
      KN=MAXA(N)
      KL=KN+1
      KU=MAXA(N+1) - 1
      KH=KU - KL
      IF(KH)110,90,50
   50 K=N-KH
      IC=0
      KLT=KU
      DO 80 J=1,KH
      IC=IC + 1
      KLT=KLT - 1
      KI=MAXA(K)
      ND=MAXA(K+1) - KI - 1
      IF(ND)80,80,60
   60 KK=MIN0(IC,ND)
      C=0.D0
      DO 70 L=1,KK
   70 C=C+A(KI+L)*A(KLT+L)
      A(KLT)=A(KLT) - C
   80 K=K+1
   90 K=N
      B=0.D0
      DO 100 KK=KL,KU
      K=K - 1
      KI=MAXA(K)
      C=A(KK)/A(KI)
      B=B + C*A(KK)
  100 A(KK)=C
      A(KN)=A(KN) - B
  110 IF (A(KN))120,120,140
  120 WRITE(6,2000) N,A(KN)
      STOP
  140 CONTINUE
C     REDUCE RIGHT-HAND-SIDE LOAD VECTOR
  150 DO 180 N=1,NN
      KL=MAXA(N) + 1
      KU=MAXA(N+1) - 1
      IF(KU-KL)180,160,160
  160 K=N
      C=0.D0
      DO 170 KK=KL,KU
      K=K - 1
  170 C=C+A(KK)*V(K)
      V(N)=V(N) - C
  180 CONTINUE
C     BACK-SUBSTITUTE
      DO 200 N=1,NN
      K=MAXA(N)
  200 V(N)=V(N)/A(K)
      IF (NN.EQ.1) RETURN
      N=NN
      DO 230 L=2,NN
      KL=MAXA(N) + 1
      KU=MAXA(N+1) - 1
      IF(KU-KL)230,210,210
  210 K=N
      DO 220 KK=KL,KU
      K=K - 1
  220 V(K)=V(K)-A(KK)*V(N)
  230 N=N-1
      DO 3 I=1,NN
    3 DEDISP(I)=V(I)
 2000 FORMAT(//,5X,'STOP-STIFFNESS MATRIX NOT POSITIVE DEFINITE',//,
     +       5X,'NONPOSITIVE PIVOT FOR EQUATION',I5,//,5X,'PIVOT =',
     +       E20.12)
      RETURN
      END
C
C
C
      SUBROUTINE CONVGE(TSTIF2,MDIA,DET,IEXP,TOLER,INCR,NCONV,ITER)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION TSTIF2(1),MDIA(1)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      DETPRE=DET
      IEXPRE=IEXP
      IEXP=0
      IDIFF=0
      CST=1.D0
      DET=1.D0
      DO 1 I=1,NNETT
      DET=DET*TSTIF2(MDIA(I))
      IF(ABS(DET).GE.1.D+15)THEN
        DET=DET*1.D-15
        IEXP=IEXP+15
      ENDIF
    1 CONTINUE
      IF(INCR.GT.1)THEN
```

```
        IF(IEXP.NE.IEXPRE)THEN
          IDIFF=ABS(IEXP-IEXPRE)
          IF(IDIFF.GT.15)THEN
            MDIFF=IDIFF/15
            NDIFF=IDIFF-15*MDIFF
            DO 2 I=1,NDIFF
    2       CST=CST*10.D0
          ELSE
            DO 3 I=1,IDIFF
    3       CST=CST*10.D0
          ENDIF
          IF(IEXP.GT.IEXPRE)THEN
            IF(IDIFF.GT.15) IEXP=IEXP-NDIFF
            IF(IDIFF.LE.15) IEXP=IEXP-IDIFF
            DET=DET*CST
          ELSEIF(IEXP.LT.IEXPRE)THEN
            IF(IDIFF.GT.15) IEXPRE=IEXPRE-NDIFF
            IF(IDIFF.LE.15) IEXPRE=IEXPRE-IDIFF
            DETPRE=DETPRE*CST
          ENDIF
        ENDIF
        WRITE(6,100) DETPRE,IEXPRE,DET,IEXP
        IF(IDIFF.GT.15) GO TO 10
        CONST=ABS((DET-DETPRE)/DETPRE*100.D0)
        IF(ITER.EQ.1)THEN
          WRITE(6,103)                          ! FORCING 2ND
   10     NCONV=0
        ELSE
          IF(CONST.LT.TOLER )THEN
            NCONV=1
            WRITE(6,101) CONST,TOLER              ! CONVERGED
          ELSE
            NCONV=0
            WRITE(6,102) CONST,TOLER              ! NOT YET
          ENDIF
        ENDIF
      ENDIF
  100 FORMAT(//,5X,' ***** INSIDE CONVGE ***** ',/,
     +        /,5X,' PREVIOUS DETERMINATE  :',E23.15,' X10',I4,
     +        /,5X,' PRESENT  DETERMINATE  :',E23.15,' X10',I4)
  101 FORMAT(/,5X,' SOLUTION HAS CONVERGED ',
     +       /,5X,' CONSTANT IS         : ',E23.15,
     +       /,5X,' TOLERANCE CRITERIA : ',E23.15,/)
  102 FORMAT(/,5X,' NO CONVERGENCE YET ',
     +      //,5X,' CONSTANT IS         : ',E23.15,
     +       /,5X,' TOLERANCE CRITERIA : ',E23.15,/)
  103 FORMAT(/,5X,' FORCING SECOND ITERATION - CHANGING STIFFNESS',
     +             ' FOR CRACKING',/)
      RETURN
      END
C
```

```
C
C
      SUBROUTINE CURRENT(IEL,NNODEL,ICO,JX,AA,BB,NVAR,LJ,X,Y,IS,IB,NEL)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION ICO(6,1),X(1),Y(1),JX(1),LJ(68),XX(4),YY(4)
      DO 1 J=1,NNODEL
      J1=(J-1)*NVAR
      J2=NVAR*(ICO(J,IEL)-1)
      XX(J)=X(ICO(J,IEL))
      YY(J)=Y(ICO(J,IEL))
      DO 1 I=1,NVAR
    1 LJ(I+J1)=JX(J2+I)
      AA=ABS(XX(3)-XX(1))
      BB=ABS(YY(3)-YY(1))
      IS=ICO(NNODEL+1,IEL)
      IB=ICO(NNODEL+2,IEL)
      RETURN
      END
C
C
C
      SUBROUTINE DIAADD(NNET,NVA,LBAND,MHT,MDIA,NNET1)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION MHT(1),MDIA(1)
      NNET1=NNET+1
      DO 20 I=1,NNET1
   20 MDIA(I)=0
      MDIA(1)=1
      MDIA(2)=2
      LBAND=0
      IF(NNET.EQ.1)GO TO 21
        DO 16 I=2,NNET
        IF(MHT(I).GT.LBAND) LBAND=MHT(I)
   16   MDIA(I+1)=MDIA(I)+MHT(I)+1
   21 LBAND=LBAND+1
      NVA=MDIA(NNET1)-MDIA(1)
      RETURN
      END
C
C
C
      SUBROUTINE ELASTIC(ICO,JX,X,Y,DEDISP,NPRINT,TSTIF2,RLOAD2,THICK,
     +                   RLOAD,INCR,ITER,TSTIF,TP,MDIA,NVA,NNET1,THETA,
     +                   ISO,SX,SY,TOTST,IBP,DSTL,BSTEL)
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
      DIMENSION BSTORE(3,20,45),DSTORE(3,3,45)
      DIMENSION THICK(45,1),TSTIF(1),TP(1),DEDISP(1),MDIA(1),
     +          X(1),Y(1),ICO(6,1),JX(1),RLOAD(1),TSTIF2(1),
     +          THETA(45,1),RLOAD2(1),TOTST(26,1),
     +          ISO(6,1),SX(1),SY(1),DSTL(3,1),IBP(1),BSTEL(6,3,200)
      COMMON/CSTIF/LJ(68),P(68),STIF(68,68),DISP(20,1)
      COMMON/CSIGS/SIG(3,45),EPS(3,45),BMATX(3,20),D(3,3),BSTORE,DSTORE
      COMMON/CSTSP/STSIG(3),STEPS(3),STIFS(6,6),SPSTIF(26,26),
     +            STLOAD(6,1),SPLOAD(26,1),TRLOAD(23,1),TRSTIF(26,26),
     +            SUSH(26),SWSH(26),UCSH(4),USSH(6),WSSH(4)
      COMMON/CELAST/TEPS1(3,1),TSIG1(3,1),BTEMP1(1,6),DTDISP(26,1),
     +             STDISP(6,1),CNDISP(10,1)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +            NMATT,NVELT,NNET,NNETT
      COMMON/CONST/E,E1,E2,ANU,BETA,FCU,FTU,ET,EP,YL,YLP,EPSU,ECX,ECY
      IF(INCR.GT.1)RETURN
      CALL COLSOL(TSTIF,TP,DEDISP,MDIA,NNETT,NVA,1,TSTIF2,RLOAD2)
      WRITE(6,100)
      REWIND 3
      CALL PRESET(TOTST,26,NELS)
      DO 1 IEL=1,NEL
       READ(3)(((BSTORE(M,L,IPT),M=1,3),L=1,20),IPT=1,45)
       CALL CURRENT(IEL,NNODEL,ICO,JX,AA,BB,NVAR,LJ,X,Y,IS,IB,NEL)
       DO 2 J=1,NVEL
       IF(LJ(J).LT.0 .OR. LJ(J).GT.0) DISP(J,1)=DEDISP(LJ(J))
    2  IF(LJ(J).EQ.0) DISP(J,1)=0.D0
       DO 3 IPT=1,45
       CALL STORE(BMATX,BSTORE,3,20,IPT,0)
       CALL STORE(D,DSTORE,3,3,IPT,1)
       CALL MULT (BMATX,DISP,TEPS1,3,20,1)
       CALL MULT (D,TEPS1,TSIG1,3,3,1)
       DO 4 L=1,3
       EPS(L,IPT)=TEPS1(L,1)
    4  SIG(L,IPT)=TSIG1(L,1)
    3  CONTINUE
       IF(NPRINT.EQ.2) CALL OUTSIG(SIG,EPS,IEL,INCR,ITER,THICK,THETA)
       WRITE(1) ((SIG(L,IPT),L=1,3),IPT=1,45),
     +           ((EPS(L,IPT),L=1,3),IPT=1,45),
     +           (((DSTORE(M,L,IPT),M=1,3),L=1,3),IPT=1,45)
*
      L=0
      DO 5 IELS=1,NELS
       IF(ISO(5,IELS).NE.IEL) GO TO 5
       ID=ISO(6,IELS)
       L=L+1
       IF(ID.EQ.2)THEN                                    ! VERTICAL
         XL=ABS(SY(ISO(2,IELS))-SY(ISO(1,IELS)))
         ECC=ECY
       ELSE                                               ! HORIZONTAL
         XL=ABS(SX(ISO(2,IELS))-SX(ISO(1,IELS)))
         ECC=ECX
       ENDIF
       CALL TRNDISP(ICO,ISO,IELS,SX,SY,X,Y,XL,ID,IBP,TOTST,
     +               DEDISP,L,JX)
       DO 6 I=1,6
    6  STDISP(I,1)=DTDISP(NVEL+I,1)
       DO 7 I=1,3
```

```
        DO 8 J=1,6
    8   BTEMP1(1,J)=BSTEL(J,I,IELS)
        CALL MULT(BTEMP1,STDISP,TSEPS,1,6,1)
        STEPS(I)=TSEPS
        STSIG(I)=DSTL(I,IELS)*STEPS(I)
    7   CONTINUE
      WRITE(7) (STSIG(I),I=1,3),(STEPS(I),I=1,3)
    5 CONTINUE
    1 CONTINUE
  100 FORMAT(1X,'***** INSIDE ELASTIC *****',/,
     +       1X,'INNITIAL STRESSES        JUST SOLVED AX=B')
      RETURN
      END
C
C
C
      SUBROUTINE INPUT(NINCR,MITER,FAC,TOLER,NPRINT,SKUIN,SLIPM,
     +                 SKWIN,NSP)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION FAC(1),ICON(10),CON(10),TITLE(9)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      COMMON/CONST/E,E1,E2,ANU,BETA,FCU,FTU,ET,EP,YL,YLP,EPSU,ECX,ECY
      WRITE(6,150)
      READ (5,100) TITLE
      WRITE(6,100) TITLE
      READ (5,*) NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL
      WRITE(6,151) NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL
      READ (5,*) NINCR,MITER,NPRINT
      WRITE(6,152) NINCR,MITER,NPRINT
      READ (5,*) (FAC(I),I=1,NINCR)
      DO 1 I=1,NINCR
    1 WRITE(6,153) I,FAC(I)
      READ (5,*) E,E1,E2,BETA
      READ (5,*) FCU,FTU,ANU,EPSU
      WRITE(6,154) E,E1,E2,ANU,FCU,FTU,BETA,EPSU
      READ (5,*) ET,EP,YL,YLP,TOLER,ECX,ECY
      WRITE(6,155) ET,EP,YL,YLP,TOLER,ECX,ECY
      READ(5,*) SKUIN,SKWIN,SLIPM
      WRITE(6,200) SKUIN,SKWIN,SLIPM
      READ(5,*) NSP
      WRITE(6,156) NSP
      NMAT=NNOD*NVAR
      NVEL=NVAR*NNODEL
      NMATT=NMAT+NNODS*3
      NVELT=NVEL+NNODSEL*3
  100 FORMAT(9A8)
  150 FORMAT(//,10X,'*****************************************',
     +        /,10X,'*****  FINITE ELEMENT PLATE PROGRAM *****',
     +        /,10X,'*****************************************',//)
  151 FORMAT(//,5X,'TOTAL NO. OF CONCRETE ELEMENTS    NEL    =',I5,
     +        /,5X,'TOTAL NO. OF STEEL ELEMENTS       NELS   =',I5,
     +        /,5X,'NO. OF CONCRETE NODES             NNOD   =',I5,
     +        /,5X,'NO. OF STEEL NODES                NNODS  =',I5,
     +        /,5X,'VARIABLES PER NODE                NVAR   =',I5,
     +        /,5X,'NO. OF NODES PER ELEM.            NNODEL =',I5,
     +        /,5X,'NO. OF STEEL NODES PER ELEM.      NNODSEL=',I5,//)
  152 FORMAT(   5X,'NO. OF LOAD INCREMENTS            NINCR  =',I5,
     +        /,5X,'NO. OF ITERATIONS                 MITER  =',I5,
     +        /,5X,'COUNTER FOR STRESS OUTPUT         NPRINT =',I5,//)
  153 FORMAT(   5X,' LOAD INCREMENT NO. ',I5,'      FACTOR = ',F10.3)
  154 FORMAT(//,5X,'MODULUS OF ELASTICITY             E       = ',F10.1,
     +        /,5X,'TANGENT MODULUS                   E1      = ',F10.1,
     +        /,5X,'TANGENT MODULUS                   E2      = ',F10.1,
     +        /,5X,'POISSION RATIO                    ANU     = ',F10.1,
     +        /,5X,'COMPRESSIVE STRENGTH              FCU     = ',F10.2,
     +        /,5X,'TENSILE STRENGTH                  FTU     = ',F10.2,
     +        /,5X,'FACTOR FOR SHEAR TERM             BETA    = ',F10.1,
     +        /,5X,'STRAIN AT PEAK UNIAXIAL STRESS    EPSU    = ',F10.7)
  155 FORMAT(//,5X,'MODULUS FOR STEEL                 ET      = ',F10.1,
     +        /,5X,'MODULUS FOR PRESTRESSED STEEL     EP      = ',F10.1,
     +        /,5X,'YIELD STRESS OF STEEL             YL      = ',F10.1,
     +        /,5X,'YIELD STRESS OF PRESTRESSED STEEL YLP     = ',F10.1,
     +        /,5X,'TOLERANCE FOR CONVERGENCE         TOLER   = ',F10.1,
     +        /,5X,'STEEL ECCENTRICITY X DIRECTION    ECX     = ',F10.7,
     +        /,5X,'STEEL ECCENTRICITY Y DIRECTION    ECY     = ',F10.7,
     +        //)
  156 FORMAT(/,5X,'NO. OF BOLTS(SPRING)               NSP     = ',I5,/)
  200 FORMAT(/,5X,'INITIAL VALUE OF SKU        SKUIN=',F12.1,
     +       /,5X,'INITIAL VALUE OF SKW        SKWIN=',F12.1,
     +       /,5X,'MAXIMUM SLIP                SLIPM=',F12.5,/)
      RETURN
      END
C
C
C
      SUBROUTINE LAYOUT(X,Y,ICO,IX,JX,MHT,LJ,MDIA,NVA,THICK,NNET1,
     +                 AREA,SX,SY,ISO,IBP,XK,ND,NSP)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION X(1),Y(1),ICO(6,1),IX(1),JX(1),THICK(45,1),MHT(1),
     +          MDIA(1),AREA(1),SX(1),SY(1),ISO(6,1),LJ(68),
     +          IBP(1),XK(1),ND(1)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      NNN=NNODEL+2
      WRITE(6,150)
      DO 10 I=1,NNOD
      I2=NVAR*I
      I1=I2-NVAR+1
      READ (5,*)   X(I),Y(I),(IX(J),J=I1,I2)
      WRITE(6,151) I,X(I),Y(I),(IX(J),J=I1,I2)
   10 CONTINUE
```

```
    WRITE(6,160)
    DO 11 I=1,NNODS
    I2=3*I
    I1=I2-3+1
    J1=I1+NMAT
    J2=I2+NMAT
    READ(5,*) SX(I),SY(I),(IX(J),J=J1,J2),IBP(I)
    WRITE(6,161) I,SX(I),SY(I),(IX(J),J=J1,J2),IBP(I)
 11 CONTINUE
    NNETT=0
    DO 12 I=1,NMATT
    IF(I.EQ.NMAT+1) NNET=NNETT
    IF(IX(I)) 1,2,3
  3 IF(IX(I)-1) 80,80,81
 81 NNETT=NNETT+IX(I)
    GO TO 82
 80 NNETT=NNETT+1
 82 JX(I)=NNETT
    GO TO 12
  1 NNETT=NNETT+IX(I)+1
    JX(I)=NNETT
    GO TO 12
  2 JX(I)=0
 12 CONTINUE
    WRITE(6,152)
    DO 19 I=1,NNETT
 19 MHT(I)=0
    DO 18 I=1,NEL
    READ(5,*)    (ICO(J,I),J=1,NNN),THICK(1,I)
 18 WRITE(6,153) I,(ICO(J,I),J=1,NNN),THICK(1,I)
    WRITE(6,162)
    DO 20 J=1,NELS
    READ(5,*) (ISO(I,J),I=1,6),AREA(J)
 20 WRITE(6,163) J,(ISO(I,J),I=1,6),AREA(J)
    DO 21 I=1,NEL
    DO 13 K=1,NNODEL
    J1=(K-1)*NVAR
    J2=NVAR*(ICO(K,I)-1)
    DO 13 L=1,NVAR
 13 LJ(L+J1)=JX(J2+L)
    L=0
    DO 22 J=1,NELS
    IF(ISO(5,J).NE.I) GO TO 22
    DO 22 K=1,2
    DO 22 KK=1,3
    L=L+1
    K1=(ISO(K,J)-1)*3+KK+NMAT
    LJ(NVEL+L)=JX(K1)
 22 CONTINUE
    CALL COLHT(NVELT,LJ,MHT)
 21 CONTINUE
```

```
      CALL DIAADD(NNETT,NVA,LBAND,MHT,MDIA,NNET1)
      WRITE(6,154) NNET,NNETT,LBAND,NVA,NMAT,NVELT,NMATT
      WRITE(6,155)
      DO 16 I=1,NSP
      READ(5,*) ND(I),XK(I)
   16 WRITE(6,156) ND(I),XK(I)
  150 FORMAT(/,4X,'NODE',6X,' X-CORD',6X,'Y-CORD',8X,'U',3X,'V',3X,
     +       'W',3X,'WX',2X,'WY',/)
  151 FORMAT(1X,I5,5X,F10.3,2X,F10.3,5X,6I4)
  152 FORMAT(//,5X,'CONCRETE ',
     +       /,5X,'ELEMENT',7X,'NODE NUMBERS',4X,'   IS','   IB','
     +        THICKNESS',/)
  153 FORMAT(5X,I5,6X,4I4,2X,2I5,5X,F10.3)
  154 FORMAT(//,5X,' NNET   = ',I5,
     +        /,5X,' NNETT  = ',I5,
     +        /,5X,' LBAND  = ',I5,
     +        /,5X,' NVA    = ',I5,
     +        /,5X,' NMAT   = ',I5,
     +        /,5X,' NVELT  = ',I5,
     +        /,5X,' NMATT  = ',I5,//)
  155 FORMAT(//,5X,'NODES      STIFFNESS OF SPRING')
  156 FORMAT(3X,I5,12X,F10.3)
  160 FORMAT(//,5X,'STEEL',
     +       /,5X,'NODES',6X,'COORD-X',7X,'COORD-Y',7X,'U'4X,
     +       'W',4X,'WX',7X,'IBP',/)
  161 FORMAT(5X,I4,4X,F10.3,4X,F10.3,3X,3(I5),5X,I5)
  162 FORMAT(//,5X,'STEEL ELEM.',4X,
     +         'ISO1   ISO2    IS     IB     IC     ID     AREA ',/)
  163 FORMAT(5X,I4,6X,6I7,F10.3)
      RETURN
      END
C
C
C
      SUBROUTINE MATRIX(S,T,R,BMATX,AA,BB,THICK,IEL,IPT,NEL)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION BMATX(3,20),THICK(45,1)
      CALL PRESET(BMATX,3,20)
      BMATX(1, 1)=-(1.D0-T)/(2.D0*AA)
      BMATX(1, 3)=-3.D0*S*R*(1.D0-T)/(2.D0*AA*AA)*THICK(IPT,IEL)
      BMATX(1, 4)=-(1.D0-T)*(3.D0*S-1.D0)*R/(4.D0*AA)*THICK(IPT,IEL)
      BMATX(1, 6)=-BMATX(1,1)
      BMATX(1, 8)=-BMATX(1,3)
      BMATX(1, 9)= BMATX(1,4)*(3.D0*S+1)/(3.D0*S-1)
      BMATX(1,11)= (1.D0+T)/(2.D0*AA)
      BMATX(1,13)=-BMATX(1,3)*(1.D0+T)/(1.D0-T)
      BMATX(1,14)= BMATX(1,9)*(1.D0+T)/(1.D0-T)
      BMATX(1,16)=-BMATX(1,11)
      BMATX(1,18)=-BMATX(1,13)
      BMATX(1,19)= BMATX(1,4)*(1.D0+T)/(1.D0-T)
      BMATX(2, 2)=-(1.D0-S)/(2.D0*BB)
      BMATX(2, 3)=-3.DO*T*R*(1.DO-S)/(2.DO*BB*BB)*THICK(IPT,IEL)
      BMATX(2, 5)=-(1.DO-S)*(3.DO*T-1.DO)*R/(4.DO*BB)*THICK(IPT,IEL)
      BMATX(2, 7)=-(1.DO+S)/(2.DO*BB)
      BMATX(2, 8)= BMATX(2,3)*(1.DO+S)/(1.DO-S)
      BMATX(2,10)= BMATX(2,5)*(1.DO+S)/(1.DO-S)
      BMATX(2,12)= (1.DO+S)/(2.DO*BB)
      BMATX(2,13)=-BMATX(2,3)*(1.DO+S)/(1.DO-S)
      BMATX(2,15)= BMATX(2,10)*(3.DO*T+1.DO)/(3.DO*T-1.DO)
      BMATX(2,17)=-BMATX(2,2)
      BMATX(2,18)=-BMATX(2,3)
      BMATX(2,20)= BMATX(2,15)*(1.DO-S)/(1.DO+S)
      BMATX(3, 1)= BMATX(2,2)
      BMATX(3, 2)= BMATX(1,1)
      BMATX(3, 3)=-(4.DO-3.DO*S*S-3.DO*T*T)*R/(2.DO*AA*BB)*
     +             THICK(IPT,IEL)
      BMATX(3, 4)=-(1.DO-S)*(1.DO+3.DO*S)*R/(4.DO*BB)*THICK(IPT,IEL)
      BMATX(3, 5)=-(1.DO-T)*(1.DO+3.DO*T)*R/(4.DO*AA)*THICK(IPT,IEL)
      BMATX(3, 6)= BMATX(2,7)
      BMATX(3, 7)= BMATX(1,6)
      BMATX(3, 8)=-BMATX(3,3)
      BMATX(3, 9)=-(1.DO+S)*(1.DO-3.DO*S)*R/(4.DO*BB)*THICK(IPT,IEL)
      BMATX(3,10)=-BMATX(3,5)
      BMATX(3,11)= BMATX(2,12)
      BMATX(3,12)= (1.DO+T)/(2.DO*AA)
      BMATX(3,13)= BMATX(3,3)
      BMATX(3,14)=-BMATX(3,9)
      BMATX(3,15)= (1.DO+T)*(1.DO-3.DO*T)*R/(4.DO*AA)*THICK(IPT,IEL)
      BMATX(3,16)=-BMATX(3,1)
      BMATX(3,17)= BMATX(1,16)
      BMATX(3,18)= BMATX(3,8)
      BMATX(3,19)=-BMATX(3,4)
      BMATX(3,20)=-BMATX(3,15)
      RETURN
      END
C
C
C
      SUBROUTINE MULT(A,B,C,M,N,L)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION A(M,N),B(N,L),C(M,L)
      DO 4 I=1,M
      DO 5 J=1,L
      X=0.DO
      DO 6 K=1,N
    6 X=X+A(I,K)*B(K,J)
    5 C(I,J)=X
    4 CONTINUE
      RETURN
      END
C
C
```

```
C
      SUBROUTINE MULT2(A,B,C,M,N,Z)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION A(M,M),B(M,N),C(N,N),Z(M,N)
      DO 1 I=1,M
      DO 2 K=1,N
      X=0.D0
      DO 3 J=1,M
    3 X=X + A(I,J)*B(J,K)
    2 Z(I,K)=X
    1 CONTINUE
      DO 4 I=1,N
      DO 5 K=I,N
      X=0.D0
      DO 6 J=1,M
    6 X=X + B(J,I)*Z(J,K)
      C(I,K)=X
    5 C(K,I)=X
    4 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE OUTSIG(SIG,EPS,IEL,INCR,ITER,THICK,THETA)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION SIG(3,45),EPS(3,45),THICK(45,1),THETA(45,1)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      WRITE(6,100) INCR,ITER,IEL   ! INCREMENT,ITERATION,ELEMENT NO.
      WRITE(6,101)
      DO 1 I=1,45
      THE=THETA(I,IEL)*180.D0/3.141592654
    1 WRITE(6,102) I,SIG(1,I),SIG(2,I),SIG(3,I),EPS(1,I),EPS(2,I),
     +             EPS(3,I),THICK(I,IEL),THE
  100 FORMAT(1X,//,' INCREMENT NO.',I5,
     +       '         ITERATION NO.',I5,
     +       '         ELEMENT   NO.',I5,/)
  101 FORMAT(1X,'INTEGRATION PT.    SIG-XX     SIG-YY',7X,
     +       'SIGXY            EXX           EYY          EXY',
     +       '    THICKNESS    ANGLE',/)
  102 FORMAT(3X,I5,6X,3F12.3,4X,3(2X,E10.4),1X,F10.6,2X,F7.2)
      RETURN
      END
C
C
C
      SUBROUTINE OUTPUT(JX,NMAT,NMATT,NNOD,NVAR,FORCE,DIS,N)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION FORCE(1),DIS(1),JX(1),AMODE(1000),BMODE(1000)
!      WRITE(6,100)
```

```
      DO 1 I=1,NMATT
!      AMODE(I)=0.D0
      BMODE(I)=0.D0
      IF(JX(I).EQ.0) GO TO 1
!      AMODE(I)=FORCE(JX(I))
      BMODE(I)=DIS(JX(I))
    1 CONTINUE
!      DO 2 I=1,NNOD
!      I2=NVAR*I
!      I1=I2-NVAR+1
!      WRITE(6,102) I,(AMODE(J),J=I1,I2)
!    2 CONTINUE
      WRITE(6,101)
      DO 3 I=1,NNOD
      I2=NVAR*I
      I1=I2-NVAR+1
      WRITE(6,102) I,(BMODE(J),J=I1,I2)
    3 CONTINUE
      IF(N.EQ.1) THEN
      WRITE(6,103)
      II=NMAT+1
      DO 4 I=II,NMATT
      K=I-NMAT
      WRITE(6,104) K,AMODE(I),BMODE(I)
    4 CONTINUE
      ENDIF
  100 FORMAT(/,' NODE ',10X,'FX',10X,'FY',10X,'FZ',10X,'MX',10X,'MY',/)
  101 FORMAT(/,' NODE ',10X,' U',10X,' V',10X,' W',10X,'WX',10X,'WY',/)
  102 FORMAT(I5,3X,5(3X,E10.4))
  103 FORMAT(/,5X,'NODE OF STEEL',5X,' FORCE '7X,'DISPLACEMENT',/)
  104 FORMAT(5X,I5,5X,E10.4,5X,E10.4)
      RETURN
      END
C
C
C
      SUBROUTINE PSET(A,M)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION A(M)
      DO 1 I=1,M
    1 A(I)=0.D0
      RETURN
      END
C
C
C
      SUBROUTINE PRESET(A,M,N)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION A(M,N)
      DO 1 J=1,N
      DO 1 I=1,M
    1 A(I,J)=0.DO
      RETURN
      END
C
C
C
      SUBROUTINE SETUP(A,B,STIF,P,MDIA,LJ,NVELT)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION A(1),B(1),MDIA(1),STIF(NVELT,NVELT),LJ(NVELT),
     +          P(NVELT)
      DO 200 I=1,NVELT
      LJR=LJ(I)
      IF(LJR) 200,200,100
100   B(LJR)=B(LJR)+P(I)
      DO 220 J=I,NVELT
      LJC=LJ(J)
      IF(LJC) 220,220,110
110   IJ=LJR-LJC
      IF(IJ) 210,215,215
210   MJ=MDIA(LJC)
      IJ=-IJ
      GO TO 216
215   MJ=MDIA(LJR)
216   KK=MJ+IJ
      A(KK)=A(KK)+STIF(I,J)
220   CONTINUE
200   CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE STORE(A,B,M,L,IPT,NUM)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION A(M,L),B(M,L,45)
      IF(NUM.EQ.1)THEN
        DO 1 I=1,L
        DO 1 J=1,M
    1   B(J,I,IPT)=A(J,I)
      ELSE
        DO 2 I=1,L
        DO 2 J=1,M
    2   A(J,I)=B(J,I,IPT)
      ENDIF
      RETURN
      END
C
C
C
      SUBROUTINE TAPE(MTAPE,NTAPE,LTAPE,KTAPE)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      MTAPE=3-MTAPE
      NTAPE=3-NTAPE
      LTAPE=15-LTAPE
      KTAPE=15-KTAPE
      ENTRY REWIND(MTAPE,NTAPE,LTAPE,KTAPE)
      REWIND MTAPE
      REWIND NTAPE
      REWIND LTAPE
      REWIND KTAPE
      REWIND 3
      RETURN
      END
C
C
C
      SUBROUTINE TRANPS(A,AT,M,N)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION A(M,N),AT(N,M)
      DO 1 I=1,M
      DO 2 J=1,N
    2 AT(J,I)=A(I,J)
    1 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE ZERO(TDISP,TLOAD,DELOAD,P,DET,IEXP,RLOAD,IPTCR,
     +                DSIGS,SIGS,DEPSS,EPSS,THICK,DSTL,IPTYL,BSTEL)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION TDISP(1),TLOAD(1),DELOAD(1),P(68),RLOAD(1),DSIGS(1),
     +          SIGS(1),DEPSS(1),EPSS(1),IPTCR(45,1),THICK(45,1),
     +          DSTL(3,1),IPTYL(3,1),BSTEL(6,3,200)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      COMMON/CTAPE/MTAPE,NTAPE,LTAPE,KTAPE
      NTAPE=2
      MTAPE=1
      LTAPE=7
      KTAPE=8
      IEXP=0
      DET=0.D0
      DO 1 IEL=1,NEL
      DO 1 IPT=1,45
      IPTCR(IPT,IEL)=0
    1 THICK(IPT,IEL)=THICK(1,IEL)
      CALL PSET (P,NVELT)
      CALL PSET (TDISP,NNETT)
      CALL PSET (TLOAD,NNETT)
      CALL PSET (RLOAD,NNETT)
      CALL PSET (DELOAD,NNETT)
      CALL PRESET (DSTL,3,NELS)         !CHECK AGAIN
      CALL PSET (DSIGS,NELS)
      CALL PSET (DEPSS,NELS)
      CALL PSET (SIGS,NELS)
      CALL PSET (EPSS,NELS)
      DO 2 I=1,NELS
      DO 2 J=1,3
      IPTYL(J,I)=0
      DO 2 K=1,6
    2 BSTEL(K,J,I)=0.D0
      RETURN
      END
C
C
C
      SUBROUTINE TRANST(ICO,ISO,IELS,SX,SY,X,Y,XL,ID)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION ISO(6,1),SX(1),SY(1),X(1),Y(1),ICO(6,1)
      COMMON/CTRAN/TRAN(10,20),TRAN1(6,23),TRANJ(26,23),TRANT(20,10),
     +             TRAN1T(23,6),C(11)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      CALL PRESET(TRAN,10,20)
      DO 1 I=1,2
      IF(ID.EQ.2)THEN                    ! VERTICAL
        X1= ABS(X(ICO(3,ISO(5,IELS))) - X(ICO(1,ISO(5,IELS))))/2.D0
        X2= SX(ISO(1,IELS)) - X(ICO(1,ISO(5,IELS)))
        S = (X2-X1)/X1
        A = X1*2.D0
        B = XL
        IF(I.EQ.1) T=-1.D0
        IF(I.EQ.2) T= 1.D0
      ELSE                                         ! HORIZONTAL
        Y1= ABS(Y(ICO(3,ISO(5,IELS))) - Y(ICO(1,ISO(5,IELS))))/2.D0
        Y2= SY(ISO(1,IELS)) - Y(ICO(1,ISO(5,IELS)))
        T = (Y2-Y1)/Y1
        A = XL
        B = Y1*2.D0
        IF(I.EQ.1) S=-1.D0
        IF(I.EQ.2) S= 1.D0
      ENDIF
      L=0
      DO 2 J=0,15,5
      L=L+1
      CALL SHAPE(C,S,T,A,B,L)
      DO 3 K=1,2
    3 TRAN((5*I-5)+K,K+J)=C(K)
      DO 4 K=1,3
      TRAN((5*I-2)   ,K+J+2)=C(2+K)
      TRAN((5*I-2)+1,K+J+2)=C(5+K)
    4 TRAN((5*I-2)+2,K+J+2)=C(8+K)
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE SHAPE(C,S,T,A,B,N)
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
      DIMENSION C(1)
      S1=1.D0-S
      S2=1.D0+S
      S3=3.D0*S+1.D0
      T1=1.D0-T
      T2=1.D0+T
      T3=3.D0*T+1.D0
      IF(N.EQ.1)THEN
        C( 1)= 0.25*S1*T1
        C( 2)= 0.25*S1*T1
        C( 3)= 1.-.25*S2*T2-0.125*(2.-S)*T1*S2*S2-0.125*(2.-T)*S1*T2*T2
        C( 4)= 0.0625*A*S2*T1*S1*S1
        C( 5)= 0.0625*B*S1*T2*T1*T1
        C( 6)= 1.D0/(4.D0*A)*(-3.D0+4.D0*T+3.D0*S*S-3.D0*S*S*T-T**3)
        C( 7)=-0.125*S3*S1*T1
        C( 8)=-0.125*B/A*T2*T1*T1
        C( 9)= 1.D0/(4.D0*B)*(-3.D0+4.D0*S+3.D0*T*T-3.D0*S*T*T-S**3)
        C(10)=-0.125*A/B*S1*S1*S2
        C(11)=-0.125*T3*T1*S1
      ELSEIF(N.EQ.2)THEN
        C( 1)= 0.25*S2*T1
        C( 2)= 0.25*S2*T1
        C( 3)= 0.125*(2.0-S)*T1*S2*S2-0.125*S2*T2*T1*T
        C( 4)=-0.0625*A*S1*T1*S2*S2
        C( 5)= 0.0625*B*S2*T2*T1*T1
        C( 6)= 1.D0/(4.D0*A)*(3.D0-4.D0*T-3.D0*S*S+3.D0*S*S*T+T**3)
        C( 7)=-0.125*(1.D0-3.D0*S)*S2*T1
        C( 8)= 0.125*B/A*T1*T1*T2
        C( 9)= 1.D0/(4.D0*B)*(-3.D0-4.D0*S+3.D0*T*T+3.D0*S*T*T+S**3)
        C(10)= 0.125*A/B*S1*S2*S2
        C(11)=-0.125*S2*T3*T1
      ELSEIF(N.EQ.3)THEN
        C( 1)= 0.25*S2*T2
        C( 2)= 0.25*S2*T2
        C( 3)= 0.125*(2.0-S)*T2*S2*S2+0.125*S2*T2*T1*T
        C( 4)=-0.0625*A*S1*T2*S2*S2
        C( 5)=-0.0625*B*S2*T1*T2*T2
        C( 6)= 1.D0/(4.D0*A)*(3.D0+4.D0*T-3.D0*S*S-3.D0*S*S*T-T**3)
        C( 7)=-0.125*(1.D0-3.D0*S)*S2*T2
        C( 8)=-0.125*B/A*T1*T2*T2
        C( 9)= 1.D0/(4.D0*B)*(3.D0+4.D0*S-3.D0*T*T-3.D0*S*T*T-S**3)
        C(10)=-0.125*A/B*S1*S2*S2
        C(11)=-0.125*(1.D0-3.D0*T)*S2*T2
      ELSEIF(N.EQ.4)THEN
        C( 1)= 0.25*S1*T2
        C( 2)= 0.25*S1*T2
        C( 3)= 0.125*(2.0-T)*S1*T2*T2-0.125*S1*S2*T2*S
        C( 4)= 0.0625*A*S2*T2*S1*S1
        C( 5)=-0.0625*B*S1*T1*T2*T2
        C( 6)= 1.D0/(4.D0*A)*(-3.D0-4.D0*T+3.D0*S*S+3.D0*S*S*T+T**3)
        C( 7)=-0.125*S3*S1*T2
        C( 8)= 0.125*B/A*T1*T2*T2
        C( 9)= 1.D0/(4.D0*B)*(3.D0-4.D0*S-3.D0*T*T+3.D0*S*T*T+S**3)
        C(10)= 0.125*A/B*S2*S1*S1
        C(11)=-0.125*(1.D0-3.D0*T)*S1*T2
      ENDIF
      RETURN
      END
C
C
C
      SUBROUTINE BTRANST(ICO,ISO,IELS,SX,SY,X,Y,XL,ID,IBP)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION ISO(6,1),SX(1),SY(1),X(1),Y(1),ICO(6,1),IBP(1)
      COMMON/CSTSP/STSIG(3),STEPS(3),STIFS(6,6),SPSTIF(26,26),
     +            STLOAD(6,1),SPLOAD(26,1),TRLOAD(23,1),TRSTIF(26,26),
     +            SUSH(26),SWSH(26),UCSH(4),USSH(6),WSSH(4)
      COMMON/CTRAN/TRAN(10,20),TRAN1(6,23),TRANJ(26,23),TRANT(20,10),
     +             TRAN1T(23,6),C(11)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      COMMON/CONST/E,E1,E2,ANU,BETA,FCU,FTU,ET,EP,YL,YLP,EPSU,ECX,ECY
      CALL PRESET(TRAN1,6,23)
      IF(ID.EQ.2)THEN                    ! VERTICAL
        X1= ABS(X(ICO(3,ISO(5,IELS))) - X(ICO(1,ISO(5,IELS))))/2.D0
        X2= SX(ISO(1,IELS)) - X(ICO(1,ISO(5,IELS)))
        S = (X2-X1)/X1
        A = X1*2.D0
        B = XL
        IF(IBP(ISO(1,IELS)).EQ.1) T=-1.D0
        IF(IBP(ISO(2,IELS)).EQ.1) T= 1.D0
      ELSE                                         ! HORIZONTAL
        Y1= ABS(Y(ICO(3,ISO(5,IELS))) - Y(ICO(1,ISO(5,IELS))))/2.D0
        Y2= SY(ISO(1,IELS)) - Y(ICO(1,ISO(5,IELS)))
        T = (Y2-Y1)/Y1
        A = XL
        B = Y1*2.D0
        IF(IBP(ISO(1,IELS)).EQ.1) S=-1.D0
        IF(IBP(ISO(2,IELS)).EQ.1) S= 1.D0
      ENDIF
      DO 2 J=1,4
      CALL SHAPE(C,S,T,A,B,J)
      IF(ID.EQ.1) THEN                             ! HORIZONTAL
```

```
        DO 3 K=1,3
        IF(IBP(ISO(1,IELS)).EQ.1)THEN
          TRAN1(2,(5*J-3)+K)=C(2+K)                        ! N(1)~N(12)
          TRAN1(3,(5*J-3)+K)=C(5+K)                        ! DN/DX
        ELSEIF(IBP(ISO(2,IELS)).EQ.1)THEN
          TRAN1(5,5*J+K)=C(2+K)
          TRAN1(6,5*J+K)=C(5+K)
        ENDIF
    3   CONTINUE
      ELSE                                                 ! VERTICAL
        DO 4 K=1,3
        IF(IBP(ISO(1,IELS)).EQ.1)THEN
          TRAN1(2,(5*J-3)+K)=C(2+K)                        ! N(1)~N(12)
          TRAN1(3,(5*J-3)+K)=C(8+K)                        ! DN/DY
        ELSEIF(IBP(ISO(2,IELS)).EQ.1)THEN
          TRAN1(5,5*J+K)=C(2+K)
          TRAN1(6,5*J+K)=C(8+K)
        ENDIF
    4   CONTINUE
      ENDIF
    2 CONTINUE
      S1=1.D0-S
      S2=1.D0+S
      T1=1.D0-T
      T2=1.D0+T
      CALL UCSHAPE(S1,S2,T1,T2,UCSH)
      IF(IBP(ISO(1,IELS)).EQ.1)THEN
        TRAN1(4,21)=1.D0
        TRAN1(5,22)=1.D0
        TRAN1(6,23)=1.D0
        DO 5 I=1,4
    5   TRAN1(1,ID+(I-1)*5)=UCSH(I)
      ELSEIF(IBP(ISO(2,IELS)).EQ.1)THEN
        TRAN1(1,1)=1.D0
        TRAN1(2,2)=1.D0
        TRAN1(3,3)=1.D0
        DO 6 I=1,4
    6   TRAN1(4,ID+(I-1)*5+3)=UCSH(I)
      ENDIF
      RETURN
      END
C
C
C
      SUBROUTINE BOND(DISP,STDISP1,ICO,ISO,IELS,SX,SY,X,Y,XL,ID,SKU,
     +                SKUIN,SLIPM,INCR)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION AN(12,2),ANX(12,2),ANY(12,2),R(22),DISP(20,1),SNDP(22),
     +          ANCE(12),ANC(12,2),STDISP1(6,1),RSLIP(2),C(11)
      DIMENSION ICO(6,1),ISO(6,1),SX(1),SY(1),X(1),Y(1)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      COMMON/CONST/E,E1,E2,ANU,BETA,FCU,FTU,ET,EP,YL,YLP,EPSU,ECX,ECY
      SKU=SKUIN
      RETURN
      END
C
C
C
      SUBROUTINE TRNDISP(ICO,ISO,IELS,SX,SY,X,Y,XL,ID,IBP,TOTST,
     +                   DEDISP,L,JX)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION X(1),Y(1),SX(1),SY(1),ICO(6,1),ISO(6,1),JX(1),IBP(1),
     +          DEDISP(1),TOTST(26,1)
      COMMON/CTRAN/TRAN(10,20),TRAN1(6,23),TRANJ(26,23),TRANT(20,10),
     +             TRAN1T(23,6),C(11)
      COMMON/CELAST/TEPS1(3,1),TSIG1(3,1),BTEMP1(1,6),DTDISP(26,1),
     +              STDISP(6,1),CNDISP(10,1)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      COMMON/CSTIF/LJ(68),P(68),STIF(68,68),DISP(20,1) !CHECK DIMENSION
      ISO1=ISO(1,IELS)
      ISO2=ISO(2,IELS)
      DO 1 J=1,NNODEL
      J1=(J-1)*NVAR
      J2=NVAR*(ICO(J,ISO(5,IELS))-1)
      DO 1 I=1,NVAR
    1 LJ(I+J1)=JX(J2+I)
      N=(L-1)*6
      DO 2 K=1,2
      DO 2 KK=1,3
      N=N+1
      K1=(ISO(K,IELS)-1)*3+KK+NMAT
      LJ(NVEL+N)=JX(K1)
    2 CONTINUE
      DO 3 I=1,NVEL
      IF(LJ(I).GT.0) DISP(I,1)=DEDISP(LJ(I))
    3 IF(LJ(I).EQ.0) DISP(I,1)=0.D0
      IF(IBP(ISO1).EQ.1 .OR. IBP(ISO2).EQ.1)THEN
        CALL TRANST(ICO,ISO,IELS,SX,SY,X,Y,XL,ID)
        CALL MULT(TRAN,DISP,CNDISP,10,20,1)
      ENDIF
      M=(L-1)*6+NVEL
      IF(IBP(ISO1).EQ.1)THEN
        DTDISP(21,1)=CNDISP(  ID,1)
        DTDISP(22,1)=CNDISP(   3,1)
        DTDISP(23,1)=CNDISP(3+ID,1)
      ELSEIF(IBP(ISO1).EQ.0)THEN
        DO 4 I=1,3
    4   DTDISP(NVEL+I,1)=DEDISP(LJ(M+I))
      ENDIF
      IF(IBP(ISO2).EQ.1)THEN
```

```
        DTDISP(24,1)=CNDISP(5+ID,1)
        DTDISP(25,1)=CNDISP(   8,1)
        DTDISP(26,1)=CNDISP(8+ID,1)
      ELSEIF(IBP(ISO2).EQ.0)THEN
        DO 5 I=4,6
    5   DTDISP(NVEL+I,1)=DEDISP(LJ(M+I))
      ENDIF
      DO 6 I=1,NVEL
    6 DTDISP(I,1)=DISP(I,1)
      DO 7 I=1,26
      TOTST(I,IELS)=TOTST(I,IELS)+DTDISP(I,1)
    7 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE SBMATX(S,T,BSTL,XL,ECC,ID)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION BSTL(1,6)
      IF(ID.EQ.2) THEN
        BSTL(1,1)=-1.DO/XL
        BSTL(1,2)= 6.DO*ECC*T/(XL*XL)
        BSTL(1,3)= ECC*(3.DO*T-1.DO)/XL
        BSTL(1,4)=-BSTL(1,1)
        BSTL(1,5)=-BSTL(1,2)
        BSTL(1,6)= ECC*(3.DO*T+1.DO)/XL
      ELSE
        BSTL(1,1)=-1.DO/XL
        BSTL(1,2)= 6.DO*ECC*S/(XL*XL)
        BSTL(1,3)= ECC*(3.DO*S-1.DO)/XL
        BSTL(1,4)=-BSTL(1,1)
        BSTL(1,5)=-BSTL(1,2)
        BSTL(1,6)= ECC*(3.DO*S+1.DO)/XL
      ENDIF
      RETURN
      END
C
C
C
      SUBROUTINE DSTEEL(DSTL,ES,IELS,NELS)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION DSTL(3,1)
      DO 1 I=1,3
    1 DSTL(I,IELS)=ES
      RETURN
      END
C
C
C
      SUBROUTINE SSTRESS(DSTL,IPTYL,BSTEL,JX,ISO,SX,SY,ICO,X,Y,DEDISP,
     +                  ITER,INCR,TOTST,IBP)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION DSTL(3,1),IPTYL(3,1),BSTEL(6,3,200),TOTST(26,1)
      DIMENSION JX(1),SX(1),SY(1),X(1),Y(1),ISO(6,1),ICO(6,1),
     +          DEDISP(1),IBP(1),JPRINT(6),DSTEPS(3),DSTSIG(3)
      COMMON/CELAST/TEPS1(3,1),TSIG1(3,1),BTEMP1(1,6),DTDISP(26,1),
     +              STDISP(6,1),CNDISP(10,1)
      COMMON/CSTIF/LJ(68),P(68),STIF(68,68),DISP(20,1)
      COMMON/CSTSP/STSIG(3),STEPS(3),STIFS(6,6),SPSTIF(26,26),
     +            STLOAD(6,1),SPLOAD(26,1),TRLOAD(23,1),TRSTIF(26,26),
     +            SUSH(26),SWSH(26),UCSH(4),USSH(6),WSSH(4)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      COMMON/CONST/E,E1,E2,ANU,BETA,FCU,FTU,ET,EP,YL,YLP,EPSU,ECX,ECY
      COMMON/CTAPE/MTAPE,NTAPE,LTAPE,KTAPE
      DATA JPRINT/70,90,110,130,150,160/
      IF(ITER.EQ.2)THEN
        DO 15 I=1,6
15      IF(INCR.EQ.JPRINT(I)) WRITE(6,100)
      ENDIF
      L=0
      DO 1 IEL=1,NEL
      DO 5 IELS=1,NELS
        IF(ISO(5,IELS).NE.IEL) GO TO 5
        READ(KTAPE) (STSIG(I),I=1,3),(STEPS(I),I=1,3)
        ID=ISO(6,IELS)
        L=L+1
        IF(ID.EQ.2)THEN                                 ! VERTICAL
          XL=ABS(SY(ISO(2,IELS))-SY(ISO(1,IELS)))
          ECC=ECY
        ELSE                                            ! HORIZONTAL
          XL=ABS(SX(ISO(2,IELS))-SX(ISO(1,IELS)))
          ECC=ECX
        ENDIF
        CALL TRNDISP(ICO,ISO,IELS,SX,SY,X,Y,XL,ID,IBP,TOTST,
     +               DEDISP,L,JX)
        DO 6 I=1,6
 6      STDISP(I,1)=DTDISP(NVEL+I,1)
        DO 8 I=1,3
        DO 9 J=1,6
 9      BTEMP1(1,J)=BSTEL(J,I,IELS)
        CALL MULT(BTEMP1,STDISP,DSEPS,1,6,1)
        DSTEPS(I)=DSEPS
        DSTSIG(I)=DSTL(I,IELS)*DSTEPS(I)
 8      CONTINUE
        DO 10 I=1,3
        STSIG(I)=STSIG(I)+DSTSIG(I)
        STEPS(I)=STEPS(I)+DSTEPS(I)
        IF(ISO(4,IELS).EQ.0)THEN
          YLST=YL
          ES=ET
        ELSEIF(ISO(4,IELS).GT.0)THEN
          YLST=YLP
          ES=EP
        ENDIF
        IF(STSIG(I).GE.YLST.AND.IPTYL(I,IELS).EQ.0) THEN
          IPTYL(I,IELS)=1
          DSTL(I,IELS)=ES/40.D0
          EPSYL=YLST/ES
          DEPSYL=STEPS(I)-EPSYL
          DSIGYL=DEPSYL*DSTL(I,IELS)
          STSIG(I)=YLST+DSIGYL
        ENDIF
   10   CONTINUE
        IF(ITER.EQ.2)THEN
          DO 20 K=1,10
   20     IF(INCR.EQ.JPRINT(K)) WRITE(6,101) IELS,(IPTYL(I,IELS),I=1,3)
        ENDIF
        WRITE(LTAPE) (STSIG(I),I=1,3),(STEPS(I),I=1,3)
    5 CONTINUE
    1 CONTINUE
  100 FORMAT(//,'   STRESS STATES OF STEEL AT INTEGRATION POINTS ',
     +       //,'   0=UNYIELDED       1=YIELDED                   ',
     +        /,'  IEL ',5X,' 1    2    3 '/)
  101 FORMAT(1X,I3,4X,3I5)
      RETURN
      END
C
C
C
      SUBROUTINE TENSTIF(TSTIF,DEDISP,TPR,THETA,THICK,ICR,MDIA,
     +                   ICO,JX,X,Y,NVA,NCR)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION BSTORE(3,20,45),ICO(6,1),THETA(45,1),THICK(45,1),
     +          ICR(46,1),DEDISP(1),X(1),Y(1),
     +          TSTIF(1),JX(1),MDIA(1),TPR(1),W(3),V(5)
      COMMON/CSTIF/LJ(68),P(68),STIF(68,68),DISP(20,1)
      COMMON/CSIGS/SIG(3,45),EPS(3,45),BMATX(3,20),D(3,3),BSTORE,DSTORE
      COMMON/CTENST/PP(20),REDISP(400),TPR1(400),TSTIF1(40000),
     +              RSIG(3,1),BRSIG(20,1),BST(20,3)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      COMMON/CONST/E,E1,E2,ANU,BETA,FCU,FTU,ET,EP,YL,YLP,EPSU,ECX,ECY
      DATA  W/ 0.555555555555556,0.888888888888889,0.555555555555556/
      DATA  V/ 0.236926885056189,0.478628670499366,0.568888888888889,
     +         0.478628670499366,0.236926885056189/
      REWIND 3
      NCR=0
      CALL PSET(TPR,NNETT)
      CALL PSET(REDISP,NNETT)
      DO 1 IEL=1,NEL
      READ(3) (((BSTORE(M,L,IPT),M=1,3),L=1,20),IPT=1,45)
      CALL CURRENT(IEL,NNODEL,ICO,JX,AA,BB,NVAR,LJ,X,Y,IS,IB,NEL)
      CALL PSET(PP,20)
      IPT=0
      CONST=AA*BB/8.D0
      DO 2 I=1,3
      DO 2 J=1,3
      DO 2 K=1,5
      IPT=IPT+1
      IF(ICR(IPT,IEL).EQ.3) THEN
      NCR=1
      CALL STORE(BMATX,BSTORE,3,20,IPT,0)
      CALL TRANPS(BMATX,BST,3,20)
      CALL COHESST(RSIG,SIG,IPT,IEL,THETA,FTU)
      CALL MULT(BST,RSIG,BRSIG,20,3,1)
      DO 3 L=1,NVEL
    3 PP(L)=PP(L)+W(I)*W(J)*V(K)*BRSIG(L,1)*CONST*THICK(IPT,IEL)
      ENDIF
    2 CONTINUE
      DO 4 I=1,NVEL
      LJR=LJ(I)
      IF(LJR) 4,4,5
    5 TPR(LJR)=TPR(LJR)+PP(I)
    4 CONTINUE
    1 CONTINUE
      IF(NCR.EQ.0) RETURN
      CALL COLSOL(TSTIF,TPR,REDISP,MDIA,NNETT,NVA,1,TSTIF1,TPR1)
      DO 6 I=1,NNETT
    6 DEDISP(I)=DEDISP(I)-REDISP(I)
      RETURN
      END
C
C
C
      SUBROUTINE COHESST(RSIG,SIG,IPT,IEL,THETA,FTU)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION RSIG(3,1),SIG(3,45),THETA(45,1)
      COMMON/CCOHZ/R(3,3),TEMPFT(3,1)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      C2=DCOS(THETA(IPT,IEL))*DCOS(THETA(IPT,IEL))
      S2=DSIN(THETA(IPT,IEL))*DSIN(THETA(IPT,IEL))
      SC=DSIN(THETA(IPT,IEL))*DCOS(THETA(IPT,IEL))
      R(1,1)= C2
      R(2,1)= S2
      R(3,1)= SC
      R(1,2)= S2
      R(2,2)= C2
      R(3,2)=-SC
      R(1,3)=-2.D0*SC
      R(2,3)= 2.D0*SC
      R(3,3)= C2-S2
      TEMPFT(1,1)= FTU
      TEMPFT(2,1)= 0.D0
      TEMPFT(3,1)= 0.D0
      CALL MULT(R,TEMPFT,RSIG,3,3,1)
      RETURN
      END
C
C
C
      SUBROUTINE BOLT(NVAR,JX,MDIA,A,XK,ND,NSP)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION JX(1),MDIA(1),A(1),XK(1),ND(1)
      DO 1 I=1,NSP
      IAD=NVAR*(ND(I)-1)+3
      JAD=JX(IAD)
      JSTIF=MDIA(JAD)
      A(JSTIF)=A(JSTIF)+XK(I)
    1 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE SSTIFF(NELS,IELS,XL,ECC,ID,AREA,DSTL,BSTEL,INCR,ISS)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION BSTL(1,6),BSTEL(6,3,200),DSTL(3,1),AREA(1),XI(3),W(3)
      COMMON/CSTSP/STSIG(3),STEPS(3),STIFS(6,6),SPSTIF(26,26),
     +            STLOAD(6,1),SPLOAD(26,1),TRLOAD(23,1),TRSTIF(26,26),
     +            SUSH(26),SWSH(26),UCSH(4),USSH(6),WSSH(4)
      COMMON/CSSTIF/BDB(6,6),DB(1,6)
      DATA XI/-0.77459666924148,0.00000000000000,0.77459666924148 /
      DATA  W/ 0.55555555555556,0.88888888888889,0.55555555555556 /
      IF(ISS.EQ.0) RETURN
      DO 10 I=1,3
      IF(INCR.EQ.1) THEN
        IF(ID.EQ.2) THEN
          S=0.D0
          T=XI(I)
        ELSE
          T=0.D0
          S=XI(I)
        ENDIF
        CALL SBMATX(S,T,BSTL,XL,ECC,ID)
        DO 15 K=1,6
   15   BSTEL(K,I,IELS)=BSTL(1,K)
      ELSE
        DO 20 K=1,6
   20   BSTL(1,K)=BSTEL(K,I,IELS)
      ENDIF
      ESTL=DSTL(I,IELS)
      CALL MULT2(ESTL,BSTL,BDB,1,6,DB)
      DO 30 J=1,6
      DO 30 K=1,6
   30 STIFS(J,K)=STIFS(J,K)+W(I)*BDB(J,K)*AREA(IELS)*XL/2.D0
   10 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE STEEL(SX,SY,AREA,JX,DEDISP,X,Y,ISO,ICO,DSIGS,INCR,
     +                 ITER,TOTST,IEL,IBP,DSTL,BSTEL,SKUIN,SLIPM,
     +                 SKWIN,TSTLOAD)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION TOTST(26,1),DSTL(3,1),BSTEL(6,3,200),XI(3),W(3),
     +          TSTLOAD(1)
      DIMENSION DSIGS(1),AREA(1),JX(1),X(1),Y(1),SX(1),SY(1),IBP(1),
     +          DEDISP(1),ISO(6,1),ICO(6,1)
      COMMON/CSTSP/STSIG(3),STEPS(3),STIFS(6,6),SPSTIF(26,26),
     +            STLOAD(6,1),SPLOAD(26,1),TRLOAD(23,1),TRSTIF(26,26),
     +            SUSH(26),SWSH(26),UCSH(4),USSH(6),WSSH(4)
      COMMON/CTRAN/TRAN(10,20),TRAN1(6,23),TRANJ(26,23),TRANT(20,10),
     +             TRAN1T(23,6),C(11)
      COMMON/CSTIF/LJ(68),P(68),STIF(68,68),DISP(20,1) !CHECK DIMENSION
      COMMON/CTRNSF/TST(23,23),ST(6,23),PT(26,23)
      COMMON/CSSTIF/BDB(6,6),DB(1,6)
      COMMON/CTRNLD/BTEMP(6,1),SPDISP(26,1),TEMPLD(20)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      COMMON/CONST/E,E1,E2,ANU,BETA,FCU,FTU,ET,EP,YL,YLP,EPSU,ECX,ECY
      COMMON/CTAPE/MTAPE,NTAPE,LTAPE,KTAPE
      DATA XI/-0.77459666924148,0.00000000000000,0.77459666924148 /
      DATA W / 0.55555555555556,0.88888888888889,0.55555555555556 /
      SKW=SKWIN
      L=0
      CALL PSET(TSTLOAD,NVELT)
      DO 1 IELS=1,NELS
      IF(ISO(5,IELS).NE.IEL) GO TO 1
      IF(INCR.GT.1) READ(LTAPE) (STSIG(I),I=1,3),(STEPS(I),I=1,3)
      ID=ISO(6,IELS)
      ISS=ISO(3,IELS)
      L=L+1
      IF(ISS.EQ.0) GO TO 1
      CALL PRESET(STIFS,6,6)
      CALL PRESET(SPSTIF,26,26)
      CALL PRESET(STLOAD,6,1)
      CALL PRESET(SPLOAD,26,1)
      CALL PRESET(TRSTIF,26,26)
      CALL PRESET(TRLOAD,23,1)
      N=(L-1)*6
      DO 2 K=1,2
      DO 2 KK=1,3
      N=N+1
      K1=(ISO(K,IELS)-1)*3+KK+NMAT
      LJ(NVEL+N)=JX(K1)
    2 CONTINUE
      IF(INCR.EQ.1. AND .ISO(4,IELS).EQ.1) READ(5,*) DSIGS(IELS)
      IF(ID.EQ.2) THEN
        XL=ABS(SY(ISO(2,IELS))-SY(ISO(1,IELS)))
        ECC=ECY
      ELSE
        XL=ABS(SX(ISO(2,IELS))-SX(ISO(1,IELS)))
        ECC=ECX
      ENDIF
      IF(INCR.EQ.1)THEN
        IF(ISO(4,IELS).EQ.0) ES=ET
        IF(ISO(4,IELS).EQ.1) ES=EP
        CALL DSTEEL(DSTL,ES,IELS,NELS)
      ENDIF
      CALL BOND(DISP,STDISP1,ICO,ISO,IELS,SX,SY,X,Y,XL,ID,SKU,
     +          SKUIN,SLIPM,INCR)
      CALL JOINT(ICO,ISO,IELS,SX,SY,X,Y,XL,ECC,ID,ISS,SKU,SKW)
      CALL TRNSTIF(ICO,ISO,IELS,SX,SY,X,Y,XL,ID,IBP,2)
      CALL SSTIFF(NELS,IELS,XL,ECC,ID,AREA,DSTL,BSTEL,INCR,ISS)
      CALL TRNSTIF(ICO,ISO,IELS,SX,SY,X,Y,XL,ID,IBP,1)
      IF(INCR.EQ.1 .AND. ISO(4,IELS).EQ.1)
     +  CALL TRNPRST(ICO,ISO,IELS,SX,SY,X,Y,XL,ID,IBP,DSIGS,TSTLOAD)
      IF(INCR.GT.1)
     +  CALL TRNLOAD(IELS,TOTST,BSTEL,AREA,ICO,ISO,SX,SY,X,Y,ID,XL,IBP)
      CALL ADDSTIF(L,NVEL,NVELT)
    1 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE JTRANST(ICO,ISO,IELS,SX,SY,X,Y,XL,ID,IBP)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION ISO(6,1),ICO(6,1),SX(1),SY(1),X(1),Y(1),IBP(1)
      COMMON/CTRAN/TRAN(10,20),TRAN1(6,23),TRANJ(26,23),TRANT(20,10),
     +             TRAN1T(23,6),C(11)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      COMMON/CONST/E,E1,E2,ANU,BETA,FCU,FTU,ET,EP,YL,YLP,EPSU,ECX,ECY
      CALL PRESET(TRANJ,26,23)
      CALL BTRANST(ICO,ISO,IELS,SX,SY,X,Y,XL,ID,IBP)
      DO 1 I=1,NVEL
    1 TRANJ(I,I)=1.D0
      IF(IBP(ISO(1,IELS)).EQ.1)THEN
        DO 2 J=1,23
        DO 2 I=1,6
    2   TRANJ(NVEL+I,J)=TRAN1(I,J)
      ELSEIF(IBP(ISO(2,IELS)).EQ.1)THEN
```

```
        DO 4 J=1,NVEL
        DO 4 I=1,6
    4   TRANJ(NVEL+I,J)=TRAN1(I,J+3)
        DO 5 I=21,23
    5   TRANJ(I,I)=1.D0
      ENDIF
      RETURN
      END
C
C
C
      SUBROUTINE TRNSTIF(ICO,ISO,IELS,SX,SY,X,Y,XL,ID,IBP,NTYPE)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION X(1),Y(1),SX(1),SY(1),ISO(6,1),ICO(6,1),IBP(1)
      COMMON/CSTSP/STSIG(3),STEPS(3),STIFS(6,6),SPSTIF(26,26),
     +            STLOAD(6,1),SPLOAD(26,1),TRLOAD(23,1),TRSTIF(26,26),
     +            SUSH(26),SWSH(26),UCSH(4),USSH(6),WSSH(4)
      COMMON/CTRNSF/TST(23,23),ST(6,23),PT(26,23)
      COMMON/CTRAN/TRAN(10,20),TRAN1(6,23),TRANJ(26,23),TRANT(20,10),
     +             TRAN1T(23,6),C(11)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      ISO1=ISO(1,IELS)
      ISO2=ISO(2,IELS)
      CALL PRESET(TST,23,23)
      IF(IBP(ISO1).NE.1. AND .IBP(ISO2).NE.1) RETURN
      IF(NTYPE.EQ.1)THEN                 ! TRANPS. FOR STEEL
        CALL BTRANST(ICO,ISO,IELS,SX,SY,X,Y,XL,ID,IBP)
        CALL MULT2(STIFS,TRAN1,TST,6,23,ST)
        CALL PRESET(STIFS,6,6)
      ELSEIF(NTYPE.EQ.2)THEN             ! TRANPS. FOR JOINT
        CALL JTRANST(ICO,ISO,IELS,SX,SY,X,Y,XL,ID,IBP)
        CALL MULT2(SPSTIF,TRANJ,TST,26,23,PT)
        CALL PRESET(SPSTIF,26,26)
      ENDIF
      IF(IBP(ISO1).EQ.1)THEN
        DO 1 I=1,NVEL
        DO 1 J=1,NVEL
    1   TRSTIF(J,I)=TST(J,I)
        DO 2 J=1,3
        DO 2 I=1,NVEL
        TRSTIF(I,23+J)=TST(I,20+J)
    2   TRSTIF(23+J,I)=TRSTIF(I,23+J)
        DO 3 I=1,3
        DO 3 J=1,3
    3   TRSTIF(23+J,23+I)=TST(20+J,20+I)
      ELSEIF(IBP(ISO2).EQ.1)THEN
        IF(NTYPE.EQ.1)THEN           ! FOR STEEL
          DO 4 I=1,NVEL
          DO 4 J=1,NVEL
    4     TRSTIF(J,I)=TST(3+J,3+I)
```

```
          DO 5 J=1,3
          DO 5 I=1,NVEL
          TRSTIF(I,20+J)=TST(3+I,J)
    5     TRSTIF(20+J,I)=TRSTIF(I,20+J)
          DO 6 I=1,3
          DO 6 J=1,3
    6     TRSTIF(20+J,20+I)=TST(J,I)
        ELSE                         ! FOR JOINT
          DO 7 I=1,23
          DO 7 J=1,23
    7     TRSTIF(J,I)=TST(J,I)
        ENDIF
      ENDIF
      IF(NTYPE.EQ.2)THEN
        DO 8 I=1,26
        DO 8 J=1,26
    8   SPSTIF(J,I)=TRSTIF(J,I)
        CALL PRESET(TRSTIF,26,26)
      ENDIF
      RETURN
      END
C
C
C
      SUBROUTINE TRNPRST(ICO,ISO,IELS,SX,SY,X,Y,XL,ID,IBP,DSIGS,
     +                   TSTLOAD)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION ICO(6,1),ISO(6,1),TSTLOAD(1),X(1),Y(1),IBP(1),
     +          DSIGS(1),SX(1),SY(1)
      COMMON/CSTSP/STSIG(3),STEPS(3),STIFS(6,6),SPSTIF(26,26),
     +            STLOAD(6,1),SPLOAD(26,1),TRLOAD(23,1),TRSTIF(26,26),
     +            SUSH(26),SWSH(26),UCSH(4),USSH(6),WSSH(4)
      COMMON/CTRAN/TRAN(10,20),TRAN1(6,23),TRANJ(26,23),TRANT(20,10),
     +             TRAN1T(23,6),C(11)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      STLOAD(1,1)= DSIGS(IELS)
      STLOAD(4,1)=-DSIGS(IELS)
      IF(IBP(ISO(1,IELS)).NE.1 .AND. IBP(ISO(2,IELS)).NE.1) RETURN
      CALL BTRANST(ICO,ISO,IELS,SX,SY,X,Y,XL,ID,IBP)
      CALL TRANPS(TRAN1,TRAN1T,6,23)
      CALL MULT(TRAN1T,STLOAD,TRLOAD,23,6,1)
      IF(IBP(ISO(1,IELS)).EQ.1)THEN
        DO 6 I=1,NVEL
    6   TSTLOAD(I)=TSTLOAD(I)+TRLOAD(I,1)
      ELSEIF(IBP(ISO(2,IELS)).EQ.1)THEN
        DO 7 I=1,NVEL
    7   TSTLOAD(I)=TSTLOAD(I)+TRLOAD(3+I,1)
      ENDIF
      RETURN
      END
```

```
C
C
C
      SUBROUTINE TRNLOAD(IELS,TOTST,BSTEL,AREA,ICO,ISO,SX,SY,X,Y,ID,
     +                   XL,IBP)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION TOTST(26,1),BSTEL(6,3,200),IBP(1),X(1),Y(1),W(3),
     +          AREA(1),ICO(6,1),ISO(6,1),SX(1),SY(1)
      COMMON/CSTSP/STSIG(3),STEPS(3),STIFS(6,6),SPSTIF(26,26),
     +            STLOAD(6,1),SPLOAD(26,1),TRLOAD(23,1),TRSTIF(26,26),
     +            SUSH(26),SWSH(26),UCSH(4),USSH(6),WSSH(4)
      COMMON/CTRAN/TRAN(10,20),TRAN1(6,23),TRANJ(26,23),TRANT(20,10),
     +             TRAN1T(23,6),C(11)
      COMMON/CSTIF/LJ(68),P(68),STIF(68,68),DISP(20,1) !CHECK DIMENSION
      COMMON/CTRNLD/BTEMP(6,1),SPDISP(26,1),TEMPLD(20)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      DATA W / 0.55555555555556,0.88888888888889,0.55555555555556 /
      CALL PSET(TEMPLD,20)
      DO 8 I=1,26
    8 SPDISP(I,1)=TOTST(I,IELS)
      CALL MULT(SPSTIF,SPDISP,SPLOAD,26,26,1)
      IF(IBP(ISO(1,IELS)).EQ.1 .OR. IBP(ISO(2,IELS)).EQ.1)THEN
        DO 9 I=1,3
        DO 9 J=1,6
        BTEMP(J,1)=BSTEL(J,I,IELS)
    9   STLOAD(J,1)=STLOAD(J,1)+W(I)*BTEMP(J,1)*STSIG(I)
     +            *AREA(IELS)*XL/2.DO
        CALL BTRANST(ICO,ISO,IELS,SX,SY,X,Y,XL,ID,IBP)
        CALL TRANPS(TRAN1,TRAN1T,6,23)
        CALL MULT(TRAN1T,STLOAD,TRLOAD,23,6,1)
        IF(IBP(ISO(1,IELS)).EQ.1)THEN
          DO 11 I=1,NVEL
   11     TEMPLD(I)=TRLOAD(I,1)
        ELSE
          DO 12 I=1,NVEL
   12     TEMPLD(I)=TRLOAD(3+I,1)
        ENDIF
      ENDIF
      DO 15 I=1,NVEL
   15 P(I)=P(I)+SPLOAD(I,1)+TEMPLD(I)
      RETURN
      END
C
C
C
      SUBROUTINE ADDSTIF(L,NVEL,NVELT)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      COMMON/CSTIF/LJ(68),P(68),STIF(68,68),DISP(20,1) !CHECK DIMENSION
      COMMON/CSTSP/STSIG(3),STEPS(3),STIFS(6,6),SPSTIF(26,26),
     +            STLOAD(6,1),SPLOAD(26,1),TRLOAD(23,1),TRSTIF(26,26),
     +            SUSH(26),SWSH(26),UCSH(4),USSH(6),WSSH(4)
      DO 1 J=1,NVEL
      DO 1 I=1,NVEL
    1 STIF(I,J)=STIF(I,J)+SPSTIF(I,J)+TRSTIF(I,J)
      I1=NVEL+(6*L-5)
      I2=NVEL+6*L
      K =NVEL
      DO 2 I=I1,I2
      K=K+1
      DO 2 J=1,NVEL
      STIF(J,I)=STIF(J,I)+SPSTIF(J,K)+TRSTIF(J,K)
      STIF(I,J)=STIF(J,I)
    2 CONTINUE
      DO 3 I=I1,I2
      K1 =I-(L-1)*6
      KK1=I-(NVEL+(L-1)*6)
      DO 3 J=I1,I2
      K2 =J-(L-1)*6
      KK2=J-(NVEL+(L-1)*6)
      STIF(J,I)=STIF(J,I)+SPSTIF(K2,K1)+TRSTIF(K2,K1)+STIFS(KK2,KK1)
    3 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE JOINT(ICO,ISO,IELS,SX,SY,X,Y,XL,ECC,ID,ISS,SKU,SKW)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION ICO(6,1),ISO(6,1),SX(1),SY(1),X(1),Y(1),XI(3),W(3)
      COMMON/CSTSP/STSIG(3),STEPS(3),STIFS(6,6),SPSTIF(26,26),
     +            STLOAD(6,1),SPLOAD(26,1),TRLOAD(23,1),TRSTIF(26,26),
     +            SUSH(26),SWSH(26),UCSH(4),USSH(6),WSSH(4)
      COMMON/CTRAN/TRAN(10,20),TRAN1(6,23),TRANJ(26,23),TRANT(20,10),
     +             TRAN1T(23,6),C(11)
      COMMON/CJWSH/TRN(4,20),TEMPWS(1,4),TEMPWC(1,20)
      COMMON/CNODE/NEL,NELS,NNOD,NNODS,NVAR,NNODEL,NNODSEL,NMAT,NVEL,
     +             NMATT,NVELT,NNET,NNETT
      COMMON/CONST/E,E1,E2,ANU,BETA,FCU,FTU,ET,EP,YL,YLP,EPSU,ECX,ECY
      DATA XI/-0.77459666924148,0.00000000000000,0.77459666924148 /
      DATA W / 0.55555555555556,0.88888888888889,0.55555555555556 /
      IF(ISS.EQ.0) RETURN
      CALL PSET(SUSH,26)
      CALL PSET(SWSH,26)
      IF(ID.EQ.1) THEN                         !HORIZONTAL
        Y1=ABS(Y(ICO(3,ISO(5,IELS)))-Y(ICO(1,ISO(5,IELS))))/2.D0
        Y2=SY(ISO(1,IELS))-Y(ICO(1,ISO(5,IELS)))
        T=(Y2-Y1)/Y1
      ELSE                                     !VERTICAL
        X1=ABS(X(ICO(3,ISO(5,IELS)))-X(ICO(1,ISO(5,IELS))))/2.D0
        X2=SX(ISO(1,IELS))-X(ICO(1,ISO(5,IELS)))
        S=(X2-X1)/X1
      ENDIF
      DO 1 K=1,3
      IF(ID.EQ.1) S=XI(K)
      IF(ID.EQ.2) T=XI(K)
      T1=1.D0-T
      T2=1.D0+T
      S1=1.D0-S
      S2=1.D0+S
      IF(ID.EQ.1) THEN
        V1=S1
        V2=S2
      ELSE
        V1=T1
        V2=T2
      ENDIF
      CALL USSHAPE(V1,V2,ECC,XL)
      CALL WSSHAPE(S,T,XL,ECC,ID)
      CALL UCSHAPE(S1,S2,T1,T2)
      CALL JUSHAPE(ID)
      CALL JWSHAPE(ICO,ISO,IELS,SX,SY,X,Y,XL,ID)
      DO 13 N=1,26
      DO 13 M=1,26
   13 SPSTIF(M,N)=SPSTIF(M,N)+XL*SKU*W(K)*SUSH(M)*SUSH(N)/2.D0
     +                      +XL*SKW*W(K)*SWSH(M)*SWSH(N)/2.D0
    1 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE USSHAPE(V1,V2,ECC,XL)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      COMMON/CSTSP/STSIG(3),STEPS(3),STIFS(6,6),SPSTIF(26,26),
     +           STLOAD(6,1),SPLOAD(26,1),TRLOAD(23,1),TRSTIF(26,26),
     +           SUSH(26),SWSH(26),UCSH(4),USSH(6),WSSH(4)
      USSH(1)= 0.5*V1
      USSH(2)=-6.D0*ECC/XL*0.25*V2*V1
      USSH(3)= ECC*(1.D0-2.D0*V2+0.75*V2*V2)
      USSH(4)= 0.5*V2
      USSH(5)= 6.D0*ECC/XL*0.25*V2*V1
      USSH(6)= ECC*(-V2+0.75*V2*V2)
      RETURN
      END
C
C
C
      SUBROUTINE WSSHAPE(S,T,XL,ECC,ID)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      COMMON/CSTSP/STSIG(3),STEPS(3),STIFS(6,6),SPSTIF(26,26),
     +           STLOAD(6,1),SPLOAD(26,1),TRLOAD(23,1),TRSTIF(26,26),
     +           SUSH(26),SWSH(26),UCSH(4),USSH(6),WSSH(4)
```

```
      IF(ID.EQ.2)THEN
        WSSH(1)= 3.D0*(1.D0+T)*(T-1.D0)/(2.D0*XL)
        WSSH(2)=-(1.D0-T)*(1.D0+3.D0*T)/4.D0
        WSSH(3)= 3.D0*(1.D0+T)*(1.D0-T)/(2.D0*XL)
        WSSH(4)=-(1.D0+T)*(1.D0-3.D0*T)/4.D0
      ELSE
        WSSH(1)= 3.D0*(1.D0+S)*(S-1.D0)/(2.D0*XL)
        WSSH(2)=-(1.D0-S)*(1.D0+3.D0*S)/4.D0
        WSSH(3)= 3.D0*(1.D0+S)*(1.D0-S)/(2.D0*XL)
        WSSH(4)=-(1.D0+S)*(1.D0-3.D0*S)/4.D0
      ENDIF
      RETURN
      END
C
C
C
      SUBROUTINE UCSHAPE(S1,S2,T1,T2)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      COMMON/CSTSP/STSIG(3),STEPS(3),STIFS(6,6),SPSTIF(26,26),
     +            STLOAD(6,1),SPLOAD(26,1),TRLOAD(23,1),TRSTIF(26,26),
     +            SUSH(26),SWSH(26),UCSH(4),USSH(6),WSSH(4)
      UCSH(1)= 0.25*S1*T1
      UCSH(2)= 0.25*S2*T1
      UCSH(3)= 0.25*S2*T2
      UCSH(4)= 0.25*S1*T2
      RETURN
      END
C
C
C
      SUBROUTINE JUSHAPE(ID)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      COMMON/CSTSP/STSIG(3),STEPS(3),STIFS(6,6),SPSTIF(26,26),
     +            STLOAD(6,1),SPLOAD(26,1),TRLOAD(23,1),TRSTIF(26,26),
     +            SUSH(26),SWSH(26),UCSH(4),USSH(6),WSSH(4)
      SUSH(ID)   = UCSH(1)
      SUSH(5+ID) = UCSH(2)
      SUSH(10+ID)= UCSH(3)
      SUSH(15+ID)= UCSH(4)
      SUSH(21)   =-USSH(1)
      SUSH(24)   =-USSH(4)
      RETURN
      END
C
C
C
      SUBROUTINE JWSHAPE(ICO,ISO,IELS,SX,SY,X,Y,XL,ID)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      DIMENSION ICO(6,1),ISO(6,1),SX(1),SY(1),X(1),Y(1)
      COMMON/CSTSP/STSIG(3),STEPS(3),STIFS(6,6),SPSTIF(26,26),
     +            STLOAD(6,1),SPLOAD(26,1),TRLOAD(23,1),TRSTIF(26,26),
 +             SUSH(26),SWSH(26),UCSH(4),USSH(6),WSSH(4)
  COMMON/CTRAN/TRAN(10,20),TRAN1(6,23),TRANJ(26,23),TRANT(20,10),
 +             TRAN1T(23,6),C(11)
  COMMON/CJWSH/TRN(4,20),TEMPWS(1,4),TEMPWC(1,20)
  CALL TRANST(ICO,ISO,IELS,SX,SY,X,Y,XL,ID)
  DO 1 I=1,20
  TRN(1,I)=TRAN(   3,I)
  TRN(2,I)=TRAN(ID+3,I)
  TRN(3,I)=TRAN(   8,I)
1 TRN(4,I)=TRAN(ID+8,I)
  DO 2 I=1,4
2 TEMPWS(1,I)=WSSH(I)
  CALL MULT(TEMPWS,TRN,TEPMWC,1,4,20)
  DO 3 I=1,20
3 SWSH(I)=TEMPWC(1,I)
  SWSH(22)=-WSSH(1)
  SWSH(23)=-WSSH(2)
  SWSH(25)=-WSSH(3)
  SWSH(26)=-WSSH(4)
  RETURN
  END
```

REFERENCES

1. Lutz, L.A. and Gergely, P., "Mechanics of Bond and Slip of Deformed Bars in Concrete", J. ACI, Vol. 64, November 1967, pp. 711-721.

2. Mirza, M.S. and Hounde, J., "Study of Bond Stress-Strain Relationships in Reinforced Concrete", ACI Journal, Proceedings, Vol. 76, No. 1, January 1979, pp. 19-46.

3. Bresler, B., "Reinforced Concrete Engineering", Wiloy-Interscience publication, New York, 1974, pp. 151-193.

4. Nilson, A.H., "Non-linear Analysis of Reinforced Concrete by the Finite Element Method", ACI Journal, Vol. 65, No. 9, Sept. 1968, pp. 757-766.

5. Nilson, A.H., "Internal Measurement of Bond Slip", ACI Journal, Vol. 69, No. 7, July 1972, pp. 439-441.

6. Nilson, A.H., "Bond Stress-Slip Relations in Reinforced Concrete", Report 345, School of Civil and Environmental Engineering, Cornell University, Ithaca, December 1971.

7. Tanner, J.A., "An Experimental Determination of Bond Slip in Reinforced Concrete", M.S. Thesis, Cornell University, Ithaca, N.Y. November 1971.

8. Ngo, D. and Scordelis, A.C., "Finite Element Analysis of Reinforced Concrete Beams", ACI Journal, Vol. 64, No. 3, March 1967, pp. 152-163.

9. Goodman, R.E., Taylor, R.L. and Brekke, T.L., "A Model for the Mechanics of Jointed Rock", Journal of the Soil Mechanics and Foundation Division, ASCE, Vol. 94, No. SM3, May 1968, pp. 637-659.

TA [illegible]

10. Chow, W., "Finite Element Modelling of Prestressed Concrete Slabs", M.Eng. Thesis, McMaster University, Hamilton, August 1984.

11. Anand, S.C. and Shaw, R.H., "Mesh-Reinforcement and Substructuring Technique in Elasto-Plastic Finite Element Analysis", Comp. and Struct., Vol. 11, 1980, pp. 13-21.

12. Taylor, R., Maher, D.R.H. and Hayes, B., "Effect of the Arrangement of Reinforcement on the Behavior of Reinforved Concrete Slabs", Magazine of Concrete Research, Vol. 98, No. 55, June 1966.

13. Jiminez-Perez, R., Gergely, P. and White, R.N., "Shear Transfer across Cracks in Reinforced Concrete", Report No. 78-4, Department of Structure Engineering, Cornell University, Ithaca, N.Y., August 1978.

14. Fardis, M.N. and Buyukozturk, O., "Shear Stiffness of Concrete by Finite Elements", Journal of the Structure Division, ASCE, Vol. 106, No. ST6, June 1980.

15. Jiminez, R., White, R.N. and Gergely, P., "Bond and Dowel Capacities of Reinforced Concrete", Journal of ACI, Vol. 76, No. 1, Janurary 1979, pp. 73-92.

16. Scanlon, A., "Time Dependent Reflections of Reinforced Concrete Slabs", PH.D Thesis, The University of Alberta, Edmonton, 1971.

17. Lin, C-S. and Scordelis, A., "Nonlinear Analysis of RC Shells of General Form", Journal of Structural Division, ASCE, No. ST3, Proc. pp. 11164, March 1975.

18. Gilbert, R.I. and Warner, R.F., "Nonlinear Analysis of Reinforced concrete Slabs with Tension Stiffening", UNICIV Report No. R-167, University of New South Wales, Kensington, N.S.W., Australia, Janurary 1977.

19. Van Greunen, J., "Nonlinear Geometric, Material and Time Dependent Analysis of Reinforced and Prestressed Concrete Slabs and Panels", UC-SESM Report No. 79-3, University of California at Berkeley, October 1979.

20. Mueler, G., "Nonlinear Problems in Nonlinear Analysis of Reinforced Concrete", UC-SESM Report No. 77-5, University of California at Berkeley, September 1977.

21. Hanna, Y.G., "Finite Element Modelling of Reinforced Concrete Structures", PH.D Thesis, McGill University, Montreal, February 1983.

22. Rice, J.R., "Mathematical Analysis in the Mechanics of Fracture", Chapter 3, "Fracture: An Advanced Treatise", Ed. by H. Liebowitz, Academic Press, 1968.

23. Rice, J.R., "A Path Independent Integral and the Approximate Analysis of Strain Concentration by Notches and Cracks", Journal of Applied Mechanics, Transactions of the ASME, June 1968.

24. Griffith, A.A., "The Phenomena of Rupture and Flow in Solids", Philosophical Transactions of the Royal Society, London, Series A, Vol. 221, 1921.

25. Barenblatt, G.I., "Mathematical Theory of Equilibrium Cracks in Brittle Fracture", Advanced in Applied Mechanics, Vol. 7, Academic Press, 1962.

26. ASCE Committee on Concrete and Masonrey Structures, "A State-of-the-Art Report on Finite Element Analysis of Reinforced Concrete", ASCE Spec. publ.

27. Zienkiewicz, O.C., "The Finite Element Method", McGraw-Hill, London, 1977.

28. Moll, E., "Investigation of Transverse Stressing in Bridge Deck", M.Eng. Thesis, McMaster University, Hamilton, 1985.