Nonlinear Finite Element Analysis

of Reinforced Concrete Planar Structures

Approved by

Dissertation Committee:

R. E. Klingner ____________________

D. L. Wheat ____________________

J. L. Tassoulas ____________________

M. E. Kreger ____________________

K. M. Liechti ____________________

Nonlinear Finite Element Analysis of Reinforced Concrete Planar Structures

by

Honggun Park, M.S.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

May, 1994

To my parents, wife, and daughter

ACKNOWLEDGMENT

The author would like to express his deepest thanks to Dr. Richard E. Klingner and Dr. Dan L. Wheat for their continuous guidance, patience, and friendship during his stay at The University of Texas at Austin. Special thanks to them for reviewing every sentence of the author's dissertation in spite of their busy schedules. Additional thanks for being given for opportunity to serve as a teaching assistant and grader at the university. Special gratitude goes to Dr. John. L. Tassoulas for giving impressive structural analysis lectures which became a basis of the author's analysis research.

The author would like to express thanks to his fellow Koreans in the Department of Civil Engineering for their friendship and helpful advise. Also, thanks to John J. Myers for his sincere friendship. The author cannot forget his parents' encouragement and support for him.

Honggun Park

Nonlinear Finite Element Analysis of Reinforced Concrete Planar Structures

Publication No. ________________

Honggun Park, Ph.D.
The University of Texas at Austin, 1994

Supervisors: Richard Evans Klingner
Dan L. Wheat

The objective of this research is to predict the complete behavior up to structural failure of reinforced concrete planar members under cyclic as well as monotonic loading. The structural members to be addressed are beams, columns, beam-column joints, and shear walls, all of which experience damage initiated by tension cracking.

The proposed analytical approach will be able to simulate the behavioral characteristics of reinforced concrete structural members, due to crack opening and closing, compressive crushing, cyclic history of reinforcing steel, and bond-slip between cracked concrete and reinforcing steel.

By simulating the complete range of structural response, the proposed analytical approach can predict behavioral characteristics such as ultimate strength, inelastic deformations, primary crack orientations, and failure mechanisms, all of which are useful for the design and evaluation of reinforced concrete structural members.

To accomplish the objectives noted above, this work includes an investigation of material models for two-dimensional finite element analysis under in-plane cyclic and monotonic loading. Also, several nonlinear solution schemes are investigated to produce a numerically reliable analysis method. The proposed material models and the numerical approach are verified by using previously reported experimental results.

For the material model of cracked concrete based on the concept of smeared cracking, the rotating orthotropic axes model with successive cracking is proposed to complement the existing rotating crack model. In addition to the proposed cracked concrete model, the existing material models of reinforcing steel and bond-slip are implemented in the numerical program. The reinforcing steel model idealizes strain hardening and the Bauschinger effects. The bond-slip model idealizes the bond-deterioration due to cyclic loading.

TABLE OF CONTENTS

LISTS OF FIGURES & TABLES

NOTATIONS

A_s	Area of reinforcement
$\mathbf{D}$	Constitutive matrix
$\mathbf{D}_c$	Constitutive matrix of concrete
$\mathbf{D}_s$	Constitutive matrix of reinforcement
$\mathbf{D}_b$	Constitutive matrix of bond-slip element
$\mathbf{E}$	Elastic modulus of concrete
G	Shear stiffness of concrete
$\mathbf{K}$, $\mathbf{K}_{ij}$	Stiffness matrix, elements of stiffness matrix
$\mathbf{k}$	Material stiffness matrix
$\mathbf{k}_c$	Material stiffness matrix of concrete
$\mathbf{k}_s$	Material stiffness matrix of reinforcement
$\mathbf{k}_b$	Material stiffness matrix of bond-slip element
$\mathbf{P}$, $\mathbf{P}_i$	Force vector
$\mathbf{P}^s$	Applied force vector
$\Delta\mathbf{R}_i$	Residual force vector
$\mathbf{U}$	Displacement vector
$\Delta\mathbf{U}$, $\Delta\mathbf{U}_i$	Incremental displacement vector
ΔU^{ps}	Prescribed displacement
u_r, s_i	Relative displacement of bond-slip element
$\delta\,\mathbf{W}_{ext}$	External virtual work

δW_{int}	Internal virtual work
$\Delta\lambda$	Incremental parameter of applied force vector
σ_1, σ_2	Principal stress of concrete
σ_c	Compressive stress of concrete
σ_c^u	Compressive strength of concrete
σ_c^f	Compressive final stress of concrete
σ'_c, f'_c	Compressive cylinder strength of concrete
σ_s	Reinforcing steel stress
σ_y	Yield stress of reinforcing steel
σ_u	Ultimate stress of reinforcing steel
ε_1, ε_2	Principal strain of concrete
ε_{ip}	Residual strain of reinforcing steel
ε_c	Compressive strain of concrete
ε_s	Reinforcing steel strain
$\bar{\varepsilon}_s$	Equivalent reinforcing steel strain
ε_t	Tensile strain of concrete
ε_t^m	Maximum tensile strain
ε_c^u	Compressive strain of concrete corresponding to compressive strength σ_c^u
ε_t^o	Tensile strain of concrete corresponding to cracking stress

ε_c^f	Compressive final strain of concrete corresponding to compressive final stress σ_c^f
ε_c^m	Maximum compressive strain
ε_c^u	Compressive strain of concrete corresponding to compressive cylinder strength of concrete σ'_c
ε_{sh}	Strain hardening strain of reinforcing steel
ε'_{sh}	Effective strain hardening strain of reinforcing steel
ε_u	Ultimate strain of reinforcing steel
ε_y	Yield strain of reinforcing steel
ρ	Reinforcement ratio
γ	Shear strain
τ	Shear stress
τ_b , τ_{bi}	Bond shear stress
θ_ε^p	Principal strain direction with respect to reference axes
θ_σ^p	Principal stress direction with respect to reference axes
θ_{cr}^p	Primary crack direction with respect to reference axes

1.0 INTRODUCTION

1.1 Motivation of This Research

Many years' experience has shown that reinforced concrete members designed using standard design codes generally perform well under normal loads. However, under extreme load conditions, reinforced concrete members behave nonlinearly; it is often difficult to predict their strengths and inelastic deformation capacities. The performance under extreme loads of structural members designed by current code provisions is sometimes questionable from the standpoint of economy and safety. Though current design provisions are being continuously refined with the goal of designing more economical and reliable structures, more specific knowledge and understanding of member behavior are still necessary. For that reason, analytical research will be helpful in the following areas:

1) Current design provisions apply an integrated design process considering the interaction between flexural and shearing actions for planar members. However, in shear-dominated members such as deep beams, short columns, beam-column joints, and low-rise shear walls, the stress-strain states across the members are very complex, and it is difficult to define member strengths in terms of combinations of flexural and shear strengths. Also, in current design codes such as ACI 318-89 [41], which define the shear strength of a member in terms of contributions from reinforcing steel and from cracked concrete, the shear contribution of cracked concrete is obtained empirically, and is open to question.

2) Retrofit of existing structural members damaged under extreme loads requires damage assessment and estimation of remaining capacity. For this purpose, more detailed information is needed regarding the ultimate strength and nonlinear behavior of damaged members.

3) To investigate the nonlinear behavior of structural members under extreme loading conditions, either laboratory experiments or analyses of the members are required. However, since laboratory experiments are not always available and affordable, predictions of member behavior using reliable analytical methods will be helpful for the design of reinforced concrete structural members. Also, to increase the reliability and to extend the application of experimental results, it is desirable that these results be verified by complementary analytical research reproducing the behavior of the test specimens.

To meet the needs noted above, considerable analytical research has been done on reinforced concrete planar members. However, most analytical research for structural behavior uses simplifying assumptions for either the crack direction or the stress-strain states. This type of analysis method is not appropriate for predicting the behavior of planar members in which various crack directions and stress-strain states exist. Therefore, two-dimensional stress-strain relations and multiple cracks should be used to reasonably predict the behavior of planar structures.

According to the variation of crack direction during loading, cracked concrete models using the concept of smeared crack and smeared reinforcement, are classified into fixed orthotropic axes model and rotating orthotropic axes model. The rotating orthotropic axes model has been frequently used because of simplicity in material modeling. In this model, the behavior of concrete is defined by the equivalent uniaxial stress-strain relations in current principal axes. Though there was a research using this approach for concrete behavior without cracking under biaxial compression, in many studies, this approach has been used for cracked concrete which has the stress-induced orthotropic characteristics. The rotating orthotropic axes model for cracked concrete is also called fictitious - or rotating - crack model.

The rotating orthotropic axes approach has been controversial by interpreting the rotation of orthotropic axes as the rotation of material defect of concrete crack. However, in many studies, the rotating orthotropic axes model show better prediction of cracked concrete behavior than the fixed orthotropic axes model. Recently, the application of the rotating orthotropic axes model has been extended to simple members and load conditions.

This research will present a more logical interpretation of the rotating orthotropic axes model, and it will extend its application to a variety of load conditions and structural members. To enhance our ability to predict the behavior of reinforced concrete members to failure, this study includes the behavioral characteristics of cracked concrete, reinforcing steel, and bond-slip effects, all of which affect overall member response. For this purpose, the material models should be simple enough for stable numerical calculation, but also accurate enough to describe material behavior. Also, the analytical process should be numerically reliable for any type of structural behavior.

1.2 Objectives

The objective of this research is to predict the complete behavior up to structural failure of reinforced concrete planar members under cyclic as well as monotonic loading. The structural members to be addressed are beams, columns, beam-column joints, and shear walls, whose structural failure is caused by material failure initiated by tension cracking.

The proposed analytical approach will be able to simulate the behavioral characteristics of reinforced concrete structural members due to crack opening and closing, compressive crushing, cyclic history of reinforcing steel, and bond-slip between cracked concrete and reinforcing steel.

By simulating the complete range of structural response, the proposed analytical approach can predict behavioral characteristics such as ultimate strength, inelastic deformations, primary crack orientations, and failure mechanisms, all of which are useful for the design and evaluation of reinforced concrete structural members.

1.3 Scope

To accomplish the objectives noted above, this work includes an investigation of cyclic material models for two-dimensional finite element analysis under in-plane cyclic and monotonic loading. Also, several nonlinear solution schemes are investigated to develop a numerically reliable analysis method. The proposed material models and the numerical approach will be verified by simulating material and structural behavior.

The complete scope is presented below, with an emphasis on the original work of this dissertation. The following tasks will be performed:

1) Starting from an existing monotonic model, develop a cracked concrete model for general behavior in the following way:
   a) Cracked concrete is idealized as an orthotropic material whose orthotropic axes rotate due to progressive cracking.
   b) A tensile post-cracking model (tension stiffening model) will be proposed to address the progressive cracking process of concrete.
   c) The concept of compression and tension damage surfaces will be introduced to define two-dimensional stress-strain damage history under cyclic loading.
2) Consider the cyclic behavior of reinforcing steel and bond-slip effects, using existing cyclic models for reinforcing steel and bond-slip behavior.
3) Develop a finite element computer program to apply the proposed cracked concrete model and the existing models of reinforcing steel and bond-slip.
4) Develop a reliable and efficient solution scheme for predicting complete structural behavior, by investigating available nonlinear solution strategies, iteration strategies, and convergence criteria.
5) Test the analysis program incorporating the material models and the solution scheme to predict the behavior of structural members under monotonic and cyclic loading.
6) Examine the range of application of the proposed analysis method.

2.0 DEVELOPMENT OF CRACKED CONCRETE MODEL

2.1 General

The proposed cracked concrete model will be developed by modifying an existing monotonic orthotropic axes model, and by extending the modified model to include cyclic behavior. For cyclic behavior, the general behavior of cracked concrete is first idealized on the basis of experiments. Then, stress-strain laws defining loading-unloading behavior will be added.

2.2 Vecchio's Orthotropic Axes Model

Vecchio and Collins [36] developed an orthotropic axes approach using equivalent uniaxial stress-strain curves for cracked concrete behavior. In their approach, the two-dimensional stress-strain behavior of cracked concrete is defined by equivalent uniaxial stress-strain curves in orthotropic axes, which rotate to principal axes during the loading history. For the equivalent uniaxial stress-strain curves, empirical stress-strain relations based on shear panel tests were proposed for compression softening and tension stiffening effects due to crack opening. In this work, compression softening and tension stiffening effects are introduced in Chapter 3.0, Constitutive Laws for Cracked Concrete. Using the concept of smeared cracking, the empirical stress-strain curves are defined in terms of average stress and strain across tension cracks.

The essence of Vecchio and Collins' analytical approach is to simplify two-dimensional stress-strain relations by using a total stress-strain relation instead of an incremental one, and by assuming that principal stress axes coincide with principal strain axes. The equivalent uniaxial stress-strain relations in rotating principal axes are defined by the principal stress-strain relation of the empirical stress-strain curves. According to Vecchio's shear panel tests [36], the principal stress and strain axes deviate from each other as tension cracks widen. Nevertheless, the orthotropic model precisely follows all aspects of overall panel behavior except tension stiffening. In the tests, the tension stiffening stress, which is small compared with the compression stress, is relatively insignificant for overall panel behavior. As a result, the existing orthotropic approach provides a simple but accurate stress-strain model for cracked concrete under monotonic loading.

To apply the existing monotonic stress-strain model for general loading conditions, improvement and further investigation are required with respect to several aspects of the existing model. First, a new tension stiffening model will be proposed to replace the existing empirically obtained tension stiffening model. Next, the assumption that principal stress axes coincide with principal strain axes will be evaluated under general loading. For the cyclic stress-strain relation of concrete, the damage surfaces providing the boundary of loading and unloading will be defined in two-dimensional strain space. Based on the damage surfaces, a cyclic stress-strain law for cracked concrete will be developed.

2.3 Development of Rotating Orthotropic Axes Model with Successive Cracking

Under tension-compression stress states, concrete is disconnected by tension cracking, and concrete struts resisting compression forces form in the crack direction. Perpendicular to the crack direction, reinforcing steel resists tensile forces, and the bonding action of reinforcing steel induces tension stiffening stresses in the concrete (Figure 2.1). Accordingly, the behavioral characteristics of concrete in the crack direction are completely different from those in the perpendicular direction.

Since concrete cracks provide directionality in the characteristics of concrete behavior like that of naturally orthotropic materials, cracked concrete is usually idealized as an orthotropic material. However, the orthotropic characteristics of cracked concrete are different from those of naturally orthotropic materials in several respects. Above all, since concrete cracking is a stress-induced defect, secondary cracks can develop in any direction, in addition to the primary or initial cracking, if the current principal tensile stresses approach the tension cracking stress. Moreover, since concrete retains tensile stresses induced by bond of reinforcing steel after tensile cracking (which is called tension stiffening stress in terms of average stress), the stresses easily approach the tensile cracking stress in the rotating principal tensile axis, and they induce secondary cracks. Next, though the primary crack direction is visualized as coinciding with a principal tensile direction, the microcracks that define the primary crack direction are not uniformly oriented, and they deviate from the principal tensile direction. Therefore, the microcracks are scattered in the neighboring directions as well as in the principal tensile direction, and they make it

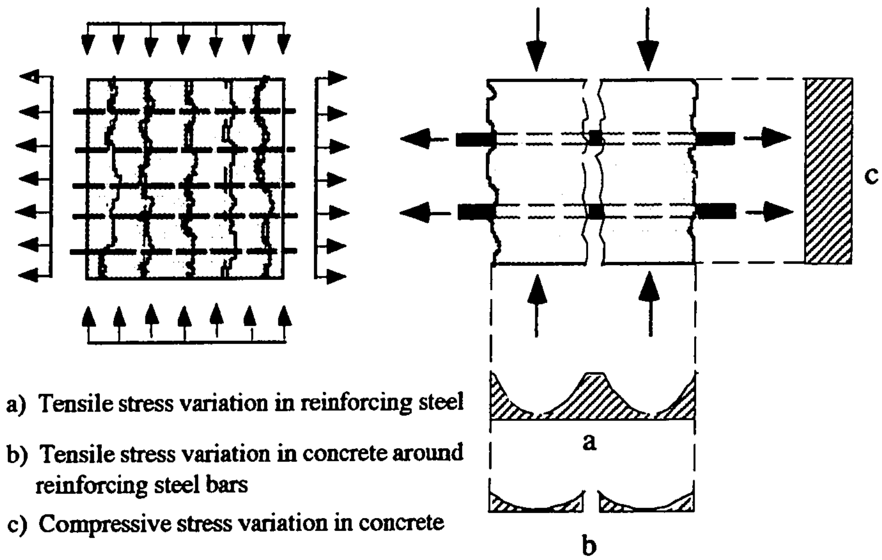


Figure 2.1 Stress variation in cracked reinforced concrete

easy for secondary cracking to occur. Accordingly, if the stress-strain state of concrete changes during loading, concrete experiences successive cracking in rotating principal tensile axes.

To model this complex concrete cracking process with conventional normal and shearing stress-strain relations, the crack orientation is usually idealized. Using the concept of smeared cracking, two different orthotropic axes models were developed, based on different assumptions for the variation of crack orientation during loading history. These are the fixed-crack model and the rotating-(or fictitious-) crack model.

In the fixed-crack model, once cracking occurs in a principal tensile direction, the crack direction does not change until the crack closes. In the rotating-crack model, the crack orientation rotates to principal stress axes or principal strain axes, depending on the assumption made. In both models, only one crack direction is allowed in an equilibrium condition; the orthotropic axes coincide with that crack direction. Also, shear stiffness is used to represent the effects of aggregate interlock and friction across cracks.

Shear panel tests under uniform shear [38] show that after primary cracking, the tension stiffening stresses in rotating principal axes are much larger than those in direct tension. If the orientation of principal axes does not change during the loading history, the tension stiffening stress is almost the same as that in direct tension. This phenomenon implies that the deviation of principal axes from the primary crack direction increases the tension stiffening stress. Also, it should be noted that, whether or not the principal axes rotate, the principal compressive stress-strain relations are the same as the uniaxial compressive stress-strain relations, including the compression softening effects due to crack opening.

Obviously, once primary cracking occurs in a principal axis, the crack orientation does not change during the loading history. However, under new equilibrium or compatibility conditions, aggregate interlock transfer shear forces across cracks. Accordingly, the stress-strain states of concrete change, and principal axes are established in directions different from the primary crack orientation. In the current principal tensile axis, if the principal tensile stress approaches the cracking stress, secondary cracking occurs. Since the primary crack opening contributes to the principal stress and strain, and since the secondary cracking occurs in the principal tensile axis, the magnitudes of principal tensile stresses in the current principal axis depend on the contributions of primary and secondary cracks. Since secondary cracking in the current principal axis requires tensile cracking stress, the tension stiffening stress becomes larger than that in direct tension at the same tensile strain.

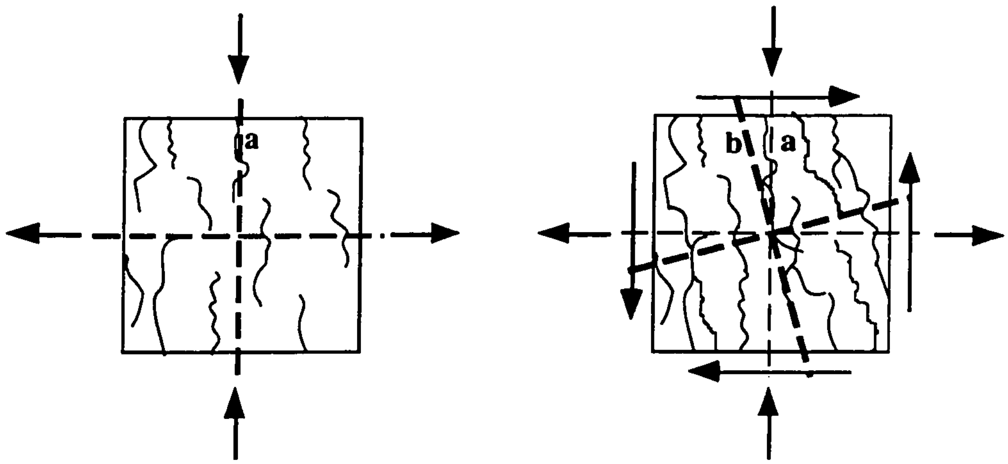
On the other hand, in cases when the nonlinear behavior of the principal compressive stress-strain relation affects overall reinforced concrete strength, the tension stiffening stress and the corresponding material strain are negligible compared with the compressive stress and strain. Also, the tension stiffening stress induced by the bonding action of reinforcing steel is localized around the reinforcing steel and the cracking zone (Figure 2.1). Since this tension stiffening stress and the corresponding material strain do not significantly affect the principal compressive stress-strain relation, the stress-strain state of cracked concrete is almost uniaxial compression. Even if the principal axes rotate after cracking, the principal tensile stresses are less than the tension cracking stress, and the uniaxial compression stress-strain state will therefore be maintained.

As mentioned before, the fixed crack model allows only primary cracking, and orthotropic axes are aligned with the primary crack direction until the primary cracks close. To consider shear behavior, such as aggregate interlock, the fixed crack model uses an effective shear stiffness, which does not involve the material strength of concrete. Therefore, the combination of the shear, compressive, and tensile stiffnesses in the primary crack direction cannot represent the material behavior of concrete under changes in equilibrium condition. Thus, the principal tensile stress induced by the shear stiffness may exceed the cracking stress, and the principal compressive stress-strain relation may be inconsistent with the material behavior of concrete. In fact, it is almost impossible to consider the material shear stiffness in terms of average stress and strain because the average strain includes crack opening in addition to material strain.

On the other hand, in the rotating crack model, orthotropic axes rotate to the current principal axes during the loading history. Once the principal tensile axis rotates from the primary crack direction or from the previous principal axis, the primary crack is assumed to rotate to the current principal tensile axis, and secondary cracking due to shear transfer between crack surfaces is neglected. Therefore, the rotating crack model underestimates tension stiffening stresses in rotating principal axes because the contribution of secondary cracking stress to the tension stiffening stresses is neglected. However, the rotating crack model maintains the uniaxial stress-strain relation in principal compressive axes. Accordingly, if the tension stiffening stress is negligible compared with the compressive stress, the rotating crack model gives a more reasonable behavior of cracked concrete than the fixed crack model.

To improve the above-mentioned shortcomings of the fixed-crack and the rotating-crack models, this research proposes a rotating orthotropic axes model with successive cracking. The fixed-and rotating-crack models idealize the primary crack direction in an equilibrium condition and they make the orthotropic axes coincide with that direction. In the proposed approach, the crack direction is not idealized. Instead, it is assumed that concrete cracking occurs progressively as the principal tensile axes rotate. The progressive cracking process due to primary and secondary cracking continuously gives behavioral directionality of concrete in rotating principal axes. Thus, the orthotropic axes rotate to the principal axes during loading (Figure 2.2). The proposed cracked concrete model defines the following material behavior:

1) If a tensile stress approaches the tension cracking stress in a principal tensile axis, primary cracking occurs, and its orientation is fixed to the principal tensile axis. Under further loading, if the principal axes rotate from the primary crack direction, the orthotropic axes follow the principal axes.



a) previous orthotropic axes

b) new orthotropic axes

Figure 2.2 Rotation of orthotropic axes under changes in equilibrium condition

2) In the principal tensile axes, since the increase in principal tensile strain requires secondary cracking in addition to the primary crack opening, the tension stiffening stresses in the orthotropic axes are determined by the contributions of secondary cracks due to reinforcement which remains elastic.

3) In the principal compressive axes, since the tension stiffening stress and the corresponding material strain do not significantly affect the principal compressive stress-strain relation, the stress-strain state of concrete is almost uniaxial compression. Therefore, the principal compressive stress-strain relation is defined by the uniaxial compressive stress-strain curve, including the compression softening due to tensile cracking.

2.4 Orientation of Orthotropic Axes under General Loading

In the existing model [38], it is assumed that principal stress axes coincide with principal strain axes, and that the two-dimensional stress-strain relation is defined by two equivalent uniaxial stress-strain curves on orthotropic axes that rotate to the principal axes during loading history. This assumption simplifies the definition of two-dimensional stress-strain relation because it eliminates the controversy over whether the orientations of orthotropic axes follow the principal stress axes or the principal strain axes. Also, since principal stress axes coincide with principal strain axes, there is no need to define shear stiffness. As a result, the two-dimensional stress-strain relation depends only on the total stress-strain relations of equivalent uniaxial stress-strain curves in the principal axes.

Although the above assumption is made for analytical convenience, it gives reasonable estimates of the actual orientation of principal axes obtained from panel tests under uniform shear [38]. According to those experiments, principal stress axes do not deviate significantly from principal strain axes until tensile cracks open wide. Though the cracks are wide, the orientations of principal axes in the analysis are close to the average directions of principal stress axes and principal strain axes in the experiments.

Under cyclic loading, shear deformation or strain in crack surfaces does not induce shear stress as the number of load cycles increases because of fatigue damage at the crack surfaces. Thus, the deviation of the two principal axes increases as the crack opening increases. If fatigue damage is severe, just after unloading, principal stress axes can deviate momentarily from principal strain axes by as much as 90 degrees. Apart from such extreme cases, as crack width increases, the deviation of principal stress and strain axes gradually increases. As cracks close, the principal axes again coincide.

For numerical convenience, the proposed material model uses the assumption that principal stress axes coincide with principal strain axes. Stevens et al. [34] failed to predict cyclic structural behavior using this assumption. In Chapter 7.0, the proposed material model will be used to demonstrate the validity of the assumption for cyclic behavior by verifying material and structural behavior.

2.5 Compressive Cyclic Behavior (Compression Damage Surface)

After cracking, concrete struts form in the primary crack direction. The concrete struts resist compressive stresses in the crack direction. Also, tension stiffening stresses are induced by bond of reinforcement across the crack. However, when compression failure affects the overall strength of reinforced concrete, the tension stiffening stress is negligible compared with the compressive stress. Even if the principal axes deviate from the primary crack direction or the concrete strut direction, the principal tensile stress will not exceed the tensile cracking stress because of secondary cracking. Therefore, the stress state of concrete strut is close to uniaxial compression.

As unloading occurs, the compressive stress in the concrete struts decreases and the cracks close. Under reversed loading, new concrete struts with closed cracks form in the previously unstressed direction. The new concrete struts are in compression. In the direction of the previous concrete struts, new cracks open, and concrete remains unstressed during reloading. During cyclic loading, concrete retains its uniaxial compressive stress state, even though the orientation of the uniaxial compression changes. Consequently, concrete experiences a series of uniaxial compressive stress states with different orientations under both monotonic and cyclic loading.

To eliminate the directional characteristics of material damage, the nonlinear behavior of plain concrete is usually defined by the relation of stress and strain invariants which are composed of principal stresses and strains. If concrete maintains uniaxial stress states, and if the uniaxial stresses have a uniform magnitude during rotation of the principal axes, the invariants due to the uniaxial stress and strain also maintain uniform magnitudes. In other words, although the principal axes rotate during loading, the magnitudes of the invariants depend on the magnitudes of uniaxial stress and strain regardless of their orientation. Accordingly, the relation of stress and strain invariants directly implies the relation of uniaxial stress and strain, and the nonlinear cyclic behavior of concrete depending on the invariants can be defined by equivalent uniaxial stress-strain relation in principal axes rotating during loading history.

Actually, the nonlinear stress-strain relation defined by the invariants determined by experimental uniaxial stress-strain curves under direct compression. Therefore, the experimental uniaxial stress-strain curves are used without modification for equivalent uniaxial stress-strain curves in the rotating principal axes. The compressive uniaxial stress-strain relation becomes the compressive principal stress-strain relation in the rotating principal axes. This can be verified by test results given by Vecchio [38]. According to the test results, the equivalent uniaxial stress-strain curve given by Vecchio follows precisely the principal compressive stress-strain relations in tests, whether or not the principal axes rotate. This verifies the fact that the uniaxial compressive stress-strain relations representing material damage remain the same in rotating principal axes, as long as the uniaxial states are maintained.

In the equivalent uniaxial stress-strain curves of the proposed material model, compressive stress is defined in terms of the corresponding compressive strain; the boundary between unloading and loading in cyclic behavior is defined by the maximum compressive strain. The maximum strain determines the magnitude of the compression damage surface. The damage surface has uniform or isotropic magnitudes in all directions because if concrete maintains uniaxial stress-strain states in any direction, the amount of uniaxial strain corresponding to the current strain invariants is uniform in all directions regardless of the orientation of uniaxial stress-strain state. Accordingly, the isotropic damage surface defined in principal strain space can be directly used in terms of uniaxial strain.

In summary, the compressive nonlinear behavior of cracked concrete is defined in the following way: Since cracked concrete maintains uniaxial compressive stress states, the experimental uniaxial nonlinear stress-strain curve is used for the equivalent uniaxial stress-strain relation in rotating principal axes. Once a compressive principal strain exceeds the compression damage surface, the equivalent stress-strain relation lies on the envelope curve or loading curve, and the compression damage surface expands uniformly in all directions to the magnitude of the compressive principal strain. As long as compressive strains remain inside the damage surface, the compressive damage surface maintains the same magnitude in all directions. Under unloading in compression, the equivalent stress-strain relation lies on the unloading and reloading curves which connect the compression and tension damage surfaces.

2.6 Tensile Cyclic Behavior (Tension Damage Surface)

If tension cracking occurs under a tension-compression stress state, the damage localizes in the principal tensile axis and the damage contribution obviously vanishes in the orthogonal axis. Under reversed loading, since the orthogonal axis has no tension crack damage under the previous loading, the orthogonal axis should experience new tensile cracking. If a principal axis in which the current principal tensile stress and strain exist experienced tensile crack damage under a previous loading history, the principal stress-strain relation would exist on the reloading curve until the principal strain reached the maximum tensile strain or the tension damage surface. Therefore, for cyclic behavior of cracked concrete, a tension damage surface is required to define the anisotropic damage distribution in two-dimensional space, which provides the boundary between unloading and loading behavior.

The initial tension damage surface forms due to primary cracking; the damage contribution due to current tensile cracking is concentrated on the current principal tensile axis and decreases sharply in neighboring directions. The tension damage surface expands from the initial tension damage surface as a tensile strain exceeds the surface. If a tensile strain exceeds the current tension damage surface, a damage influence surface which is the same shape as the initial tension damage surface, forms due to the tensile strain. If the damage influence surface exceeds the current tension damage surface in a given direction, the tension damage surface expands to the damage influence surface in that direction. Otherwise, the tension damage surface retains its previous shape. Thus, if the principal axes rotate during loading, the tension damage surface become anisotropic, different from that of the isotropic compression damage surface.

In the proposed cracked concrete model, tension behavior is defined by the tension damage surface in the following way: Once tension cracking occurs in a principal axis, the tension damage surface of the primary cracking forms in the principal axis and the neighboring directions. If a principal strain exceeds the initial tension damage surface under further loading, the tension damage surface expands to the damage influence surface due to the principal tensile strain. Under unloading, the tension damage surface does not change, and the equivalent stress-strain relation exists on unloading curves which connect the tension damage surface and the compression damage surface.

As reloading occurs, the equivalent stress-strain relation lies on the reloading curves until the tensile strain reaches the tension damage surface. If the tensile strain exceeds the tension damage surface, the equivalent stress-strain relation follows the tensile envelope curve or the loading curve, and the tension damage surface expands.

2.7 Definition of Two-Dimensional Stress-Strain Behavior of Cracked Concrete.

In the proposed cracked concrete model, cracked concrete behavior is idealized based on several basic assumptions:

1) The concept of smeared cracking is assumed to be valid. The smeared crack is regarded as a continuous material strain. Based on the concept of smeared cracking, the tensile stress and strain of cracked concrete are defined in terms of average stress and strain across tension cracks.
2) Principal stress axes coincide with principal strain axes.
3) Cracked concrete is idealized as an orthotropic material, and the orthotropic axes coincide with principal axes. The progressive cracking process due to primary and secondary cracking continuously gives behavioral directionality of concrete in rotating principal axes. Accordingly, the orthotropic axes rotate to the principal axes during loading.
4) In the orthotropic axes, the equivalent uniaxial stress-strain relations in two orthogonal principal axes are uncoupled in terms of material strain. In cracked concrete, the tension stiffening stress is negligible compared with the compressive strength of concrete, and the tension stiffening stress induced by bonding action of reinforcing steel is localized around the reinforcing steel and the cracking zone. Accordingly, the reciprocal effect of the two stress-material strain relations is neglected. To address the effect of crack opening, the equivalent uniaxial stress-strain relations are coupled in terms of average strain.

On the basis of the above assumptions, the general behavior of the proposed cracked concrete model is defined in the following way:

1) The two-dimensional stress-strain relation is defined by two equivalent uniaxial stress-strain curves in orthotropic axes. The orthotropic axes rotate to current principal axes during loading history.

2) The equivalent uniaxial stress-strain curve consists of envelope curves (loading curve) and unloading-reloading curves connecting the envelope curves (Figure 2.3). The compressive envelope curve depends on the uniaxial stress-strain relation, including the compression softening effect

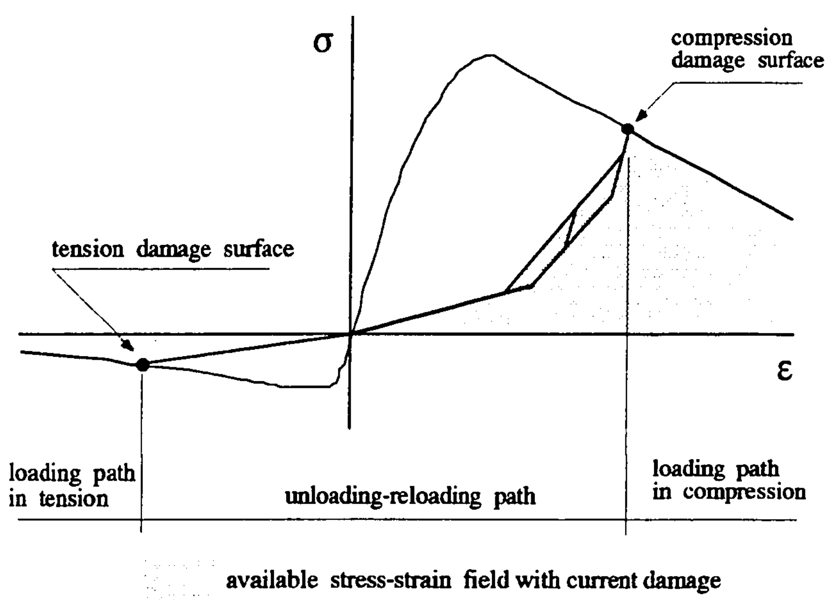


Figure 2.3 Equivalent uniaxial stress-strain curve

due to crack opening. The tension stiffening stress of the tensile envelope curve is determined by the influence of each reinforcement layer which remain elastic.

3) The equivalent uniaxial strain induces either isotropic damage in compression or anisotropic damage in tension. If the equivalent uniaxial strain exceeds compression or tension damage surface, the damage surface expands according to its expansion rule, and the equivalent stress-strain relation follows the compressive or tension envelope curve or loading curve.

4) If the equivalent uniaxial strain exists inside the damage surfaces, the equivalent stress-strain relation exists on the unloading-reloading curves connecting compressive and tensile envelope curves at the damage surfaces.

3.0 CONSTITUTIVE MODEL FOR CRACKED CONCRETE

3.1 General

In Chapter 2.0, on the basis of that idealization of cracked concrete behavior, the concepts of the proposed cracked concrete model were introduced. In this chapter, the stress-strain relations of the cracked concrete model will be specified, and the general behavior will be presented in detail.

In the proposed cracked concrete model, cracked concrete is regarded as an orthotropic material showing tensile and compressive behavioral characteristics, and with orthotropic axes that coincide with current principal axes. In the orthotropic model, the interaction between the material compressive and tensile strains is neglected, as mentioned in Section 2.2. Accordingly, the tensile and compressive behaviors in orthotropic axes depend only on principal tensile strains representing cracking opening, and can be defined by independent equivalent uniaxial stress-strain curves in tension and compression. In other words, the two-dimensional stress-strain relation is defined by two independent equivalent uniaxial stress-strain curves in orthotropic axes which rotate to principal stress axes.

The previous monotonic model of cracked concrete [38] includes compression softening and tension stiffening effects due to crack opening. The proposed cracked concrete model adds the following behavioral definitions for the general behavior of cracked concrete:

1) A two-dimensional tension stiffening curve considering the progressive cracking process;

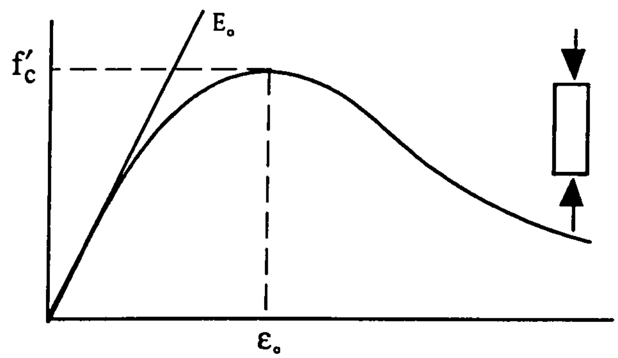


Figure 3.1 Uniaxial compressive stress-strain curve

2) Equivalent uniaxial cyclic stress-strain curves, composed of envelope and unloading-reloading curves;

3) Tensile and compressive damage surfaces in two-dimensional strain field, defining the boundary between loading and unloading.

3.2 Equivalent Uniaxial Stress-Strain Curve in Compression

Stress-strain relations in principal compressive axes are defined by an equivalent uniaxial compressive stress-strain curve, composed of envelope and unloading-reloading curves. Since the stress-strain states of concrete are almost uniaxial, the fundamental form of the envelope curve is based on a compressive uniaxial stress-strain curve, shown in Figure 3.1. However, it has been acknowledged by several researchers [11, 38] that, unlike the pure uniaxial stress-strain relation in compression, the compressive strength of cracked concrete significantly decreases due to tension cracking.

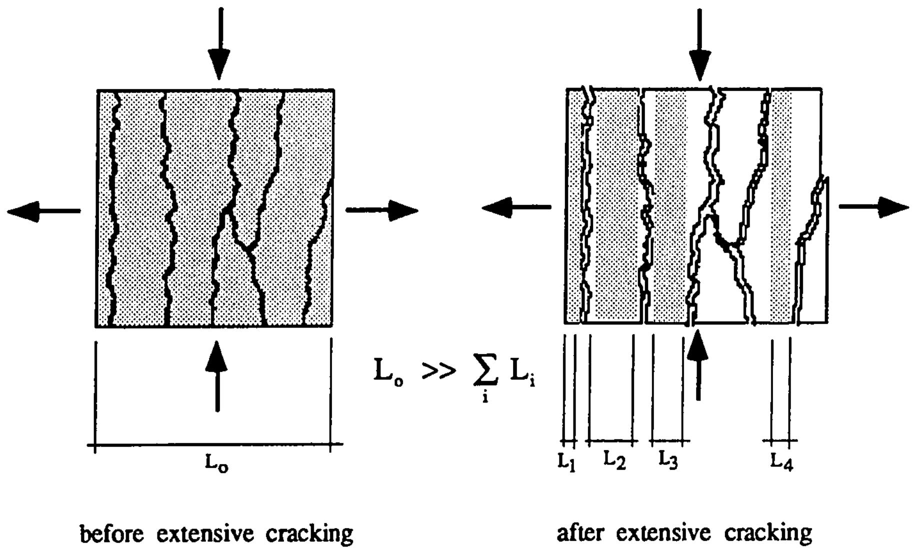


Figure 3.2 Compression softening effect

After tension cracking, reinforcing steel resists the tensile stress, and concrete struts separated from each other by tension cracks resist compressive stress (Figure 2.1). Usually, primary crack directions are visualized as coinciding with the current principal tensile direction. However, the microcracks composing the primary crack are not uniformly oriented; they deviate from the principal tensile direction. Therefore, as the cracks widen, the concrete struts are disconnected and finally crush (Figure 3.2). In other words, the deviation of microcracks from the principal axes reduce the effective area of concrete struts. Accordingly, the compressive strength of concrete decreases due to crack opening. This phenomenon is called compression softening due to crack opening.

Vecchio and Collins [36, 38] verified this phenomenon clearly by shear panel tests under in-plane loading. They showed that tension cracking causes compression

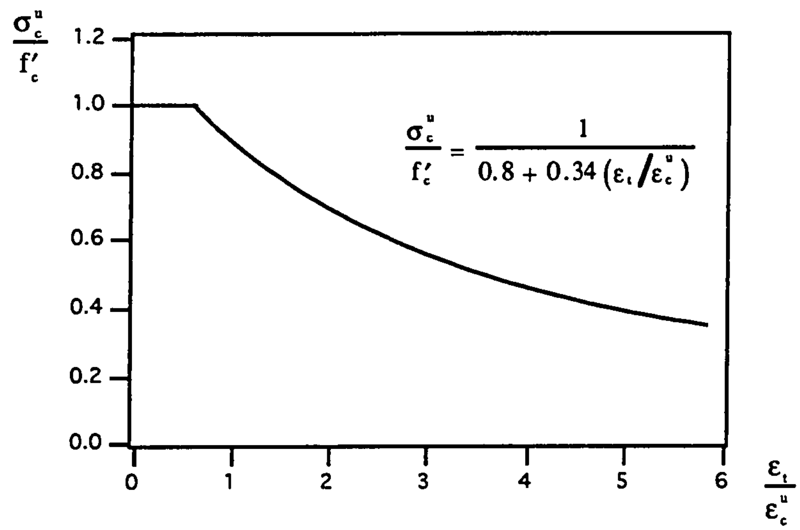


Figure 3.3 Compression softening equation proposed by Vecchio [38]

softening in tension-compression stress fields, and that the compressive strength depends on the crack width. Finally, they proposed an equivalent uniaxial stress-strain curve including the compression softening effect. According to their proposed stress-strain relation, the compressive strength in a compressive principal axis decreases as the principal tensile strain representing the current crack width increases in the orthogonal principal tensile axis. The relation between the compressive strength and the principal tensile strain is defined by the following equation (Figure 3.3):

$$\sigma_c^u = \frac{f'_c}{0.8 - 0.34\,(\varepsilon_t / \varepsilon_c^u)} \quad \text{and} \quad \sigma_c^u \le f'_c, \tag{3.1}$$

where f'_c is the cylinder strength, σ_c^u is the compressive strength, ε_c^u is the compressive strain corresponding to σ_c^u, and ε_t is the principal tensile strain.

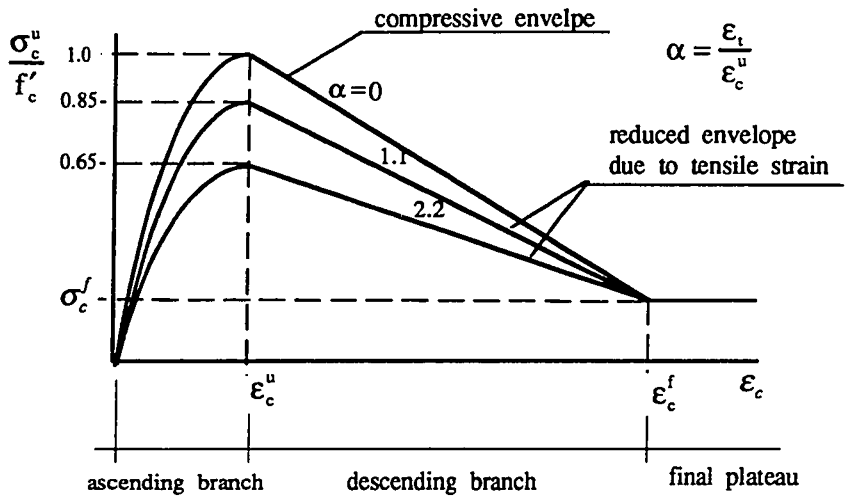


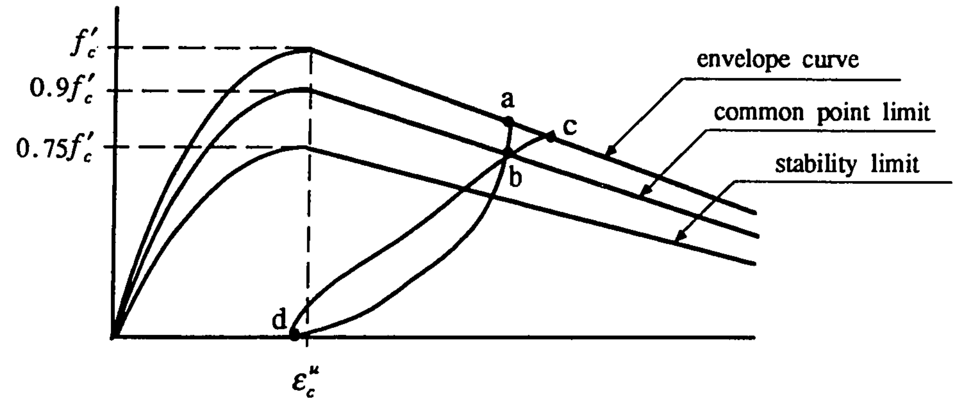
Figure 3.4 Compressive envelope curve

Vecchio and Collins applied this empirical equation for the compressive equivalent uniaxial stress-strain curve in their orthotropic axes model.

As shown in Figure 3.4, the proposed compressive envelope curve consists of three parts: an ascending branch, a descending branch, and a final plateau. The ascending branch is defined by a widely used parabolic equation for uniaxial stress-strain relations in compression [32, 36]:

$$\sigma_c = \sigma_c^u \left[2\left(\frac{\varepsilon_c}{\varepsilon_c^u}\right) - \left(\frac{\varepsilon_c}{\varepsilon_c^u}\right)^2 \right]. \tag{3.2}$$

In the proposed envelope curve including compression softening due to crack opening, the compressive strength, σ_c^u, in a principal compressive axis, depends on the principal tensile strain, ε_t, in the orthogonal axis, as defined by Equation 3.1.



a) the maximum strain
b) common point
c) restoring point
d) permanent strain

Figure 3.5 Cyclic stress-strain relation in uniaxial compression proposed by Karsan and Jirsa [22]

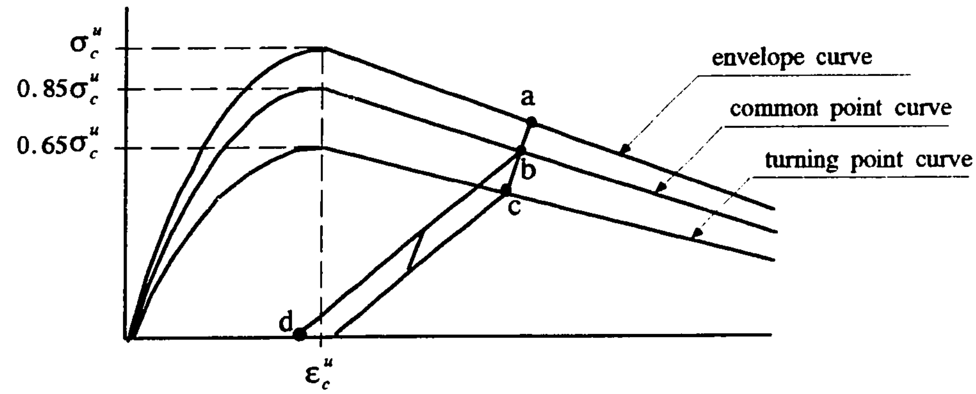
The descending branch is defined by a linear equation connecting the ascending branch and the final plateau. Beyond the final strain, ε^f_c, the compressive stress is assumed to be a constant, σ^f_c. The stress-strain relation in the descending branch represents material ductility which usually depends on the confinement due to reinforcement. Therefore, the final stress and strain should be appropriately determined according to the confinement. In the analysis program developed here, the final stress and strain are user-specified.

Though the unloading-reloading behavior of cracked concrete is very complex, this research uses two compressive cyclic models based on the unloading-reloading behavior in uniaxial compression: a hysteresis model, and a simplified model.

Karsan and Jirsa [22] performed an experiment to characterize unloading-reloading behavior under repeated uniaxial compression. According to their tests, the unloading-reloading stress-strain relation exists in the stress-strain field bounded by envelope curves, and the cyclic behavior can be defined by the unloading-reloading curves connecting several key points. As shown in Figure 3.5, these key points are the maximum strain, the common point, the permanent strain, and the restoring strain. The maximum strain on the envelope curve defines the boundary between loading and unloading. The permanent strain is irrecoverable under complete unloading, and is defined by a function of the maximum strain. At the common point, the reloading curve crosses over the unloading curve. According to the experiments of Karsan and Jirsa, the common point is not fixed, but moves depending on the previous unloading-reloading history. If unloading-reloading occurs beyond the common point limit, the common point lies at the common point limit. If repeated loading occurs within the common point limit, the common point stabilizes at the stability limit. The common point limit and the stability limit lie on curves whose shapes are equivalent to the compressive envelope curve with reduced maximum strengths of $0.9f_c'$ for the common point limit, and $0.75f_c'$ for the stability limit.

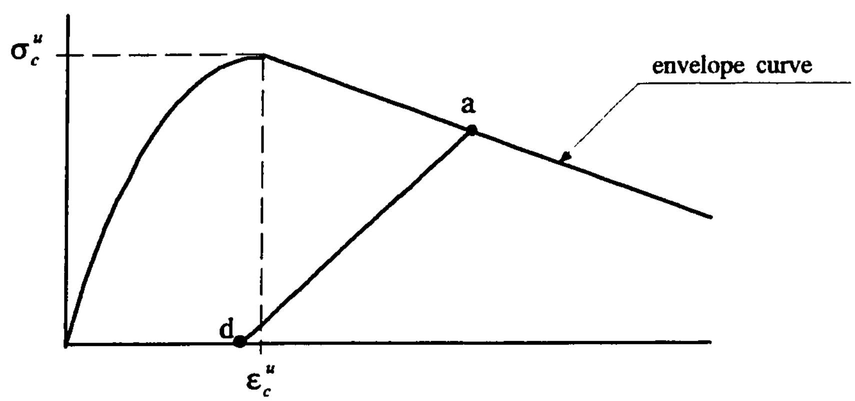
According to the above experiments, the unloading-reloading curve is defined based on the compression envelope curve. However, in the equivalent uniaxial stress-strain curve, since the envelope curve including compression softening depends on the current principal tensile strain, it is difficult to define the key points in the same way as the experimental results. As shown in Figure 3.6 (a), the proposed hysteresis model uses a simplified unloading-reloading behavior based on the cyclic model of Darwin and Pecknold [15, 16, 17].

The common point is set to the common point limit. The stability limit is defined as the turning point. At the turning point, the stiffness of the unloading curve changes from the initial elastic stiffness to a degraded stiffness. Eliminating the



a) the maximum strain
b) common point
c) turning point
d) permanent strain

(a) Hysteresis curve



(b) Simplified curve

Figure 3.6 Proposed cyclic stress-strain curve in principal compressive axes restoring point, the reloading curve beyond the common point follows the same path as the unloading curve. The unloading-reloading curves are composed of a series of straight lines connecting the key points. In the stress-strain field bounded by the unloading-reloading curves, the stress-strain relations follow the transition curve connecting the unloading and reloading curves. When the compressive strength of the envelope curve decreases by the tensile strain in the orthogonal axis, the maximum strengths of the common point curve and the turning point curve also decrease proportionally to the reduced compressive strength of the envelope curve. The reduced maximum strengths are $0.85f_c'$ for the common point curve and $0.65f_c'$ for the turning point curve, slightly different from the proposal of Karsan and Jirsa. The permanent strain, ε_c^p, is defined by the following function of the maximum compressive strain, ε_c^m, and the strain corresponding to the ultimate stress, ε_c^u:

$$\varepsilon_c^p = \varepsilon_c^u \left[0.145 \left(\frac{\varepsilon_c^m}{\varepsilon_c^u} \right) + 0.13 \left(\frac{\varepsilon_c^m}{\varepsilon_c^u} \right)^2 \right] \quad \text{for} \left(\frac{\varepsilon_c^m}{\varepsilon_c^u} \right) \leq 3.0 \qquad (3.3.a)$$

$$\varepsilon_c^p = \varepsilon_c^u \left[-1.305 + \left(\frac{\varepsilon_c^m}{\varepsilon_c^u} \right) \right] \quad \text{for} \left(\frac{\varepsilon_c^m}{\varepsilon_c^u} \right) > 3.0 \qquad (3.3.b)$$

In the proposed simplified model as shown in Figure 3.6.(b), the unloading-reloading behavior is simplified by a straight line connecting the maximum compressive strain and the permanent strain, so that the stress-strain paths of unloading and reloading are the same.

3.3 Equivalent Uniaxial Stress-Strain Curve in Tension

Like the compression envelope curve, the proposed tensile envelope curve consists of an ascending branch, a descending branch, and a final plateau (Figure 3.7). The ascending branch defines elastic tensile stress-strain relations before cracking, and the descending branch defines post-cracking behavior or tension stiffening behavior.

Until now, tension stiffening effects have been studied primarily for uniaxial stress states; current tension stiffening models for two-dimensional stress states use either uniaxial tension models or empirical equations.

In two-dimensional stress states, the stress states of concrete can change after initial cracking. In the new equilibrium conditions, secondary cracking occurs in the new principal axes, which differ from the previous principal axes. Accordingly, the tension stiffening behavior associated with new equilibrium should differ from that associated with the previous equilibrium condition. This research proposes a two-dimensional tension stiffening model based on the variation of two-dimensional strain states.

First, uniaxial tension stiffening behavior will be discussed, as the basis for the proposed tension stiffening model.

In plain concrete, tension cracking occurs abruptly by the formation of one dominant crack. Therefore, as soon as cracking occurs, the cracking energy is quickly released, and concrete tensile stresses decrease sharply with respect to average strain (Figure 3.8). On the other hand, in reinforced concrete, bond with reinforcement prevents cracked concrete from releasing the existing tensile stress quickly. As a

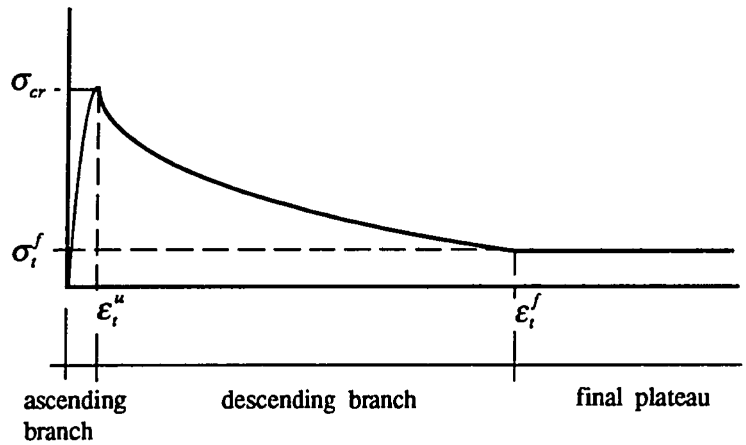


Figure 3.7 Tensile envelope curve

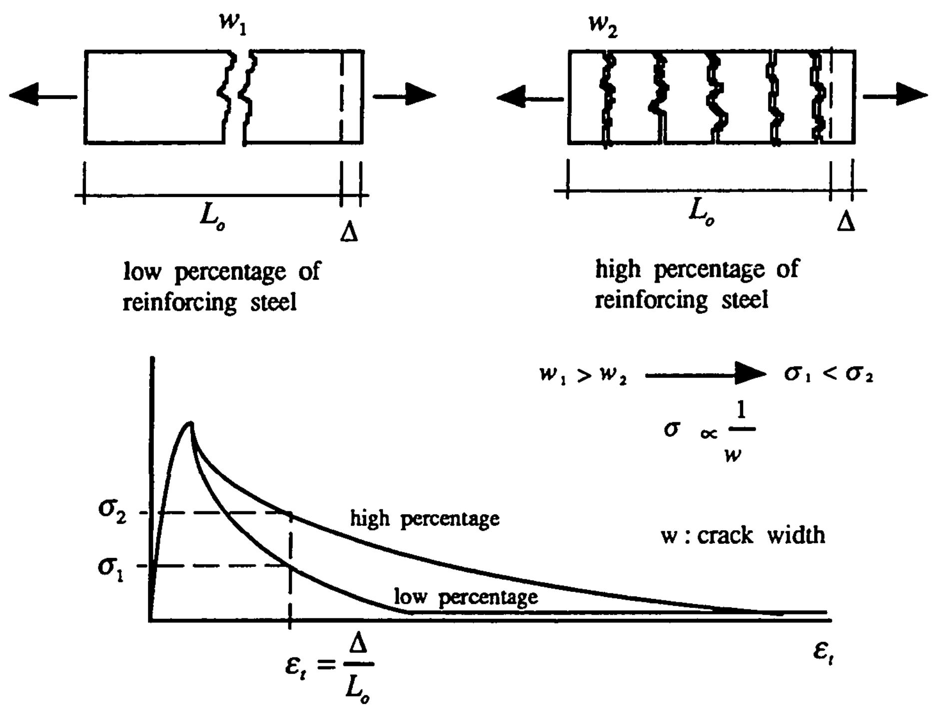


Figure 3.8 Tensile stiffening effect

result, tensile cracks spread over a large area, and the tensile stresses decease gradually. This phenomenon is represented by tension stiffening.

If the reinforcement between crack surfaces yields, the cracks widen, and new tensile cracking does not occur. Thus, the tension stiffening stresses rapidly disappear or retain very small amount of tensile stress. In the proposed tension stiffening model, the basic tension stiffening unit corresponding to a reinforcement layer is defined as a simple uniaxial tension stiffening model in Figure 3.9.

According to the shear panel tests performed at The University of Toronto [38], the details of which will be given in Chapter 7.0, tension stiffening stresses are much larger than those under uniaxial tension at the same tensile strain (Figures 7.2 - 7.9). Shear panel PV4 in Figure 7.2 is reinforced by two orthogonal reinforcement layers with the same reinforcement ratios. The uniform shear load is resisted by concrete in compression and reinforcement in tension. Since shear panel PV4 is isotropically reinforced, the principal stress directions do not change during loading. As the uniform shear increases, the reinforcement quickly approaches yield.

The other shear panels are anisotropically reinforced by two orthogonal reinforcement layers with different reinforcement ratios. As a reinforcement layer yields first, the principal tensile axes gradually rotate to the direction of the other reinforcement layer, which remain elastic. As the principal axes deviate from the previous principal axes, the load capacities of the shear panels gradually increase. At the same time, principal compressive strains significantly increase. As a result, the reinforcement strain does not increase as fast as the principal tensile strains. Therefore, the tensile strain of the reinforcement still remains within the elastic range, even with a large principal tensile strain; the reinforcement in the elastic range induces the tension stiffening stress.

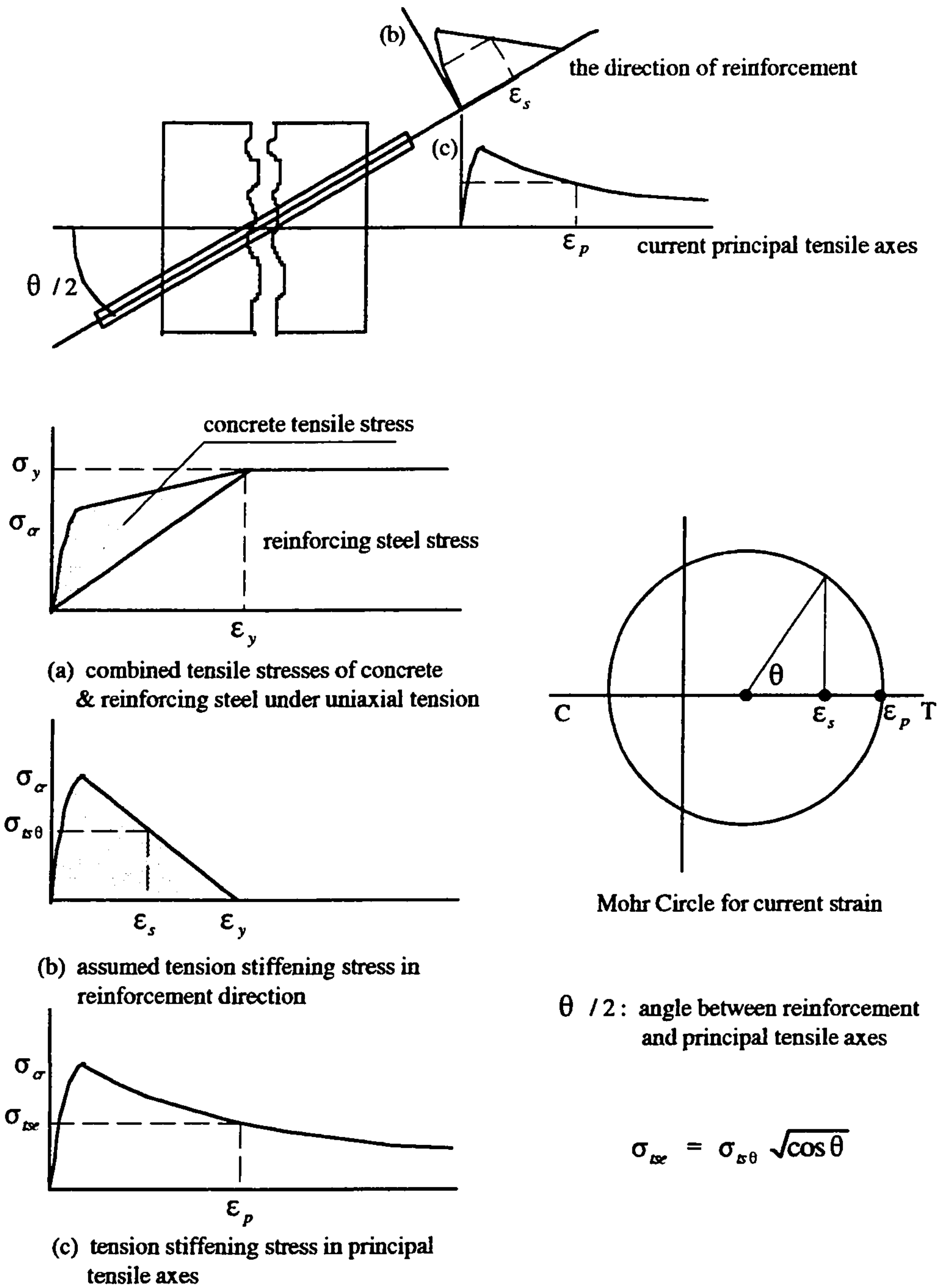


Figure 3.9 Proposed tension stiffening model showing its variation due to rotation of principal axes

According to uniaxial tension stiffening models, tension stiffening is affected by the reinforcing steel ratio, the yield stress and strain, the bar spacing, and the bar diameter. However, in two-dimensional space, it is difficult to assess accurately the influence of the above factors on the rotating principal axes. In the author's research, a simple tension stiffening model is proposed and it satisfies the following minimum requirements for two-dimensional tension stiffening effect:

1) Cracked concrete retains significant tension stiffening stresses as long as at least one reinforcement layer remains unyielded.
2) After yielding of reinforcement, the combined stresses of cracked concrete and the reinforcement should be the same as the yield stress of the reinforcement.

To idealize the two-dimensional tension stiffening stress-strain relation, it is assumed that each reinforcement layer has its own tension stiffening stress corresponding to the tensile strain in the reinforcement direction (Figure 3.9 (b)). The effect of the hypothetical tension stiffening stress, $\sigma_{ts\theta}$, on the principal tensile stress axes is defined as follows (Figure 3.9 (c));

$$\sigma_{tse} = \sigma_{ts\theta} \sqrt{\cos\theta} \,, \qquad (3.4)$$

where σ_{tse} is the equivalent tension stiffening stress in the current principal tensile axis, and θ is the angle between the current principal tensile axis and the reinforcement direction. The largest equivalent tension stiffening stress is assigned to the current tension stiffening stress.

In Figures 7.2 - 7.9, the proposed tension stiffening model is compared with the shear panel tests, Series PV and PB, and with previous analyses by Vecchio [38] and Stevens [7]. As shown in those figures, the proposed model results in reasonable tension stiffening stresses whether or not the principal axes rotate during loading. More detailed comparison with the shear panel tests will be given in Chapter 7.0.

According to experiments in direct tension, cyclic behavior in tension is very similar to that in compression. Therefore, the same definition of cyclic behavior as in compression is possible. However, since the small variation of tension stiffening stresses does not affect overall member behavior, the proposed tensile cyclic stress-strain behavior is simplified as follows. The maximum strain defining the boundary between loading and unloading is determined by the tension damage surface defined in Section 3.4. The secant which connects the maximum strain and the origin (or reference point) defines the unloading-reloading stress-strain relation as shown in Figure 3.10; hysteresis under repeated and cyclic loading is not considered.

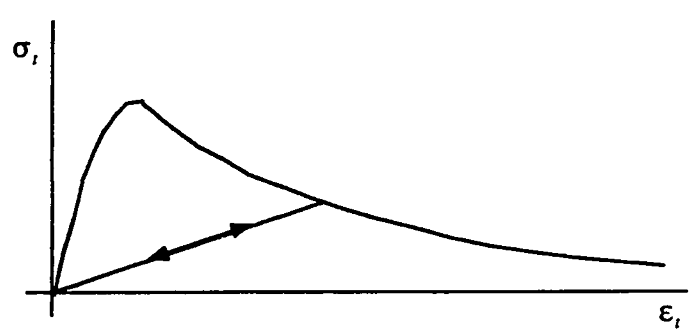
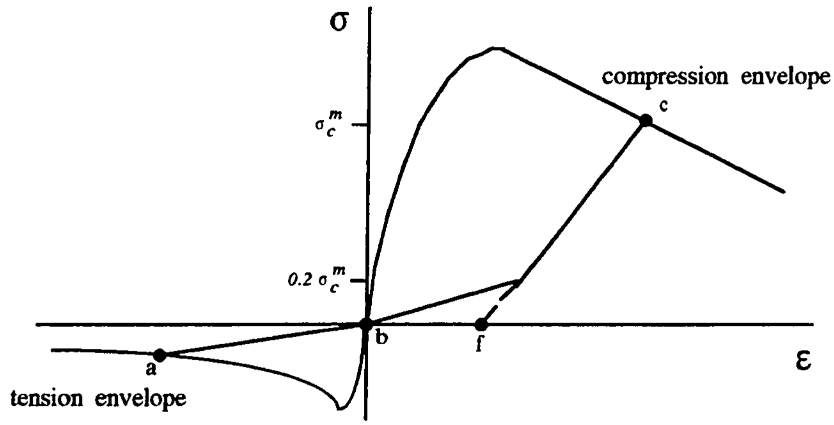


Figure 3.10 Tensile unloading-reloading curve

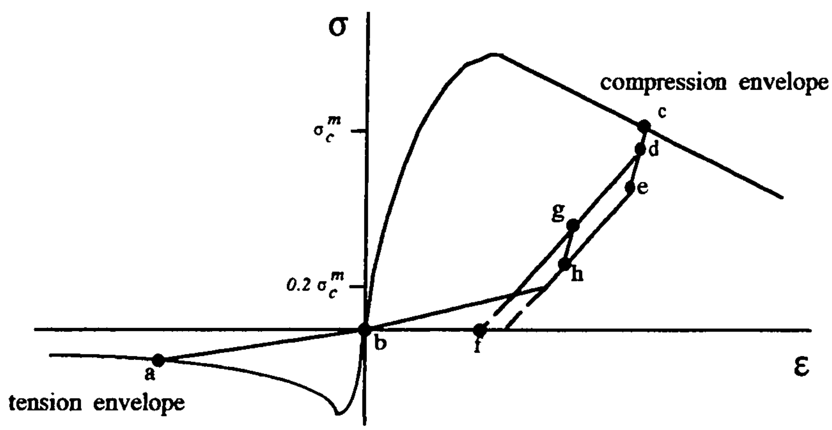
3.4 Definition of Cyclic Stress-Strain Relations

According to shear panel tests under uniform cyclic shear [34], as cracks close, the stress-strain relation is very complex due to contact at the crack surface and the interaction between cracked concrete and reinforcing steel. In Ref. 34, a complex cyclic model of cracked concrete is proposed. The model is developed from the experimental data which are interpreted on the basis of the concept of smeared crack and smeared reinforcement. However, the concept of smeared crack and smeared reinforcement has shortcomings in idealizing the interaction between multiple cracks and reinforcement, which will be explained in Section 4.1. In this study, since the complex stress-strain relation is not yet generalized, and for computational convenience, simplified cyclic models of cracked concrete is used.

In the proposed cracked concrete model, the compressive and tensile stress-strain relations in material principal axes are independent in terms of material stress-strain behavior. Accordingly, two-dimensional stress-strain relations are defined by the two independent equivalent uniaxial stress-strain curves in material principal axes or in principal stress axes. Based on the cyclic uniaxial compressive and tensile stress-strain relations previously defined in Sections 3.2 and 3.3, this research uses two cyclic equivalent uniaxial stress-strain curves; a simplified model and a hysteresis model. The simplified model in Figure 3.11(a) consists of the tension and compression envelope curves, and the unloading-reloading curves connecting the two envelope curves. The hysteresis model in Figure 3.11(b) uses the transition curve connecting unloading and reloading curves, in addition to the envelop curves and the unloading-reloading curves, so that the hysteresis model allows different unloading and reloading paths in compression.



(a) Simplified Model



(b) Hysteresis Model

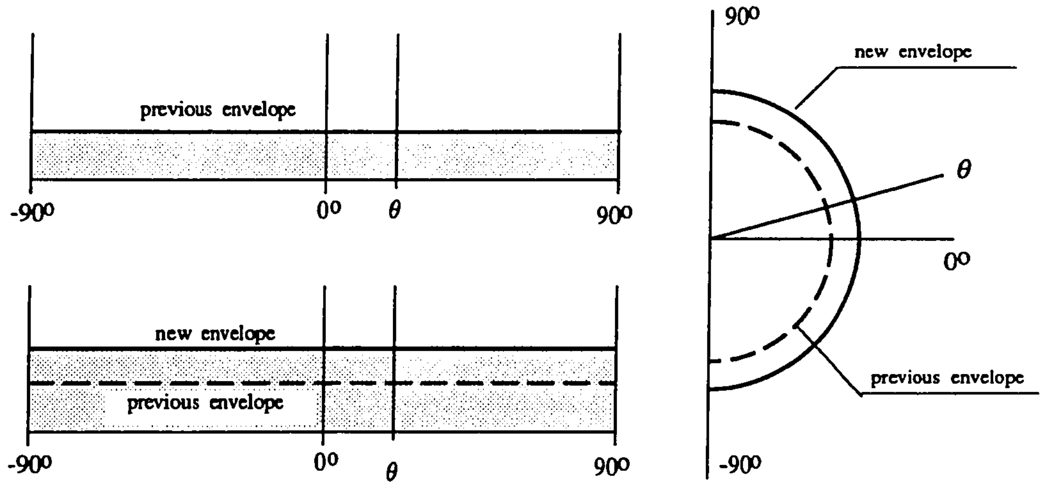
a) maximum strain in tension
b) reference point
c) maximum strain in compression
d) common point
e) turning point
f) permanent strain
g) unloading point
h) reloading point

Figure 3.11 Equivalent uniaxial stress-strain curve

As shown in Figure 3.11, the cyclic equivalent uniaxial stress-strain curve in a principal axis is defined by several key points (or strains), such as the maximum strains, reference point, permanent strain, common point, turning point, and unloading and reloading points. Since principal axes rotate during loading history, these key strains need to be defined in a two-dimensional strain field. For this purpose, the proposed cracked concrete model introduces tension and compression damage surfaces, a reference point surface, and unloading and reloading surfaces.

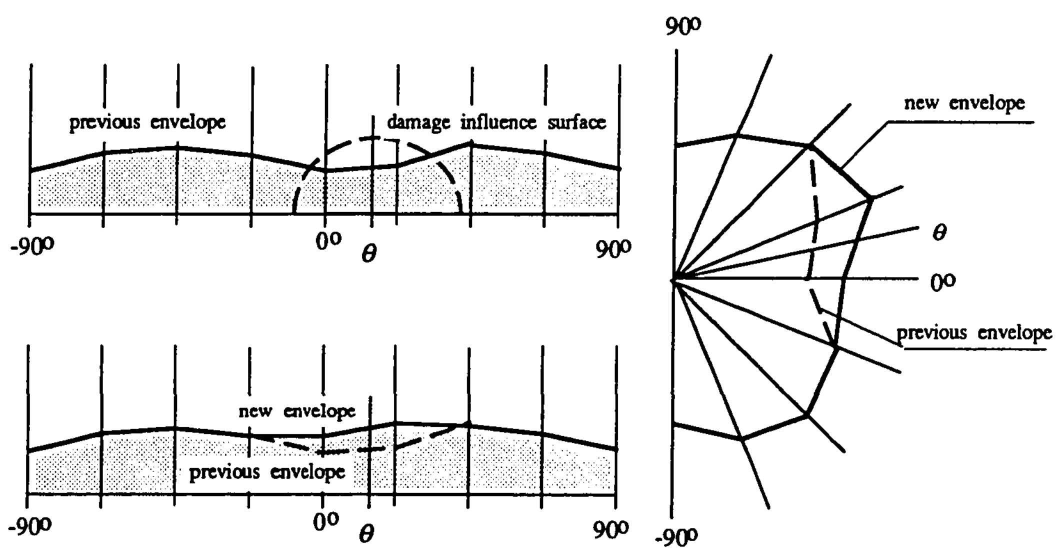
The compression damage surface defines the maximum strain representing compression damage in principal compressive axes. As mentioned in Section 2.5, though principal compressive axes rotate in a new equilibrium condition, the compressive strain representing the current concrete damage is uniform in all directions because the rotating principal axes maintain uniaxial stress-strain states in compression due to successive tensile cracking in principal tensile axes. Therefore, if a principal compressive strain exceeds the maximum strain or the compression damage surface, the surface expands isotropically to the magnitude of the compressive strain (Figure 3.12). Otherwise, the compression damage surface maintains the magnitude of the current maximum strain. In the same way, the unloading and reloading surfaces defined only in compressive stress-strain field are uniform in all directions.

On the other hand, tension cracking inducing tension damage is obviously limited to the current principal tensile axis and the neighboring directions. In the proposed model, the damage in the neighboring directions is defined by the damage influence surface due to current tension cracking or principal tensile strain. If a principal tensile strain exceeds the tension damage surface, the tensile strain forms its



θ : principal compressive axis

Figure 3.12 Compression damage surface in two-dimensional strain space



θ : principal tensile axis

Figure 3.13 Tension damage surface in two-dimensional strain space damage influence surface within 30 degrees in either side of the current principal tensile axis. The damage influence surface is defined by

$$\varepsilon^m_{t\Delta\theta} = \varepsilon_t^m \cos(3\Delta\theta) , \tag{3.5}$$

where ε_t^m is the maximum strain in the current principal tensile axis, and $\varepsilon^m_{t\Delta\theta}$ is the maximum strain in the direction deviating by $\Delta\theta$ from the current principal tensile axis. The tensile damage surface expands to the current damage influence surface. Since the damage influence surface is not uniform, the tension damage surface is anisotropic, unlike the compression damage surface. If principal tensile axes continue to rotate under further loading, and if the tension damage surface becomes more irregular, considerable computer memory is required to define the entire irregular surface. Thus, the proposed cracked concrete model designates eight reference directions in a two-dimensional strain field, each separated from each other by 22.5 degrees. In each reference direction, if the damage influence surface due to current principal tensile strain exceeds the tension damage surface, the tension damage surface expands to the current damage influence surface (Figure 3.13). The maximum strain in a principal tensile axis is linearly interpolated between the maximum strains or the tension damage surface in the reference directions. Although the interpolated maximum strain underestimates or overestimates the exact maximum strain, the discrepancy between the exact maximum strain and the interpolated maximum strain is indiscernible in member behavior.

As shown in Figure 3.11, the reference point is the starting point from which the tensile envelope initiates. If concrete experiences compression damage before tensile cracking, under unloading, the permanent strain due to the compression damage remains irrecoverable, and under reversed loading, the tensile strain relation starts at the permanent strain. Once the tensile strain exceeds the cracking strain, the reference point is set to the current permanent strain. If there is no compression damage before tension cracking, the reference point is set to the origin or zero strain. Afterwards, even though additional compression damage develops under further loading, the reference point does not change. The position of the reference point in a principal axis depends on the compression damage of concrete when the tensile strain in the principal axis initially exceeds the cracking strain. As with the tension damage surface, the reference strain is determined independently in eight reference directions. If the tensile strain in a reference direction exceeds the cracking strain, the reference strain in the direction is set to the permanent strain due to the current compression damage. The reference strain in a principal axis is determined by linear interpolation between the reference strains in the neighboring reference directions.

3.5 Strategy for Cyclic Behavior

As in Section 3.4, this section describes how the tension and compression damage surfaces are defined according to the progression of concrete damage, and how the stress-strain relation of cracked concrete is defined in two-dimensional space on the basis of those definitions.

The proposed cyclic stress-strain behavior is defined in three regions, divided by the maximum strains, and by the unloading and reloading points (Figures 2.3 and 3.11). Beyond the maximum strain in either tension or compression, the stress-strain relation follows the envelope curve. Between the maximum strains, the stress-strain relation exists in the stress-strain field bounded by the unloading and reloading curves. Between the unloading and reloading points, the stress-strain relation exists on the transition curve. Beyond the unloading and reloading points up to the maximum strains in compression and tension, the stress-strain relation follows the unloading-reloading curves.

According to the progression of concrete damage, the cyclic behavior of cracked concrete is classified into five developmental stages (Figure 3.14):

I) Elastic range without permanent damage;

II) Initial tension damage without compression damage;

III) Initial compression damage without tension damage;

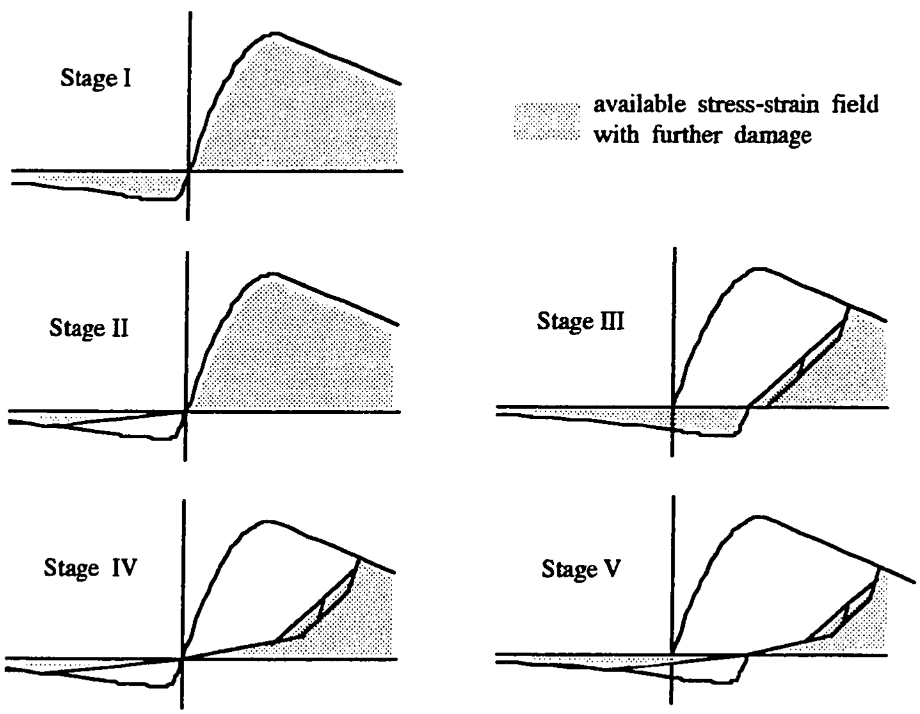
IV) Damage in both tension and compression after initial tension damage; and

V) Damage in both tension and compression after initial compression damage.

As mentioned in Section 3.3, since the material damage due to the maximum strains is determined in each reference direction, the damage development stage proceeds independently in each reference direction.

In the elastic range, the stress-strain relation either in tension or in compression follows the envelope curves under loading and unloading, and the compressive strain is completely recovered under unloading, without permanent strain.

If the tensile strain in a principal axis exceeds the tensile elastic limit, tension cracking (or damage) occurs, and Stage I shifts to Stage II. At this stage, the tension damage surface and the reference surface form in eight reference directions. Also, the primary crack direction becomes fixed to the principal tensile axis. As the tensile strains in rotating principal axes progress under further loading, the tension damage surface continues to expand in the reference directions, and the principal stress axes deviate from the principal strain axes. The reference surface is determined in the eight reference directions when the tension damage surface in each reference direction exceeds the cracking strain. When unloading occurs, the equivalent uniaxial stress-strain relations in the principal tensile axes follow the unloading curves



I) Elastic range without permanent damage

II) Initial tension damage without compression damage

III) Initial compression damage without tension damage

IV) Damage in both tension and compression after initial tension damage

V) Damage in both tension and compression after initial compression damage

Figure 3.14 Development of concrete damage connecting the tensile envelope curve at the maximum strain and the reference point (which is the origin in Stage II), and the principal stress and strain axes coincide. Under reversed loading in compression, the stress-strain relation follows the compressive envelope curve. If the strain exceeds the elastic limit in compression, this stage shifts to Stage IV.

At Stage III, beyond the elastic limit in compression, the compression damage surface forms uniformly in all directions. If unloading occurs, the stress-strain relation follows the unloading curve connecting the compressive envelope curve at the maximum strain and the permanent strain due to the current maximum strain. Also, the reloading point or reloading surface follows the current strain. If reloading occurs while the material is on the unloading path, the stress-strain relation follows the transition curve connecting the reloading point on the unloading curve and the unloading point on the reloading curve. Beyond the unloading point, the compressive stress-strain relation follows the reloading curve. Simultaneously, the reloading surface follows the compressive strain. Under reversed loading in tension, beyond the permanent strain due to the current maximum strain in compression, the stress-strain relation follows the tensile envelope curve. If the strain exceeds the elastic limit, then the tension damage surface, the reference surface and the primary crack direction are established, and the damage development Stage III shifts to Stage V. The magnitude of the reference surface is set to the current permanent strain.

In Stages IV and V, both the compression and tension damage surface exist, and the reference surface is permanently set. Between the damage surfaces or the maximum strains, the stress-strain relation follows the unloading-reloading curves. Beyond the maximum strains, it follows the envelope curves, and the damage surfaces continue to expand.

3.6 Stiffness Matrix

In the proposed cracked concrete model, the two-dimensional stress-strain relation is defined by two equivalent uniaxial stress-strain curves in orthotropic axes or principal stress axes. In most material models of concrete, concrete stress-strain behavior is defined by an incremental stress-strain relation which depends on the incremental material stiffness. In the proposed cracked concrete model, the equivalent uniaxial stress-strain curves are defined in terms of total stress and strain. Accordingly, the two-dimensional stress-strain relation in equilibrium and compatibility condition does not depend on the type of the material stiffness. The material stiffness only helps to achieve overall equilibrium and compatibility in nonlinear member behavior. Therefore, whatever type of material stiffness matrix is used, the two-dimensional stress-strain relation under a given load condition should be the same if the equilibrium and compatibility conditions are satisfied. However, stability and speed of convergence in satisfying equilibrium and compatibility conditions are critical in nonlinear computation, and they depend on the type of material stiffness used. In this research, it is found that, although the material behavior is defined by total stress-strain relation, an incremental stiffness formulation has the advantage of fast convergence for the proposed cracked concrete model. For that reason, the incremental stiffness matrix will be discussed here.

The incremental or tangent stiffness matrix is constructed in the current principal stress axes or orthotropic axes, and consists of the derivatives of the equivalent uniaxial stress-strain curves in two orthogonal principal axes and the shear stiffness. Actually, shear stresses and strains do not exist in the principal axes. However, the rotation of the principal axes induces shear stresses and strains from the current stress and strain combinations. The shear stresses and strains need to be eliminated in new principal axes. The shear stiffness is devised for that purpose.

In Figure 3.15, if the current principal stress axes rotate by $\Delta\theta_\sigma$ from the previous principal stress axes, the current shear stress is defined by the combination of previous stress components, the current incremental stress components, and the angle change. The current shear stress should vanish in the current principal stress axes:

$$\tau' = -\frac{1}{2}(\sigma_1 + \Delta\sigma_1 - \sigma_2 - \Delta\sigma_2)\sin 2\Delta\theta_\sigma + \Delta\tau\cos 2\Delta\theta_\sigma = 0\ . \tag{3.5}$$

Generally, the principal strain axes differ from the principal stress axes, and the shear strain in the principal stress axes or orthotropic axes does not vanish. However, the shear strain in the principal strain axes transformed from the principal

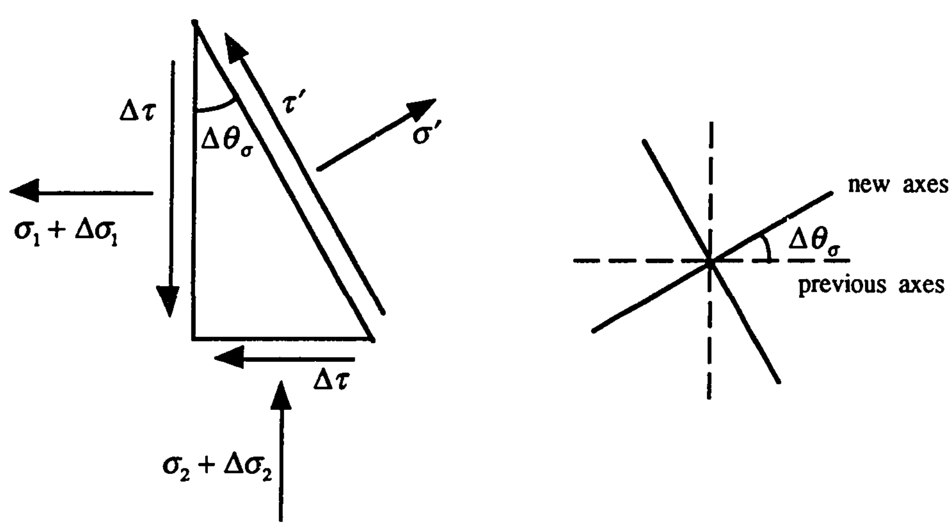


Figure 3.15 Stress variation due to rotation of principal stress axes

stress axes should be eliminated. If the current principal stress axes deviate from the principal strain axes by θ_d, the current shear strain in the principal strain axes is defined by the combinations of the strains in the principal stress axes or orthotropic axes. Also, the shear strain should vanish:

$$\gamma' = -\frac{1}{2}(\varepsilon_1 + \Delta\varepsilon_1 - \varepsilon_2 - \Delta\varepsilon_2)\sin 2(\theta_d + \Delta\theta_\sigma) + \frac{1}{2}(\gamma + \Delta\gamma)\cos 2(\theta_d + \Delta\theta_\sigma) = 0. \tag{3.7}$$

If $\Delta\theta_\sigma$ is eliminated in Equations 3.6 and 3.7, then

$$-\Delta\tau\left(\frac{A}{B}\cos 2\theta_d + \frac{\gamma}{B}\sin 2\theta_d\right) + \Delta\gamma\cos 2\theta_d - A\sin 2\theta_d + \gamma\cos 2\theta_d = 0, \tag{3.8}$$

where $A = (\varepsilon_1 + \Delta\varepsilon_1 - \varepsilon_2 - \Delta\varepsilon_2)$ and $B = \frac{1}{2}(\sigma_1 + \Delta\sigma_1 - \sigma_2 - \Delta\sigma_2)$.

In this equation, since $-A\sin 2\theta_d + \gamma\cos 2\theta_d = 0$, the relation between the shear stress and strain increments is defined by

$$G = \frac{\Delta\tau}{\Delta\gamma} = \frac{B\cos 2\theta_d}{(A\cos 2\theta_d + \gamma\sin 2\theta_d)}, \tag{3.9}$$

where G is the incremental shear stiffness. If the differences of the stress and strain increments are very small compared with those of the total stresses and strains, then $A = (\varepsilon_1 - \varepsilon_2)$, and $B = \frac{1}{2}(\sigma_1 - \sigma_2)$.

If the principal strain axes coincide with the principal stress axes, then the current shear strain, γ, and the deviation of the principal stress and strain axes, θ_d, should vanish in the principal axes. Accordingly, the shear stiffness, G, is simplified as

$$G = \frac{\Delta\tau}{\Delta\gamma} = \frac{A}{B} = \frac{(\sigma_1 - \sigma_2)}{2(\varepsilon_1 - \varepsilon_2)}. \tag{3.10}$$

The incremental shear stiffness of Equation 3.10 works effectively only if the difference between the principal stresses and strains is much larger than the difference between the principal stress or strain increments. Under complete unloading during cyclic loading, the principal stresses and strains are small. It is therefore difficult to achieve convergence. However, this shear stiffness is generally effective for fast convergence.

Mathematically, this shear stiffness can be very large, very small, or negative. Physically, the shear stiffness cannot be negative or very large. However, the shear stiffness does not have a physical meaning, and it only plays a role of eliminating shear stresses and strains in new principal axes.

Finally, in the orthotropic axes or principal stress axes, the incremental stress-strain relation is

$$\begin{Bmatrix} \Delta\sigma_1 \\ \Delta\sigma_2 \\ \Delta\tau \end{Bmatrix} = \begin{bmatrix} \frac{\partial\sigma_1}{\partial\varepsilon_1} & 0 & 0 \\ 0 & \frac{\partial\sigma_2}{\partial\varepsilon_2} & 0 \\ 0 & 0 & G \end{bmatrix} \begin{Bmatrix} \Delta\varepsilon_1 \\ \Delta\varepsilon_2 \\ \Delta\gamma \end{Bmatrix}, \tag{3.11.a}$$

or

$$\mathbf{s} = \mathbf{D}^{m} \cdot \mathbf{e}, \tag{3.11.b}$$

where $\mathbf{D}^m$ is the material stiffness matrix and G is the shear modulus.

In fact, the equivalent uniaxial stress-strain relation is a function of the strains not only in the current principal axis but also in the corresponding orthogonal axis. Therefore, non-zero off-diagonal terms may exist in the stiffness matrix, and the stiffness matrix then becomes unsymmetric. By using the symmetric stiffness matrix (Equation 3.11) and the unsymmetric stiffness matrix, it is found that the unsymmetric matrix has no advantage over the symmetric stiffness matrix for the speed of convergence in numerical computation.

4.0 REINFORCING STEEL AND BOND-SLIP MODELS

4.1 Reinforcing Steel Model

Reinforcement is idealized by either smeared or discrete elements. Reinforcement that is uniformly distributed over a relatively large area compared with the finite element size is idealized by two-dimensional smeared elements; otherwise, it is idealized by discrete line elements. The stress-strain relation of reinforcing steel is defined in terms of average stress and strain.

To idealize reinforcing steel behavior in this study, two constitutive models are used: a bilinear model including a kinematic hardening rule; and a strain hardening model including the Bauschinger effect. In the bilinear model, shown in Figure 4.1, the stress-strain relations for unloading and reloading are bounded by the upper and lower yield limits. Stiffness degradation due to cyclic loading is not included.

The strain hardening model is that proposed by Brown and Jirsa [9]. The stress-strain curve under monotonic loading, shown in Figure 4.2, consists of an elastic part, a yield plateau, and a strain hardening region. The strain hardening curve is that originally proposed by Burns and Seiss [9]:

$$\sigma_s = \sigma_y \left[\frac{112(\varepsilon_s - \varepsilon_{sh}) + 2}{60(\varepsilon_s - \varepsilon_{sh}) + 2} + \frac{\varepsilon_s - \varepsilon_{sh}}{\varepsilon_u - \varepsilon_{sh}} \left(\frac{\sigma_u}{\sigma_y} - 1.7 \right) \right], \tag{4.1}$$

where ε_{sh} is the strain-hardening strain, and ε_u and σ_u are the ultimate strain and stress respectively. From any stress-strain state, the allowable stress-strain path is

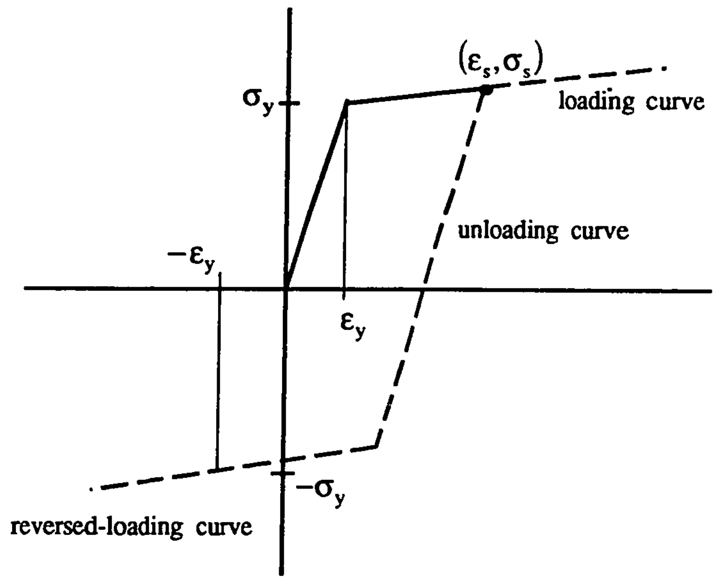


Figure 4.1 Bilinear model including a kinematic hardening rule, used for steel reinforcement

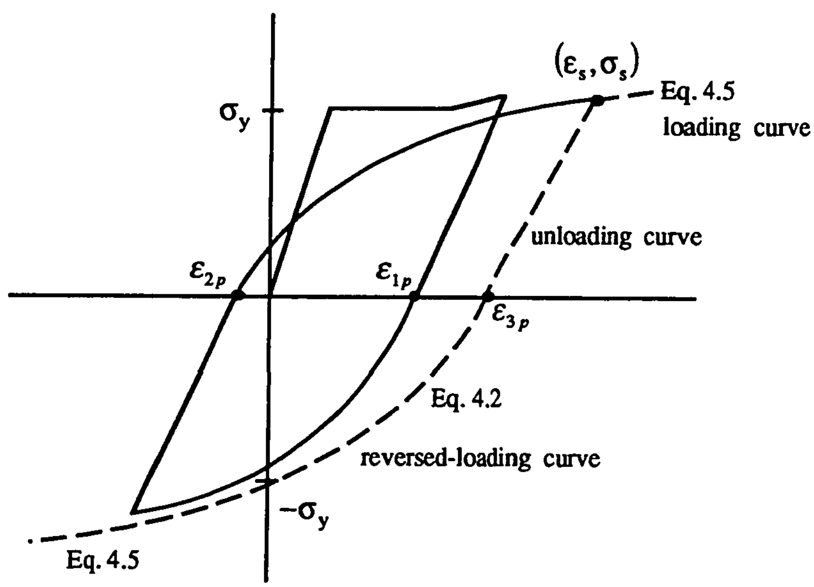


Figure 4.2 Strain hardening model including the Bauschinger effect, used for steel reinforcement (Brown and Jirsa [9])

composed of loading, unloading, and reversed-loading curves. The loading and reversed-loading curves consist of the transition curve representing the Bauschinger effect and the strain hardening curve. The transition curve defines the stress-strain relation between zero stress and yield stresses:

$$\sigma_s = \sigma_y \left[1 - \exp\left(\frac{-2.05\bar{\varepsilon}_s}{\varepsilon'_{sh}} \right) + \frac{0.129\bar{\varepsilon}_s}{\varepsilon'_{sh}} \right], \tag{4.2}$$

where $\bar{\varepsilon}_s$ is an equivalent strain, and ε'_{sh} is the effective strain-hardening strain. The equivalent strain and the effective strain hardening strain are defined by the residual strains due to the previous loading history.

$$\bar{\varepsilon}_s = \varepsilon_s - \varepsilon_{2p}, \text{ and} \tag{4.3}$$

$$\varepsilon'_{sh} = \frac{\varepsilon_{sh}}{1.38} \ln\left(\frac{\varepsilon_{1p} - \varepsilon_{2p}}{\varepsilon_y} \right), \tag{4.4}$$

where ε_{1p} is the maximum or minimum strain, and ε_{2p} is the current residual strain. In this research, the effective strain hardening strain is limited by $\varepsilon'_{sh} \geq 0.3\varepsilon_{sh}$. Beyond the yield stress, the stress-strain relation follows the strain hardening curve of Equation 4.1, whose parameters are adjusted to the current stress-strain position.

$$\sigma_s = \sigma_y \left[\frac{112(\bar{\varepsilon}_s - \varepsilon'_{sh}) + 2}{60(\bar{\varepsilon}_s - \varepsilon'_{sh}) + 2} + \frac{\bar{\varepsilon}_s - \varepsilon'_{sh}}{\varepsilon_u - \varepsilon_{sh}} \left(\frac{\sigma_u}{\sigma_y} - 1.7 \right) \right], \tag{4.5}$$

where ε'_{sh} is equivalent to ε_s at $\sigma_s = \sigma_y$ and the effective steel strain is defined by $\bar{\varepsilon}_s = \varepsilon_s - \varepsilon'_{sh} + \varepsilon_{sh}$. The unloading curve is defined by the straight line connecting the current stress-strain and the next residual strain, ε_{3p}.

$$\varepsilon_{3p} = 0.8\left(\varepsilon_s - \varepsilon_{2p}\right) + \varepsilon_{2p}, \tag{4.6}$$

For smeared steel, the tangent stiffness $\mathbf{D}_s$ is constructed by the tangent of the stress-strain curve and the reinforcement ratio in the direction of the reinforcing steel.

$$\begin{Bmatrix} \Delta\sigma_{1s} \\ \hline \Delta\sigma_{2s} \\ \hline \Delta\tau_s \end{Bmatrix} = \left[\begin{array}{c:c:c} \rho\dfrac{\partial\sigma_s}{\partial\varepsilon_s} & 0 & 0 \\ \hdashline 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 \end{array}\right] \begin{Bmatrix} \Delta\varepsilon_{1s} \\ \hline \Delta\varepsilon_{2s} \\ \hline \Delta\gamma_s \end{Bmatrix}, \tag{4.7}$$

In the case of perfect bond, displacements of the reinforcement elements are compatible with those of cracked concrete elements. Otherwise, reinforcement elements are connected to the cracked concrete elements via bond-slip elements.

Here, it is worth while to note a shortcoming of the concept of smeared cracking and smeared reinforcement, for idealizing the interaction between multiple cracks and reinforcement.

Using the concept of smeared cracking and smeared reinforcement, the strain of reinforcement is obtained by transforming the current strain combination to the reinforcement direction. Accordingly, the reinforcement strain is not related to the opening and closing of any specific crack. In other words, if the reinforcement layer crossing the both cracks, the reinforcement strain when one crack opens and the other crack closes can be the same as that when the former crack closes and the latter opens.

However, the reinforcement deformation is directly related to the opening width of a specific crack. Even for one reinforcement layer, the behavior of the reinforcement at a crack should be independent of that at the other crack. Current concept of smeared cracking and smeared reinforcement cannot consider the independent reinforcement behavior at each crack.

In usual planar members such as beams and shear walls under bending, shear, and axial loads, the stress states across the members are very complex, and the overall behavior depends on the stress-strain behavior of the flanges which are subjected to uniaxial tension or compression. In this type of member, the cyclic stress-strain behavior of the web which is a multiply cracked zone does not significantly affect the overall behavior.

However, if the stress states across the members are uniform under cyclic loading, the multiple cracks open and close simultaneously across the entire member, and the overall member behavior depends on the cyclic history of the reinforcement at each crack. Therefore, to precisely predict the member behavior with multiple cracks, more research is required for the interaction between multiple cracks and reinforcement so that the reinforcement behavior is related not to the reinforcement direction but to each crack direction.

4.2 Bond-Slip Model

Debonding phenomena of reinforcing steel are classified into pullout failure and splitting failure. Pullout failure usually occurs in anchorage zones of reinforcement in which the surrounding concrete is well confined. Splitting failure, however, occurs along reinforcement. In a splitting failure, since debonding follows spalling of the concrete cover due to splitting, bond failure occurs abruptly, and the bond strength is much lower than that of pullout failure. Also, in cracked concrete, the bond strength is lower than pullout failure strength. This is because the bond strength near the crack surface is much lower than that in the uncracked region which is well confined by surrounding concrete, and because the deterioration of the bond strength due to cyclic loading is severe.

In this research, the bond-slip model is based on an existing pullout failure model. For splitting failure and pullout failure in cracked concrete, the bond strength and ductility are assumed to be much lower than those of normal pullout failure.

Eligehausen et al. [18] developed a cyclic bond-slip model for pullout failure. The relation between bond stress, τ_b, and relative displacement, s, is composed of an envelope curve for slip in either direction, of unloading-reloading curves connecting the envelope curves, and of a transition curve connecting the unloading and reloading curves (Figure 4.3). The maximum strength of the envelope curve, τ_{b1}, decreases under fatigue damage due to cyclic loading. After cycles of loading, the bond-slip relation follows the reduced envelope curve (Figure 4.4).

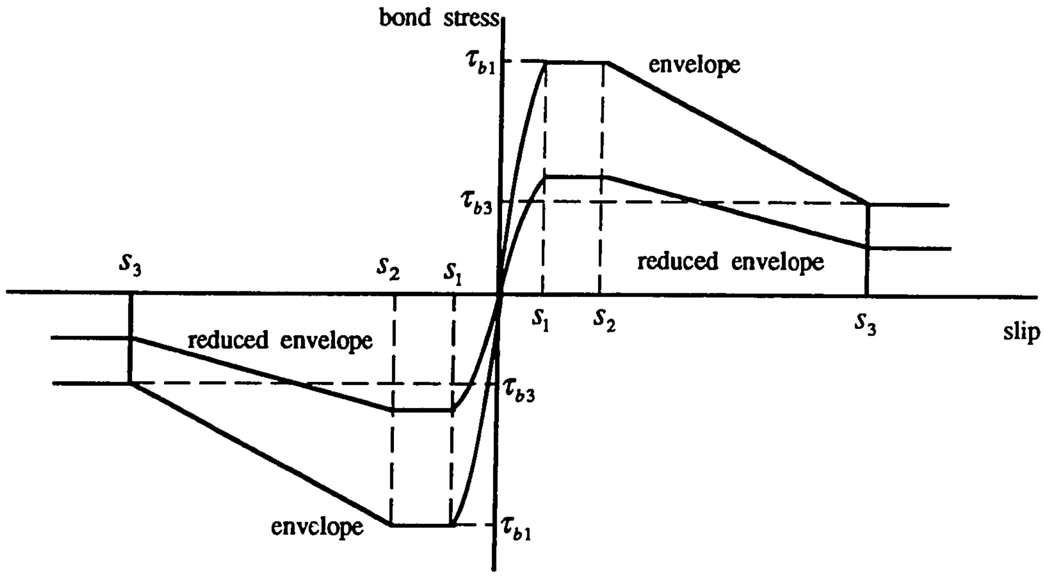


Figure 4.3 Envelope curves in bond-slip model

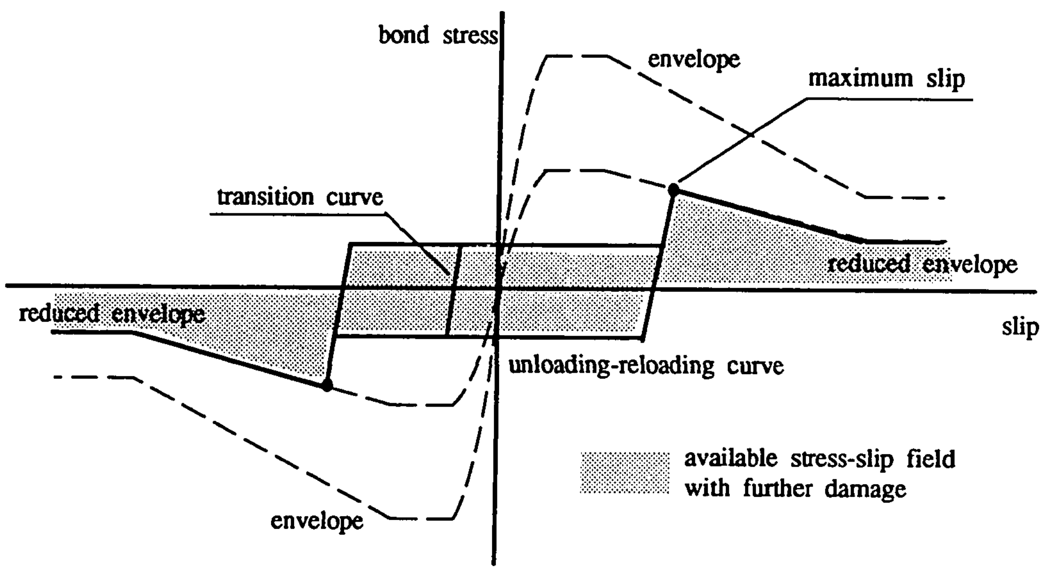


Figure 4.4 General behavior of bond-slip model

Eligehausen et al. suggest the following parameters for pullout failure:

$$s_1 = 1.0\,mm$$

$$s_2 = 3.0\,mm$$

$$s_3 = 10.5\,mm$$

$$\tau_{b1} = 13.5\ N/mm^2$$

$$\tau_{b3} = 5.0\ N/mm^2$$

In the analysis program developed here, the parameters will be chosen by program users according to the predicted failure types and other conditions. *ACI 318-63* specifies the ultimate bond strength associated with splitting failure as less than 5.6 MPa (800 psi) [42].

5.0 FINITE ELEMENT FORMULATION

5.1 Derivation of Structural Stiffness Matrix

The applied finite element formulation is derived by virtual work theory. The derivation will be demonstrated for the 4-node rectangular finite element. In the concept of virtual work, when a virtual or very small displacement is applied to a system with an existing force field in equilibrium, the internal virtual work should equal to the external virtual work.

$$\delta W_{int} = \delta W_{ext}. \tag{5.1}$$

The internal virtual work is done by the existing internal stress and the internal virtual strain due to the virtual displacement.

$$\delta W_{int} = \int_V (\delta\varepsilon \cdot \sigma)\, dV. \tag{5.2}$$

The external virtual work is expressed by the existing external force and the virtual displacement.

$$\delta W_{ext} = \delta U \cdot P. \tag{5.3}$$

For a finite element, the displacement field within the element is defined by the summation of weighted nodal displacements, expressed using polynomial interpolation functions, f_i, for each nodal displacement. The displacement at a position in the element is defined by

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} f_1 & 0 & f_2 & 0 & f_3 & 0 & f_4 & 0 \\ 0 & f_1 & 0 & f_2 & 0 & f_3 & 0 & f_4 \end{bmatrix} \begin{Bmatrix} U_1 \\ V_1 \\ \bullet \\ \bullet \\ U_4 \\ V_4 \end{Bmatrix}, \text{ or} \tag{5.4.a}$$

$$\mathbf{u} = \mathbf{N}^T \cdot \mathbf{U}. \tag{5.4.b}$$

Strains are defined by first derivatives of the displacements:

$$\varepsilon_x = \frac{\partial u}{\partial x}, \quad \varepsilon_y = \frac{\partial v}{\partial y}, \quad \gamma = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}. \tag{5.5}$$

From Equations 5.4.a and 5.5, the strains are redefined in a matrix form by the nodal displacements:

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma \end{Bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & 0 & \frac{\partial f_2}{\partial x} & 0 & \frac{\partial f_3}{\partial x} & 0 & \frac{\partial f_4}{\partial x} & 0 \\ 0 & \frac{\partial f_1}{\partial y} & 0 & \frac{\partial f_2}{\partial y} & 0 & \frac{\partial f_3}{\partial y} & 0 & \frac{\partial f_4}{\partial y} \\ \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial x} \end{bmatrix} \begin{Bmatrix} U_1 \\ V_1 \\ \bullet \\ \bullet \\ U_4 \\ V_4 \end{Bmatrix}, \text{ or} \tag{5.6.a}$$

$$\mathbf{e} = \mathbf{B} \cdot \mathbf{U}. \tag{5.6.b}$$

The constitutive relation of stresses and strains is defined by

$$\mathbf{s} = \mathbf{D} \cdot \mathbf{e}, \tag{5.7}$$

where **D** is the material stiffness matrix. The internal virtual work in Equation 5.3 is redefined by the scalar product of the virtual strain and the stress vectors.

$$\delta \mathrm{W}_{int} = \int_V (\delta\varepsilon \cdot \sigma)\, dV = \int_V \left(\delta\mathbf{e}^{\mathrm{T}} \cdot \mathbf{s}\right) dV. \tag{5.8}$$

Hence, from Equation 5.5.b, the virtual strains due to the virtual displacements are defined by

$$\delta\mathbf{e} = \mathbf{B} \cdot \delta\mathbf{U}. \tag{5.9}$$

Using compatibility and constitutive conditions in Equations 5.9 and 5.7 , the internal virtual work in Equation 5.8 is redefined in terms of nodal displacements;

$$\begin{aligned} \delta \mathrm{W}_{int} &= \int_V \left(\delta\mathbf{e}^{\mathrm{T}} \cdot \mathbf{s}\right) dV = \int_V \left(\delta\mathbf{U}^{\mathrm{T}} \cdot \mathbf{B}^{\mathrm{T}} \cdot \mathbf{D} \cdot \mathbf{B} \cdot \mathbf{U}\right) dV \\ &= \delta\mathbf{U}^{\mathrm{T}} \cdot \left(\int_V \left(\mathbf{B}^{\mathrm{T}} \cdot \mathbf{D} \cdot \mathbf{B}\right) dV\right) \cdot \mathbf{U} \end{aligned} \tag{5.10}$$

If the external virtual work in Equation 5.3 is defined by nodal displacements and forces,

$$\delta\mathrm{W}_{ext} = \delta\mathbf{U}^{\mathrm{T}} \cdot \mathbf{P}. \tag{5.11}$$

In virtual work theory, the virtual internal work in Equation 5.10 should equal to the virtual external work in Equation 5.11. Eliminating the virtual displacements in both equations, the load-deformation relation is defined by

$$\mathbf{P} = \left(\int_V \left(\mathbf{B}^{\mathrm{T}} \cdot \mathbf{D} \cdot \mathbf{B}\right) dV\right) \cdot \mathbf{U}, \text{ or} \tag{5.12.a}$$

$$\mathbf{P} = \mathbf{k} \cdot \mathbf{U}. \tag{5.12.b}$$

This gives the stiffness relation between the element forces and displacements.

For bond-slip elements, the internal virtual work is defined in terms of relative displacement (or slip), u_r, and the corresponding shear stress, τ_b.

$$\delta \mathrm{W}_{int} = \int \left(\delta u_r \cdot \tau_b\right) dl = \int \left(\delta \mathbf{u_r}^{\mathrm{T}} \cdot \mathbf{t}_b\right) dl. \tag{5.13}$$

The external virtual work is expressed by the existing external force, P_b, and the virtual displacement, δU_r.

$$\delta \mathrm{W}_{ext} = \delta U_r \cdot P_b = \delta \mathbf{U}_r^{\mathrm{T}} \cdot \mathbf{P_b}. \tag{5.14}$$

Using the compatibility condition, $\mathbf{u_r} = \mathbf{N_b} \cdot \mathbf{U_r}$, and the constitutive equation, $\mathbf{t_b} = \mathbf{D_b} \cdot \mathbf{u_r}$, Equation 5.13 is redefined as

$$\delta \mathrm{W}_{int} = \int \left(\delta u_r \cdot \tau_b\right) dl = \delta \mathbf{U}_r^{\mathrm{T}} \left(\int \mathbf{N_b}^{\mathrm{T}} \cdot \mathbf{D_b} \cdot \mathbf{N_b}\, dl\right) \mathbf{U_r}. \tag{5.15}$$

Since the external virtual work in Equation 5.14 equals the internal virtual work in Equation 5.15, the relation between element shear forces and the relative displacements is

$$\mathbf{P_b} = \left(\int \mathbf{N_b}^T \cdot \mathbf{D_b} \cdot \mathbf{N_b}\, dl\right) \mathbf{U_r}, \text{ or} \tag{5.16a}$$

$$\mathbf{P_b} = \mathbf{k_b} \cdot \mathbf{U_r}. \tag{5.16b}$$

A structural stiffness matrix is developed by assembling the corresponding element stiffness at each degree of freedom. In material nonlinear analysis, a large main memory is required for the information of the history of stress and strain as well as the basic information about the member and the loading condition. Therefore, it is important to manage main memory effectively. As an effective matrix solver, the frontal method, which eliminates the stiffness element by element, is used.

5.2 Finite Element Types

Analytical load-deflection characteristics depend on modeling characteristics such as the element types, the number of elements, and the number of gaussian points. Accordingly, the model should be chosen carefully.

In the proposed model, 4- and 8-node rectangular elements are used as shown in Figure 5.1. Based on comparison of the two elements in analysis, it is recommended to use the 8-node rectangular element since its high order displacement field assures smoother and more continuous structural behavior than the 4-node element.

As shown in Equation 5.12, the element stiffness of concrete, $\mathbf{k}_c$, is constructed by the integration of the material stiffness and the displacement field matrices.

$$\mathbf{k}_c = \int_V \left(\mathbf{B}^T \mathbf{D}_c \mathbf{B}\right) dV \tag{5.17}$$

For the integration, a 3x3 mesh of gaussian points is used for the 8-node element, and a 2x2 mesh is used for the 4-node element. The smaller number of the gaussian points is advantageous in saving main memory. However, underintegration can cause divergence in iteration.

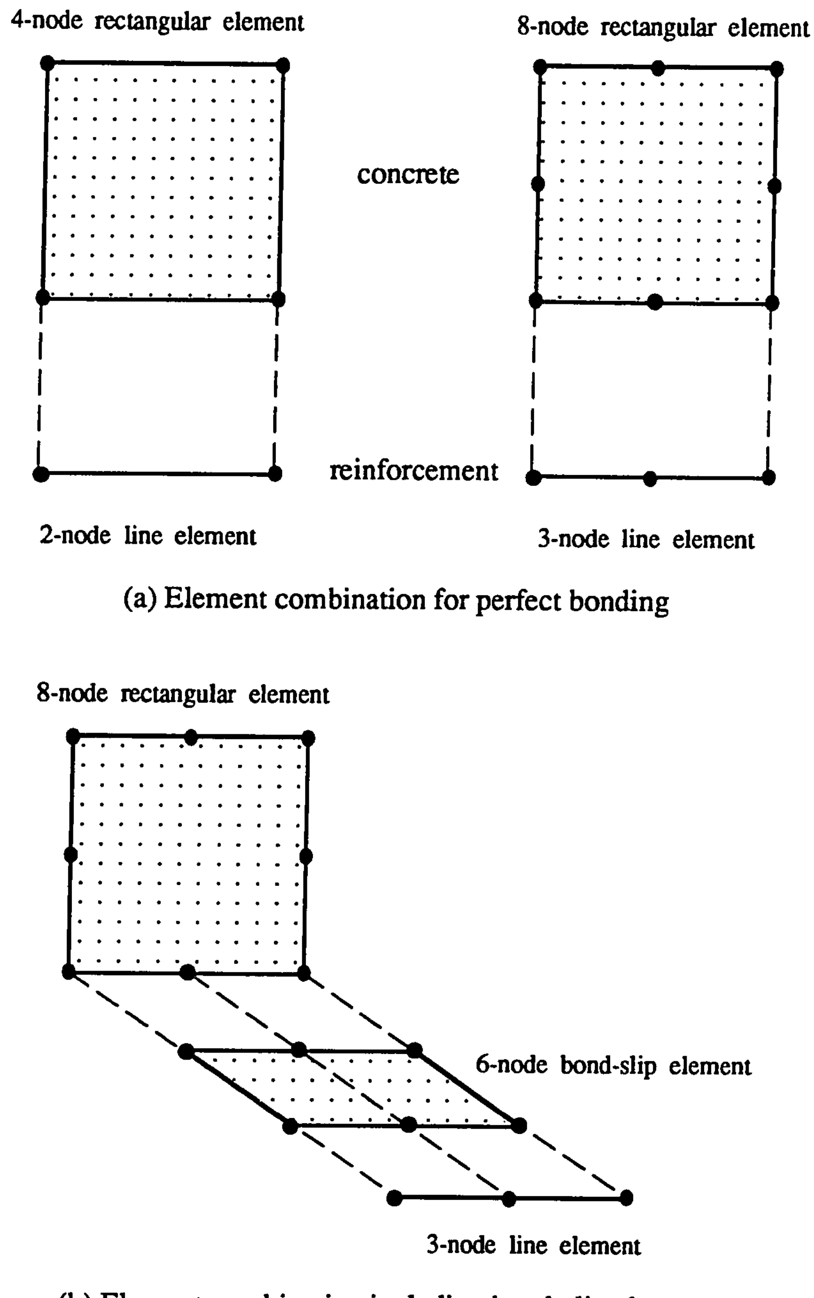
The 8-node and 4-node elements are also used for smeared reinforcement. The material stiffness, $\mathbf{k}_s$, is defined by

$$\mathbf{k}_s = \int_V \left(\mathbf{B}^T \mathbf{D}_s \mathbf{B}\right) dV \ . \tag{5.18}$$

In the proposed model, the finite element formation is constructed using isoparametric elements. However, since the smeared reinforcement is not likely to be distributed non-uniformly, it is reasonable that the element shape is rectangular.

A total element stiffness matrix is made by the summation of the stiffness matrices of concrete $\mathbf{k}_c$ and reinforcement $\mathbf{k}_s$.

$$\mathbf{k} = \mathbf{k}_c + \mathbf{k}_s \tag{5.19}$$



(a) Element combination for perfect bonding

(b) Element combination including bond-slip element

Figure 5.1 Finite element combinations

For the discrete reinforcement model of Figure 5.1 (a), a 2-node truss element and a 3-node line element are used. To satisfy compatibility at the boundary between the line elements and the rectangular elements, the 2-node line element is used for the 4-node rectangular element, and the 3-node line element is used for the 8-node rectangular element. In the 3-node line element, 3 gaussian points are used for numerical integration.

As bond-slip elements, a 6-node rectangular element is used to connect the 3-node line element and the 8-node rectangular element; and a 4-node rectangular element is used to connect the 2-node line element and the 4-node rectangular element (Figure 5.1 (b)). As shown in Figure 5.2, using compatibility and equilibrium conditions, the 6-node element is condensed into a 3-node bond-slip line element, and the 4-node element is condensed into a 2-node bond-slip line element.

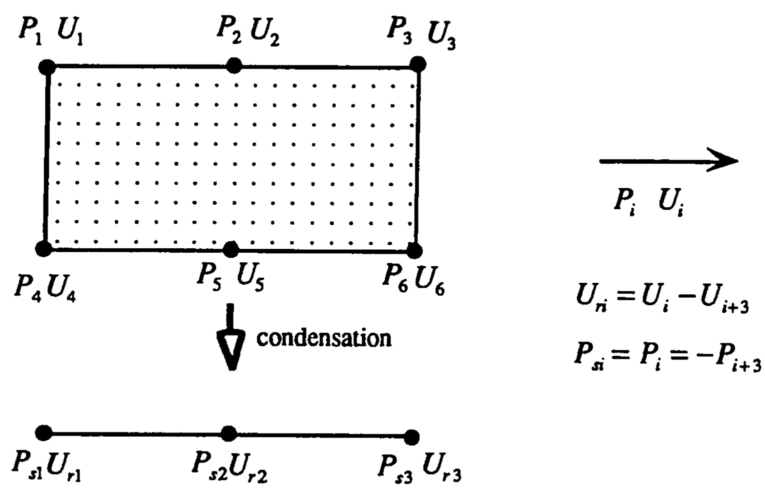


Figure 5.2 Bond-slip element

6.0 SOLUTION TECHNIQUE

6.1 General

Under general loading, a reinforced concrete member repeatedly experiences crack opening and closing, reinforcing steel yielding, and concrete crushing. Under these conditions, member behavior becomes highly nonlinear. During the analysis of the member, convergence may not be accomplished at a certain load level, called the critical load. Such a convergence problem can occur at the maximum load capacity of the analyzed member, or it can simply be a numerical difficulty. If there are no experimental data to verify the analysis results, one cannot tell whether or not the critical load corresponding to the convergence problem is actually the maximum load capacity of the member.

Previous researchers [33, 38] using the orthotropic axes approach with the equivalent uniaxial stress-strain curves report that a critical load is detected in the analysis of beam tests [8], as shown in Figure 7.11. At the critical load, numerical difficulty is detected, and the load-deflection curve is discontinuous. By extensive computer and programming work, it is found that the critical load or the discontinuity can occur depending on various conditions, such as the type of the tension stiffening model, the size of loading step, the target tolerance, and the finite element mesh. Also, it is found that the critical load can occur at any load level lower than the maximum load capacity of the analyzed member.

Therefore, to achieve the ultimate strength and ductility of the member, one needs a reliable numerical scheme to provide complete member behavior up to the target displacement.

6.2 Displacement Control Method

The displacement control method presented here successfully follows member behavior up to any target displacement, so that the ultimate strength and ductility of the member can be predicted. At a critical load, numerical difficulty is sometimes detected. However, in the displacement control method, numerical failure can be avoided by using an appropriate iteration scheme the detail of which are presented in Section 6.4.

A general method and a simplified method of displacement control given by Ramm [30] will now be introduced.

In the ith iteration, the tangent formulations of the load-deflection relation is rearranged so that the prescribed displacement, $\Delta U_2 = \Delta U^{ps}$, is separated from the other displacement components.

$$\begin{bmatrix} \mathbf{K}_{11}^i & \mathbf{K}_{12}^i \\ \mathbf{K}_{21}^i & K_{22}^i \end{bmatrix} \begin{bmatrix} \Delta\mathbf{U}_1^i \\ \Delta U_2 \end{bmatrix} = \Delta\lambda^i \begin{bmatrix} \mathbf{P}_1^s \\ P_2^s \end{bmatrix} + \begin{bmatrix} \Delta\mathbf{R}_1^i \\ \Delta R_2^i \end{bmatrix}, \tag{6.1}$$

where the force vector consists of the applied incremental force vector, $\mathbf{P}$, and the residual force vector, $\Delta\mathbf{R}$.

If the known variables are moved to the right hand side,

$$\begin{bmatrix} \mathbf{K}_{11}^i & -\mathbf{P}_1^s \\ \mathbf{K}_{21}^i & -P_2^s \end{bmatrix} \begin{bmatrix} \Delta\mathbf{U}_1^i \\ \Delta\lambda^i \end{bmatrix} = \begin{bmatrix} \Delta\mathbf{R}_1^i \\ \Delta R_2^i \end{bmatrix} - \begin{bmatrix} \mathbf{K}_{12}^i \\ K_{22}^i \end{bmatrix} \Delta U_2 \,. \tag{6.2}$$

The first equilibrium equation in Equation 6.2 is

$$\mathbf{K}_{11}^{i} \bullet \Delta\mathbf{U}_{1}^{i} = \Delta\lambda^{i} \cdot \mathbf{P}_{1}^{s} + \Delta\mathbf{R}_{1}^{i} - \mathbf{K}_{12}^{i} \cdot \Delta U_{2} . \tag{6.3}$$

In the equation, the displacement vector $\Delta\mathbf{U}_{1}^{i}$ can be divided into two parts: $\left(\Delta\mathbf{U}_{1}^{i}\right)^{\mathrm{I}}$ for the applied force; and $\left(\Delta\mathbf{U}_{1}^{i}\right)^{\mathrm{II}}$ for the residual force.

$$\Delta\mathbf{U}_{1}^{i} = \Delta\lambda^{i} \cdot \left(\Delta\mathbf{U}_{1}^{i}\right)^{\mathrm{I}} + \left(\Delta\mathbf{U}_{1}^{i}\right)^{\mathrm{II}}. \tag{6.4}$$

The relations between the separated displacement vectors and the force vectors are

$$\mathbf{K}_{11}^{i} \bullet \left(\Delta\mathbf{U}_{1}^{i}\right)^{\mathrm{I}} = \mathbf{P}_{1}^{s}, \tag{6.5}$$

and

$$\mathbf{K}_{11}^{i} \bullet \left(\Delta\mathbf{U}_{1}^{i}\right)^{\mathrm{II}} = \Delta\mathbf{R}_{1}^{i} - \mathbf{K}_{12}^{i} \cdot \Delta U_{2}. \tag{6.6}$$

Using the displacement vectors, the incremental parameter of the applied force vector is solved in the second equilibrium equation of Equation 6.2:

$$\Delta\lambda^{i} = \frac{-\Delta R_{2}^{i} + \mathbf{K}_{21}^{i} \bullet \left(\Delta\mathbf{U}_{1}^{i}\right)^{\mathrm{II}} + K_{22}^{i} \cdot \Delta U_{2}}{P_{2}^{s} - \mathbf{K}_{21}^{i} \bullet \left(\Delta\mathbf{U}_{1}^{i}\right)^{\mathrm{I}}} . \tag{6.7}$$

The total displacement increment and the total force increment are obtained by

$$\Delta\mathbf{U} = \sum_{i} \left[\Delta\lambda^{i} \cdot \left(\Delta\mathbf{U}^{i}\right)^{I} + \left(\Delta\mathbf{U}^{i}\right)^{II}\right]$$

and (6.8)

$$\Delta\mathbf{P} = \sum_i \left[\Delta\lambda^i \cdot \mathbf{P}^s\right]. \tag{6.9}$$

The total displacement and force vectors in each loading step are obtained by

$$\mathbf{U}^j = \mathbf{U}^{j-1} + \Delta\mathbf{U} \tag{6.10}$$

and

$$\mathbf{P}^j = \mathbf{P}^{j-1} + \Delta\mathbf{P}\ . \tag{6.11}$$

This general method can be simplified by removing the process of stiffness modification. Instead of the modified stiffness $\mathbf{K}_{11}^i$, $\mathbf{K}^i$ is used in Equations 6.5 and 6.6:

$$\mathbf{K}^i \bullet \left(\Delta\mathbf{U}_1^i\right)^{\mathrm{I}} = \mathbf{P}^s \tag{6.12}$$

and

$$\mathbf{K}^i \bullet \left(\Delta\mathbf{U}_1^i\right)^{\mathrm{II}} = \Delta\mathbf{R}\ , \tag{6.13}$$

where the prescribed displacement term is also removed.

Again, the incremental displacement vector is defined by the two displacement vectors obtained in Equations 6.12 and 6.13.

$$\Delta\mathbf{U}^i = \Delta\lambda^i \cdot \left(\Delta\mathbf{U}^i\right)^{\mathrm{I}} + \left(\Delta\mathbf{U}^i\right)^{\mathrm{II}}. \tag{6.14}$$

Of the incremental displacement vector components, the controlled incremental displacement should be the prescribed value:

$$\Delta U_2^i = \Delta\lambda^i \cdot \left(\Delta U_2^i\right)^{\mathrm{I}} + \left(\Delta U_2^i\right)^{\mathrm{II}} = \Delta U^{ps}. \tag{6.15}$$

In the first iteration, the incremental load parameter is obtained from Equation 6.15:

$$\Delta\lambda^1 = \frac{\Delta U^{ps} - \left(\Delta U_2^1\right)^{\mathrm{II}}}{\left(\Delta U_2^1\right)^{\mathrm{I}}}. \tag{6.16}$$

After the first iteration, further incremental displacement is eliminated so that the total incremental displacement is equivalent to the prescribed value:

$$\Delta\lambda^i = -\frac{\left(\Delta U_2^i\right)^{\mathrm{II}}}{\left(\Delta U_2^i\right)^{\mathrm{I}}} \quad (i \geq 2). \tag{6.17}$$

As shown before, since it eliminates the modification of the stiffness matrix, the simplified displacement control method can save main memory.

By comparing the two displacement control methods, it is found that the two methods produce the identical convergence rate, and that the simplified method is more efficient for computer memory and running time.

6.3 Iteration Strategy

In analyzing a reinforced concrete structure, the choice of solution strategy is one of the most important factors determining the practicality of a analysis method. Since the material behavior of concrete is highly nonlinear, it is very difficult to ensure that the applied iteration scheme always converges in a stable manner. The convergence speed is also important. Generally, as much study is required to select the iteration scheme, as the material model. Stability and speed of convergence depend on the applied material modeling and the assumptions on which the material model is based. Convergence is also sensitive to the solution technique and the tolerance limit.

In the analysis program developed here, tangent stiffness is used for incremental displacement stepping. The tangent stiffness is composed of the slope of each equivalent uniaxial stress-strain curves and shear stiffness as shown in Equation 3.11. For this tangent stiffness, numerical difficulty frequently occurs in the following situations.

1) In softening material, where the slope of stress-strain relation in loading (increase of strain) is opposite to that in unloading (decrease of strain). Either stiffness in the direction of a incremental strain cannot follow the stress increment in the other incremental strain.
2) When a structural load capacity suddenly decreases, or when the load transfer mechanism suddenly changes, loading and unloading (increase and decrease of strain) occur across the entire structure.

3) Near zero stresses or strains, principal directions change significantly even with small incremental strains. The proposed cracked concrete model is very sensitive to the orientation of principal axes. However, the shear stiffness, which make the principal stress and strain axes coincide, is not measured accurately with small stresses and strains.

To prevent numerical difficulties associated with the above, the following guidelines are recommended for stable and fast convergence:

1) The Modified Newton-Raphson Method does not always produce convergence in each loading step. In softening material, the current equilibrium position does not always lie near the tangent stiffness at the previous equilibrium position. As a result, the initial tangent stiffness or once-modified tangent stiffness sometimes fails to converge. Therefore, the tangent stiffness should be modified in every iteration.

2) A very small stiffness element or a negative stiffness element can cause divergence. To avoid such problems, it is recommended that individual diagonal elements not be less than E/1000 in value [31], where E is the elastic modulus of concrete .

3) Convergence is sometimes difficult even when the strategies in 1) and 2) are used. Use of the initial elastic stiffness matrix is found to give the most stable convergence [31]. The elastic stiffness is the largest possible stiffness, and is constant regardless of axis orientation. Therefore, in most iterations,

convergence can be accomplished monotonically using the elastic stiffness. However, such convergence requires a considerable number of iteration. The initial elastic stiffness is therefore used only when convergence is not accomplished by the tangent stiffness. After convergence is accomplished, the iteration scheme is switched back to the tangent stiffness method.

4) As a tolerance limit, an incremental displacement criterion is applied:

$$Tol = \sqrt{\frac{\Delta \mathbf{U}^i \bullet \Delta \mathbf{U}^i}{\Delta \mathbf{U}^T \bullet \Delta \mathbf{U}^T}} \tag{6.18}$$

where $\Delta \mathbf{U}^T$ is the vector of the total displacement increment in current loading step and $\Delta \mathbf{U}^i$ is the vector of the displacement increment in ith iteration. This criterion provides an indirect measure of the incremental force tolerance. Generally, it allows more residual forces than does the incremental force criterion with a given tolerance limit. The incremental displacement criterion is suitable for both monotonic and cyclic loads. Though a stricter tolerance limit gives a more exact representation of the true load-deflection curve, as the tolerance limit, acceptable accuracy and faster convergence are obtained with displacement tolerance limit of 1%.

7.0 VERIFICATION OF MATERIAL MODEL

7.1 General

In this chapter, the proposed material model will be verified by analyzing structural members under monotonic and cyclic load conditions. The behavior of structural members which exhibit flexure-dominated behavior can be analyzed by various methods, assuming either uniaxial stress-strain relations or crack directions across the member section. To verify the effectiveness of the proposed method of analysis, the structural members analyzed herein exhibit not only flexure-dominated behavior but also shear-dominated behavior, for which more general analysis methods are required.

7.2 Shear Panel Tests (Vecchio and Bhide)

Two series of shear panels were tested at The University of Toronto in the early 1980's. The shear panels were tested under in-plane loading: Series PV panels, tested by Vecchio [38], were primarily subjected to uniform shear; Series PB panels, tested by Bhide [7], were subjected to uniaxial tension and shear. The stress and strain states across a tested panel were intended to be uniform, so that the test data would give a basis for developing the smeared stress-strain relation of cracked concrete.

Table 7.1 and Figure 7.1 show the dimensions and the material properties of the shear panels analyzed herein. Most test panels are anisotropically reinforced by steel layers so that the variation of the principal stress-strain relation can be observed in rotating principal axes. For the analytical model, one 4-node element, shown in Figure

7.1, is used. Since the stress-strain states are uniform across the panels, one element is sufficient to estimate the stress-strain relations induced by the applied loads.

The analyses are compared with the test results in Figures 7.2 - 7.9. They are also compared with the analyses of Vecchio (Series PV) in Figures 7.2 - 7.5 and Stevens (Series PB) in Figures 7.8 and 7.9. The figures show the relations of principal compressive stress versus principal compressive strain, principal tensile stress versus principal tensile strain, maximum shear stress versus maximum shear strain, and the orientations of principal stress and strain axes versus maximum shear strain.

Before comparing the analysis results and the experiments, the concept of the proposed material model for monotonic loading will be restated here:

1) Orthotropic axes rotate to current principal axes.
2) In the principal compressive axis, the compressive strength is reduced by the corresponding principal tensile strain which represents crack opening.
3) As long as at least one reinforcement layer remains elastic, cracked concrete has considerable tension stiffening stresses in the current principal tensile axis. Once the reinforcement exceeds the yield strain, the tension stiffening stresses disappear.

As shown in Figure 7.2, since PV4 is isotropically reinforced, and since the reinforcement is symmetric with respect to the principal axes, the principal axes do not rotate during loading; the two reinforcement layers reach the yield stresses simultaneously so that the tension stiffening stress disappears quickly. The shear stress-strain curve is trilinear, the key points of which are defined by cracking and the yielding of the two reinforcing steel layers. The shear stress of concrete reaches its maximum capacity when the reinforcement layers yield.

On the other hand, shear panels PV10 and PV12 are anisotropically reinforced. If one reinforcement layer yields after cracking, the principal axes rotate to the direction of the other reinforcement layer which remains elastic. The shear stress-strain curves are trilinear, the key points of which are defined by cracking and the yielding of one reinforcing steel layer. In the singly reinforced shear panels PV13, PB16, PB19, PB21, and PB22, once tensile cracking occurs, the principal axes rotate to the direction of the reinforcement layer. The shear stress-strain curves are bilinear, which is defined by tensile cracking.

In the anisotropically or singly reinforced shear panels, it is observed that considerable tension stiffening stresses are maintained even at large tensile strain. This is because at least one reinforcing steel layer remains elastic. However, even with the considerable tension stiffening stresses, the increase of the shear stress of concrete is not conspicuous because the compressive stresses do not increase much due to crack opening (compression softening due to crack opening).

By comparing PV 4 in Figure 7.2 and the other experimental results in Figures 7.3 - 7.9, it is obvious that if a reinforcement layer remain elastic, cracked concrete has considerable tension stiffening stresses. The proposed tension stiffening model idealizes this phenomenon appropriately. The analytical results show good agreement with Series PV, PB 21, and PB22, but some discrepancy is shown in Series PB16 and PB 19. However, as shown in Figures 7.6 and 7.7, the principal stress and strain axes deviate from each other by about 10 degrees, even before cracking. By investigating the test results, it is found that since the shear panels of Series PB are subjected to uniaxial tension and shear forces, the stress distribution across the section of the shear panels is not assured to be uniform as expected. As a result, tensile cracking occurs locally even before the average principal stress reaches the tensile cracking stress.

As shown in the figures, the Vecchio and Stevens' models generally give reasonable results. In Series PV, Vecchio's model and the proposed model show close predictions for the tests. In Series PB 21 and PB 22, the proposed model provides better prediction than Stevens' model. Also, it is noted that the assumption that principal stress axes coincide with principal strain axes gives more rigidity to the shear panels than actual stress-strain relations in principal axes deviated from each other. As a result, the analyses slightly overestimate the actual shear stress-strain relations at large strains.

Table 7.1 Loading conditions and material properties of shear panels

Panel	Loading	Concrete		Reinforcing steel			
	$\sigma_x:\sigma_y:v_{xy}$	f'_c (MPa)	ε_o (%)	f_{xy} (MPa)	f_{yy} (MPa)	ρ_{sx} (%)	ρ_{sy} (%)
PV 4	0 : 0 : 1	26.6	0.25	242	242	1.056	1.056
PV 10	0 : 0 : 1	14.5	0.27	276	276	1.785	0.99
PV 12	0 : 0 : 1	16.0	0.25	469	269	1.785	0.446
PV 13	0 : 0 : 1	18.2	0.27	248	0.0	1.785	0.0
PB 16	1.96 : 0 : 1	41.7	0.3225	502	0.0	2.023	0.0
PB 19	1.01 : 0 : 1	20	0.1913	402	0.0	2.195	0.0
PB 21	3.1 : 0 : 1	21.8	0.18	402	0.0	2.195	0.0
PB 22	6.1 : 0 : 1	17.6	0.203	433	0.0	2.195	0.0

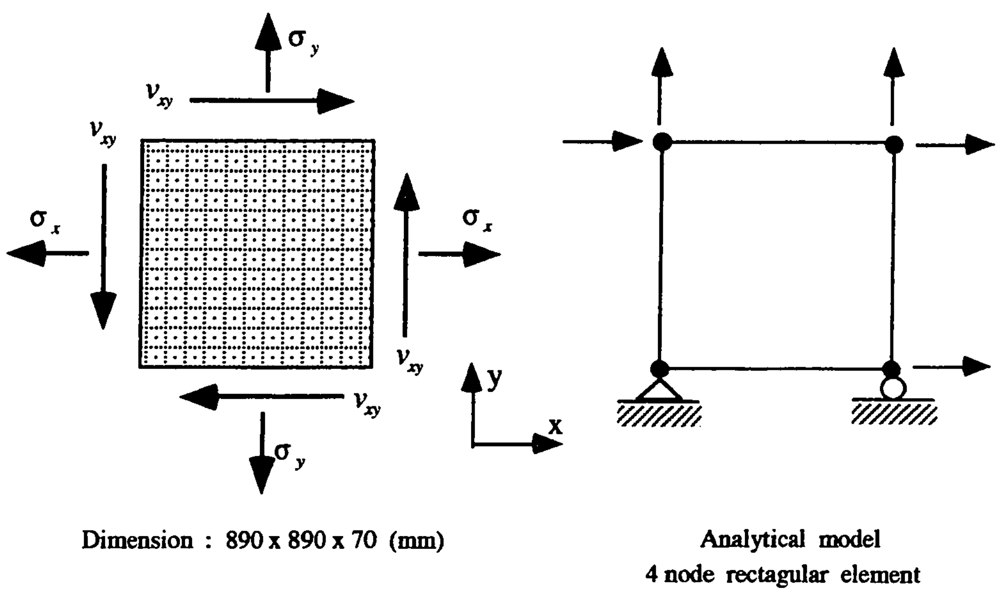


Figure 7.1 Shear panels tested at The University of Toronto [7, 38]

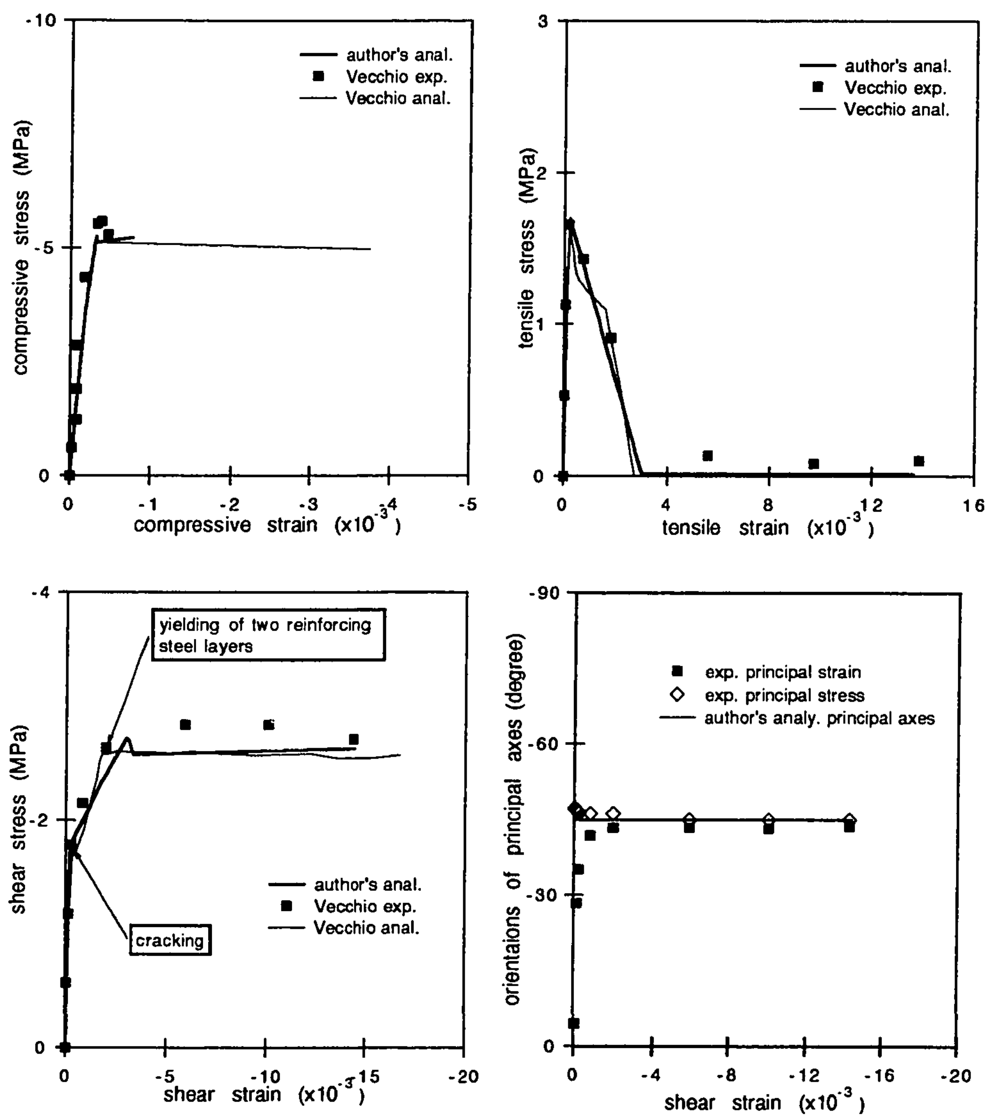


Figure 7.2 Comparison of analytical predictions and test results for Shear Panel PV4 (Vecchio [38])

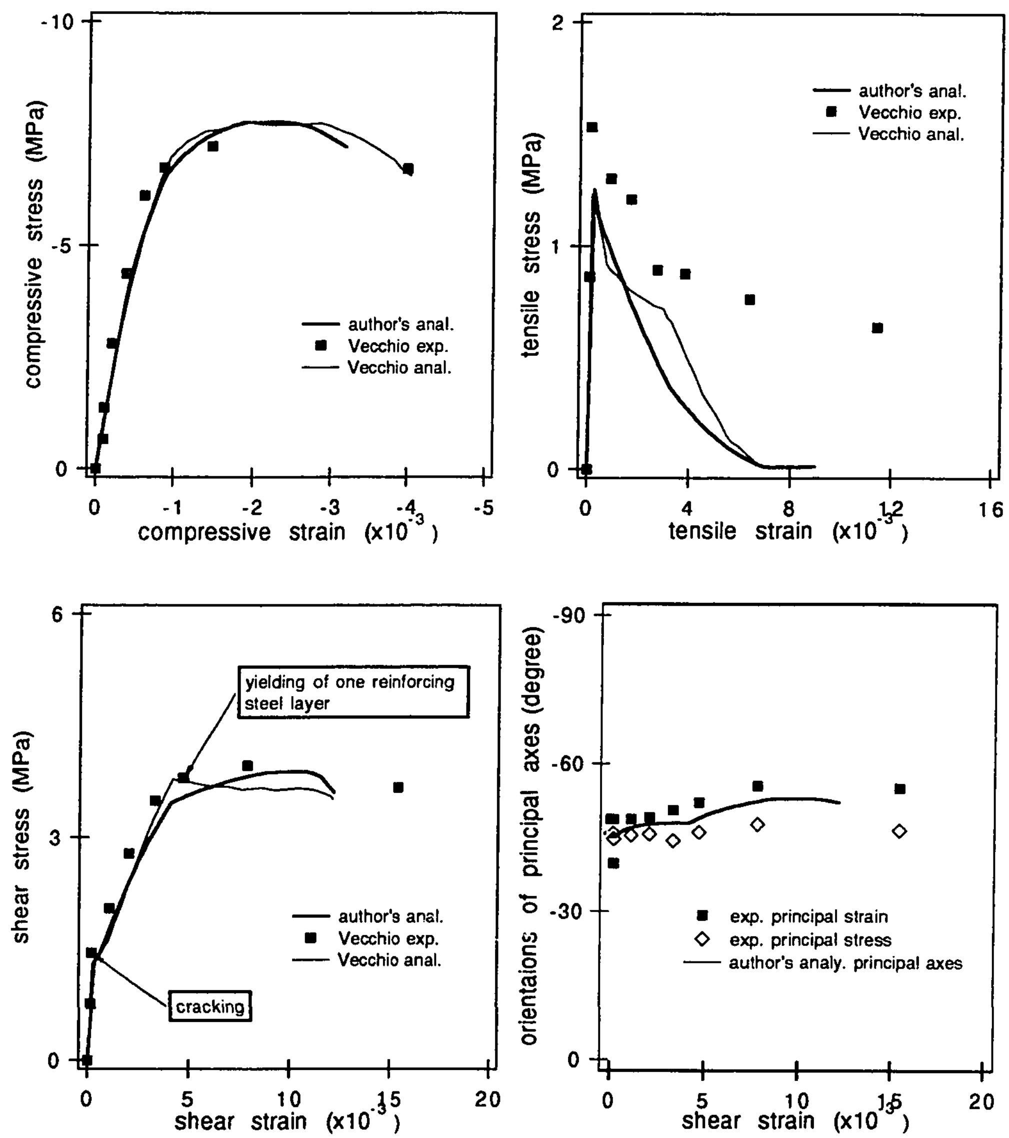


Figure 7.3 Comparison of analytical predictions and test results for Shear Panel PV10 (Vecchio [38])

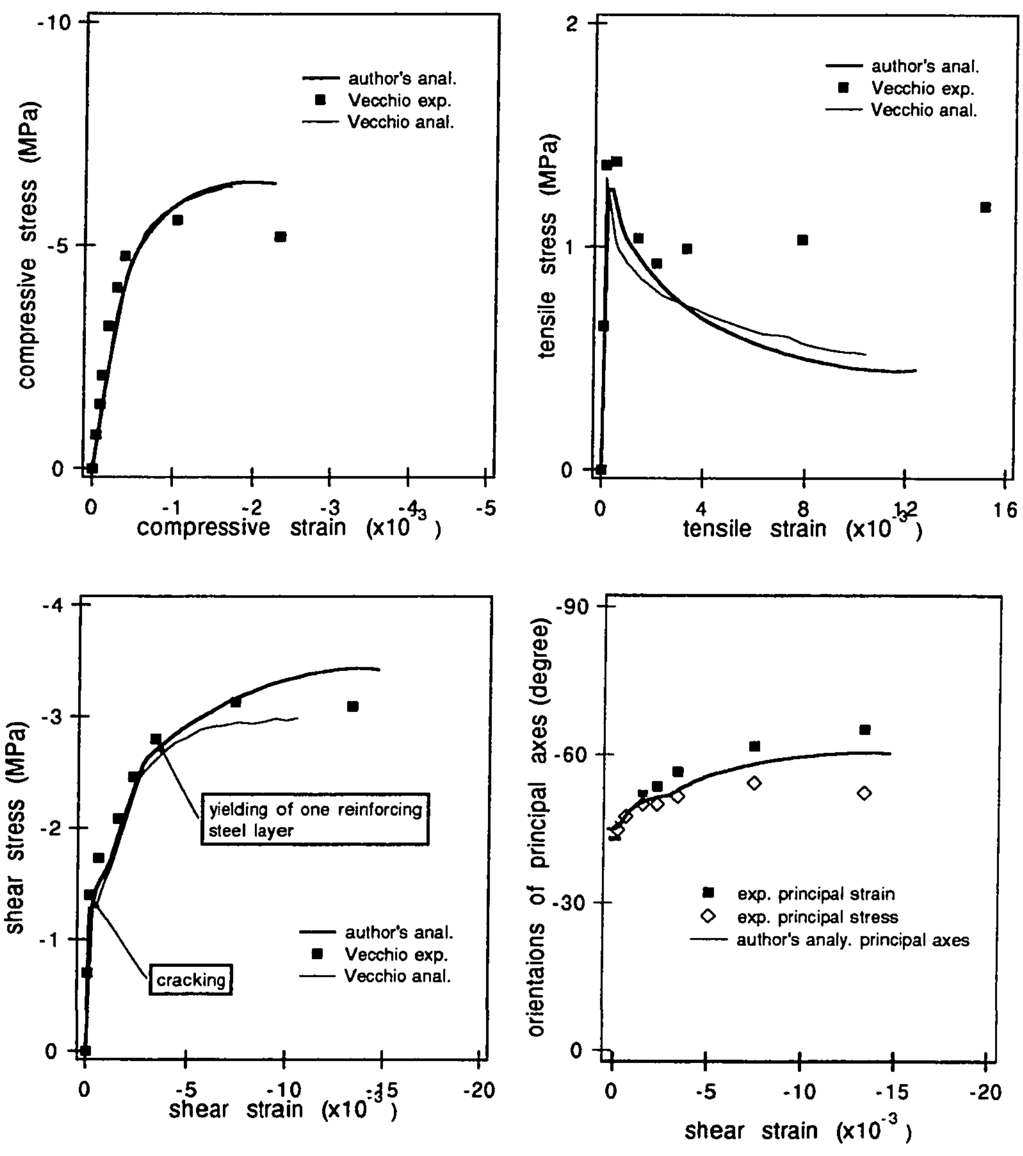


Figure 7.4 Comparison of analytical predictions and test results for Shear Panel PV12 (Vecchio [38])

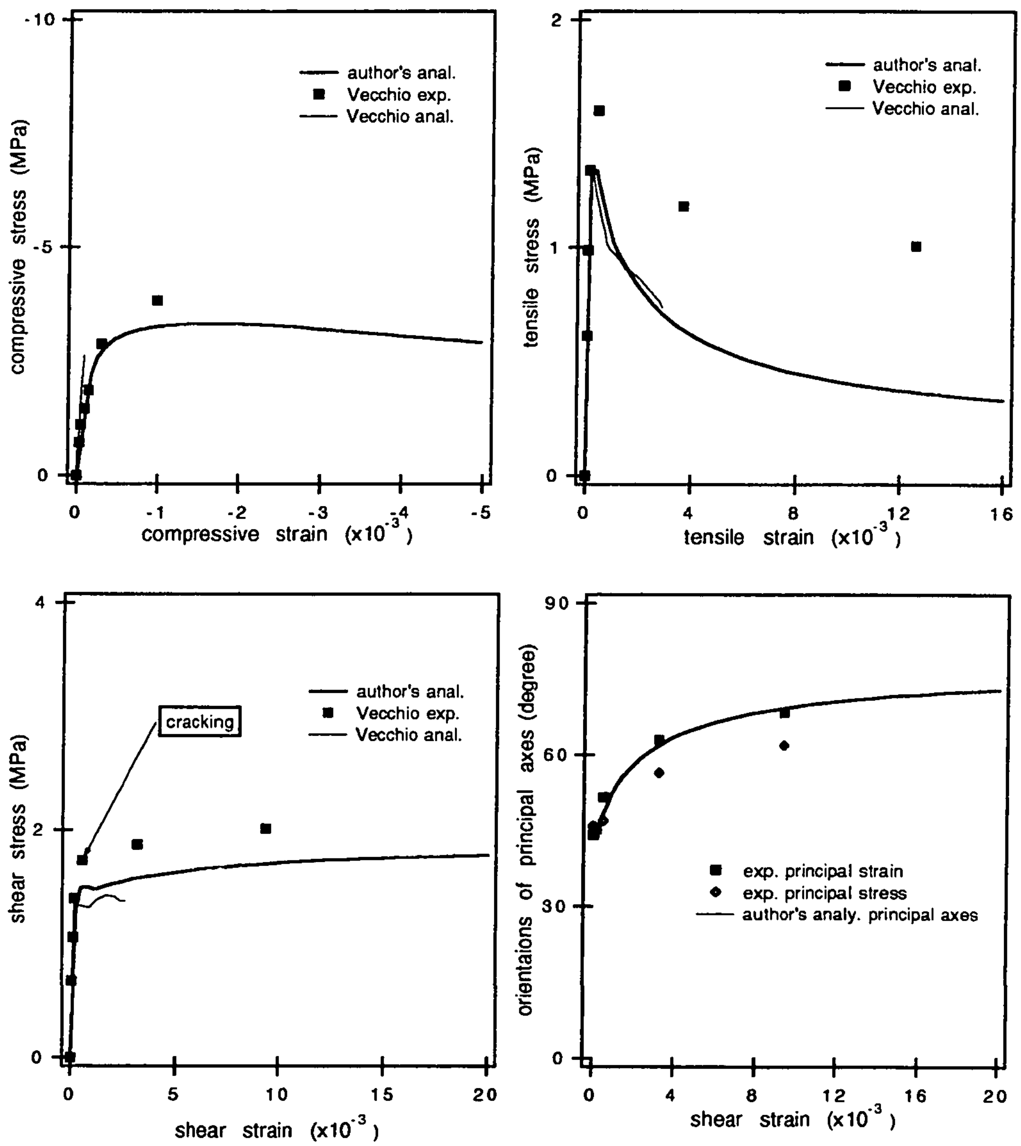


Figure 7.5 Comparison of analytical predictions and test results for Shear Panel PV13 (Vecchio [38])

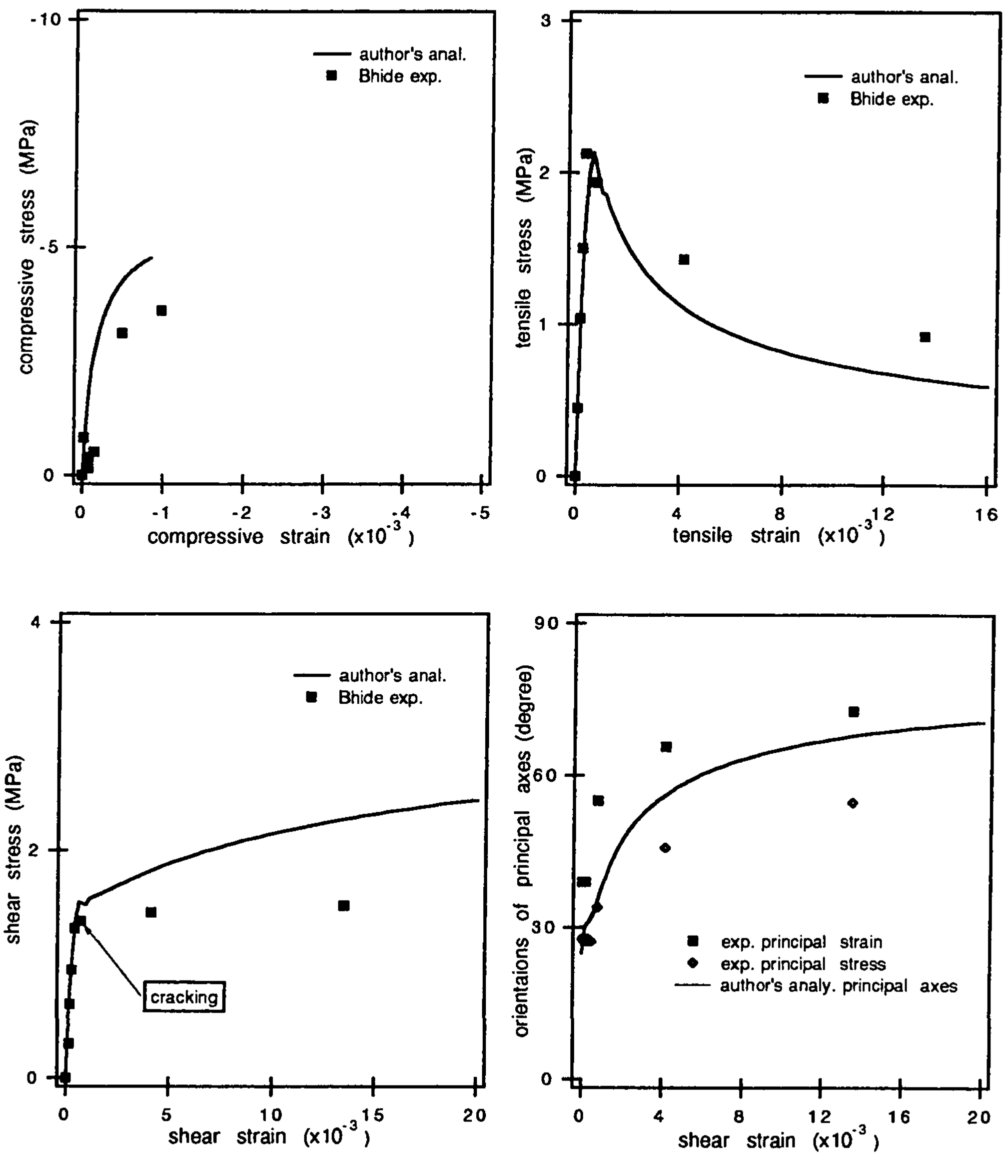


Figure 7.6 Comparison of analytical predictions and test results for Shear Panel PB16 (Bhide [7])

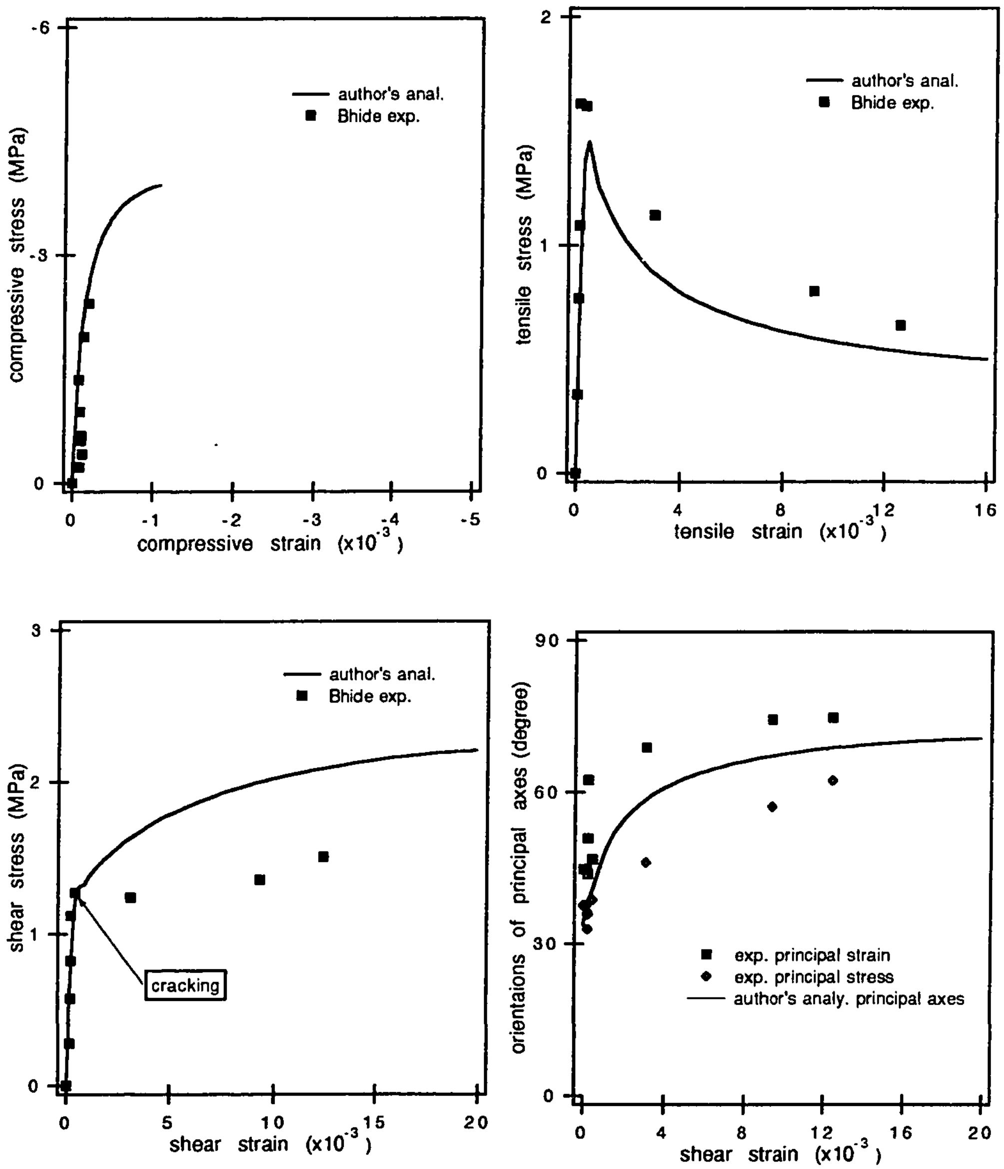


Figure 7.7 Comparison of analytical predictions and test results for Shear Panel PB19 (Bhide [7])

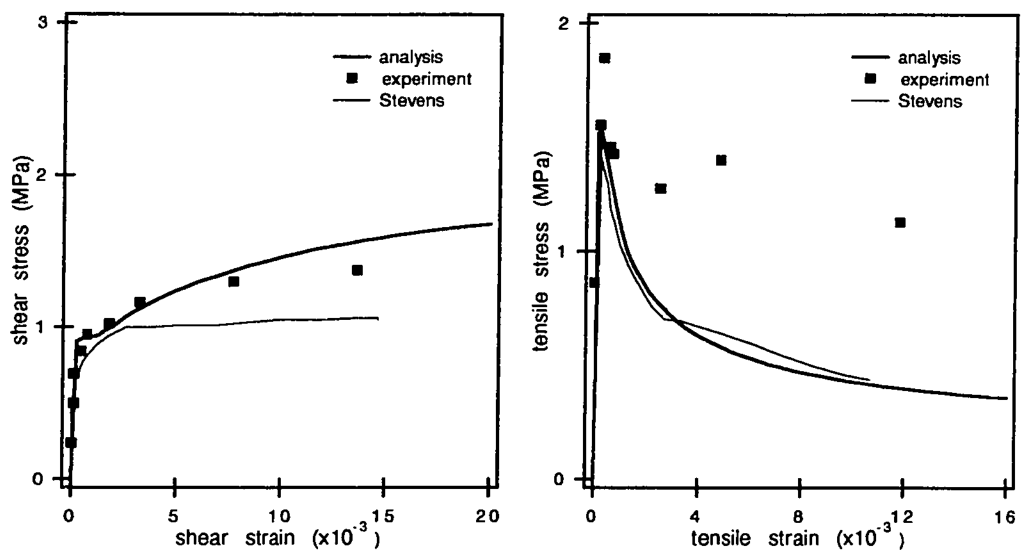


Figure 8.8 Comparison of analytical predictions and shear panel tests for shear panel PB21 (Bhide [7])

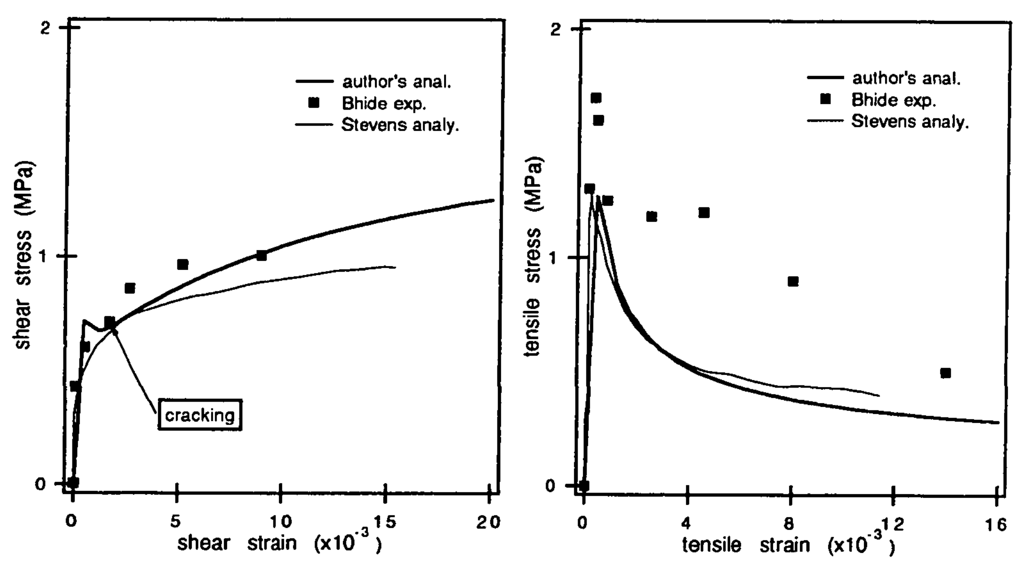


Figure 7.9 Comparison of analytical predictions and test results for Shear Panel PB22 (Bhide [7])

7.3 Reinforced Concrete Beam Tests under Monotonic Loading (Bresler and Scordelis)

Bresler and Scordelis [8] investigated the shear capacity of a series of beam specimens. Their Beams A-1 and A-2 are analyzed here. Span-to-depth ratios are 7.0 for Beam A-1 and 10.0 for Beam A-2, typical of shallow beams. However, these beams have heavy longitudinal reinforcement at the bottom, so that inelastic flexural deformation due to yielding of reinforcing steel is prevented. On the other hand, the reinforcement ratio of the vertical bars is low, inviting a shear failure due to diagonal tension cracking.

Due to symmetry, these beams are idealized by an equivalent half-beam, due to symmetry. As shown in Figure 7.10, the half-beam model is composed of twenty 8-node rectangular elements and ten 3-node line elements. The horizontal bars at the bottom and the top of the beam are idealized by discrete line elements. The vertical bars are assumed uniformly distributed, and are idealized by rectangular elements for smeared reinforcement. The analysis was performed by the displacement control method, and a 1-percent displacement tolerance was used for the convergence criterion. The influence of the main bars in the bottom of the beam is included in the tension stiffening effect in the web of the beam.

By comparing the analytical and the experiments results, shown in Figure 7.11, the following observations are made:

1) The analytical results using the proposed approach are close to the experiments. The results clearly show that lack of shear capacity causes brittle member failure without much ductility.

2) The load capacity of the beam falls between the flexural and the shear load capacity values calculated according to ACI 318-89 [41]. The ACI code underestimates the actual shear capacity of the beam.

Figure 7.12 presents the variation of principal tensile axes in which principal tensile strain exceeds the cracking strain. The principal tensile axes represent primary crack directions. In the figure, it is noticed that tensile cracks spread from the beam web to the compressive region in the top of the beam section, and that tensile cracks suddenly increase when the brittle failure occurs.

By investigating the computational stress-strain relations throughout the beam, the failure mechanism of Beam A-1 is interpreted as shown in Figure 7.12:

Fig. 7.12 (a) The deformation of the beam is dominated by diagonal tension cracks in the beam web. Flexural cracking in the bottom of the beam is resisted by large longitudinal bottom bars, which remain elastic.

Fig. 7.12 (b) Due to lack of shear reinforcement, the diagonal tension cracks widen and spread over the compression zone of the beam. As the result, the effective compression area available to resist the existing load capacity is reduced, and concrete crushing occurs.

Fig. 7.12 (c) Finally, the load capacity of the beam maintained by bending action is no longer effective due to the large crack opening, and the load capacity decreases abruptly.

According to the analysis, the sudden decrease of the load capacity is caused by change of the load transfer mechanism of the beam. As the cracks spread over the entire cross section, the bending action of the beam is no longer effective. Instead, as the cracks spread from the bottom to the top of the beam, the deformation of the beam depends on the crack widths. At the displacement of the maximum load capacity, the load capacity due to the crack or shear deformation is much lower than that due to bending action. Accordingly, the load-deflection curve become discontinuous.

The reduced load capacity after brittle failure is maintained by the vertical reinforcing steel across the cracks. In fact, the load capacity of the reinforcing steel bars at large crack opening is meaningless, because the reinforcing steel cannot retain its capacity without bond to concrete.

As shown in Figure 7.11, the load-deflection curve is discontinuous at the brittle failure, and any other continuous load-deflection path is not found to connect the maximum load capacity and the reduced load capacity. According to this analytical research about solution technique, when an arc-length method is used with monotonic stress-strain relations of materials, a continuous load-deflection path, like snap-back phenomenon in geometry nonlinear problem, can be found. The monotonic stress-strain relations restrict the member behavior or the load-deflection path. Just after the maximum load capacity, the load-deflection curve is on an unloading path with decrease of displacement, and regains then equilibrium positions with increase of displacement. This is because the load-deflection curve cannot go on the unloading path due to the restriction of the monotonic stress-strain laws. If cyclic constitutive laws are used with the arc-length method, the load-deflection curve continues to be on an unloading path with decrease of displacement after the maximum load capacity, because the cyclic material laws allow equilibrium on the unloading path. As a result, the equilibrium position with increase of displacement cannot be found. Therefore, to accomplish complete behavior up to a target displacement, the displacement control method is used in these analyses.

Comparing Figure 7.12 (c) and Figure 7.13, the crack pattern represented by the orientations of the principal axes are very similar to the widely used strut-tie model. However, different from the strut-tie model, which is defined in a force-displacement field, the rotating orthotropic axes model can consider the nature of the interaction between cracked concrete and reinforcing steel, such as tension stiffening and compression softening due to crack opening. Also, the rotating orthotropic axes model, which can adjust the directions of strut-tie to current principal axes, can be used for cyclic behavior.

During computation of the response of these members, the following observations are made:

1) The predictions of several researchers [34, 37] are almost the same as in the author's analysis. The only difference is that their predictions stop at the maximum load capacity, while the author's analysis clearly shows the sudden decrease of load capacity.
2) The compression softening effect does not significantly influence member behavior because the compressive stress and strain in the web are small. Except for deep beams with small shear span, compression softening does not significantly affect member behavior.
3) The maximum load capacity of a shear-dominated member is affected by the characteristics of the tension stiffening model. It is obvious that using the tension stiffening model for direct tension underestimates the load capacity of

the beams. Accordingly, the tension stiffening model considering the variation of two-dimensional stress-strain states should be used.

4) For the tension stiffening stress in the web of the beam, the influence of the main bar in the bottom of the beam should be considered. Otherwise, the analysis underestimates the actual load capacity. The diagonal crack width in the web is directly affected by the deformation of the main reinforcement in the bottom of the beam.

5) In this analysis, it is sometimes difficult to achieve convergence when there is a sudden decrease in the load capacity. However, as shown in Figure 7.11, such numerical problems do not affect the overall load-deflection history of the beams. In the next loading step, convergence can be accomplished.

To investigate the effect of bond-slip relations on beam members, Beam A-1 is idealized in two different discretizations shown in Figure 7.14. Discretization 1 is that used by Stevens [33]. Two of the four bottom reinforcing steel bars are cut off at 12 inches (30.5 cm) from the supports. The parameters for the bond-slip model are shown in Figure 7.14. In Discretization 2, all of the bottom reinforcing steel bars are cut off at 12 inches (30.5 cm) from the supports.

The comparison between the analyses of Stevens and of the author for Discretization 1 is given in Figure 7.15 (a). Since the two analyses adopt the same bond-slip model given by Eligehausen [18], and since both use the same bond-slip parameters, the predictions are expected to be the same. However, the ultimate strength predicted by Stevens' model is about 75 kips, only two-thirds of the strength found in the original experiment. On the other hand, the ultimate strength predicted by using the proposed model is only slightly lower than that of the original experiment.

In Figure 7.15 (b), the results of Discretizations 1 and 2 analyzed by the proposed model are compared. The ultimate strength of Discretization 2 is much lower than that of Discretization 1, and is almost the same as Stevens' analysis for Discretization 1. This analysis shows that the development length of the bottom steel bars is insufficient.

As shown the above analysis examples, the proposed model can precisely predict shear failure under monotonic loading without numerical difficulties. It can also predict bond failure of discrete reinforcing bars if the bond strength can be accurately estimated, and it can predict the impact of that bond failure on member behavior.

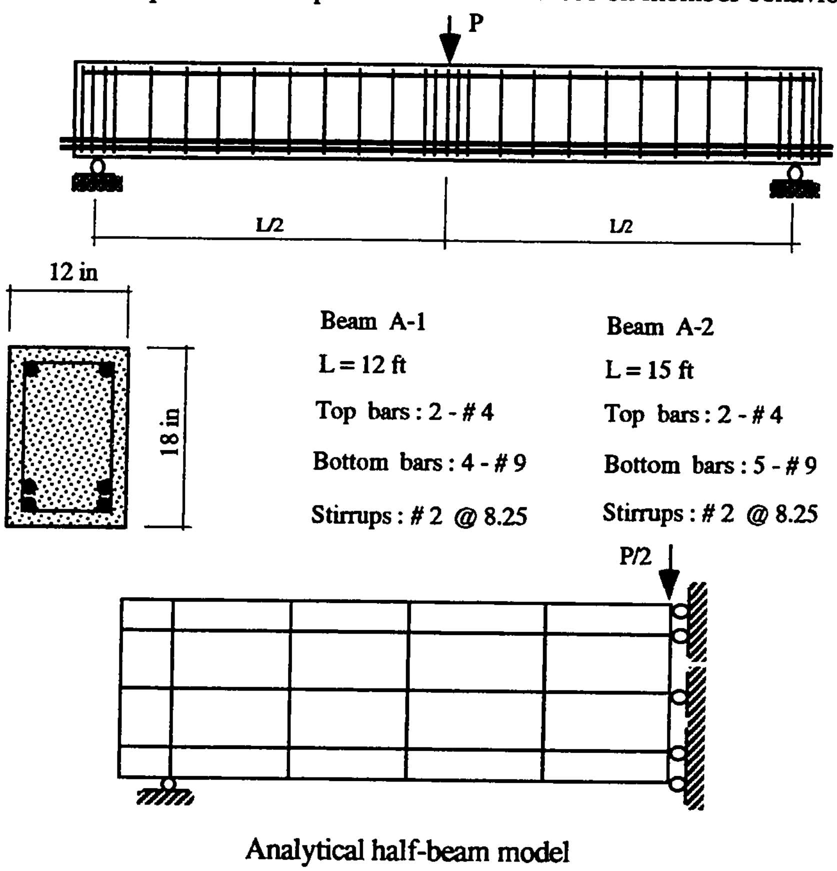
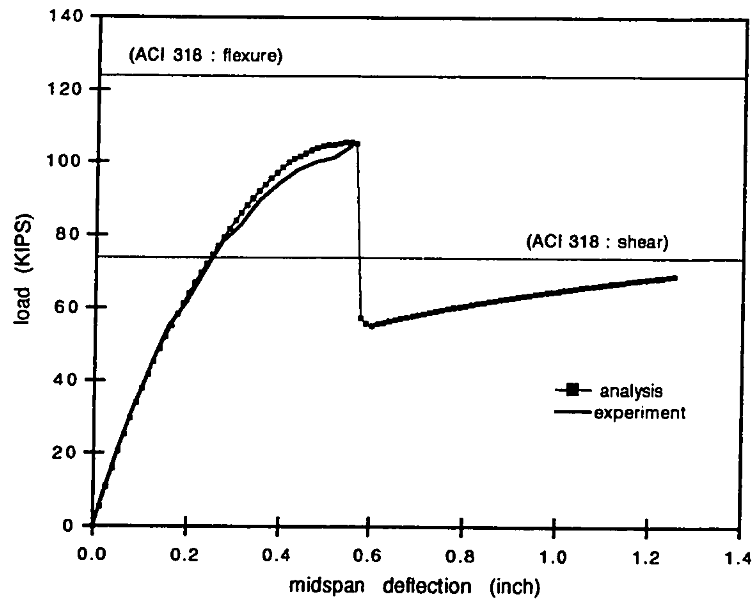
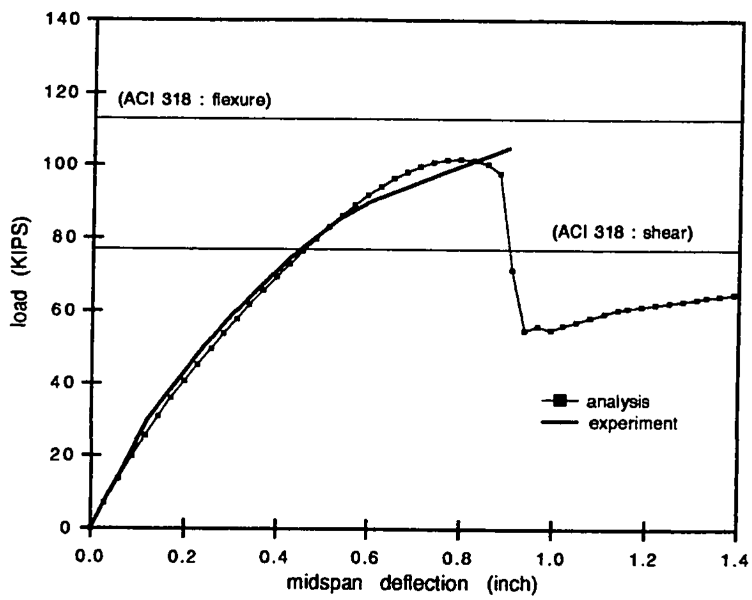


Figure 7.10 Reinforced concrete beam tested by Bresler and Scordelis [8]

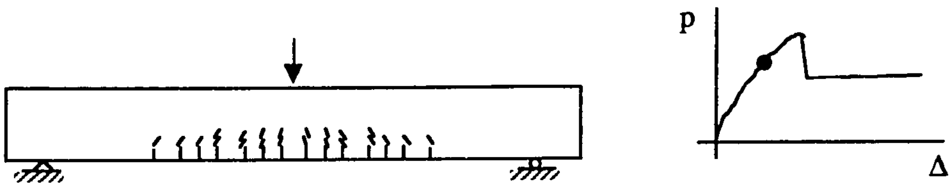


(a) Beam A-1

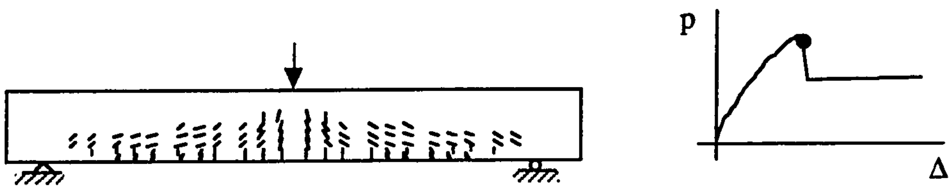


(b) Beam A-2

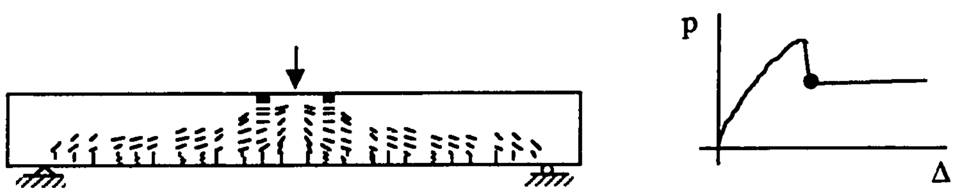
Figure 7.11 Comparison between analysis and experiment of reinforced concrete beam (Bresler and Scordelis [8])



a) Diagonal tensile cracking initiates.



b) The cracking spreads from the middle to the top of the beam, and compression crushing occurs.



c) Flexural behavior is no longer effective, and the deformation is governed by the crack opening

Figure 7.12 Development of failure mechanism of Beam A-1 [8]

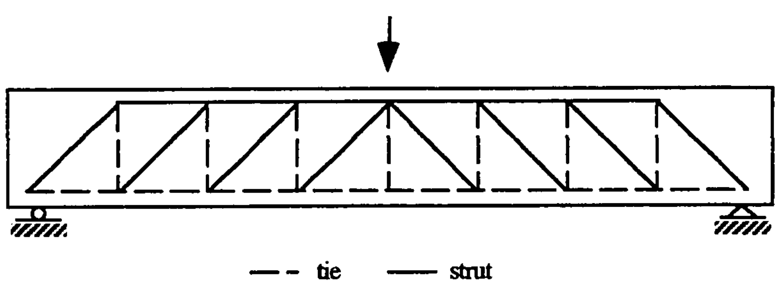
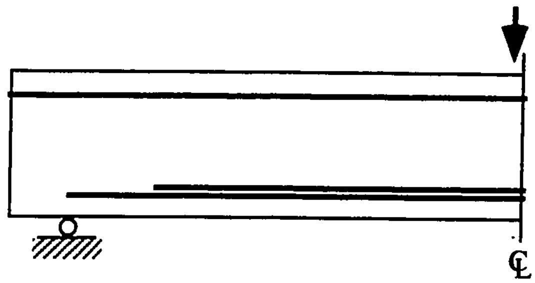
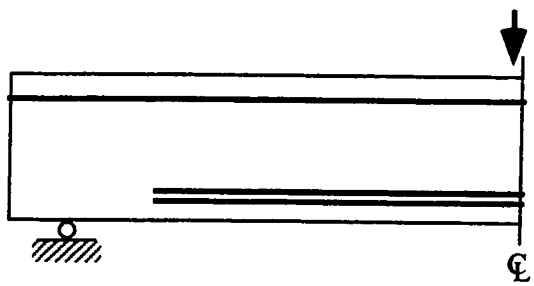


Figure 7.13 Strut-tie model



(a) Discretization 1 (Stevens)



(b) Discretization 2

$\tau_1 = 5.7$ MPa (0.83 ksi)

$\tau_2 = 0.1$ MPa (0.015 ksi)

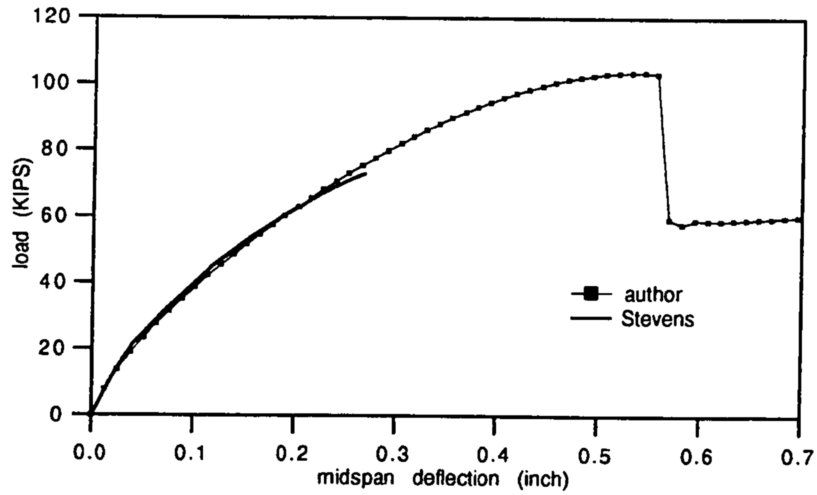
$s_1 = 0.63$ mm (0.025 in)

$s_2 = 1.27$ mm (0.5 in)

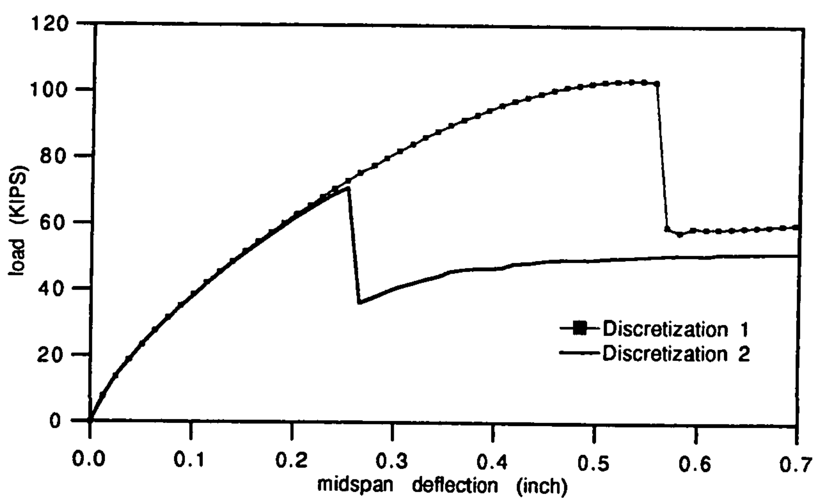
$s_3 = 11.0$ mm (4.3 in)

Bond parameters

Figure 7.14 Discretization of Beam A-1 for bond-slip behavior



(a) Comparison between the analyses of the author and of Stevens for Discretization 1



(b) Bond-slip effect on Discretizations 1 and 2

Figure 7.15 Bond-slip effect on beam behavior according to development length

7.4 Reinforced Concrete Beam Tests under Reversed Cyclic Loading (Brown and Jirsa)

Brown and Jirsa [9] performed a series of beam tests to determine the effect of cyclic load history on the strength, ductility, and mode of failure of beams. Here, two of their test beams are selected for analysis, representing the two types of cyclic behavior of the test beams: flexure-dominated and shear-dominated behaviors. As shown in Figure 7.16, each beam has two cross sections: they measure 6 x 12 inches at the free end, and 10 x 12 inches at the fixed support. These beams, designated as Beam 88-34-RV5-30 and Beam 66-35-RV10-60, have 30- and 60-inch shear spans, respectively. Their material properties are shown in Figure 7.16.

The beams are idealized using two discretizations. Mesh 1, shown in Figure 7.17 (a), idealizes the actual dimensions of the entire structure, including the two cross sections. Mesh 2, shown in Figure 7.17 (b), simplifies the actual beam by assuming the 6 x 12 beam has a fixed support at the interface between the 6 x 12 and the 10 x 18 sections. The main bars at top and bottom are modeled as discrete elements. The vertical reinforcement is treated as a smeared steel layer.

The analytical prediction of Mesh 2 using the equivalent reinforcement are compared with the test results in Figures 7.18 and 7.19 for Beam 88-34-RV5-30 and in Figures 7.20 and 7.21 for Beam 66-35-RV10-60. The following observations are made by comparing the load-displacement curves from the analyses and the experiments.

Beam 66-35-RV10-60 shows flexure-dominated behavior, and the load-displacement relations after the second half-cycle are very close to the cyclic characteristics of the reinforcing steel (Bauschinger effect). On the other hand, Beam 88-34-RV5-30 shows pinching during reversed cycles.

According to Ref. 9, the characteristics of cyclic behavior are influenced primarily by shear effects, and the third half-cycle of behavior significantly affects the subsequent cyclic behavior. The behavior of Beam 88-34-RV5-30, with a short shear span, is strongly influenced by shear forces and the corresponding web deformations. During unloading and reloading, the web cracks open and close suddenly. As a result, the member behavior shows pinching.

On the other hand, the behavior of Beam 66-35-RV10-60, with a larger shear span, is affected by bending rather than shear action. The web cracks remain narrow during the loading history, and do not affect member behavior. As a result, the member behavior is affected by the cyclic characteristics of horizontal reinforcing steel bars in the top and bottom of the beam section.

As shown in Figures 7.18 and 7.20, it is very difficult to predict the exact behavior of the specimen, because after several load reversals, bond-slip, concrete spalling, and early contact of crack surfaces significantly affect behavior. However, the general characteristics of the cyclic behavior can be predicted analytically up to the third half-cycle of the behavior.

The analyses after the second half-cycle underestimate the test capacities. This is because the cyclic stress-strain relation of reinforcing steel, proposed by Brown and Jirsa [9] underestimates the actual one, and because early contact of crack surfaces, inducing compressive stresses before complete crack closing, is not idealized in the proposed cracked concrete model.

In Figure 7.22, the load-deflection curve of Mesh 1 is compared with that of Mesh 2 for Beam 88-34-RV5-30. In the first load cycle, the member behavior of Mesh

2 is stiffer than that of Mesh 1, while after the first cycle, the member behavior of Mesh 2 is more flexible. The analysis results using Mesh 1 are closer to the test results.

This indicates that the member behavior is very sensitive to idealized boundary conditions, and that the analysis models should be close to the actual structures. Clearly, the fixed support condition in Mesh 2 provides stiffer boundary conditions than those of the actual specimens. Also, it is observed that the discrepancy between the two meshes for Beam 66-35-RV10-60 is much larger than that for Beam 88-34-RV5-30. According to Ref. 9, the plastic hinge zone that developed during cyclic loading was concentrated at the fixed end. The beam deformation depends on the hinge rotation which occurs within one-half of the effective depth (10 inches). Therefore, the member behavior of Beam 66-35-RV10-60 is more sensitive to the boundary conditions.

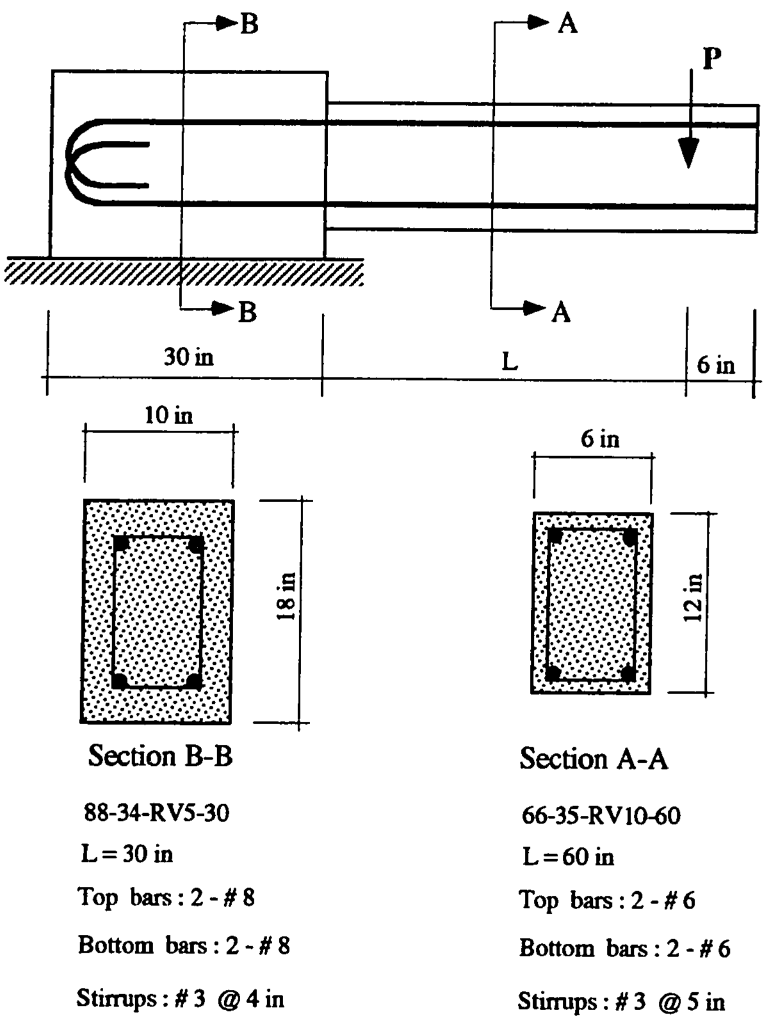
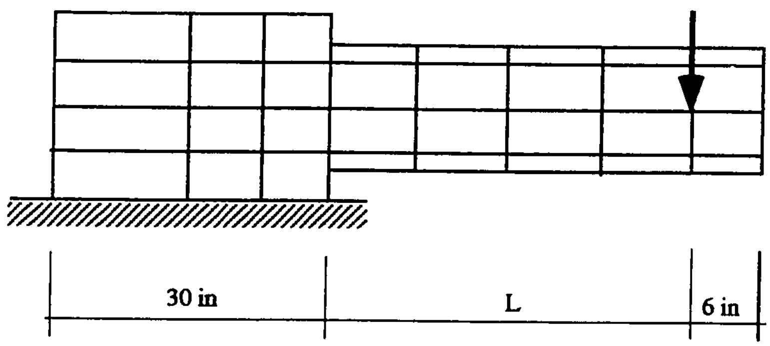
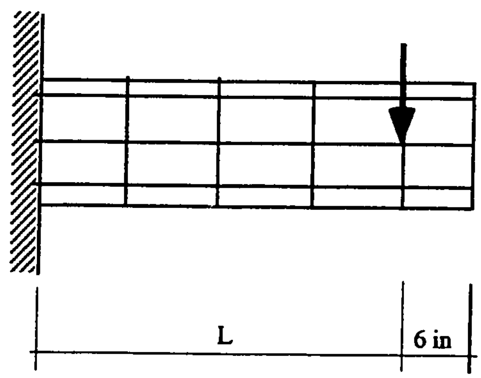


Figure 7.16 Cantilever beam tested by Brown and Jirsa [9]

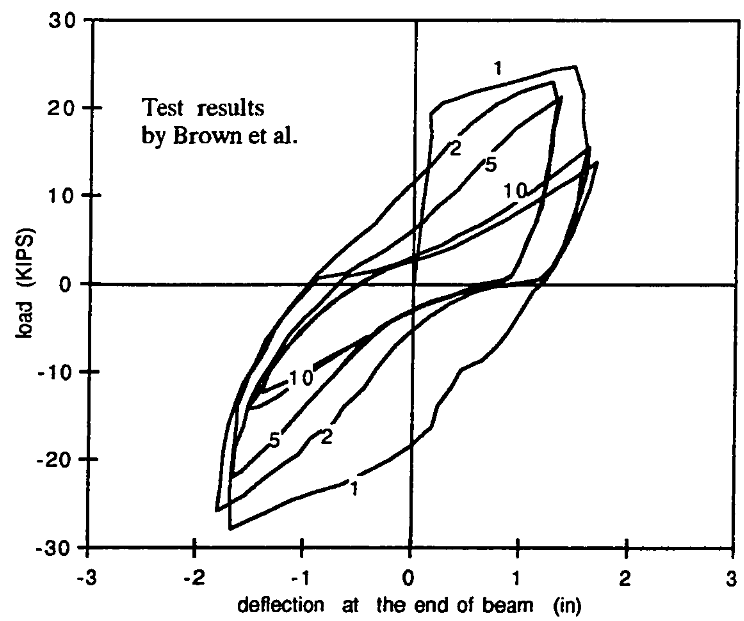


(a) Mesh 1



(b) Mesh 2

Figure 7.17 Member discretization of cantilever beam tested by Brown and Jirsa [8]



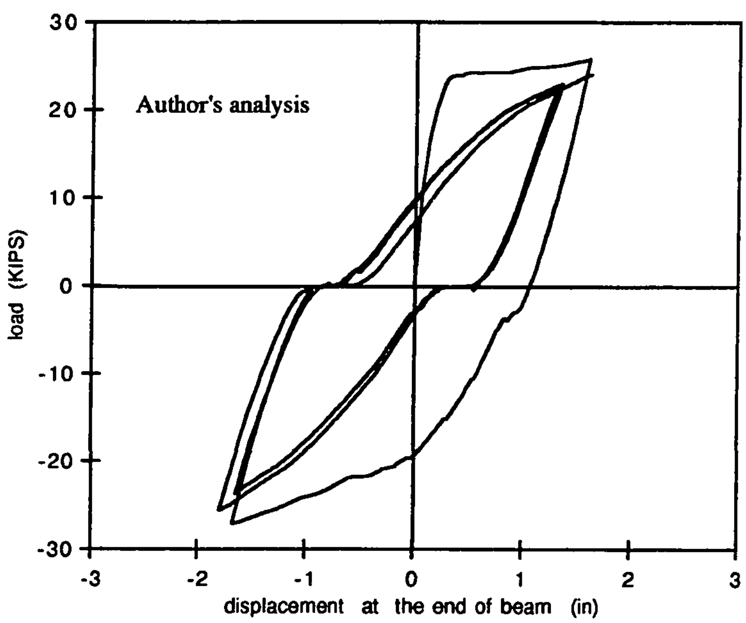


Figure 7.18 Load-deflection relations at the end of beam for Beam 88-34-RV5-30 (Brown and Jirsa [9])

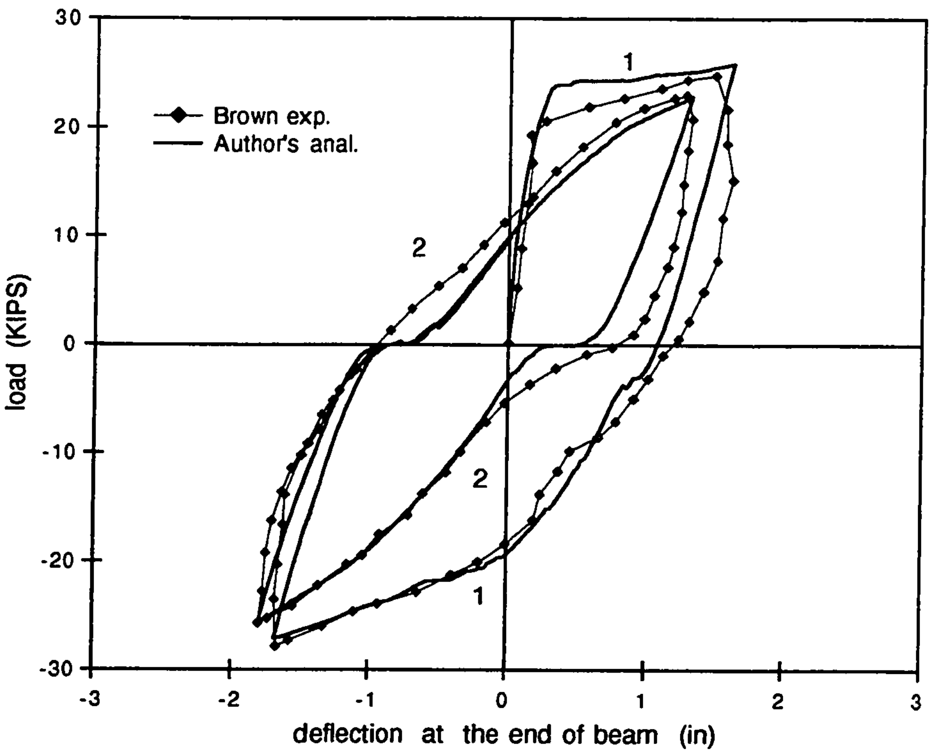


Figure 7.19 Comparison between analysis and test up to the second cycle for Beam 88-34-RV5-30 (Brown and Jirsa [9])

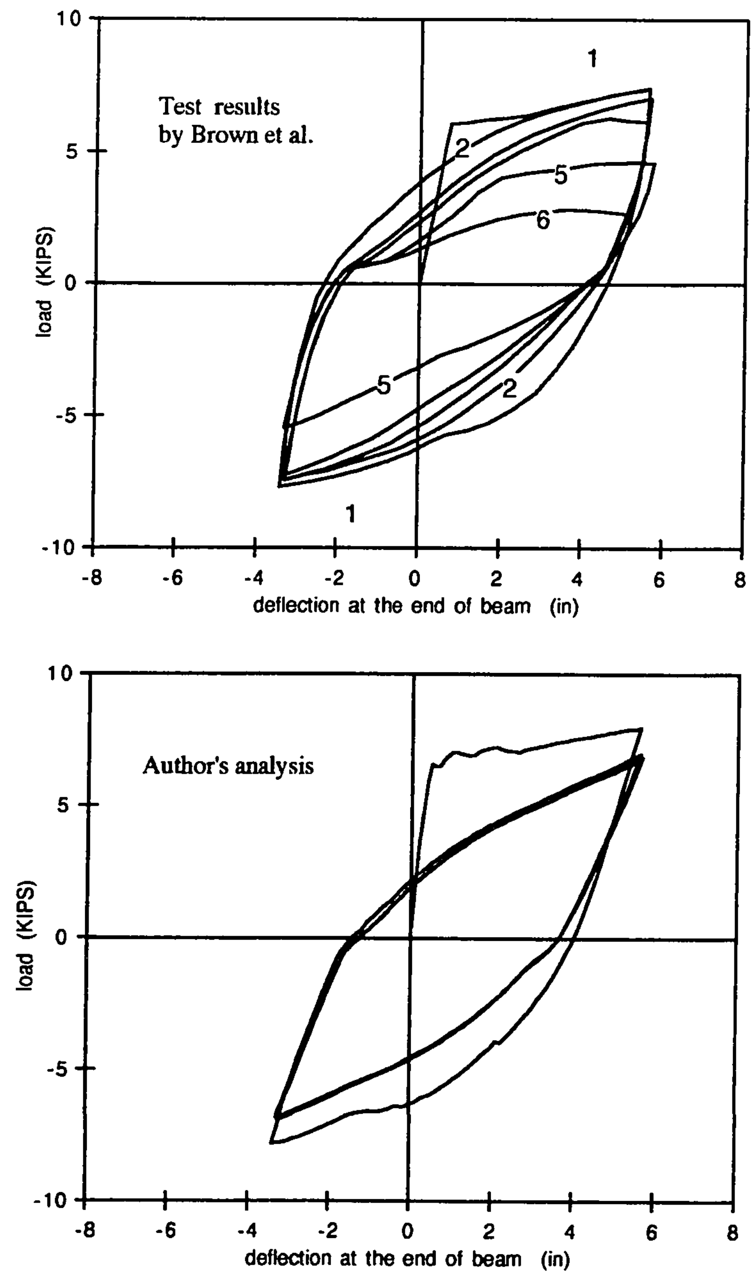


Figure 7.20 Load-deflection relations at the end of beam for Beam 66-35-RV10-60 (Brown and Jirsa [9])

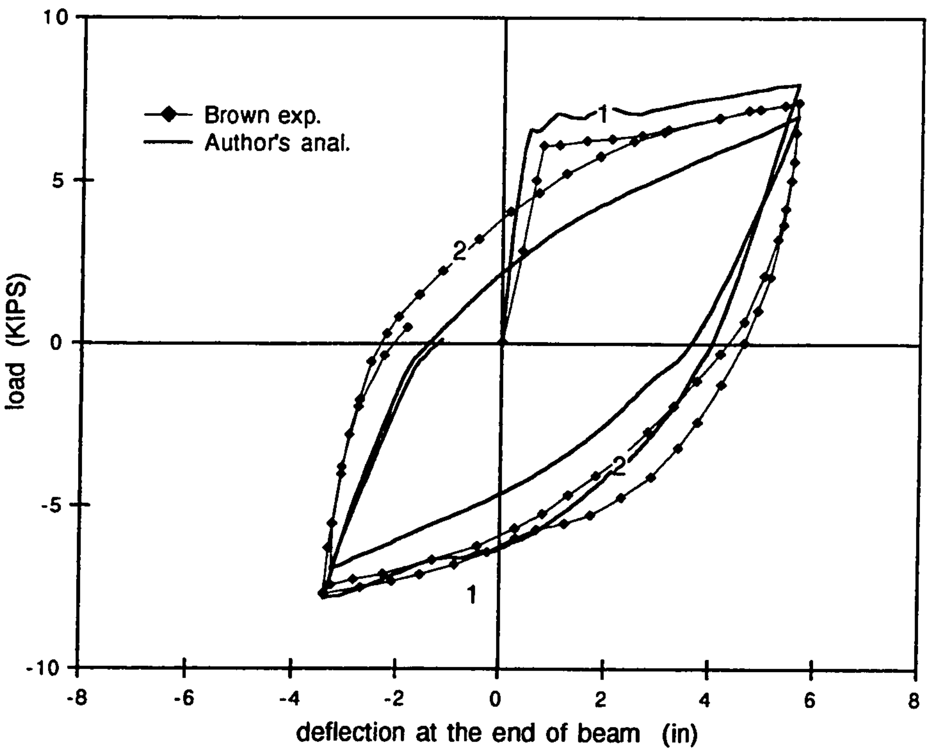


Figure 7.21 Comparison between analysis and test up to the second cycle for Beam 66-35-RV10-60 (Brown and Jirsa [9])

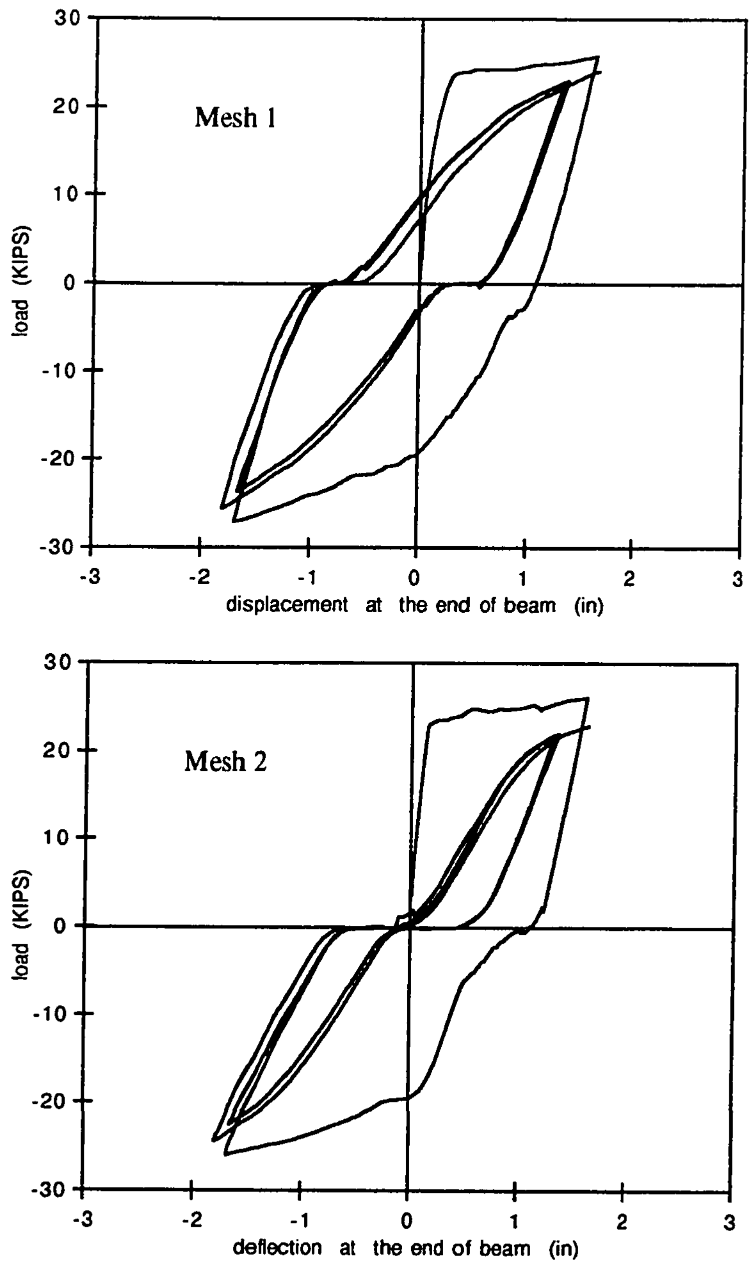


Figure 7.22 Comparison between Mesh 1 and Mesh 2
for Beam 88-34-RV5-30 (Brown and Jirsa [9])

7.5 Reinforced Concrete Masonry Wall Tests under Cyclic Loading (Shing et al.)

The analytical model is applied for the reinforced concrete masonry shear wall tests performed by Shing et al. at The University of Colorado [31]. The experiments were also analyzed by de la Rovere [31].

Shing's Walls 6, 10, and 12 are analyzed here. As shown in Figure 7.23, the shear walls have a rigid base and a top slab. They are subjected to uniformly distributed vertical loads and a concentrated horizontal load at the top slab. The vertical loads remain constant during loading, while the lateral load varies. The shear walls are reinforced by uniformly distributed vertical and horizontal steel layers. The loading conditions and the properties of materials are shown in Table 7.2 [31].

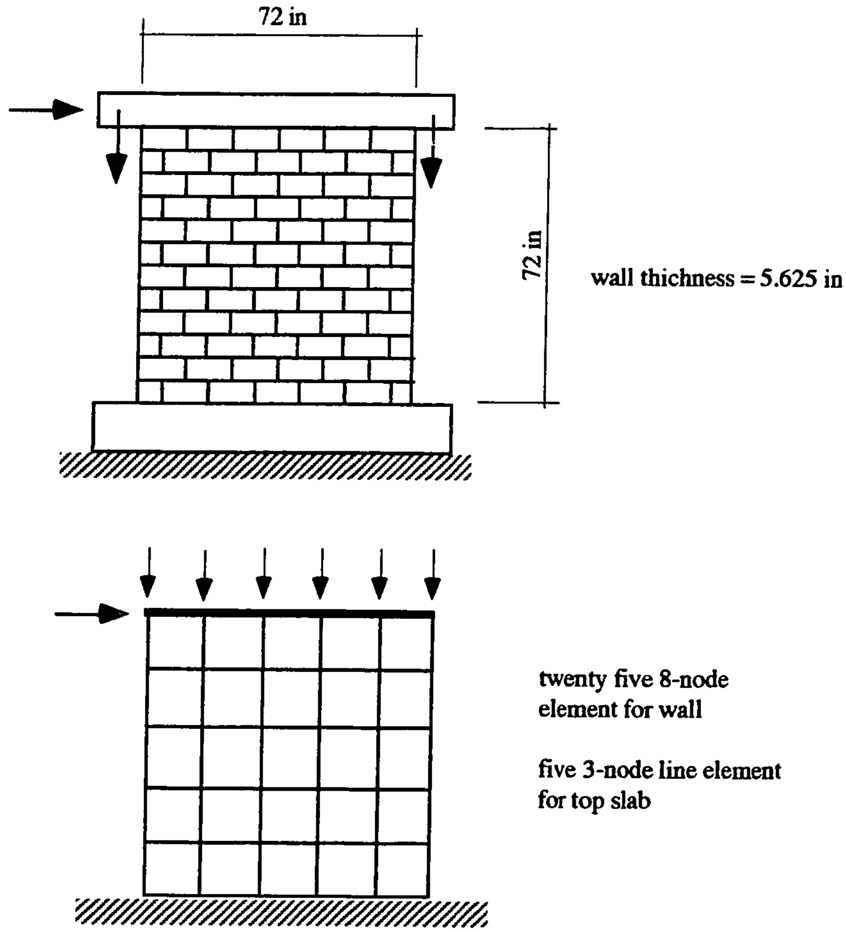
For the analytical model, the shear wall and the top slab are idealized by twenty-five 8-node rectangular elements and five 3-node line elements. The line elements idealize the top slab with large axial stiffness; and the flexural stiffness of the top slab is neglected. The vertical and horizontal reinforcing steel layers are idealized by smeared reinforcement. The vertical load is idealized as equivalent joint loads, and the lateral load is assumed to act on the middle of the top slab. At first, the vertical loads increase up to the constant amount shown in Table 7.2, under force control, and the cyclic lateral load then increases under displacement control.

In Figures 7.24 - 7.26, the analyses are compared with those from experiment on Walls 7, 10, and 12. The experimental cyclic curves are picked up from the entire history curves to clearly compare the analysis results. In the shear walls, the well-distributed reinforcing steel layers and the vertical loads prevent the tensile cracks from widening. As a result, the shear walls fail due to compressive crushing of concrete.

Wall 7 with heavy vertical reinforcement draws large horizontal load. However, compression crushing occurs suddenly just after the maximum horizontal load due to the relatively small horizontal reinforcement. On the other hand, Walls 10 and 12 with the reinforcement balanced horizontally and vertically have less load capacity for horizontal load than Wall 7, but show ductile behavior after the maximum horizontal load.

For all specimens, the analytical results follow the experiments reasonably well. This is because the well-distributed reinforcement and the vertical load prevent the tensile cracks from widening so that the tensile cracks spread over large area and material deterioration due to cyclic loading is minimized. The member behavior after the maximum member capacity depends heavily on the descending slope of the compressive softening stress-strain relation of concrete. In these analyses, $\sigma_c^f = \sigma_c^u / 20$ and $\varepsilon_c^f = 15\,\varepsilon_c^u$ are used for the final stress and strain in Figure 3.4. Referring to Ref. 31, the author's analyses produce better predictions than de la Rovere's analysis, especially for Wall 7.

As shown above, the proposed material model, using the concept of smeared crack and smeared reinforcement, can predict the maximum load capacity of the shear wall structures, and can also predict the post-failure behavior accurately, provided that the softening relation in the descending branch of the concrete stress-strain curve is well estimated.



Analytical Model

Figure 7.23 Reinforced concrete masonry wall tested by Shing et al.

Table 7.2 Loading conditions and material properties of shear wall [31]

Wall No.	Masonry σ'_m (psi)	Horizontal Steel		Vertical Steel		Axial load (KIPS)
		ρ_x (%)	f_{xy} (ksi)	ρ_y (%)	f_{yy} (ksi)	
7	3000	0.14	56	0.74	70	0.01056
10	3200	0.14	56	0.38	63	0.01785
12	3200	0.24	66	0.38	63	0.01785

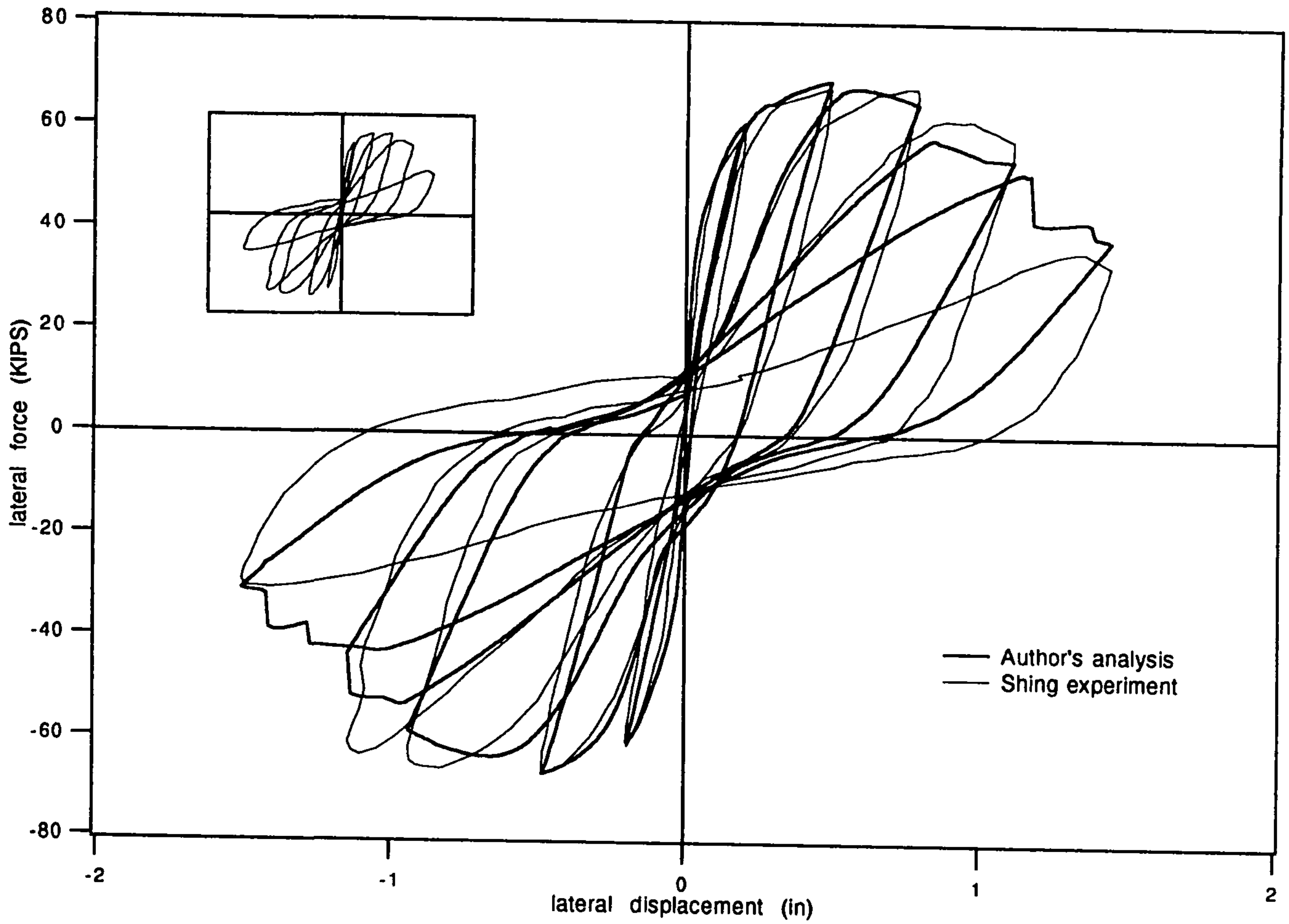


Figure 7.24 Comparison between cyclic analysis and experiments for Wall 10

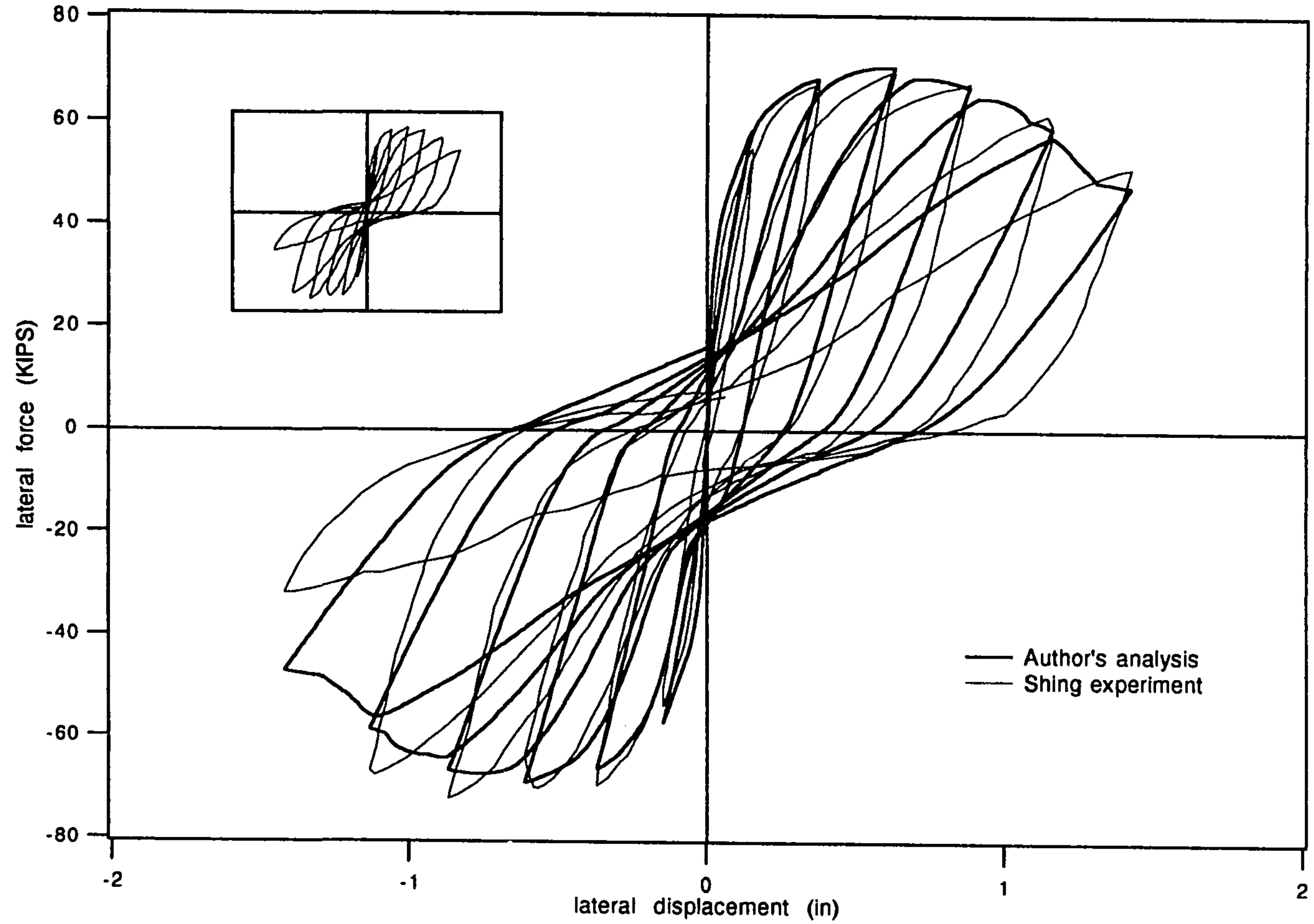


Figure 7.25 Comparison between cyclic analysis and experiments for Wall 12

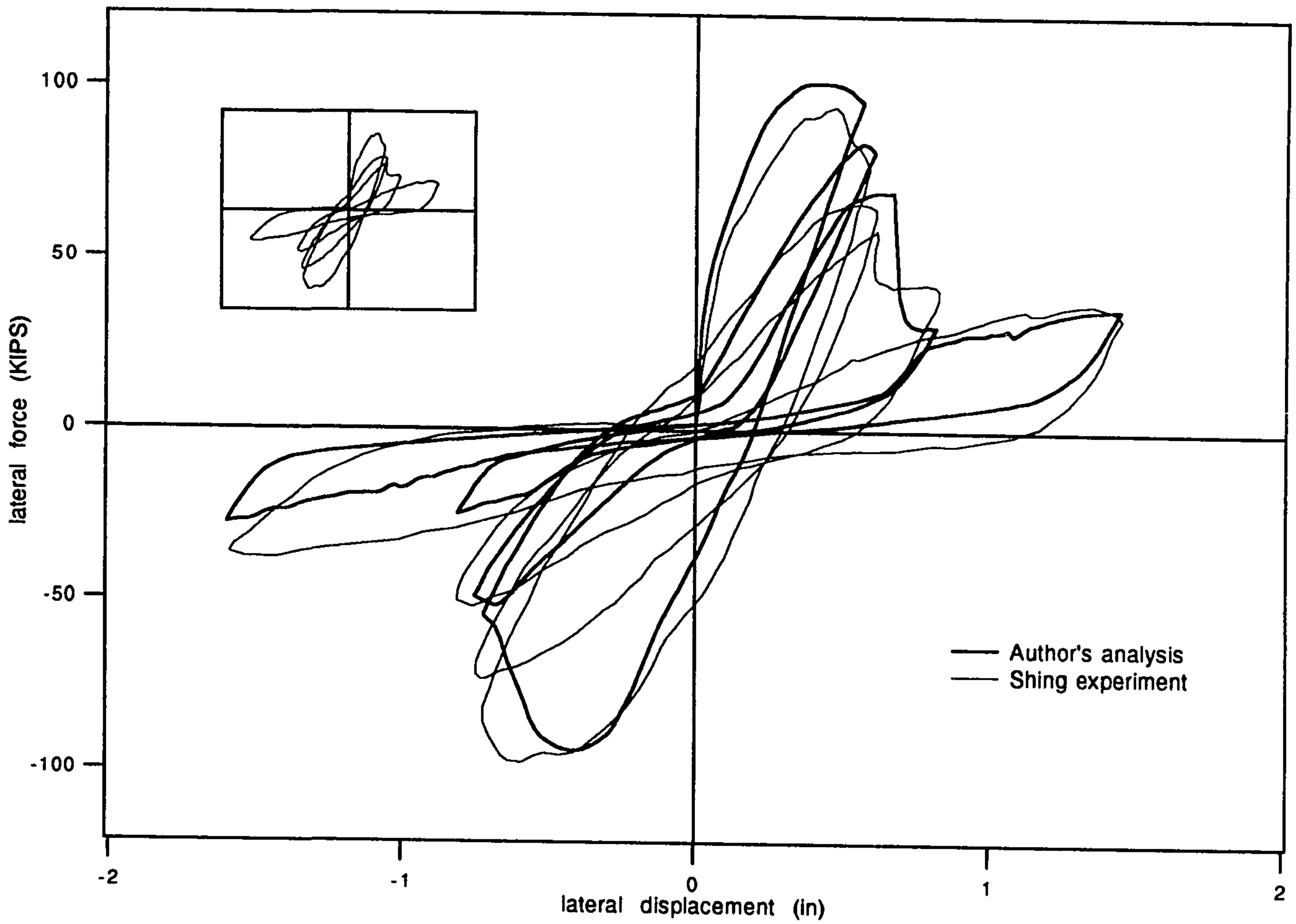


Figure 7.26 Comparison between cyclic analysis and experiments for Wall 7

8.0 SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

8.1 General

The purpose of this research is to develop a reliable analysis method which is able to predict the complete behavior up to structural failure of reinforced concrete planar members under cyclic as well as monotonic loading. The structural members to which the analysis applies are beams, columns, beam-column joints, and shear walls, all of which experience damage initiated by tension cracking.

The proposed analytical approach can simulate the behavior of reinforced concrete structural members due to crack opening and closing, compressive crushing, cyclic history of reinforcing steel and bond-slip between cracked concrete and reinforcing steel.

By simulating the complete structural response, the proposed analytical approach predicts behavioral characteristics such as ultimate strength, inelastic deformations, primary crack orientations, and failure mechanisms, and is useful for the design and retrofit of reinforced concrete structural members.

To accomplish the objectives noted above, this work includes an investigation of material models for two-dimensional finite element analysis under in-plane cyclic and monotonic loading. Also, several nonlinear solution schemes are investigated to develop a numerically reliable analysis method. The proposed material models and the numerical approach are verified using previously reported experimental results.

8.2 Summary of Proposed Analysis Method

A cracked concrete material model, referred to as a rotating orthotropic axes model with successive cracking, is proposed. This model complements existing rotating crack models. In the proposed model, the following assumptions are used to idealize the behavior of cracked reinforced concrete:

1) The concept of smeared cracking is assumed to be valid. The smeared crack is regarded as a continuous material strain. Based on the concept of smeared cracking, the tensile stress and strain of cracked concrete are defined in terms of average stress and strain across tension cracks.
2) Principal stress axes coincide with principal strain axes.
3) Cracked concrete is idealized as an orthotropic material, and the orthotropic axes coincide with principal axes. The progressive cracking process due to primary and secondary cracking continuously gives behavioral directionality of concrete in rotating principal axes. The orthotropic axes rotate to the principal axes during loading.
3) In the orthotropic axes, equivalent uniaxial stress-strain relations in two orthogonal principal axes are uncoupled in terms of material strain. In cracked concrete, the tension stiffening stress induced by bonding action of reinforcing steel is negligible compared with the compressive strength of concrete, and it is localized around the reinforcing steel and the cracking zone. Accordingly, the reciprocal effect of the compressive and tensile material stress-strain relations is neglected. To consider the effect of crack opening,

the equivalent uniaxial stress-strain relations are coupled in terms of average strain.

On the basis of those assumptions, the concept of the proposed approach can be summarized as follows:

1) Since a uniaxial stress-strain state is maintained in cracked concrete, isotropic compression damage representing concrete crushing is uniform in any rotating principal direction. On the other hand, anisotropic tension damage, which represents tensile cracking, localizes in the initial crack direction.
2) Compressive strength of concrete is reduced by the principal tensile strain representing the existing crack opening.
3) Cracked concrete has considerable tension stiffening stresses as long as at least one reinforcement layer crossing the existing cracks remains elastic. Accordingly, the tension stiffening stress is not directly related to the current principal strain, but the tensile strain of the reinforcement and its direction.

The general behavior of the proposed cracked concrete model is defined in the following way:

1) The two-dimensional stress-strain relation is defined by two equivalent uniaxial stress-strain curves in orthotropic axes. The orthotropic axes rotate to current principal axes during the loading history.
2) The equivalent uniaxial stress-strain curve consists of envelope curves (loading curves) and unloading-reloading curves connecting the envelope

curves. The compressive envelope curve depends on the uniaxial stress-strain relation, including the compression softening effect due to crack opening. The tension stiffening stress of the tensile envelope curve is determined by the influence of each reinforcement layer which remain elastic.

3) If the equivalent uniaxial strains exceed the compression or tension damage surface, the damage surface expands according to its expansion rule, and the equivalent stress-strain relation follows the compressive or tensile envelope curve or the loading curve.

4) If the equivalent uniaxial strain lies inside the damage surfaces, the equivalent stress-strain relation lies on the unloading-reloading curves connecting the compressive and tensile envelope curves at the damage surfaces.

In addition to the proposed cracked concrete model, existing material models of reinforcing steel and bond-slip are implemented in the analysis program. To idealize reinforcing steel behavior in this study, two constitutive models are used: a bilinear model including a kinematic hardening rule; and a strain hardening model including the Bauschinger effect. The reinforcing steel is idealized by either discrete or smeared elements. The bond-slip model idealizes the bond deterioration due to cyclic loading. The bond-slip elements, which are out-of-plane rectangular elements, connect the in-plane rectangular elements representing concrete and smeared reinforcement, to line elements representing discrete reinforcement.

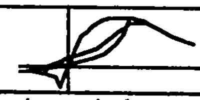
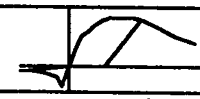
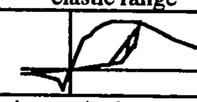
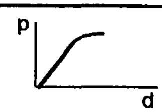
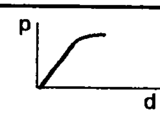
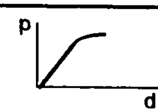
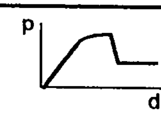
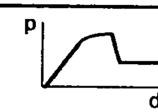
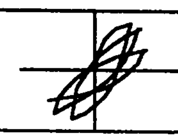
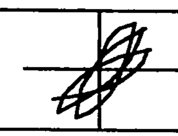
A finite element computer program is developed to apply the proposed cracked concrete model and the existing models of reinforcing steel and bond-slip. To produce a reliable solution scheme for the applied material models, extensive programming and computer work has been performed. As a result, a simplified displacement-control method is used for the nonlinear numerical procedure, and Newton-Rapshon method with tangent stiffness is used for the iteration scheme.

In Table 8.1, the various orthotropic axes models are compared with respect to material modeling, analysis results, and the range of their application. The original works of the proposed analysis method are the following considerations in material models:

1) Two-dimensional tension stiffening model;
2) Isotropic damage due to compressive crushing and anisotropic damage due to tensile cracking;
3) Cyclic characteristics of reinforcing steel including Baushinger effect; and
4) Bond-slip between concrete and reinforcing steel.

Also, the proposed analysis method predicts the complete behavior up to structural failure, and extends its application to a variety of load conditions and structural members.

Table 8.1 Comparison of various orthotropic axes models

		Darwin & Pecknold	Vecchio	Stevens	de la Rovere	Proposed model
applied structure types		R/C beam & shear wall	R/C beam	R/C beam & shear wall	masonry shear wall	R/C beam, R/C and masonry shear wall
concrete model	orientation of ortho-tropic axes	principal stress axes	principal stress and strain axes coincide	principal incremental stress and strain axes coincide	principal stress axes	principal stress and strain axes coincide
	compression softening	N/A	equation proposed by Vecchio	equation proposed by Vecchio	equation proposed by Vecchio	equation proposed by Vecchio
	tension stiffening	N/A	empirical equation	consideration of reinforcement direction	trial of various empirical equations	consideration of reinforcement in elastic range
	cyclic behavior		N/A			
	damage history	uniaxial stress-strain relation in terms of equivalent uniaxial strain	N/A	anisotropic damage in compression and tension	two separate damage history regardless of the orientation of orthotropic axes	isotropic damage in compression, anisotropic damage in tension
reinforcing steel		bilinear model	bilinear model	Bauschinger effect	bilinear model	Bauschinger effect
bond-slip		N/A	N/A	Eligehausen Model	N/A	Eligehausen Model
analysis result	monotonic loading	 not successful in post-failure behavior	 not successful in post-failure behavior	 not successful in post-failure behavior	 post-failure behavior	 post-failure behavior
	cyclic loading	only one cycle	N/A	not always successful		

8.3 Conclusions

1) The rotating orthotropic axes model with successive cracking complements the rotating crack model which is controversial. The concept of successive cracking process justifies the fact that the orthotropic axes are established in the current principal axes rotating during loading.

2) Fixed crack model is not appropriate to define the stress-induced orthotropic characteristics of cracked concrete because secondary cracks are developed in current principal axes different from primary crack direction.

3) The assumption that principal stress axes coincide with principal strain axes, can overestimate the load capacity of structural members.

4) The proposed material model can predict the characteristics of shear-dominated as well as flexure-dominated member behavior under monotonic loading.

5) The proposed material model can predict the brittle failure of shear-dominated members. By following member behavior up to a given target displacement without numerical failure, the proposed model can clearly define a member's maximum load capacity at which the brittle failure occurs.

6) The maximum load capacity of a shear-dominated member is affected by the characteristics of the tension stiffening model. It is obvious that using the tension stiffening model for direct tension underestimates the load capacity of the beams. Accordingly, a tension stiffening model for two-dimensional stress states, such as that proposed here, should be used.

7) The proposed model includes bond-slip behavior of discrete reinforcing steel bars, so that it can predict the impact of bond-slip on the overall member

behavior. Therefore, the proposed model can be used to investigate the effects of an anchorage length on the member behavior.

8) The analysis using the proposed material model can predict the various types of cyclic characteristics of planar structures. It can predict the types of the member failure initiated by either reinforcing steel yielding or concrete crushing, and can predict unloading-reloading behavior which is either shear-dominated or flexure-dominated.

9) It is difficult to predict exactly the behavior due to fatigue failure, because after several load reversals, bond deterioration and concrete spalling significantly affect the behavior. The existing bond-slip model is not sufficient to predict fatigue failure due to cyclic loading.

10) Since reasonably predicting most planar member behavior of cracked concrete, the proposed analysis method can be applied for complex combination of structural members, such as the substructure of beam, column, and their joints.

11) The proposed analysis method provides a basis on implementing and investigating the effect of the various phenomena of cracked concrete on the overall member behavior, such as slip on the interface of supports, the relation of crushed concrete and bond to reinforcement, the development length of reinforcement, bond deterioration due to cyclic loading, and so on.

12) As far as the basic concept is concerned, the rotating orthotropic axes model proposed here is an extension of the strut-and-tie model defined in a load-displacement field, frequently used as an approximate analysis and design method. However, the rotating orthotropic axes model can consider the nature of cracked concrete behavior, such as compression softening due to

crack opening and tension stiffening effects. Also, since it is possible to adjust the direction of strut-and-tie to current principal axes by considering equilibrium and compatibility conditions, the proposed model can reasonably predict cyclic as well as monotonic behavior.

8.4 Recommendations for Further Research

1) The proposed analysis program should be extended to address member behavior governed by concrete crushing under biaxial compression.
2) To be used for general loading, the proposed material model should be verified for non-proportional loading.
3) Since the cyclic characteristics of cracked concrete depends on the cyclic stress-strain relation of reinforcing steel, it is recommended that more accurate reinforcing steel model able to show reasonable Bauschinger effects be used.
4) In multiply cracked concrete, the behavior of reinforcement in a crack direction is independent of that in the other crack direction, even for a single reinforcement layer. Accordingly, the behavior of reinforcing steel should be related not to the smeared strain in the reinforcement direction but to the corresponding crack width. Since the current concept of smeared cracking and smeared reinforcement does not permit consideration of individual reinforcement behavior each crack direction, more research on the interaction between cracks and reinforcement is required.
5) The proposed cracked concrete models, simplifying the actual stress-strain relations, use the same stress-strain path in both unloading and reloading. To

consider the material fatigue due to repeated loading, a more sophisticated material model should be used.

6) For the member behavior governed by the yielding of reinforcing steel, bond-deterioration due to cyclic loading is much more serious than that predicted by the existing bond-slip model. For this reason, bond-slip between concrete with closely spaced cracks and yielded reinforcing steel should be investigated.

7) The bond-slip element, if the parameters are appropriately adjusted, can be used for the various bond-slip behavior between prestressed tendon and concrete in prestressed concrete, between different materials in composite members, at the base of shear walls, and so on.

APPENDICES (Finite Element Analysis Program)

A.1 Introduction of Program RCCRAK

The finite element analysis program RCCRAK (Two-Dimensional Analysis of Reinforced Concrete with Crack Damage) was developed to complement the author's research of cracked concrete behavior. The program can be used for analysis of two-dimensional reinforced concrete structural members subjected to either monotonic or cyclic loading, such as beams, beam-column joints, and shear walls.

This program was written in Fortran 77 by the author. The matrix solution subroutines and the memory array of this program are based on the program distributed by Professor J. L. Tassoulas in the University of Texas at Austin. As a solution method of equations, the Frontal Method is implemented to save main memory. As a nonlinear solution scheme, the Newton-Rapshon method with tangent stiffness is used. Each equilibrium position during analysis is controlled by the step size of either forces or displacements, initially given by elastic analysis. The final equilibrium position in each load cycle is controlled by target forces or displacements.

This program uses 4- and 8-node rectangular elements for concrete with smeared reinforcement, 2- and 3-node line elements for discrete reinforcement, and 4- and 6-node out-of-plane rectangular elements for bond-slip. This program uses the following material models as introduced in the main chapters:

1) Cracked concrete model referred to as the rotating orthotropic axes model with successive cracking, including compression softening and tension stiffening;

2) Smeared and discrete reinforcing steel model of either bilinear model including kinematic hardening or nonlinear model including Bauschinger effect; and

3) Bond-slip model proposed by Eligehausen in the University of California at Berkeley.

Next, several recommendations will be given for the program users.

1) This program does not accurately predict the member behavior governed by biaxial compression stress states.

2) If convergence problems occur frequently during analysis, or if the load-deflection curve is not smooth, reduce the loading step size.

3) A target tolerance of 1% is recommended. Smaller tolerances may increase computer running time considerably.

4) It is recommended that large capacity computers, such as work-stations be used, rather than micro-personal computer.

5) The structure should be idealized close to the actual support and loading conditions. Concentrated loads can cause local failure or local large deformation. The fixed supports can overestimate actual member constraints.

6) The material models implemented in the program use simplified unloading-reloading stress-strain paths. As a result, the analysis results

under repeated loading (not complete cyclic loading) may not be close to the experimental results.

7) This program does not idealize the strength deterioration due to fatigue phenomena, such as concrete spalling.

This program automatically produces two output files with the suffixes of '.GEN' and '.SPE'. The output file, 'filename.GEN', contains general information of input data and displacement- and force- tolerances each iteration. The output file, 'filename.SPE', contains the load and deformation at the selected node. Also, the program produces unformatted files to be used for post-processing. This program has a post-processing sub-program, RCPOST, to output analysis results, such as applied loads, deformations, stress-strain relations, orientation of principal axes, and bond stress-slip relations. The post-processing will be presented in Section A.3.

A.2 Example Input File

A.2.1 General Information

'*' marks indicate selection options recommended by the author

1) **General Input**

1. NYEXIST : restarting code

   = 0 (restart)

   = 1 (new input file)

2. ICOMP : element type

   = 1 (4 node rectangular & 2 node line elements)

   = 2 (8 node rectangular & 3 node line elements) *

3. NDIM : no. of dimension
4. NN : no. of nodes
5. NUMEL1 : no. of line elements
6. NUMEL2 : no. of rectangular elements
7. NUMEL3 : no. of bond slip elements
8. NMAT1 : no. material types of reinforcing steel
9. NMAT2 : no. material types of concrete with smeared steel
10. MNDOFN : maximum no. of degree of freedom per node
11. MNNE : maximum number of nodes per element
12. MNCM : maximum no. of constants per material type
13. NGAU : no. of Gaussian points per axis in a element

    = 2 (4 node rectangular & 2 node line elements)

    = 3 (8 node rectangular & 3 node line elements)

14. NCYCLE : no. of half load cycles

2) **Material Input for Reinforcing Steel (NMAT1)**

1. NCM : no. of constant
2. CONSTM(1) : yield stress
3. CONSTM(2) : Young's modulus
4. CONSTM(3) : reinforcement ratio

5. CONSTM(4) : direction with respect to x axis
6. CONSTM(5) : diameter
7. CONSTM(6) : area
8. CONSTM(7) : strain hardening strain
9. CONSTM(8) : ultimate strain
10. CONSTM(9) : ultimate stress
11. CONSTM(10) : ultimate bond stress, τ_1
12. CONSTM(11) : final bond stress, τ_3
13. CONSTM(12) : bond slip, s_1
14. CONSTM(13) : bond slip, s_2
15. CONSTM(14) : final bond slip, s_3

3) Material Input for Concrete with Smeared Steel (NMAT2)

1. NCM : no. of constant
2. CONSTM(1) : maximum stress
3. CONSTM(2) : Poisson ratio
4. CONSTM(3) : thickness
5. CONSTM(4) : void
6. CONSTM(5) : unit weight (positive direction in y axis)
7. CONSTM(6) : maximum stress in compression
8. CONSTM(7) : initial tangent stiffness in compression
9. CONSTM(8) : secant stiffness for maximum stress in compression
10. CONSTM(9) : secant stiffness for final stress in compression
11. CONSTM(10) : final stress in compression
12. CONSTM(11) : maximum stress in tension
13. CONSTM(12) : initial tangent stiffness in tension
14. CONSTM(13) : secant stiffness for maximum stress in tension
15. CONSTM(14) : secant stiffness for final stress in tension
16. CONSTM(15) : final stress in compression
17. CONSTM(16) : material type no. of reinforcement layer 1
18. CONSTM(17) : material type no. of reinforcement layer 2
19. CONSTM(18-21) : material type no. of discrete reinforcement bars affecting tension stiffening in the corresponding element.

4) **Coordinate Input (NN)**

1. K : no. of node
2. X(1) : x coordinate
3. X(2) : y coordinate
4. 2 (default)
5. IS(1) : constraint code for x degree of freedom

   = 0 (free)

   = 1 (fixed)
6. IS(2) : constraint code for y degree of freedom

5) **Element Input (NUMEL1 & NUMEL2 & NUMEL3)**

1. K : no. of node
2. IELT : element type

   = 1 : line element

   = 2 : rectangular element

   = 3 : bond slip element
3. IELM : element material type
4. NNE : no. of node
5. ICONN(NNE) : connected node no.

6) **LOAD CASE 1 & LOAD CASE 2**

1. NODE : node no.
2. P1(1) : joint load in x axis
3. P1(2) : joint load in y axis
4. -999999 : indication of the end of the load case

7) **Nonlinear Information**

1. ISOL : selection of compressive cyclic model of concrete

   = 1 : simplified model *

   = 2 : hysteresis model
2. NS : maximum no. of load step
3. MIT : maximum no. of iteration

4. TOL : tolerance
5. ISYM : symmetric matrix

   = 1 (default)
6. NRHS : no. of right hand side (force vector)

   = 1 (default)
7. EEFT : concrete final strain in tension
8. FFT : ratio of strain hardening stiffness to elastic stiffness for bilinear model of reinforcing steel
9. IBAU : reinforcing model select code

   = 1 : Bilinear model

   = 2 : Strain hardening model with Bauschinger effect *

8) Load Conditions (NCYCLE)

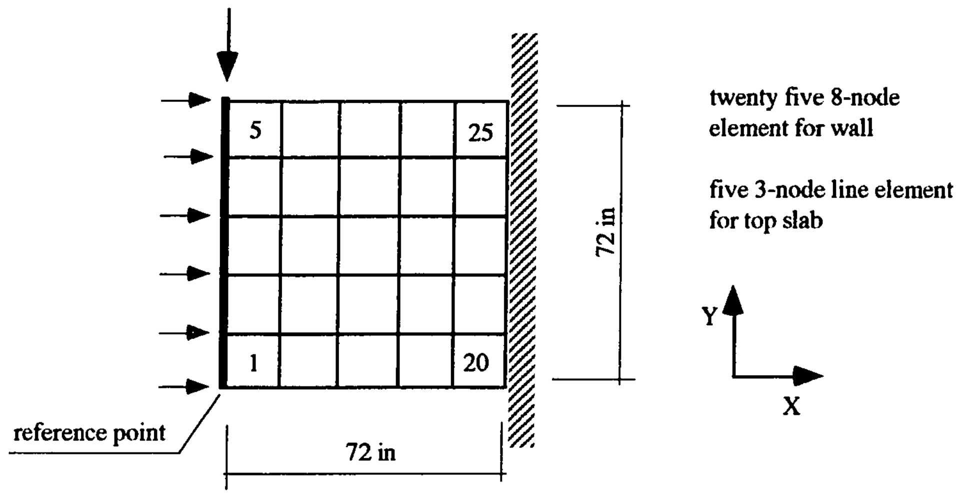
1. IFDIS : controlled variable selection code

   = 1 : load control

   = 2 : displacement control *
2. NODE : controlled node
3. NNUDOF : controlled degree of freedom of NODE
4. CYLOAD : target load or displacement of NNUDOF
5. FLD1 : load factor of LOAD CASE 1
6. FLD2 : load factor of LOAD CASE 2

A.2.2 Example Input Data for Wall 7

Graphical Description of Wall 7



INPUT DATA

0 2	NYEXIST ICOMP
2	NDIM
96 5 25 0	NN NUMEL1 NUMEL2 NUMEL
3 1 2	NMAT1 NMAT2 MNDOFN
8 21 3 12	MNNE MNCM NGAU NCYCLE
9 56.D0 2.9D4 0.14d-2 90.D0 1.59D0 1.D0 0.01D0 0.2D0	NCM CONSTM(1-14)
80.D0	
9 63.D0 2.9D4 0.38D-2 -0.D0 1.59D0 1.D0 0.01D0 0.2D0	NCM CONSTM(1-14)
87.D0	
9 58.D0 2.9D4 0.0d-2 90.D0 1.59D0 1.D8 0.01D0 0.2D0	NCM CONSTM(1-14)
87.D0	
21 3.2D0 0.15D0 5.625D0 2.D0 0.D-2	NCM CONSTM(1-21)
3.2D0 2460.D0 1230.D0 4.D0 .15D0	
1.D-1 2460.D0 500.D0 100.D-1 0.6D-1	
1.D0 2.D0 0.D0 0.D0 0.D0 0.D0	
1 0.D0 0.D0 2 0 0	K X(1) X(2) 2 IS(1) IS(2)
2 0.D0 6.D0 2 0 0	
3 0.D0 12.D0 2 0 0	
4 0.D0 20.D0 2 0 0	
5 0.D0 28.D0 2 0 0	
6 0.D0 36.D0 2 0 0	
7 0.D0 44.D0 2 0 0	
8 0.D0 52.D0 2 0 0	
9 0.D0 60.D0 2 0 0	
10 0.D0 66.D0 2 0 0	
11 0.D0 72.D0 2 0 0	

```
12 10.D0 0.D0  2  0  0
13 10.D0 12.D0  2  0  0
14 10.D0 28.D0  2  0  0
15 10.D0 44.D0  2  0  0
16 10.D0 60.D0  2  0  0
17 10.D0 72.D0  2  0  0
18 20.D0 0.D0  2  0  0
19 20.D0 6.D0  2  0  0
20 20.D0 12.D0  2  0  0
21 20.D0 20.D0  2  0  0
22 20.D0 28.D0  2  0  0
23 20.D0 36.D0  2  0  0
24 20.D0 44.D0  2  0  0
25 20.D0 52.D0  2  0  0
26 20.D0 60.D0  2  0  0
27 20.D0 66.D0  2  0  0
28 20.D0 72.D0  2  0  0
29 30.D0 0.D0  2  0  0
30 30.D0 12.D0  2  0  0
31 30.D0 28.D0  2  0  0
32 30.D0 44.D0  2  0  0
33 30.D0 60.D0  2  0  0
34 30.D0 72.D0  2  0  0
35 40.D0 0.D0  2  0  0
36 40.D0 6.D0  2  0  0
37 40.D0 12.D0  2  0  0
38 40.D0 20.D0  2  0  0
39 40.D0 28.D0  2  0  0
40 40.D0 36.D0  2  0  0
41 40.D0 44.D0  2  0  0
42 40.D0 52.D0  2  0  0
43 40.D0 60.D0  2  0  0
44 40.D0 66.D0  2  0  0
45 40.D0 72.D0  2  0  0
46 48.D0 0.D0  2  0  0
47 48.D0 12.D0  2  0  0
48 48.D0 28.D0  2  0  0
49 48.D0 44.D0  2  0  0
50 48.D0 60.D0  2  0  0
51 48.D0 72.D0  2  0  0
52 56.D0 0.D0  2  0  0
53 56.D0 6.D0  2  0  0
54 56.D0 12.D0  2  0  0
55 56.D0 20.D0  2  0  0
56 56.D0 28.D0  2  0  0
57 56.D0 36.D0  2  0  0
58 56.D0 44.D0  2  0  0
59 56.D0 52.D0  2  0  0
60 56.D0 60.D0  2  0  0
61 56.D0 66.D0  2  0  0
62 56.D0 72.D0  2  0  0
63 61.D0 0.D0  2  0  0
64 61.D0 12.D0  2  0  0
65 61.D0 28.D0  2  0  0
```

```
66 61.D0 44.D0 2 0 0
67 61.D0 60.D0 2 0 0
68 61.D0 72.D0 2 0 0
69 66.D0 0.D0 2 0 0
70 66.D0 6.D0 2 0 0
71 66.D0 12.D0 2 0 0
72 66.D0 20.D0 2 0 0
73 66.D0 28.D0 2 0 0
74 66.D0 36.D0 2 0 0
75 66.D0 44.D0 2 0 0
76 66.D0 52.D0 2 0 0
77 66.D0 60.D0 2 0 0
78 66.D0 66.D0 2 0 0
79 66.D0 72.D0 2 0 0
80 69.D0 0.D0 2 0 0
81 69.D0 12.D0 2 0 0
82 69.D0 28.D0 2 0 0
83 69.D0 44.D0 2 0 0
84 69.D0 60.D0 2 0 0
85 69.D0 72.D0 2 0 0
86 72.D0 0.D0 2 1 1
87 72.D0 6.D0 2 1 1
88 72.D0 12.D0 2 1 1
89 72.D0 20.D0 2 1 1
90 72.D0 28.D0 2 1 1
91 72.D0 36.D0 2 1 1
92 72.D0 44.D0 2 1 1
93 72.D0 52.D0 2 1 1
94 72.D0 60.D0 2 1 1
95 72.D0 66.D0 2 1 1
96 72.D0 72.D0 2 1 1
1 2 1 8 20 3 1 18 13 2 12 19          K IELT IELM NNE ICONN(NNE)
2 2 1 8 22 5 3 20 14 4 13 21
3 2 1 8 24 7 5 22 15 6 14 23
4 2 1 8 26 9 7 24 16 8 15 25
5 2 1 8 28 11 9 26 17 10 16 27
6 2 1 8 37 20 18 35 30 19 29 36
7 2 1 8 39 22 20 37 31 21 30 38
8 2 1 8 41 24 22 39 32 23 31 40
9 2 1 8 43 26 24 41 33 25 32 42
10 2 1 8 45 28 26 43 34 27 33 44
11 2 1 8 54 37 35 52 47 36 46 53
12 2 1 8 56 39 37 54 48 38 47 55
13 2 1 8 58 41 39 56 49 40 48 57
14 2 1 8 60 43 41 58 50 42 49 59
15 2 1 8 62 45 43 60 51 44 50 61
16 2 1 8 71 54 52 69 64 53 63 70
17 2 1 8 73 56 54 71 65 55 64 72
18 2 1 8 75 58 56 73 66 57 65 74
19 2 1 8 77 60 58 75 67 59 66 76
20 2 1 8 79 62 60 77 68 61 67 78
21 2 1 8 88 71 69 86 81 70 80 87
22 2 1 8 90 73 71 88 82 72 81 89
23 2 1 8 92 75 73 90 83 74 82 91
```

```
24  2  1  8  94  77  75  92  84  76  83  93
25  2  1  8  96  79  77  94  85  78  84  95
26  1  3  3  1  2  3
27  1  3  3  3  4  5
28  1  3  3  5  6  7
29  1  3  3  7  8  9
30  1  3  3  9  10  11
1                                          NODE
2.d0  0.d0                                 P1(1) P1(2)
2
4.d0  0.d0
3
4.d0  0.d0
4
4.d0  0.d0
5
4.d0  0.d0
6
4.d0  0.d0
7
4.d0  0.d0
8
4.d0  0.d0
9
4.d0  0.d0
10
4.d0  0.d0
11
2.d0  0.d0
-999999
6                                          NODE
0.D0  20.D0                                P1(1) P1(2)
-999999
1                                          ISOL
4000 50 1.D-2                              NS MIT TOL
1  1                                       ISYM  NRHS
1.d-3  200.D0                              EEFT FFT
2                                          IBAU
1  6  1  4.d0  .1d0  0.d0                  IFDIS NODE NNUDOF CYLOAD
                                           FLD1 FLD2
2  6  2  0.1987d0  0.d0  1.d0
2  6  2  -0.1955d0  0.d0  1.d0
2  6  2  0.4887d0  0.d0  1.d0
2  6  2  -0.48d0  0.d0  1.d0
2  6  2  0.7865d0  0.d0  1.d0
2  6  2  -0.9366d0  0.d0  1.d0
2  6  2  1.111d0  0.d0  1.d0
2  6  2  -1.1436d0  0.d0  1.d0
2  6  2  1.4425d0  0.d0  1.d0
2  6  2  -1.5056d0  0.d0  1.d0
2  6  2  0.d0  0.d0  1.d0
END
```

A.3 Post-Processing (RCPOST)

A.3.1 General

Post-processing is carried out after execution of the analysis program, RCCRAK. To execute the post-processing program, RCPOST, the original input file, unformatted files produced by RCCRAK, and a post-processing input file should exist in the working directory. The unformatted files have suffixes of '.003' and '.004' with the original input file name. The post-processing input file should have the same file name as the original input file name with a suffix of '.PSP'. The output files contains the following analysis results after the post-processing.

filename.DIS : force and deformation

filename.STR : stress and strain of concrete

filename.REN : stress and strain of discrete and smeared reinforcing steel

filename.AST : shear stress and slip of bond-slip element

A.3.2 Input Information of 'filename.PSP'

1. IMED
   - = 0 : load-deflection information
   - = 1 : stress-strain information
2. ICODE :
   - = 0 : node basis (nodal displ. and forces at all load steps)
   - = 1 : load step basis (all nodal displ. and forces at the specified load step)
3. NLN :
   - If ICODE = 0, select desired nodal no.

If ICODE = 1, select desired load step no.

4. NELP : select desired element no.

5. ICODE1 : desired no. of loading steps

   = 0 : Gauss point basis

   (stresses and strains at the specified Gauss point in all load steps)

   = 1 : load step basis

   (stresses and strains at all Gauss points in the specified load step)

6. NGAP :

   If ICODE1 = 0, select desired Gauss point no.

   If ICODE1 = 1, select desired load step no.

A.4 Program Listings

```
      PROGRAM RCCRAK
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(40000),IA(4000)
      CHARACTER*12 IN,OUT,DISFIL,STRFIL
      CHARACTER*12 STORE1,STORE2,STORE3,STORE4
      CHARACTER*1 TAB
      LOGICAL YESNO
      COMMON /CNTL/ ISYM,NUMEL,IRESOL,NRHS,NTAPEB,NTAPEU,NTAPEL,
     *              MA,IWRT,IPRINT,IERR,NNEGP,NPOSP,NRHSF,
     *              IB,IU,IL,IFB,IFU,IFL,MBUF,MW,MKF,
     *              MELEM,MFWR,MB,MDOF,MFW,MLDEST
      COMMON /INDS/ INDR(60),INDI(30)
      COMMON /DIMS/ MNCM,MNDOFN,MNNE,NDIM,NMAT1,NMAT2,NN,MNDOFE,MNDOF,
     *              NUMEL1,NUMEL2,NUMEL3,ICOMP,NGAU,IARC,IBAU,ISTL
      COMMON /CONSTS/ ZERO,ONE,TWO
      COMMON /ITRN/ JST,IST
      COMMON /CL/ ISOL,ISP
      COMMON /CNTL1/ TAB
      COMMON /CNTL2/ EEFT,FFT,TOL
      DATA NRA/40000/
      DATA NIA/4000/
      TAB=CHAR(9)
      CALL CLEAR(A,NRA)
      CALL ICLEAR(IA,NIA)
      CALL CNCLEAR
    1 WRITE(*,*) 'ENTER INPUT FILE NAME:'
      READ(*,2) IN
    2 FORMAT(A12)
      INQUIRE(FILE=IN,EXIST=YESNO)
      IF(YESNO) GO TO 3
      WRITE(*,*) 'INPUT FILE DOES NOT EXIST.'
      GO TO 1
    3 IDUM=0
      DO 25 I=1,12
      IF(IN(I:I).EQ.' '.OR.IN(I:I).EQ.'.') GOTO 27
   25 IDUM=IDUM+1
   27 STORE1=IN(1:IDUM)
      STORE2=IN(1:IDUM)
      STORE3=IN(1:IDUM)
      STORE4=IN(1:IDUM)
      STORE1(IDUM+1:IDUM+4)='.001'
      STORE2(IDUM+1:IDUM+4)='.002'
      STORE3(IDUM+1:IDUM+4)='.003'
      STORE4(IDUM+1:IDUM+4)='.004'
      WRITE(*,*) 'ENTER OUTPUT FILE NAME:'
      READ(*,2) OUT
    5 IDUM=0
      DO 7 I=1,12
      IF(OUT(I:I).EQ.' '.OR.OUT(I:I).EQ.'.') GOTO 9
    7 IDUM=IDUM+1
    9 DISFIL=OUT(1:IDUM)
      STRFIL=OUT(1:IDUM)
      DISFIL(IDUM+1:IDUM+4)='.GEN'
      STRFIL(IDUM+1:IDUM+4)='.SPE'
      INQUIRE(FILE=DISFIL,EXIST=YESNO)
      IF(YESNO) THEN
```

```
      WRITE(*,*) 'OUTPUT FILE ALREADY EXISTS.'
      WRITE(*,*) 'WARNING: UNLESS YOU SPECIFY A DIFFERENT NAME, ',
     *           'THE FILE WILL BE OVERWRITTEN.'
      WRITE(*,*) 'ENTER OUTPUT FILE NAME:'
      READ(*,2) OUT
      GOTO 5
      ENDIF
      OPEN (UNIT=5,FILE=IN,STATUS='OLD')
      OPEN (UNIT=50,FILE=DISFIL,STATUS='UNKNOWN')
      OPEN (UNIT=51,FILE=STRFIL,STATUS='UNKNOWN')
      OPEN (UNIT=40,FILE=STORE1,STATUS='UNKNOWN',FORM='UNFORMATTED')
      OPEN (UNIT=41,FILE=STORE2,STATUS='UNKNOWN',FORM='UNFORMATTED')
      OPEN (UNIT=42,FILE=STORE3,STATUS='UNKNOWN',FORM='UNFORMATTED')
      OPEN (UNIT=43,FILE=STORE4,STATUS='UNKNOWN',FORM='UNFORMATTED')
      OPEN (UNIT=20,STATUS='SCRATCH',FORM='UNFORMATTED')
      OPEN (UNIT=21,STATUS='SCRATCH',FORM='UNFORMATTED')
      OPEN (UNIT=22,STATUS='SCRATCH',FORM='UNFORMATTED')
      OPEN (UNIT=23,STATUS='SCRATCH',FORM='UNFORMATTED')
      READ(5,*) NYEXIST,ICOMP
      WRITE(50,11) NYEXIST,ICOMP
   11 FORMAT(1X,'FILE EXIST MODE:',1X,I1,/,
     *       1X,'INCOMPATIBLE ELEMENT MODE:',1X,I1,/)
      READ(5,*) NDIM
      WRITE(50,10) NDIM
   10 FORMAT(1X,'NUMBER OF DIMENSIONS:',1X,I1,/)
      READ(5,*) NN,NUMEL1,NUMEL2,NUMEL3,
     *          NMAT1,NMAT2,MNDOFN,MNNE,MNCM,NGAU
      READ(5,*) NCYCLE
      WRITE(50,20) NN,NUMEL1,NUMEL2,NUMEL3,
     *             NMAT1,NMAT2,MNDOFN,MNNE,MNCM,NGAU,
     *             NCYCLE
   20 FORMAT(1X,'NUMBER OF NODES:',1X,I4,/,
     *       1X,'NUMBER OF TRUSS ELEMENTS:',1X,I3,/,
     *       1X,'NUMBER OF PLANE STRESS ELEMENTS:',1X,I3,/,
     *       1X,'NUMBER OF BOND-SLIP ELEMENTS:',1X,I3,/,
     *       1X,'NUMBER OF MATERIALS OF TRUSS ELEMENTS:',1X,I3,/,
     *       1X,'NUMBER OF MATERIALS OF PLANE STRESS ELEMENTS:',1X,I3,/,
     *       1X,'MAX. NUMBER OF DEGREES OF FREEDOM PER NODE:',1X,I3,/,
     *       1X,'MAX. NUMBER OF NODES PER ELEMENT:',1X,I3,/,
     *       1X,'MAX. NUMBER OF CONSTANTS PER MATERIAL:',1X,I3,/,
     *       1X,'NUMBER OF GAUSSIAN POINTS:',1X,I3,/,
     *       1X,'NUMBER OF CYCLE OF LOAD:',1X,I3,/)
      NUMEL=NUMEL1+NUMEL2+NUMEL3
      MNDOFE=MNNE*MNDOFN
      MNDOF=NN*MNDOFN
C
C.....REAL STORAGE ALLOCATION
C
      INDR(1)=1
C
C.....X  COORDINATE
C
      INDR(2)=INDR(1)+NN*NDIM
C
C.....ARRAY TO STORE THE SOLUTION (DISPLACEMENT/ROTATION VECTOR)
C
      INDR(3)=INDR(2)+NN*MNDOFN
C
C.....SM(ESM)  ELEMENT STIFFNESS
```

```
C
      INDR(4)=INDR(3)+(MNNE*MNDOFN)**2
C
C.....ELRHS  ELEMENT RIGHT HAND SIDE
C
      INDR(5)=INDR(4)+MNNE*MNDOFN
C
C.....CONSTM1  CONSTANTS FOR LINE ELEMENT (REINFORCING STEEL)
C
      INDR(6)=INDR(5)+NMAT1*MNCM
C
C.....WORKING COPY OF P
C
      INDR(7)=INDR(6)+NN*MNDOFN
C
C.....Y(EX)  NODAL COORDINATE IN A ELEMENT
C
      INDR(8)=INDR(7)+MNNE*NDIM
C
C.....V(EU)  NODAL DISPLACEMENT IN A ELEMENT
C
      INDR(9)=INDR(8)+MNNE*MNDOFN
C
C.....PERMANENT COPY OF P1(DP)
C
      INDR(10)=INDR(9)+NN*MNDOFN
C
C.....R  INTERNAL NODAL FORCE
C
      INDR(11)=INDR(10)+MNDOF
C
C.....DR  UNBALANCE FORCE
C
      INDR(12)=INDR(11)+MNDOF
C
C.....U  NODAL DISPLACEMENT
C
      INDR(13)=INDR(12)+MNDOF
C
C.....CONSTM2  CONSTANTS FOR CONCRETE WITH SMEARED REINFORCEMENT
C
      INDR(14)=INDR(13)+NMAT2*MNCM
C
C.....DU1  INCREMENTAL DISPLACEMENT 1
C
      INDR(15)=INDR(14)+MNDOF
C
C.....DU2  INCREMENTAL DISPLACEMENT 2
C
      INDR(16)=INDR(15)+MNDOF
C
C.....EEU  NODAL INCREMENTAL DISPLACEMENT IN A ELEMENT
C
      INDR(17)=INDR(16)+MNDOFE
C
C.....EEP  NODAL INCREMENTAL FORCE IN A ELEMENT
C
      INDR(18)=INDR(17)+MNDOFE
C
```

```
C.....P  TOTAL FORCE
C
      INDR(19)=INDR(18)+MNDOF
C
C.....DDU  TOTAL INCREMENTAL DISPLACEMENT
C
      INDR(20)=INDR(19)+MNDOF
C
C.....DDP  TOTAL INCREMENTAL FORCE
C
      INDR(21)=INDR(20)+MNDOF
C
C.....ST1   PREVIOUS TOTAL STRESS
C
      INDR(22)=INDR(21)+3*NGAU*NGAU*NUMEL2
C
C.....PST   PREVIOUS TOTAL STRESS BEFORE ITERATION
C
      INDR(23)=INDR(22)
C
C.....AGP    PREVIOUS  PRINCIPAL STRESS  AXIS
C
      INDR(24)=INDR(23)+NGAU*NGAU*NUMEL2
C
C.....PMAX   PREVIOUS  MAX OR MIN  STRAIN
C
      INDR(25)=INDR(24)+28*NGAU*NGAU*NUMEL2
      INDR(26)=INDR(25)
C
C.....EMAX    PREVIOUS  MAX OR MIN STRAIN OF SMEARED STEEL
C
      INDR(27)=INDR(26)+2*6*NGAU*NGAU*NUMEL2
      INDR(28)=INDR(27)
C
C.....RST1 PREVIOUS TOTAL STRESS OF SMEARED STEEL 1
C
      INDR(29)=INDR(28)+NGAU*NGAU*NUMEL2
C
C.....RDST1  TOTAL STRESS INCREMENT OF SMEARED STEEL 1
C
      INDR(30)=INDR(29)
C
C.....RST2 PREVIOUS TOTAL STRESS OF SMEARED STEEL 2
C
      INDR(31)=INDR(30)+NGAU*NGAU*NUMEL2
C
C.....RDST2  TOTAL STRESS INCREMENT OF SMEARED STEEL 2
C
      INDR(32)=INDR(31)
C
C.....EMAX1  PREVIOUS  MAX  STRAIN OF SEPERATED STEEL
C
      INDR(33)=INDR(32)+6*NGAU*NUMEL1
C
C.....EMAX3  PREVIOUS  MAX  STRAIN OF BOND-SLIP
C
      INDR(34)=INDR(33)+11*NGAU*NUMEL3
C
C.....RST PREVIOUS TOTAL STRESS OF SEPERATED STEEL
```

```
C
      INDR(35)=INDR(34)+NGAU*NUMEL1
C
C.....BRST PREVIOUS TOTAL STRESS OF BOND
C
      INDR(36)=INDR(35)+NGAU*NUMEL3
      INDR(37)=INDR(36)
C
C.....CYLOAD  TARGET DISPLACEMENT OR FORCE EACH LOAD CYCLE
C
      INDR(38)=INDR(37)+NCYCLE
C
C.....PERMANENT COPY OF P2(DP)
C
      INDR(39)=INDR(38)+NN*MNDOFN
C
C.....FLD1  LOAD FACTOR FOR LOAD CASE 1
C
      INDR(40)=INDR(39)+NCYCLE
C
C.....FLD2  LOAD FACTOR FOR LOAD CASE 2
C
      INDR(41)=INDR(40)+NCYCLE
      INDR(42)=INDR(41)
      INDR(43)=INDR(42)
      INDR(44)=INDR(43)
      INDR(45)=INDR(44)
      INDR(46)=INDR(45)
      INDR(47)=INDR(46)
      INDR(48)=INDR(47)
      INDR(49)=INDR(48)
      INDR(50)=INDR(49)
      INDR(51)=INDR(50)
      INDR(52)=INDR(51)
      INDR(53)=INDR(52)
      INDR(54)=INDR(53)
      INDR(55)=INDR(54)
      INDR(56)=INDR(55)
      INDR(57)=INDR(56)
      INDR(58)=INDR(57)
      INDR(59)=INDR(58)
C
C.....ARRAY FOR SUBROUTINE SOLVE
C
      INDR(60)=INDR(59)
      MAXRA=INDR(60)-1
C
C.....INTEGER STORAGE ALLOCATION
C
      INDI(1)=1
C
C.....NDOFN  NO. OF D.O.F. PER NODE
C
      INDI(2)=INDI(1)+NN
C
C.....IS  SUPPORT CONDITIONS
C
      INDI(3)=INDI(2)+NN*MNDOFN
C
```

```
C.....ICONN  CONNECTIVITY
C
      INDI(4)=INDI(3)+NUMEL*MNNE
C
C.....IELT  ELEMENT TYPES
C
      INDI(5)=INDI(4)+NUMEL
C
C.....NNE  NO. OF NODE PER ELEMENT
C
      INDI(6)=INDI(5)+NUMEL
C
C.....INTEGER ARRAY USED IN SUBROUTINE PREFNT
C
      INDI(7)=INDI(6)+NUMEL
C
C.....ANOTHER INTEGER ARRAY USED IN SUBROUTINE PREFNT
C
      INDI(8)=INDI(7)+2*(NUMEL*MNNE+MNNE)
C
C.....IDEST
C
      INDI(9)=INDI(8)+NUMEL*MNNE
C
C.....NDOFE  NO. OF D.O.F. PER ELEMENT
C
      INDI(10)=INDI(9)+NUMEL
C
C.....IELM  MATERIAL NO. FOR EACH ELEMENT
C
      INDI(11)=INDI(10)+NUMEL
C
C.....IEL1  NO. OF LINE ELEMENT
C
      INDI(12)=INDI(11)+NUMEL
C
C.....IEL2  NO. OF RECTANGULAR ELEMENT
C
      INDI(13)=INDI(12)+NUMEL
      INDI(14)=INDI(13)
C
C.....IEL3  NO. OF BOND-SLIP ELEMENT
C
      INDI(15)=INDI(14)+NUMEL
C
C.....NUDOF  SPECIFIED DEGREE OF FREEDOM
C
      INDI(16)=INDI(15)+NCYCLE
C
C.....IFDIS  INDICATION OF FORCE OR DISPL. CONTROL EACH LOAD CYCLE
C
      INDI(17)=INDI(16)+NCYCLE
      INDI(18)=INDI(17)
      INDI(19)=INDI(18)
      INDI(20)=INDI(19)
      INDI(21)=INDI(20)
      INDI(22)=INDI(21)
      INDI(23)=INDI(22)
      INDI(24)=INDI(23)
```

```
      INDI(25)=INDI(24)
      INDI(26)=INDI(25)
      INDI(27)=INDI(26)
      INDI(28)=INDI(27)
      INDI(29)=INDI(28)
      INDI(30)=INDI(29)
      MAXIA=INDI(30)-1
      IF(MAXRA.GT.NRA) THEN
      WRITE(50,30) MAXRA
   30 FORMAT(1X,'INSUFFICIENT REAL MEMORY LOCATIONS',/,
     *       1X,'REQUIRED LENGTH OF ARRAY A:',1X,I7)
      STOP
      ELSE
      WRITE(50,40) NRA-MAXRA
   40 FORMAT(1X,'NUMBER OF UNUSED REAL MEMORY WORDS:',1X,I7)
      ENDIF
      IF(MAXIA.GT.NIA) THEN
      WRITE(50,50) MAXIA
   50 FORMAT(1X,'INSUFFICIENT INTEGER MEMORY LOCATIONS',/,
     *       1X,'REQUIRED LENGTH OF INTEGER ARRAY A:',1X,I7)
      STOP
      ELSE
      WRITE(50,60) NIA-MAXIA
   60 FORMAT(1X,'NUMBER OF UNUSED INTEGER MEMORY WORDS:',1X,I7)
      ENDIF
      CALL INMAT(A(INDR(5)),A(INDR(13)),NMAT1,NMAT2,MNCM)
      CALL INNOD(A(INDR(1)),IA(INDI(1)),IA(INDI(2)),
     *           NDIM,NN,MNDOFN)
      CALL INEL(IA(INDI(4)),IA(INDI(10)),IA(INDI(5)),IA(INDI(3)),
     *          IA(INDI(1)),IA(INDI(9)),NN,NUMEL,MNNE,
     *          NUMEL1,NUMEL2,NUMEL3,IA(INDI(11)),IA(INDI(12)),
     *          IA(INDI(14)))
      CALL LOAD(A(INDR(9)),A(INDR(38)),IA(INDI(1)),
     *    NN,MNDOFN)
      CALL NONE(ISOL,NS,NIT,MIT,TOL,ISYM,NRHS,
     *          EEFT,FFT,IARC,NCYCLE,A(INDR(37)),IBAU,ISTL,
     *  MNDOFN,A(INDR(39)),A(INDR(40)),IA(INDI(15)),IA(INDI(16)))
      IF(NYEXIST.EQ.1) THEN
      READ(41) JSTP
      REWIND 41
      IF(NS.LT.JSTP) THEN
      WRITE(*,*) 'ERROR! NO OF STEP SHOULD BE INCREASED'
      STOP
      ENDIF
      ENDIF
C
C.....PREPARE FOR ASSEMBLY AND SOLUTION
C
      CALL PREP(IA(INDI(6)),IA(INDI(7)),
     *          IA(INDI(5)),IA(INDI(1)),IA(INDI(3)),
     *          NUMEL,NN,MNNE)
      IRESOL=0
      NTAPEB=20
      NTAPEU=21
      NTAPEL=22
      IPRINT=1
      JST=0
      CALL PREFNT(IA(1),IA(INDI(6)),IA(INDI(7)),MS,MU,MR)
      MAMIN=(MDOF*(MDOF+1))/2+MDOF*NRHS+
```

```
     *       (MFW*(MFW+1))/2+MFW*NRHS+
     *       NUMEL+MLDEST+2*MDOF+MFW+NRHS
      WRITE(50,70) MAMIN
   70 FORMAT(1X,'MINIMUM MEMORY (REQUIRED) BY THE SOLVER:',1X,I7)
      MA=NRA-MAXRA
      WRITE(50,80) MA
   80 FORMAT(1X,'MEMORY AVAILABLE TO THE SOLVER:',1X,I7)
      IF(MAMIN.GT.MA) THEN
      WRITE(50,90) MAXRA+MAMIN
   90 FORMAT(1X,'LENGTH OF REAL ARRAY A MUST BE AT LEAST:',1X,I7)
      STOP
      ENDIF
      WRITE(51,1000)
 1000 FORMAT(//,'*** DISPL. AND FORCE OF SPECIFIED NODE ***',/,
     *      'LOAD STEP   ','DISPLACEMENT     ','FORCE   ')
      WRITE(50,2000)
 2000 FORMAT(//,'*** INCREMENTAL DISPL. AND FORCE TOLERANCES ***',/,
     *      'LOAD STEP   ','ITER. NO.','FORCE TOL.    ','DISPL. TOL. ')
      CALL ARCDIS(A,IA,NS,NIT,MIT,TOL,NYEXIST,
     *            NCYCLE)
      END

      SUBROUTINE ARCDIS(A,IA,NS,NIT,MIT,TOL,NYEXIST,
     *          NCYCLE)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTL/ ISYM,NUMEL,IRESOL,NRHS,NTAPEB,NTAPEU,NTAPEL,
     *              MA,IWRT,IPRINT,IERR,NNEGP,NPOSP,NRHSF,
     *              IB,IU,IL,IFB,IFU,IFL,MBUF,MW,MKF,
     *              MELEM,MFWR,MB,MDOF,MFW,MLDEST
      COMMON /INDS/ INDR(60),INDI(30)
      COMMON /DIMS/ MNCM,MNDOFN,MNNE,NDIM,NMAT1,NMAT2,NN,MNDOFE,MNDOF,
     *              NUMEL1,NUMEL2,NUMEL3,ICOMP,NGAU,IARC,IBAU,ISTL
      COMMON /CONSTS/ ZERO,ONE,TWO
      COMMON /ITRN/ JST,IST
      COMMON /CL/ ISOL,ISP
      COMMON /CNTL1/ TAB
      DIMENSION A(1),IA(1)
      DIMENSION SSPU(20),SSPP(20)
C
C.....SET STEP SIZE
C
      IST=0
      JST=0
      DTOL=1.D0
      DSD=1.D0
      PDSD=1.D0
      DS=DSQRT(1.D0+1.D0)
      DLM2=1.D0
      DO 100 ICYL=1,NCYCLE
      NUDOF=IA(INDI(15)+ICYL-1)
C
C.....GET A WORKING COPY OF P
C
      CALL AEBC(A(INDR(6)),A(INDR(9)),A(INDR(38)),
     *      A(INDR(39)+ICYL-1),A(INDR(40)+ICYL-1),NN*MNDOFN)
C
C.....(ASSEMBLE AND) SOLVE
C
      CALL SOLVE(A(1),IA(1),A(INDR(59)))
```

```
C
C.....SET STEP SIZE EACH LOAD CYCLE
C
      NDUM=INDR(2)+NUDOF-1
      SSPU(ICYL)=A(NDUM)
      NDUM=INDR(6)+NUDOF-1
      SSPP(ICYL)=A(NDUM)
  100 CONTINUE
C
C.....CLEARING ARRAY
C
      IDUM=INDR(5)-INDR(2)
      CALL CLEAR(A(INDR(2)),IDUM)
      IDUM=INDR(9)-INDR(6)
      CALL CLEAR(A(INDR(6)),IDUM)
      IDUM=INDR(13)-INDR(10)
      CALL CLEAR(A(INDR(10)),IDUM)
      IDUM=INDR(23)-INDR(14)
      CALL CLEAR(A(INDR(14)),IDUM)
      IDUM=INDR(37)-INDR(24)
      CALL CLEAR(A(INDR(24)),IDUM)
      CALL CLEAR(A(INDR(41)),(INDR(60)-INDR(41)+1))
C
C.....READING EXISTING FILE
C
      PFORCE=0.D0
      ICYLP=1
      DSIG=1.D0
      SSDU=0.D0
      SSDP=0.D0
      SSDUP=0.D0
      SSDPP=0.D0
      IF(NYEXIST.EQ.1) THEN
      READ(40)  (A(II),II=INDR(18),INDR(18)+MNDOF)
      READ(40)  (A(II),II=INDR(10),INDR(10)+MNDOF)
      READ(40)  (A(II),II=INDR(11),INDR(11)+MNDOF)
      READ(40)  (A(II),II=INDR(12),INDR(12)+MNDOF)
      READ(40)  (A(II),II=INDR(21),INDR(21)+3*NGAU*NGAU*NUMEL2)
      READ(40)  (A(II),II=INDR(23),INDR(23)+NGAU*NGAU*NUMEL2)
      READ(40)  (A(II),II=INDR(24),INDR(24)+28*NGAU*NGAU*NUMEL2)
      READ(40)  (A(II),II=INDR(26),INDR(26)+2*6*NGAU*NGAU*NUMEL2)
      READ(40)  (A(II),II=INDR(28),INDR(28)+NGAU*NGAU*NUMEL2)
      READ(40)  (A(II),II=INDR(30),INDR(30)+NGAU*NGAU*NUMEL2)
      READ(40)  (A(II),II=INDR(32),INDR(32)+6*NGAU*NUMEL1)
      READ(40)  (A(II),II=INDR(33),INDR(33)+11*NGAU*NUMEL3)
      READ(40)  (A(II),II=INDR(34),INDR(34)+NGAU*NUMEL1)
      READ(40)  (A(II),II=INDR(35),INDR(35)+NGAU*NUMEL3)
      READ(41) JSTP,IST,ICYLP
      READ(41) DS,CON2,DSIG,CZ,SSDU,SSDP,PDSD
      READ(41) SSPECU,SSDUP,SSDPP,DDTOL,PFORCE
      REWIND 40
      REWIND 41
      ELSE
      JSTP=0
      ENDIF
      JST=JSTP
C
C....LOAD CYCLES
C
```

```
      DO 4600 ICYL=ICYLP,NCYCLE
      NUDOF=IA(INDI(15)+ICYL-1)
      IFDIS=IA(INDI(16)+ICYL-1)
      SSPECU=SSPU(ICYL)
      SSPECP=SSPP(ICYL)
      SCAL=1.D0
      IF(IFDIS.EQ.2) THEN
      NDUM=INDR(12)+NUDOF-1
      PFORCE=A(NDUM)
      ELSEIF(IFDIS.EQ.1) THEN
      NDUM=INDR(18)+NUDOF-1
      PFORCE=A(NDUM)
      ENDIF
      NDUM=INDR(37)+ICYL-1
      TFORCE=A(NDUM)
      WRITE(*,2405) ICYL,TFORCE
 2405 FORMAT(/,2X,'CYCLE  NO  =  ',I5,10X,D13.5)
      IF(IFDIS.EQ.1) THEN
      NDUM=INDR(18)+NUDOF-1
      IF((TFORCE-A(NDUM))*SSPECP.GE.0.D0) THEN
      PDSD=1.D0
      ELSE
      PDSD=-1.D0
      ENDIF
      ELSEIF(IFDIS.EQ.2) THEN
      NDUM=INDR(12)+NUDOF-1
      IF((TFORCE-A(NDUM))*SSPECU.GE.0.D0) THEN
      PDSD=1.D0
      ELSE
      PDSD=-1.D0
      ENDIF
      ENDIF

 1000 JST=JST+1
      DSD=PDSD
      CZ=1.D0
      WRITE(23) (A(II),II=INDR(21),INDR(21)+3*NGAU*NGAU*NUMEL2)
      WRITE(23) (A(II),II=INDR(28),INDR(28)+NGAU*NGAU*NUMEL2)
      WRITE(23) (A(II),II=INDR(30),INDR(30)+NGAU*NGAU*NUMEL2)
      WRITE(23) (A(II),II=INDR(23),INDR(23)+NGAU*NGAU*NUMEL2)
      WRITE(23) (A(II),II=INDR(34),INDR(34)+NGAU*NUMEL1)
      WRITE(23) (A(II),II=INDR(35),INDR(35)+NGAU*NUMEL3)
      WRITE(23) (A(II),II=INDR(11),INDR(11)+MNDOF)
      REWIND 23
      EMIT=MIT
      ENIT=NIT
      IF(JST.GT.1) THEN
      EIST=IST
      SPECU=SSPECU*DSD
      ENDIF
      CALL CLEAR(A(INDR(19)),MNDOF)
      IST=0
      IRESOL=2
      CALL SOLVE(A(1),IA(1),A(INDR(59)))
      IST=0
      ISP=0
      IRESOL=1
      PDTOL=1.D0
      ISR=1
```

```
      ISTMIN=0
      DTOLMIN=1.D0
      IF(IFDIS.EQ.1) DLM2=DSD
  450 IF(IST.NE.0) THEN
      READ(23) (A(II),II=INDR(21),INDR(21)+3*NGAU*NGAU*NUMEL2)
      READ(23) (A(II),II=INDR(28),INDR(28)+NGAU*NGAU*NUMEL2)
      READ(23) (A(II),II=INDR(30),INDR(30)+NGAU*NGAU*NUMEL2)
      READ(23) (A(II),II=INDR(23),INDR(23)+NGAU*NGAU*NUMEL2)
      READ(23) (A(II),II=INDR(34),INDR(34)+NGAU*NUMEL1)
      READ(23) (A(II),II=INDR(35),INDR(35)+NGAU*NUMEL3)
      READ(23) (A(II),II=INDR(11),INDR(11)+MNDOF)
      REWIND 23
      ENDIF
      IF(ISP.EQ.1.AND.ISTMIN.EQ.0) ISP=2
      IF(ISP.EQ.2) THEN
      IF(IFDIS.EQ.2) DSD=PDSD
      CZ=1.D0
      IF(IFDIS.EQ.1) DLM2=DSD
      ENDIF
      ISQ=0
      IST=0
      DS=DS*CZ
      IF(IFDIS.EQ.2) THEN
      DSD=DSD*CZ
      SPECU=SSPECU*DSD*SCAL
      ELSE
      DLM2=DLM2*CZ*SCAL
      ENDIF
      DTOL=1.D0
      DLM1=0.D0
      CZ=1.D0
      CALL CLEAR(A(INDR(3)),(MNNE*MNDOFN)**2)
      CALL CLEAR(A(INDR(4)),MNNE*MNDOFN)
      CALL CLEAR(A(INDR(7)),MNNE*NDIM)
      CALL CLEAR(A(INDR(8)),MNNE*MNDOFN)
      CALL CLEAR(A(INDR(10)),MNDOF)
      CALL CLEAR(A(INDR(14)),MNDOF)
      CALL CLEAR(A(INDR(15)),MNDOF)
      CALL CLEAR(A(INDR(16)),MNDOFE)
      CALL CLEAR(A(INDR(17)),MNDOFE)
      CALL CLEAR(A(INDR(19)),MNDOF)
      CALL CLEAR(A(INDR(20)),MNDOF)
C
C.....ITERATION
C
  500 IST=IST+1
      IF(ISP.GT.12) THEN
      WRITE(50,2700)
 2700 FORMAT(/,5X,'CONVERGENCE IS NOT ACCOMPLISHED',/)
      STOP
      ENDIF
      PDLM2=DLM2
      DDTOL=DTOL
      IF(IST.EQ.1) THEN
      NDUM=1
C
C.....GET A WORKING COPY OF DP
C
      CALL AEBC(A(INDR(6)),A(INDR(9)),A(INDR(38)),
```

```
     *       A(INDR(39)+ICYL-1),A(INDR(40)+ICYL-1),NN*MNDOFN)
C
C.....SOLVE (ONLY BACKSUBSTITUTION)
C
      CALL SOLVE(A(1),IA(1),A(INDR(59)))
C
C.....GET A COPY OF DU1
C
      CALL AEB(A(INDR(14)),A(INDR(2)),NN*MNDOFN)
      ENDIF

      IF(IST.GE.2.AND.ISP.LE.1.AND.IFDIS.EQ.2) THEN
      IRESOL=0
C
C.....GET A WORKING COPY OF DP
C
      CALL AEBC(A(INDR(6)),A(INDR(9)),A(INDR(38)),
     *       A(INDR(39)+ICYL-1),A(INDR(40)+ICYL-1),NN*MNDOFN)
C
C.....SOLVE (ONLY BACKSUBSTITUTION)
C
      CALL SOLVE(A(1),IA(1),A(INDR(59)))
C
C.....GET A COPY OF DU1
C
      CALL AEB(A(INDR(14)),A(INDR(2)),NN*MNDOFN)
      ENDIF
      IRESOL=1
      IF(IFDIS.EQ.1.AND.ISP.LE.1) THEN
      IRESOL=0
      ELSE
      IRESOL=1
      ENDIF
C
C.....GET A WORKING COPY OF DR
C
      CALL AEB(A(INDR(6)),A(INDR(11)),NN*MNDOFN)
C
C.....SOLVE (ONLY BACKSUBSTITUTION)
C
      CALL SOLVE(A(1),IA(1),A(INDR(59)))
C
C.....GET A COPY OF DU2
C
      CALL AEB(A(INDR(15)),A(INDR(2)),NN*MNDOFN)
      IF(IFDIS.EQ.2) THEN
      CALL DISPSUB(A(INDR(14)),A(INDR(15)),SPECU,
     *     DST,MNDOF,DS,DSD,SSPECU,DLM1,DLM2,NUDOF)
      ENDIF
C
C.....COMPARING P WITH R
C
      IF(IFDIS.EQ.1) THEN
      CALL COMPFC(A(INDR(9)),A(INDR(38)),
     *       A(INDR(39)+ICYL-1),A(INDR(40)+ICYL-1),
     *       A(INDR(14)),A(INDR(15)),MNDOF,DLM1,DLM2,
     *       A(INDR(19)),A(INDR(20)),NUDOF,
     *       SPECU,SPECP,A(INDR(11)),DDTOL,IA(INDI(2)),
     *       DTOL1,DTOL2,A(INDR(18)))
```

```
      ELSE
      CALL COMP(A(INDR(9)),A(INDR(38)),
     *       A(INDR(39)+ICYL-1),A(INDR(40)+ICYL-1),
     *       A(INDR(14)),A(INDR(15)),MNDOF,DLM1,DLM2,
     *       A(INDR(19)),A(INDR(20)),NUDOF,
     *       SPECU,SPECP,A(INDR(11)),DDTOL,IA(INDI(2)),
     *       DTOL1,DTOL2,A(INDR(18)))
      ENDIF
      DTOL=DTOL2
C
C.....CHECK TOLERANCE
C
      IF(ISP.EQ.0) THEN
      IF(DTOL.LT.DTOLMIN.AND.DTOL.LE.0.1D0) THEN
      ISTMIN=IST
      DTOLMIN=DTOL
      ENDIF

      IF(ISP.LE.1.AND.IST.GE.5.AND.DDTOL.LT.DTOL.AND.
     *   DTOL.GT.10.D0*TOL) THEN
      ISQ=ISQ+1
      IF(ISQ.EQ.3) THEN
      WRITE(*,2100) IST,DTOL
      ISP=ISP+1
      IRESOL=0
      GOTO 450
      ENDIF
      ENDIF

      IF(IST.GE.5.AND.DTOL2.GT.0.9D0) THEN
      WRITE(*,2100) IST,DTOL
      ISP=ISP+1
      IF(IFDIS.EQ.1) CZ=0.001D0
      IF(IFDIS.EQ.1.AND.ISP.EQ.2) ISP=3
      IRESOL=0
      GOTO 450
      ENDIF

      IF(ISP.LE.1.AND.IST.GE.MIT.AND.DTOL.GT.TOL) THEN
      WRITE(*,2100) IST,DTOL
      ISP=ISP+1
      IF(IFDIS.EQ.1) CZ=0.001D0
      IF(IFDIS.EQ.1.AND.ISP.EQ.2) ISP=3
      IRESOL=0
      GOTO 450
      ENDIF

      ELSEIF(ISP.EQ.2) THEN
      IF(IST.GE.5.AND.DTOL2.GT.0.9D0) THEN
      WRITE(*,2100) IST,DTOL
      ISP=ISP+1
      IF(IFDIS.EQ.1) CZ=0.001D0
      IF(IFDIS.EQ.1.AND.ISP.EQ.2) ISP=3
      IRESOL=0
      GOTO 450
      ENDIF
      ENDIF
 2100 FORMAT(/,5X,'ITERATION  NO  =  ',I5,5X,'TOLERANCE = ',D13.6)
C
```

```
C.....UPDATE THE INCREMENTS OF LOAD & DISPLACEMENT
C
      CALL UPDT(A(INDR(1)),A(INDR(3)),A(INDR(5)),A(INDR(7)),
     *        IA(INDI(3)),IA(INDI(10)),IA(INDI(5)),A(INDR(21)),
     *        NDIM,NN,NUMEL,NMAT1,NMAT2,MNDOFN,MNNE,MNCM,MNDOFE,MNDOF,
     *        NGAU,A(INDR(12)),A(INDR(8)),
     *        A(INDR(18)),A(INDR(10)),A(INDR(11)),A(INDR(17)),
     *        IA(INDI(2)),A(INDR(19)),A(INDR(20)),A(INDR(16)),
     *        A(INDR(14)),A(INDR(15)),DLM2,
     *        IA(INDI(1)),IA(INDI(4)),IA(INDI(9)),
     *        A(INDR(4)),A(INDR(23)),A(INDR(26)),A(INDR(28)),
     *        A(INDR(30)),A(INDR(24)),NUMEL1,NUMEL2,NUMEL3,
     *        A(INDR(32)),A(INDR(34)),A(INDR(35)),
     *        IA(INDI(11)),IA(INDI(12)),ICOMP,A(INDR(13)),
     *        A(INDR(33)),IA(INDI(14)),IARC,IBAU,ISTL)
C
C.....JUMP CURRENT STEP
C
      IF(ISP.EQ.1.AND.IST.EQ.ISTMIN) GOTO 600
      IF(ISP.LE.1.AND.IST.GE.5.AND.DDTOL.LT.DTOL.AND.
     *   DTOL.LT.10.D0*TOL) GOTO 600
      IF(ISP.GE.2.AND.DDTOL.LT.DTOL.AND.DTOL.LT.TOL) THEN
      GOTO 600
      ENDIF
C
C.....CHECK CONVERGENCE
C
      IF(ISP.GE.2) THEN
      IF(IST.EQ.MIT) THEN
      GOTO 600
      ELSE
      GOTO 500
      ENDIF
      ENDIF
      IF(DTOL.GT.TOL) GOTO 500
C
C.....CHECK TARGET DISP. OR FORCE EACH LOAD CYLCE
C
  600 IF(IFDIS.EQ.2) THEN
      NDUM=INDR(12)+NUDOF-1
      NNDUM=INDR(19)+NUDOF-1
      SHST=A(NDUM)+A(NNDUM)
      IF(SCAL.EQ.1.D0) THEN
      IF(TFORCE.GT.PFORCE.AND.TFORCE.LT.SHST) THEN
      SCAL=DABS(TFORCE-PFORCE)/DABS(SHST-PFORCE)
      CZ=1.D0
      IRESOL=0
      GOTO 450
      ELSEIF(TFORCE.LT.PFORCE.AND.TFORCE.GT.SHST) THEN
      SCAL=DABS(TFORCE-PFORCE)/DABS(SHST-PFORCE)
      CZ=1.D0
      IRESOL=0
      GOTO 450
      ENDIF
      ENDIF
      ELSEIF(IFDIS.EQ.1) THEN
      NDUM=INDR(18)+NUDOF-1
      NNDUM=INDR(20)+NUDOF-1
      SHST=A(NDUM)+A(NNDUM)
```

```
      IF(SCAL.EQ.1.D0) THEN
      IF(TFORCE.GT.PFORCE.AND.TFORCE.LT.SHST) THEN
      SCAL=DABS(TFORCE-PFORCE)/DABS(SHST-PFORCE)
      CZ=1.D0
      IRESOL=0
      GOTO 450
      ELSEIF(TFORCE.LT.PFORCE.AND.TFORCE.GT.SHST) THEN
      SCAL=DABS(TFORCE-PFORCE)/DABS(SHST-PFORCE)
      CZ=1.D0
      IRESOL=0
      GOTO 450
      ENDIF
      ENDIF
      ENDIF
      PFORCE=SHST
      ISP=0
      WRITE(*,2400) JST,IST,DSD
 2400 FORMAT(/,5X,'STEP  NO  =  ',I5,10X,'COUNT = ',I5,5X,D13.5)
C
C.....COMPUTE MEMBER LOADS AND DISPL.
C

      CALL MODIFX(A(INDR(18)),A(INDR(20)),A(INDR(18)),NN,NDIM)
      CALL MODIFX(A(INDR(12)),A(INDR(19)),A(INDR(12)),NN,NDIM)
C
C.....PRINT RESULTS (DISPLACEMENTS/ROTATIONS)
C
      CALL PRNT(A(INDR(12)),A(INDR(18)),NN,MNDOFN,DTOL,NUDOF,
     *    A(INDR(11)))
C
C.....COMPUTE AND PRINT STRESSES AND STRAINS
C
      CALL STRESS(A(INDR(1)),A(INDR(5)),A(INDR(7)),
     *       IA(INDI(3)),IA(INDI(10)),IA(INDI(5)),A(INDR(21)),
     *       NDIM,NN,NUMEL,NMAT1,NMAT2,MNDOFN,MNNE,MNCM,MNDOFE,
     *       MNDOF,NGAU,A(INDR(12)),A(INDR(8)),A(INDR(17)),
     *       IA(INDI(1)),IA(INDI(2)),IA(INDI(4)),IA(INDI(9)),
     *       A(INDR(23)),A(INDR(26)),A(INDR(28)),
     *       A(INDR(30)),A(INDR(24)),NUMEL1,NUMEL2,NUMEL3,
     *       A(INDR(32)),A(INDR(34)),A(INDR(35)),IA(INDI(11)),
     *       IA(INDI(12)),ICOMP,A(INDR(13)),SHDUM,
     *       A(INDR(33)),IA(INDI(14)),IARC,IBAU,ISTL)
      EEIST=IST
      IF(EEIST.LE.EIST.AND.EIST.LE.ENIT) THEN
      CON2=2.D0
      ELSE
      CON2=1.D0
      ENDIF
      WRITE(40)  (A(II),II=INDR(18),INDR(18)+MNDOF)
      WRITE(40)  (A(II),II=INDR(10),INDR(10)+MNDOF)
      WRITE(40)  (A(II),II=INDR(11),INDR(11)+MNDOF)
      WRITE(40)  (A(II),II=INDR(12),INDR(12)+MNDOF)
      WRITE(40)  (A(II),II=INDR(21),INDR(21)+3*NGAU*NGAU*NUMEL2)
      WRITE(40)  (A(II),II=INDR(23),INDR(23)+NGAU*NGAU*NUMEL2)
      WRITE(40)  (A(II),II=INDR(24),INDR(24)+28*NGAU*NGAU*NUMEL2)
      WRITE(40)  (A(II),II=INDR(26),INDR(26)+2*6*NGAU*NGAU*NUMEL2)
      WRITE(40)  (A(II),II=INDR(28),INDR(28)+NGAU*NGAU*NUMEL2)
      WRITE(40)  (A(II),II=INDR(30),INDR(30)+NGAU*NGAU*NUMEL2)
      WRITE(40)  (A(II),II=INDR(32),INDR(32)+6*NGAU*NUMEL1)
```

```
      WRITE(40)  (A(II),II=INDR(33),INDR(33)+11*NGAU*NUMEL3)
      WRITE(40)  (A(II),II=INDR(34),INDR(34)+NGAU*NUMEL1)
      WRITE(40)  (A(II),II=INDR(35),INDR(35)+NGAU*NUMEL3)
      WRITE(41) JST,IST,ICYL
      WRITE(41) DS,CON2,DSIG,CZ,SSDU,SSDP,PDSD
      WRITE(41) SSPECU,SSDUP,SSDPP,DDTOL,PFORCE
      REWIND 40
      REWIND 41
      IF(JST.EQ.NS) STOP
      IF(SCAL.EQ.1.D0) GOTO 1000
 4600 CONTINUE
      RETURN
      END

      SUBROUTINE FCLEAR(NFILE,NT)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CONSTS/ ZERO,ONE,TWO
      DO 100 I=1,NT
      WRITE(NFILE) ZERO
  100 CONTINUE
      REWIND NFILE
      END

      SUBROUTINE DISPSUB(DU1,DU2,SPECU,
     *     DST,MNDOF,DS,DSD,SSPECU,DLM1,DLM2,NUDOF)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ITRN/ JST,IST
      DIMENSION DU1(MNDOF),DU2(MNDOF)
      IF(IST.EQ.1) THEN
      DLM2=(SPECU-DU2(NUDOF))/DU1(NUDOF)
      DS=DSQRT(SPECU*SPECU/SSPECU/SSPECU+DLM2*DLM2)
      ELSE
      DLM2=-DU2(NUDOF)/DU1(NUDOF)
      ENDIF
      RETURN
      END

      SUBROUTINE AEBC(A,B,C,FB,FC,N)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(N),B(N),C(N)
      DO 10 I=1,N
      A(I)=B(I)*FB+C(I)*FC
   10 CONTINUE
      RETURN
      END

      SUBROUTINE AEB(A,B,N)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(N),B(N)
      DO 10 I=1,N
      A(I)=B(I)
   10 CONTINUE
      RETURN
      END

      BLOCK DATA
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CONSTS/ ZERO,ONE,TWO
      COMMON /XGWGT/ XG(4,4),WGT(4,4)
```

```
      DATA ZERO,ONE,TWO/0.D0,1.D0,2.D0/
C
C     MATRIX XG STORES GAUSS - LEGENDRE SAMPLING POINTS
C
      DATA XG/  0.D0,   0.D0,   0.D0,   0.D0,   -.5773502691896D0,
     1 .5773502691896D0,   0.D0,   0.D0,   -.7745966692415D0,   0.D0,
     2 .7745966692415D0,   0.D0,   -.8611363115941D0,
     3 -.3399810435849D0,   .3399810435849D0,   .8611363115941D0 /
C
C     MATRIX WGT STORES GAUSS - LEGENDRE WEIGHTING FACTORS
C
      DATA WGT /  2.D0,   0.D0,   0.D0,   0.D0,   1.D0,   1.D0,
     1 0.D0,   0.D0,   .5555555555556D0,   .8888888888889D0,
     2 .5555555555556D0,   0.D0,   .3478548451375D0,   .6521451548625D0,
     3 .6521454548625D0,   .3478548451375D0 /
      END

      SUBROUTINE CLEAR(A,NA)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(NA)
      COMMON /CONSTS/ ZERO,ONE,TWO
      DO 10 I=1,NA
      A(I)=ZERO
   10 CONTINUE
      RETURN
      END

      SUBROUTINE ICLEAR(IA,NA)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION IA(NA)
      COMMON /CONSTS/ ZERO,ONE,TWO
      DO 10 I=1,NA
      IA(I)=0
   10 CONTINUE
      RETURN
      END

      SUBROUTINE MODIFX(X,U,XI,NN,NDIM)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(NDIM,NN),U(NDIM,NN),XI(NDIM,NN)
      DO 10 I=1,NN
      DO 10 J=1,NDIM
   10 XI(J,I)=X(J,I)+U(J,I)
      RETURN
      END

      SUBROUTINE INEL(IELT,IELM,NNE,ICONN,NDOFN,NDOFE,NN,NUMEL,MNNE,
     *                NUMEL1,NUMEL2,NUMEL3,IEL1,IEL2,IEL3)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION IELT(NUMEL),IELM(NUMEL),NNE(NUMEL),ICONN(MNNE,NUMEL)
      DIMENSION NDOFN(NN),NDOFE(NUMEL),IEL1(NUMEL1),IEL2(NUMEL2)
      DIMENSION IEL3(NUMEL3)
      K1=0
      K2=0
      K3=0
      DO 20 IEL=1,NUMEL
      READ(5,*) K,IELT(K),IELM(K),NNE(K),(ICONN(J,K),J=1,NNE(K))
      IF(IELT(K).EQ.1) THEN
      K1=K1+1
```

```
      IEL1(K)=K1
      ENDIF
      IF(IELT(K).EQ.2) THEN
      K2=K2+1
      IEL2(K)=K2
      ENDIF
      IF(IELT(K).EQ.3) THEN
      K3=K3+1
      IEL3(K)=K3
      ENDIF
      NDOFE(K)=0
      DO 20 J=1,NNE(K)
      NDOFE(K)=NDOFE(K)+NDOFN(ICONN(J,K))
   20 CONTINUE
      RETURN
      END

      SUBROUTINE INMAT(CONSTM1,CONSTM2,NMAT1,NMAT2,MNCM)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION CONSTM1(MNCM,NMAT1),CONSTM2(MNCM,NMAT2)
      DO 10 IMAT=1,NMAT1
      READ(5,*) NCM,(CONSTM1(ICM,IMAT),ICM=1,NCM)
   10 CONTINUE
      DO 20 IMAT=1,NMAT2
      READ(5,*) NCM,(CONSTM2(ICM,IMAT),ICM=1,NCM)
   20 CONTINUE
      RETURN
      END

      SUBROUTINE INNOD(X,NDOFN,IS,NDIM,NN,MNDOFN)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(NDIM,NN)
      DIMENSION NDOFN(NN),IS(MNDOFN,NN)
      DO 10 I=1,NN
      READ(5,*) K,(X(J,K),J=1,NDIM),
     *           NDOFN(K),(IS(J,K),J=1,NDOFN(K))
   10 CONTINUE
      RETURN
      END

      SUBROUTINE LOAD(P1,P2,NDOFN,NN,MNDOFN)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION NDOFN(NN)
      DIMENSION P1(1),P2(1)
      CALL CLEAR(P1,MNDOFN*NN)
      CALL CLEAR(P2,MNDOFN*NN)
   10 READ(5,*) NODE
      IF(NODE.NE.-999999) THEN
      I1=(NODE-1)*MNDOFN+1
      I2=I1+NDOFN(NODE)-1
      READ(5,*) (P1(I),I=I1,I2)
      GO TO 10
      ENDIF
  100 READ(5,*) NODE
      IF(NODE.NE.-999999) THEN
      I1=(NODE-1)*MNDOFN+1
      I2=I1+NDOFN(NODE)-1
      READ(5,*) (P2(I),I=I1,I2)
      GO TO 100
```

```
      ENDIF
      RETURN
      END

      SUBROUTINE NONE(ISOL,NS,NIT,MIT,TOL,ISYM,
     *       NRHS,EEFT,FFT,IARC,NCYCLE,
     *       CYLOAD,IBAU,ISTL,MNDOFN,FLD1,FLD2,NUDOF,IFDIS)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION CYLOAD(NCYCLE),FLD1(NCYCLE),FLD2(NCYCLE)
      DIMENSION NUDOF(NCYCLE),IFDIS(NCYCLE)
C
C      NS: NO OF STEP
C      MIT: MAX NO OF ITERATION
C      TOL: TOLERANCE
C      ISYM=1 : SYMMETRIC SOLVER
C           3 : UNSYMMETRIC SOLVER
C      NRHS=1 (DEFAULT): NO. OF RIGHT HAND SIDE
C      IARC=1:PRINCIPAL STRESS AND STRAIN AXES COINCIDE
C           2: NOT
C      IBAU=1:BILINEAR
C           2: BAUSCHINGER  EFFECT
C      ISTL=1:REAL STEEL LAYER
C           2:EQUIVALENT  STEEL LAYER
C      IFDIS=1:FORCE  CONTROL
C            2:DISPLACEMENT  CONTROL
C
      READ(5,*) ISOL
      READ(5,*) NS,MIT,TOL
      READ(5,*) ISYM,NRHS
      READ(5,*) EEFT,FFT
      READ(5,*) IARC,IBAU,ISTL
      NIT=MIT
      DO 100 I=1,NCYCLE
      READ(5,*) IFDIS(I),NODE,NNUDOF,CYLOAD(I),FLD1(I),FLD2(I)
      NUDOF(I)=(NODE-1)*MNDOFN+NNUDOF
  100 CONTINUE
      RETURN
      END

      SUBROUTINE COMP(DP1,DP2,FLD1,FLD2,DU1,DU2,MNDOF,DLM1,DLM2,
     *                DDU,DDP,NUDOF,SPECU,SPECP,DR,
     *                DDTOL,IS,DTOL1,DTOL2,P)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ITRN/ JST,IST
      COMMON /CNTL1/ TAB
      DIMENSION DP1(MNDOF)
      DIMENSION DP2(MNDOF),DR(MNDOF),IS(MNDOF),P(MNDOF)
      DIMENSION DU1(MNDOF),DU2(MNDOF),DDU(MNDOF),DDP(MNDOF)
      CP1=0.D0
      CP2=0.D0
      CD1=0.D0
      CD2=0.D0
      CR1=0.D0
      CR2=0.D0
      DO 100 I=1,MNDOF
      DDU(I)=DDU(I)+DLM2*DU1(I)+DU2(I)
      DDP(I)=DDP(I)+DLM2*(DP1(I)*FLD1+DP2(I)*FLD2)
  100 CONTINUE
      DO 200 I=1,MNDOF
```

```
      IF(IS(I).EQ.0) THEN
      CP1=CP1+(DLM2*DU1(I)+DU2(I))*(DLM2*DU1(I)+DU2(I))
      CP2=CP2+DDU(I)*DDU(I)
      CD1=CD1+DR(I)*DR(I)
      CD2=CD2+DDP(I)*DDP(I)
      ENDIF
  200 CONTINUE
      DTOL2=DSQRT(CP1)/DSQRT(CP2)
      DTOL1=DSQRT(CD1)/DSQRT(CD2)
      WRITE(50,7500) JST,TAB,IST,TAB,DTOL1,TAB,DTOL2
 7500 FORMAT(I5,A1,I5,5(A1,E12.6))
      IF(IST.EQ.1) THEN
      DTOL1=1.D0
      DTOL2=1.D0
      ENDIF
      RETURN
      END

      SUBROUTINE COMPFC(DP1,DP2,FLD1,FLD2,DU1,DU2,MNDOF,DLM1,DLM2,
     *                  DDU,DDP,NUDOF,SPECU,SPECP,DR,
     *                  DDTOL,IS,DTOL1,DTOL2,P)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ITRN/ JST,IST
      COMMON /CNTL1/ TAB
      DIMENSION DP1(MNDOF)
      DIMENSION DP2(MNDOF),DR(MNDOF),IS(MNDOF),P(MNDOF)
      DIMENSION DU1(MNDOF),DU2(MNDOF),DDU(MNDOF),DDP(MNDOF)
      CP1=0.D0
      CP2=0.D0
      CD1=0.D0
      CD2=0.D0
      CR1=0.D0
      CR2=0.D0
      DO 100 I=1,MNDOF
      IF(IST.EQ.1) THEN
      DDU(I)=DDU(I)+DLM2*DU1(I)
      DDP(I)=DDP(I)+DLM2*(DP1(I)*FLD1+DP2(I)*FLD2)
      ELSE
      DDU(I)=DDU(I)+DU2(I)
      ENDIF
  100 CONTINUE
      DO 200 I=1,MNDOF
      IF(IS(I).EQ.0) THEN
      CP1=CP1+DU2(I)*DU2(I)
      CP2=CP2+DDU(I)*DDU(I)
      CD1=CD1+DR(I)*DR(I)
      CD2=CD2+DDP(I)*DDP(I)
      ENDIF
  200 CONTINUE
      DTOL2=DSQRT(CP1)/DSQRT(CP2)
      DTOL1=DSQRT(CD1)/DSQRT(CD2)
      WRITE(50,7500) JST,TAB,IST,TAB,DTOL1,TAB,DTOL2
 7500 FORMAT(I5,A1,I5,5(A1,E12.6))
      IF(IST.EQ.1) THEN
      DTOL1=1.D0
      DTOL2=1.D0
      ENDIF
      RETURN
      END
```

```
      SUBROUTINE MODIF(SM,ELRHS,ELEM,P,IS,NDOFN,ICONN,
     *                 NNE,NDOF,NN,MNDOFN,IEL)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SM(NDOF,NDOF),ELRHS(NDOF)
      DIMENSION P(MNDOFN,NN)
      DIMENSION IS(MNDOFN,NN),ICONN(NNE),NDOFN(NN)
      DIMENSION ELEM(1)
      COMMON /CONSTS/ ZERO,ONE,TWO
      COMMON /CNTL/ ISYM,NUMEL,IRESOL,IDUM(26)
      K=0
      DO 30 I=1,NNE
      NODE=ICONN(I)
      DO 30 J=1,NDOFN(NODE)
      K=K+1
      IF(IS(J,NODE).EQ.0) THEN
      ELRHS(K)=ELRHS(K)+P(J,NODE)
      P(J,NODE)=ZERO
      ELSE
      DISP=P(J,NODE)
      IF(ISYM.EQ.1) THEN
      DO 10 L=1,K
      ELRHS(L)=ELRHS(L)-SM(L,K)*DISP
      SM(L,K)=ZERO
   10 CONTINUE
      DO 20 L=K,NDOF
      ELRHS(L)=ELRHS(L)-SM(K,L)*DISP
      SM(K,L)=ZERO
   20 CONTINUE
      ELSE
      DO 15 L=1,NDOF
      ELRHS(L)=ELRHS(L)-SM(L,K)*DISP
      SM(L,K)=ZERO
      SM(K,L)=ZERO
   15 CONTINUE
      ENDIF
      SM(K,K)=ONE
      ELRHS(K)=DISP
      ENDIF
   30 CONTINUE
      K=0
      IF(IRESOL.EQ.1) GOTO 150
      DO 48 J=1,NDOF
      IF(ISYM.EQ.1) THEN
      IK=J
      DO 40 I=1,IK
      K=K+1
      ELEM(K)=SM(I,J)
   40 CONTINUE
      ELSE
      IK=NDOF
      DO 45 I=1,IK
      K=K+1
      ELEM(K)=SM(J,I)
   45 CONTINUE
      ENDIF
   48 CONTINUE
  150 DO 50 I=1,NDOF
      K=K+1
```

```
      ELEM(K)=ELRHS(I)
   50 CONTINUE
      RETURN
      END

      SUBROUTINE PICK(X,Y,ICONN,NNE,NDIM,NN)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(NDIM,NN)
      DIMENSION Y(NDIM,NNE)
      DIMENSION ICONN(NNE)
      DO 10 J=1,NNE
      NODE=ICONN(J)
      DO 10 I=1,NDIM
      Y(I,J)=X(I,NODE)
   10 CONTINUE
      RETURN
      END

      SUBROUTINE PREOUT(INTA,IEL,N,IA,IB)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION INTA(1)
      DIMENSION IA(1),IB(1)
      COMMON /INDS/ INDR(60),INDI(30)
      COMMON /DIMS/ MNCM,MNDOFN,MNNE,NDIM,NMAT1,NMAT2,NN,MNDOFE,MNDOF,
     *              NUMEL1,NUMEL2,NUMEL3,ICOMP,NGAU,IARC,IBAU,ISTL
      J=INDI(8)+MNNE*(IEL-1)-1
      DO 10 I=1,N
      J=J+1
      INTA(J)=IB(I)
   10 CONTINUE
      RETURN
      END

      SUBROUTINE PREP(IN,IA,NNE,NDOFN,ICONN,NUMEL,NN,MNNE)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION IN(1),IA(1)
      DIMENSION NNE(NUMEL),NDOFN(NN),ICONN(MNNE,NUMEL)
      K=0
      L=0
      DO 10 I=1,NUMEL
      K=K+1
      IN(K)=NNE(I)
      DO 10 J=1,NNE(I)
      L=L+1
      NODE=ICONN(J,I)
      IA(L)=10*NODE+NDOFN(NODE)
   10 CONTINUE
      RETURN
      END

      SUBROUTINE PRNT(U,P,NN,MNDOFN,DTOL,NUDOF,DR)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ITRN/ JST,IST
      COMMON /CNTL1/ TAB
      DIMENSION U(MNDOFN,NN),P(MNDOFN,NN),DR(MNDOFN,NN)
      DO 20 I=1,NN
      WRITE(43) (U(IN,I),IN=1,MNDOFN),(P(IN,I),IN=1,MNDOFN)
   20 CONTINUE
      NODE=NUDOF/MNDOFN+1
```

```
      IDOF=NUDOF-(NODE-1)*MNDOFN
      WRITE(51,7500) JST,TAB,U(IDOF,NODE),TAB,P(IDOF,NODE),TAB,
     *           DTOL
 7500 FORMAT(I5,4(A1,E12.6))
      RETURN
      END

      SUBROUTINE SOLIN(A,IA,IEL,IFG,NRHS,NUMDES,LDEST,ELEM)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(1),IA(1)
      DIMENSION LDEST(1),ELEM(1)
      COMMON /INDS/ INDR(60),INDI(30)
      COMMON /DIMS/ MNCM,MNDOFN,MNNE,NDIM,NMAT1,NMAT2,NN,MNDOFE,MNDOF,
     *              NUMEL1,NUMEL2,NUMEL3,ICOMP,NGAU,IARC,IBAU,ISTL
      NUMDES=IA(INDI(5)+IEL-1)
      J=INDI(8)+MNNE*(IEL-1)-1
      DO 10 I=1,NUMDES
      J=J+1
      LDEST(I)=IA(J)
   10 CONTINUE
      IF(IFG.EQ.1) RETURN
      CALL STIFF(A(INDR(1)),A(INDR(3)),A(INDR(5)),A(INDR(7)),
     *       IA(INDI(3)),IA(INDI(10)),IA(INDI(5)),A(INDR(21)),
     *       NDIM,NN,NUMEL,NMAT1,NMAT2,MNDOFN,MNNE,MNCM,MNDOFE,MNDOF,
     *       NGAU,ELEM,IEL,A(INDR(12)),A(INDR(8)),IA(INDI(1)),
     *       IA(INDI(2)),IA(INDI(4)),IA(INDI(9)),A(INDR(4)),
     *       A(INDR(6)),A(INDR(23)),A(INDR(26)),A(INDR(24)),
     *       A(INDR(28)),A(INDR(30)),NUMEL1,NUMEL2,NUMEL3,
     *       A(INDR(32)),A(INDR(34)),A(INDR(35)),IA(INDI(11)),
     *       IA(INDI(12)),ICOMP,A(INDR(13)),A(INDR(19)),
     *       A(INDR(33)),IA(INDI(14)),IARC,IBAU,ISTL)
      RETURN
      END

      SUBROUTINE SOLOUT(A,IA,IEL,NDOF,NRHS,ELEM)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ELEM(1)
      DIMENSION A(1),IA(1)
      COMMON /INDS/ INDR(60),INDI(30)
      COMMON /DIMS/ MNCM,MNDOFN,MNNE,NDIM,NMAT1,NMAT2,NN,MNDOFE,MNDOF,
     *              NUMEL1,NUMEL2,NUMEL3,ICOMP,NGAU,IARC,IBAU,ISTL
      J=INDI(3)+MNNE*(IEL-1)-1
      NNE=IA(INDI(5)+IEL-1)
      M=0
      DO 20 I=1,NNE
      NODE=IA(J+I)
      NDOFN=IA(INDI(1)+NODE-1)
      K=INDR(2)+MNDOFN*(NODE-1)-1
      DO 10 L=1,NDOFN
      A(K+L)=ELEM(M+L)
   10 CONTINUE
      M=M+NDOFN
   20 CONTINUE
      RETURN
      END

      SUBROUTINE UPDT(X,ESM,CONSTM1,EX,ICONN,IELM,NNE,ST1,
     *       NDIM,NN,NUMEL,NMAT1,NMAT2,MNDOFN,MNNE,MNCM,
     *       MNDOFE,MNDOF,NGAU,U,EU,P,R,DR,EEP,IS,DDU,DDP,EEU,
```

```
     *        DU1,DU2,DLM2,NDOFN,IELT,NDOFE,ELRHS,AGP,
     *        EMAX,RST1,RST2,PMAX,NUMEL1,NUMEL2,NUMEL3,
     *        EMAX1,RST,BRST,IEL1,IEL2,ICOMP,CONSTM2,
     *        EMAX3,IEL3,IARC,IBAU,ISTL)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ITRN/ JST,IST
      DIMENSION X(NDIM,NN),EX(NDIM,MNNE)
      DIMENSION ICONN(MNNE,NUMEL),IELM(NUMEL),NNE(NUMEL)
      DIMENSION CONSTM1(MNCM,NMAT1),CONSTM2(MNCM,NMAT2)
      DIMENSION U(MNDOFN,NN),ESM(MNDOFE,MNDOFE)
      DIMENSION EU(MNDOFN,MNNE),IS(MNDOFN,NN)
      DIMENSION EEP(MNDOFN,MNNE),P(MNDOFN,NN),R(MNDOFN,NN)
      DIMENSION DR(MNDOFN,NN)
      DIMENSION DDU(MNDOFN,NN),DDP(MNDOFN,NN),EEU(MNDOFN,MNNE)
      DIMENSION DU1(MNDOFN,NN),DU2(MNDOFN,NN)
      DIMENSION ST1(3*NGAU*NGAU,NUMEL2),AGP(NGAU*NGAU,NUMEL2)
      DIMENSION RST1(NGAU*NGAU,NUMEL2)
      DIMENSION RST2(NGAU*NGAU,NUMEL2),ELRHS(MNDOFN,MNNE)
      DIMENSION EMAX(2*6*NGAU*NGAU,NUMEL2)
      DIMENSION PMAX(28*NGAU*NGAU,NUMEL2)
      DIMENSION NDOFN(NN),IELT(NUMEL),NDOFE(NUMEL)
      DIMENSION RST(NGAU,NUMEL1),BRST(NGAU,NUMEL3)
      DIMENSION EMAX1(6*NGAU,NUMEL1),EMAX3(11*NGAU,NUMEL3)
      DIMENSION IEL1(NUMEL),IEL2(NUMEL),IEL3(NUMEL)
      CALL CLEAR(R,MNDOFN*NN)
      DO 20 IEL=1,NUMEL
      CALL PICK(X,EX,ICONN(1,IEL),NNE(IEL),NDIM,NN)
      DO 10 I=1,NNE(IEL)
      NODE=ICONN(I,IEL)
      DO 10 J=1,NDOFN(NODE)
      EU(J,I)=U(J,NODE)+DDU(J,NODE)
      EEU(J,I)=DLM2*DU1(J,NODE)+DU2(J,NODE)
   10 CONTINUE
      IF(IELT(IEL).EQ.1) THEN
C
C.....LINE ELEMENT
C
      IDUM=IEL1(IEL)

      CALL UPSS1(EX,CONSTM1(1,IELM(IEL)),EMAX1(1,IDUM),
     *      NCM,IEL,EU,ELRHS,RST(1,IDUM),ICOMP,
     *      NGAU,MNDOFN,MNNE,NDOFE(IEL),NDIM,EEP,IBAU)

      ELSEIF(IELT(IEL).EQ.2) THEN
C
C     RECTANGULAR ELEMENT
C
      IDUM=IEL2(IEL)
      CALL UPQD4(EX,CONSTM1,ST1(1,IDUM),AGP(1,IDUM),EMAX(1,IDUM),
     *      RST1(1,IDUM),RST2(1,IDUM),MNCM,EU,IEL,EEP,EEU,
     *      PMAX(1,IDUM),NMAT1,NMAT2,IELM(IEL),
     *      ICOMP,ELRHS,CONSTM2,NGAU,MNDOFN,MNNE,MNDOFE,NDIM,
     *      SHST,IARC,IBAU,ISTL)

      ELSEIF(IELT(IEL).EQ.3) THEN
C
C.....BOND-SLIP ELEMENT
C
      IDUM=IEL3(IEL)
```

```
      CALL UPBOND(EX,CONSTM1(1,IELM(IEL)),EMAX3(1,IDUM),
     *       NCM,IEL,EU,ELRHS,BRST(1,IDUM),ICOMP,
     *       NGAU,MNDOFN,MNNE,NDOFE(IEL),NDIM,EEP)
      ENDIF
      DO 30 I=1,NNE(IEL)
      NODE=ICONN(I,IEL)
      DO 30 J=1,NDOFN(NODE)
      R(J,NODE)=R(J,NODE)+EEP(J,I)
   30 CONTINUE
   20 CONTINUE
      DO 200 I=1,NN
      DO 200 J=1,MNDOFN
      IF(IS(J,I).EQ.0) THEN
      DR(J,I)=(P(J,I)+DDP(J,I))-R(J,I)
      ELSE
      DR(J,I)=0.D0
      ENDIF
  200 CONTINUE
      RETURN
      END

      SUBROUTINE STIFF(X,ESM,CONSTM1,EX,ICONN,IELM,NNE,ST1,
     *       NDIM,NN,NUMEL,NMAT1,NMAT2,MNDOFN,MNNE,MNCM,MNDOFE,MNDOF,
     *       NGAU,ELEM,IEL,U,EU,NDOFN,IS,IELT,NDOFE,ELRHS,P,AGP,
     *       EMAX,PMAX,RST1,RST2,NUMEL1,NUMEL2,NUMEL3,
     *       EMAX1,RST,BRST,IEL1,IEL2,ICOMP,CONSTM2,
     *       DDU,EMAX3,IEL3,IARC,IBAU,ISTL)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ITRN/ JST,IST
      DIMENSION X(NDIM,NN),EX(NDIM,MNNE)
      DIMENSION ESM(MNDOFE,MNDOFE)
      DIMENSION ICONN(MNNE,NUMEL),IELM(NUMEL),NNE(NUMEL)
      DIMENSION CONSTM1(MNCM,NMAT1),CONSTM2(MNCM,NMAT2)
      DIMENSION ST1(3*NGAU*NGAU,NUMEL2),AGP(NGAU*NGAU,NUMEL2)
      DIMENSION U(MNDOFN,NN),RST(NGAU,NUMEL1),DDU(MNDOFN,NN)
      DIMENSION BRST(NGAU,NUMEL3)
      DIMENSION EU(MNDOFN,MNNE),RST1(NGAU*NGAU,NUMEL2)
      DIMENSION NDOFN(NN),IS(MNDOFN,NN),RST2(NGAU*NGAU,NUMEL2)
      DIMENSION IELT(1),NDOFE(1),ELRHS(1),P(1)
      DIMENSION EMAX(2*6*NGAU*NGAU,NUMEL2)
      DIMENSION PMAX(28*NGAU*NGAU,NUMEL2)
      DIMENSION EMAX1(6*NGAU,NUMEL1),EMAX3(11*NGAU,NUMEL3)
      DIMENSION IEL1(NUMEL),IEL2(NUMEL),IEL3(NUMEL)
      CALL PICK(X,EX,ICONN(1,IEL),NNE(IEL),NDIM,NN)
      DO 10 I=1,NNE(IEL)
      NODE=ICONN(I,IEL)
      DO 10 J=1,NDOFN(NODE)
      EU(J,I)=U(J,NODE)+DDU(J,NODE)
   10 CONTINUE
      IF(IELT(IEL).EQ.1) THEN
C
C.....LINE ELEMENT
C
      IDUM=IEL1(IEL)

      CALL SF1(EX,CONSTM1(1,IELM(IEL)),ESM,EMAX1(1,IDUM),
     *         NCM,IEL,EU,ELRHS,RST(1,IDUM),
     *         ICOMP,NGAU,MNDOFN,MNNE,NDOFE(IEL),NDIM,IBAU)
      ELSEIF(IELT(IEL).EQ.2) THEN
```

```
C
C     RECTANGULAR ELEMENT
C
      IDUM=IEL2(IEL)
      CALL QUADS4(EX,CONSTM1,ESM,ST1(1,IDUM),AGP(1,IDUM),
     *            EMAX(1,IDUM),MNCM,IEL,EU,ELRHS,
     *            PMAX(1,IDUM),RST1(1,IDUM),RST2(1,IDUM),
     *            NMAT1,NMAT2,IELM(IEL),ICOMP,CONSTM2,
     *            NGAU,MNDOFN,MNNE,MNDOFE,
     *            NDIM,IARC,IBAU,ISTL)
      ELSEIF(IELT(IEL).EQ.3) THEN
C
C.....BOND-SLIP ELEMENT
C
      IDUM=IEL3(IEL)
      CALL SFBOND(EX,CONSTM1(1,IELM(IEL)),ESM,EMAX3(1,IDUM),
     *              NCM,IEL,EU,ELRHS,BRST(1,IDUM),
     *              ICOMP,NGAU,MNDOFN,MNNE,NDOFE(IEL),NDIM)
      ENDIF
C
C.....MODIFY ELEMENT STIFFNESS MATRIX FOR SUPPORT CONDITIONS
C
      CALL MODIF(ESM,ELRHS,ELEM,P,IS,NDOFN,
     *             ICONN(1,IEL),NNE(IEL),NDOFE(IEL),NN,MNDOFN,IEL)
      RETURN
      END
C
      SUBROUTINE STRESS(X,CONSTM1,EX,ICONN,IELM,NNE,ST1,
     *        NDIM,NN,NUMEL,NMAT1,NMAT2,MNDOFN,MNNE,MNCM,MNDOFE,MNDOF,
     *        NGAU,U,EU,EEP,NDOFN,IS,IELT,NDOFE,AGP,EMAX,
     *        RST1,RST2,PMAX,NUMEL1,NUMEL2,NUMEL3,
     *        EMAX1,RST,BRST,IEL1,IEL2,ICOMP,CONSTM2,
     *        SHDUM,EMAX3,IEL3,IARC,IBAU,ISTL)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ITRN/ JST,IST
      COMMON /CNTL1/ TAB
      DIMENSION X(NDIM,NN),EX(NDIM,MNNE)
      DIMENSION ICONN(MNNE,NUMEL),IELM(NUMEL),NNE(NUMEL)
      DIMENSION CONSTM1(MNCM,NMAT1),CONSTM2(MNCM,NMAT2)
      DIMENSION U(MNDOFN,NN)
      DIMENSION EU(MNDOFN,MNNE),EEP(MNDOFE)
      DIMENSION ST1(3*NGAU*NGAU,NUMEL2),AGP(NGAU*NGAU,NUMEL2)
      DIMENSION RST1(NGAU*NGAU,NUMEL2),RST2(NGAU*NGAU,NUMEL2)
      DIMENSION NDOFN(NN),IS(MNDOFN,NN),IELT(NUMEL),NDOFE(NUMEL)
      DIMENSION EMAX(2*6*NGAU*NGAU,NUMEL2)
      DIMENSION PMAX(28*NGAU*NGAU,NUMEL2)
      DIMENSION ASTRESS(3),ASTRAIN(3),P(2)
      DIMENSION RST(NGAU,NUMEL1),BRST(NGAU,NUMEL3)
      DIMENSION EMAX1(6*NGAU,NUMEL1),EMAX3(11*NGAU,NUMEL3)
      DIMENSION IEL1(NUMEL),IEL2(NUMEL),IEL3(NUMEL)
      PAI=2.D0*DASIN(1.D0)
      DO 30 I=1,3
      ASTRESS(I)=0.D0
   30 ASTRAIN(I)=0.D0
      DO 20 IEL=1,NUMEL
      CALL PICK(X,EX,ICONN(1,IEL),NNE(IEL),NDIM,NN)
      DO 10 I=1,NNE(IEL)
      NODE=ICONN(I,IEL)
      DO 10 J=1,NDOFN(NODE)
```

```
      EU(J,I)=U(J,NODE)
   10 CONTINUE
      IF(IELT(IEL).EQ.1) THEN
C
C.....LINE ELEMENT
C
      IDUM=IEL1(IEL)
      CALL SS1(EX,CONSTM1(1,IELM(IEL)),EMAX1(1,IDUM),
     *         NCM,IEL,EU,RST(1,IDUM),
     *         ICOMP,NGAU,MNDOFN,MNNE,NDOFE(IEL),NDIM,IBAU)
      ELSEIF(IELT(IEL).EQ.2) THEN
C
C     RECTANGULAR ELEMENT
C
      IDUM=IEL2(IEL)
      CALL EFQD4(EX,CONSTM1,ST1(1,IDUM),AGP(1,IDUM),EMAX(1,IDUM),
     *           RST1(1,IDUM),RST2(1,IDUM),MNCM,EU,IEL,EEP,
     *           ASTRESS,ASTRAIN,PMAX(1,IDUM),
     *           NMAT1,NMAT2,IELM(IEL),ICOMP,
     *           CONSTM2,NGAU,MNDOFN,MNNE,MNDOFE,NDIM,
     *           SHDUM,IEL,IARC,IBAU,ISTL)
      ELSEIF(IELT(IEL).EQ.3) THEN
C
C.....BOND-SLIP ELEMENT
C
      IDUM=IEL3(IEL)
      CALL SSBOND(EX,CONSTM1(1,IELM(IEL)),EMAX3(1,IDUM),
     *            NCM,IEL,EU,BRST(1,IDUM),
     *         ICOMP,NGAU,MNDOFN,MNNE,NDOFE(IEL),NDIM)
      ENDIF
   20 CONTINUE
      RETURN
      END

      SUBROUTINE CNCLEAR
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CNTL/IDUM(29)
      DO 100 I=1,30
      IDUM(I)=0
  100 CONTINUE
      RETURN
      END

      SUBROUTINE QUADS4(XX,CONSTM1,S,ST1,AGP,EMAX,
     *                  NCM,NEL,EU,ELRHS,PMAX,RST1,RST2,
     *                  NMAT1,NMAT2,IELM,ICOMP,CONSTM2,
     *                  NGAU,MNDOFN,MNNE,MNDOFE,NDIM,
     *                  IARC,IBAU,ISTL)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CNTL/ ISYM,NUMEL,IRESOL,IIDUM(26)
      COMMON /ITRN/ JST,IST
      COMMON /XGWGT/ XG(4,4),WGT(4,4)
      DIMENSION D(4,4),B(4,16),XX(NDIM,MNNE),S(MNDOFE,MNDOFE)
      DIMENSION DB(4)
      DIMENSION CONSTM1(NCM,NMAT1),ST1(3,NGAU*NGAU),D1(3,3),D2(3,3)
      DIMENSION CONSTM2(NCM,NMAT2),AGP(NGAU*NGAU)
      DIMENSION EPSN(3),SIGM(4),EMAX(2*6,NGAU*NGAU),EPS(3)
      DIMENSION EU(MNDOFE),ELRHS(MNDOFE),P(2),H(8)
      DIMENSION RST1(NGAU*NGAU),RST2(NGAU*NGAU)
```

```
      DIMENSION PMAX(28,NGAU*NGAU),IREN(6)
      DIMENSION US(6),YS(6),RS(6),AS(6),DS(6)
      DIMENSION REP(6),REPP(6)
C
C          CONSTM2(1) = YOUNGS MODULUS
C          CONSTM2(2) = POISSONS RATIO
C          CONSTM2(3) = THICHNESS
C          CONSTM2(4) = VOID
C          CONSTM2(5) = UNIT WEIGHT
C          CONSTM2(6) = ULTIMATE STRENGTH IN COMPRESSION
C          CONSTM2(7) = INITIAL MODULUS IN COMPRESSION
C          CONSTM2(8) = SECANT MODULUS IN COMPRESSION
C          CONSTM2(9) = FINAL SECANT MODULUS IN COMPRESSION
C          CONSTM2(10) = FINAL STRENGTH IN COMPRESSION
C          CONSTM2(11) = ULTIMATE STRENGTH IN TENSION
C          CONSTM2(12) = INITIAL MODULUS IN TENSION
C          CONSTM2(13) = SECANT MODULUS IN TENSION
C          CONSTM2(14) = FINAL SECANT MODULUS IN TENSION
C          CONSTM2(15) = FINAL STRENGTH IN TENSION
C          CONSTM2(16) = TYPE OF SMEARED STEEL 1
C          CONSTM2(17) = TYPE OF SMEARED STEEL 2
C          CONSTM2(18-21) = TYPE OF DISCRETE REINF. BARS AFFECTING
C                              TENSION STIFFENING

C          CONSTM1(1) = YIELD STRESS
C          CONSTM1(2) = YOUNGS MODULUS
C          CONSTM1(3) = REINFORCEMENT RATIO
C          CONSTM1(4) = DIRECTION WITH RESPECT TO X AXIS
C          CONSTM1(5) = DIAMETER
C          CONSTM1(6) = AREA
C          CONSTM1(7) = STRAIN HARDENING STRAIN
C          CONSTM1(8) = ULTIMATE STRAIN
C          CONSTM1(9) = ULTIMATE STRESS
C          CONSTM1(10) = ULTIMATE BOND STRESS
C          CONSTM1(11) = FINAL BOND STRESS
C          CONSTM1(12) = BOND-SLIP 1
C          CONSTM1(13) = BOND-SLIP 2
C          CONSTM1(14) = FINAL BOND-SLIP
C
      PAI=2.D0*DASIN(1.D0)
      CALL SCONS(CONSTM1,CONSTM2,YM,PR,THIC,NINT,UWT,
     *       USC,YOC,YSC,YFC,UFC,UST,YOT,YST,YFT,UFT,
     *       IREN,US,YS,RS,AS,DS,NCM,IELM,NMAT1,NMAT2)
      NINT=NGAU
      NDOF=MNDOFE
      CALL CLEAR(ELRHS,NDOF)
      IF(IRESOL.EQ.1) RETURN
      ITYPE=2
   20 DO 30 I=1,MNDOFE
      DO 30 J=1,MNDOFE
   30 S(I,J)=0.D0
      KK=0
      DO 80 LX=1,NINT
      RI=XG(LX,NINT)
      DO 80 LY=1,NINT
      SI=XG(LY,NINT)
      KK=KK+1
C
C     EVALUATE DERIVATIVE OPERATOR B AND THE JACOBIAN DETERMINANT DET
```

```
C
      IF(ICOMP.EQ.2) THEN
      CALL STDM8(XX,B,H,DET,RI,SI,XBAR,NEL,ITYPE,NDIM,MNNE)
      ELSE
      CALL STDM4(XX,B,H,DET,RI,SI,XBAR,NEL,ITYPE,NDIM,MNNE)
      ENDIF
      IF(ITYPE.GT.0) XBAR=THIC
      WT=WGT(LX,NINT)*WGT(LY,NINT)*XBAR*DET
      DO 810 J=1,3
      SIGM(J)=0.0D0
  810 EPSN(J)=0.0D0
      DO 815 J=2,MNDOFE,2
      JJ=J-1
      EPSN(1)=EPSN(1)+B(1,JJ)*EU(JJ)
      EPSN(2)=EPSN(2)+B(2,J )*EU(J )
      EPSN(3)=EPSN(3)+B(3,JJ)*EU(JJ)+B(3,J)*EU(J)
  815 CONTINUE
C
C.....PRINCIPAL STRAINS
C
      CC=(EPSN(1)+EPSN(2))*0.5D0
      BB=(EPSN(1)-EPSN(2))*0.5D0
      DUM=AGP(KK)
      EPSN(3)=EPSN(3)/2.D0
      CALL PRINCIPAL(EPSN,P,AG,DUM)
      EPSN(3)=EPSN(3)*2.D0
      EPSN(1)=P(1)
      EPSN(2)=P(2)
      AGS=AG
      EPS(1)=EPSN(1)
      EPS(2)=EPSN(2)
  950 CONTINUE
      DO 955 I=1,6
      IF(IREN(I).NE.0) THEN
      REP(I)=CC+BB*DCOS(2.D0*AS(I))+EPSN(3)*DSIN(2.D0*AS(I))/2.D0
      REPP(I)=REP(I)
      CALL REPST(REP(I),UST,YST,USC,YSC,
     *  AS(I),PMAX(1,KK),PMAX(9,KK),PMAX(17,KK))
      ENDIF
  955 CONTINUE
C
C      SMEARED REINFORCING STEEL IN AXIS 1
C
      CALL DMATS1(D1,REPP(1),EMAX(1,KK),
     *              CONSTM1(1,IREN(1)),NCM,NMAT1,IREN(1),
     *   RST1(KK),UST,YST,IBAU,EPS,PMAX(1,KK),USC,YSC,UFC,YFC)
C
C      SMEARED REINFORCING STEEL IN AXIS 2
C
      CALL DMATS1(D2,REPP(2),EMAX(7,KK),
     *              CONSTM1(1,IREN(2)),NCM,NMAT1,IREN(2),
     *  RST2(KK),UST,YST,IBAU,EPS,PMAX(1,KK),USC,YSC,UFC,YFC)
C
C      CONCRETE
C
      CALL DMAT(D,AGS,EPS,CONSTM1,NCM,PMAX(1,KK),
     *           NMAT1,NMAT2,IELM,CONSTM2,
     *    AGP(KK),AG,REP)
C
```

```
C     ADD CONTRIBUTION TO ELEMENT STIFFNESS
C
      DO 370 J=1,MNDOFE
      DO 340 K=1,3
      DB(K)=0.0D0
      DO 340 L=1,3
  340 DB(K)=DB(K)+(D(K,L)+D1(K,L)+D2(K,L))*B(L,J)
      DO 360 I=1,MNDOFE
      STIFF=0.0D0
      DO 350 L=1,3
  350 STIFF=STIFF+B(L,I)*DB(L)
  360 S(I,J)=S(I,J)+STIFF*WT
  370 CONTINUE
   80 CONTINUE
      RETURN
      END

      SUBROUTINE REPST(REP,UST,YST,USC,YSC,
     *  AG,CMST,TMST,RMST)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION TMST(8),RMST(8)
      PAI=2.D0*DASIN(1.D0)
      ESC=DABS(USC/YSC)
      EST=DABS(UST/YST)
      CALL AGPICK(AG,KK,TH)
      CALL FINMAX(AG,TMST,TM,KK,TH)
      CALL REFPICK(AG,CMST,RMST,ESC,KK,TH,ECT)
      IF(REP.LT.(TM+ECT)) REP=(TM+ECT)
      RETURN
      END

      SUBROUTINE STDM4 (XX,B,H,DET,R,S,XBAR,NEL,ITYPE,NDIM,MNNE)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XX(NDIM,MNNE),B(4,16),H(8),P(2,4),XJ(2,2),XJI(2,2)
      ITYPE=2
      RP=1.0D0+R
      SP=1.0D0+S
      RM=1.0D0-R
      SM=1.0D0-S
C
C     INTERPOLATION FUNCTIONS
C
      H(1)=0.25D0*RP*SP
      H(2)=0.25D0*RM*SP
      H(3)=0.25D0*RM*SM
      H(4)=0.25D0*RP*SM
C
C     NATURAL COORDINATE DERIVATIVE OF THE INTERPOLATION FUNCTIONS
C
C        1. WITH RESPECT TO R
C
      P(1,1)=0.25D0*SP
      P(1,2)=-P(1,1)
      P(1,3)=-0.25D0*SM
      P(1,4)=-P(1,3)
C
C        2. WITH RESPECT TO S
C
      P(2,1)=0.25D0*RP
```

```
      P(2,2)=0.25D0*RM
      P(2,3)=-P(2,2)
      P(2,4)=-P(2,1)
C
C     EVALUATE THE JACOBIAN MATRIX AT POINT (R,S)
C
   10 DO 30 I=1,2
      DO 30 J=1,2
      DUM=0.0D0
      DO 20 K=1,4
   20 DUM=DUM+P(I,K)*XX(J,K)
   30 XJ(I,J)=DUM
C
C     COMPUTE THE DETERMINANT OF JACOBIAN MATRIX AT POINT (R,S)
C
      DET=XJ(1,1)*XJ(2,2)-XJ(2,1)*XJ(1,2)
      IF(DET.GT.0.00000001D0) GO TO 40
      WRITE(50,2000) NEL
      STOP
C
C     COMPUTE INVERSE OF THE JACOBIAN MATRIX
C
   40 DUM=1.0D0/DET
      XJI(1,1)=XJ(2,2)*DUM
      XJI(1,2)=-XJ(1,2)*DUM
      XJI(2,1)=-XJ(2,1)*DUM
      XJI(2,2)=XJ(1,1)*DUM
C
C     EVALUATE GLOBAL DERIVATIVE OPERATOR B
C
      K2=0
      DO 60 K=1,4
      K2=K2+2
      B(1,K2-1)=0.D0
      B(1,K2  )=0.D0
      B(2,K2-1)=0.D0
      B(2,K2  )=0.D0
      DO 50 I=1,2
      B(1,K2-1)=B(1,K2-1)+XJI(1,I)*P(I,K)
   50 B(2,K2  )=B(2,K2  )+XJI(2,I)*P(I,K)
      B(3,K2  )=B(1,K2-1)
   60 B(3,K2-1)=B(2,K2  )
      RETURN
 2000 FORMAT (///'*** ERROR',
     1       52H ZERO OR NEGATIVE JACOBIAN DETERMINANT FOR ELEMENT (,I4,
     2       1H)   )
      END

      SUBROUTINE STDM8(XX,B,H,DET,R,S,XBAR,NEL,ITYPE,NDIM,
     *                  MNNE)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XX(NDIM,MNNE),B(4,16),H(8),P(2,8),XJ(2,2),XJI(2,2)
      RP=1.0D0+R
      SP=1.0D0+S
      RM=1.0D0-R
      SM=1.0D0-S
      H(5) = 0.5D0*RP*RM*SP
      H(6) = 0.5D0*RM*SP*SM
      H(7) = 0.5D0*RP*RM*SM
```

```
      H(8) = 0.5D0*RP*SP*SM
      H(1) = .25D0*RP*SP-0.5D0*H(5)-0.5D0*H(8)
      H(2) = .25D0*RM*SP-0.5D0*H(5)-0.5D0*H(6)
      H(3) = .25D0*RM*SM-0.5D0*H(6)-0.5D0*H(7)
      H(4) = .25D0*RP*SM-0.5D0*H(7)-0.5D0*H(8)
      P(1,1) = .25D0*SP*(2.D0*R+S)
      P(1,2) = .25D0*SP*(2.D0*R-S)
      P(1,3) = .25D0*SM*(2.D0*R+S)
      P(1,4) = .25D0*SM*(2.D0*R-S)
      P(1,5) = -R*SP
      P(1,6) = -0.5D0*SP*SM
      P(1,7) = -R*SM
      P(1,8) = 0.5D0*SP*SM
      P(2,1) = .25D0*RP*(2.D0*S+R)
      P(2,2) = .25D0*RM*(2.D0*S-R)
      P(2,3) = .25D0*RM*(2.D0*S+R)
      P(2,4) = .25D0*RP*(2.D0*S-R)
      P(2,5) = 0.5D0*RP*RM
      P(2,6) = -S*RM
      P(2,7) = -0.5D0*RP*RM
      P(2,8) = -S*RP
   10 DO 30 I=1,2
      DO 30 J=1,2
      DUM=0.0D0
      DO 20 K=1,8
   20 DUM=DUM+P(I,K)*XX(J,K)
   30 XJ(I,J)=DUM
      DET=XJ(1,1)*XJ(2,2)-XJ(2,1)*XJ(1,2)
      IF(DET.GT.0.00000001D0) GO TO 40
      WRITE(50,2000) NEL
      STOP
   40 DUM=1.0D0/DET
      XJI(1,1)=XJ(2,2)*DUM
      XJI(1,2)=-XJ(1,2)*DUM
      XJI(2,1)=-XJ(2,1)*DUM
      XJI(2,2)=XJ(1,1)*DUM
      K2=0
      DO 60 K=1,8
      K2=K2+2
      B(1,K2-1)=0.0D0
      B(1,K2  )=0.0D0
      B(2,K2-1)=0.0D0
      B(2,K2  )=0.0D0
      DO 50 I=1,2
      B(1,K2-1)=B(1,K2-1)+XJI(1,I)*P(I,K)
   50 B(2,K2  )=B(2,K2  )+XJI(2,I)*P(I,K)
      B(3,K2  )=B(1,K2-1)
   60 B(3,K2-1)=B(2,K2  )
      RETURN
 2000 FORMAT (10H0*** ERROR,
     1        52H ZERO OR NEGATIVE JACOBIAN DETERMINANT FOR ELEMENT (,I4,
     2        1H)   )
      END

      SUBROUTINE DMAT(D,AG,EPSN,CONSTM1,NCM,PMAX,
     *       NMAT1,NMAT2,IELM,CONSTM2,AGP,AGE,REP)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CL/ ISOL,ISP
      COMMON /CNTL/ ISYM,IDUM(28)
```

```
      COMMON /ITRN/ JST,IST
      DIMENSION CONSTM2(NCM,NMAT2),CONSTM1(NCM,NMAT1)
      DIMENSION D(4,4),EPSN(3),TP(3,3),TT(3,3),SST1(3)
      DIMENSION US(6),YS(6),RS(6),AS(6),DS(6),REP(6)
      DIMENSION PMAX(28),IREN(6)
      PAI=2.D0*DASIN(1.D0)
      CALL SCONS(CONSTM1,CONSTM2,YM,PR,THIC,NINT,UWT,
     *      USC,YOC,YSC,YFC,UFC,UST,YOT,YST,YFT,UFT,
     *      IREN,US,YS,RS,AS,DS,NCM,IELM,NMAT1,NMAT2)
      IF(AG.GT.0.D0) AG1=AG-PAI/2.D0
      IF(AG.LE.0.D0) AG1=AG+PAI/2.D0
      IF(AGE.GT.0.D0) AGE1=AGE-PAI/2.D0
      IF(AGE.LE.0.D0) AGE1=AGE+PAI/2.D0
      DO 10 I=1,3
      SST1(I)=0.D0
      DO 10 J=1,3
   10 D(I,J)=0.D0
C
C.....INITIAL STIFFNESS
C
      IF(JST.EQ.0.OR.ISP.GE.2) THEN
      DUM=YOC/(1.D0-PR*PR)
      D(1,1)=DUM
      D(1,2)=DUM*PR
      D(2,1)=DUM*PR
      D(2,2)=DUM
      D(3,3)=YOC/2.D0/(1.D0+PR)
      RETURN
      ENDIF
C
C....PRINCIPAL DIRECTION 1
C
      CALL DSTRESS(EPSN(1),EPSN(2),USC,YOC,YSC,YFC,UFC,
     *      D(1,1),D(1,2),ISYM,
     *      UST,YOT,YST,YFT,UFT,US,YS,RS,
     *      AS,DS,AG,SST1(1),
     *    PMAX(1),PMAX(9),PMAX(17),PMAX(25),AGP,
     *    PMAX(2),AGE,REP,IREN)
      IF(D(1,1).LE.0.D0) THEN
      D(1,1)=YOC/1.D3
      ENDIF
C
C....PRINCIPAL DIRECTION 2
C
      CALL DSTRESS(EPSN(2),EPSN(1),USC,YOC,YSC,YFC,UFC,
     *      D(2,2),D(2,1),ISYM,
     *      UST,YOT,YST,YFT,UFT,US,YS,RS,
     *      AS,DS,AG1,SST1(2),
     *    PMAX(1),PMAX(9),PMAX(17),PMAX(25),AGP,
     *    PMAX(2),AGE1,REP,IREN)
      IF(D(2,2).LE.0.D0) THEN
      D(2,2)=YOC/1.D3
      ENDIF
C
C....SHEAR MODULUS
C
  700 IF(EPSN(1).EQ.0.D0.AND.EPSN(2).EQ.0.D0) THEN
      D(3,3)=YOC/2.D0
      ELSE
```

```
            IF(AG.EQ.AGE) THEN
                D(3,3)=(SST1(1)-SST1(2))
                D(3,3)=D(3,3)/(2.D0*(EPSN(1)-EPSN(2)))
            ELSE
                  DTH=AGE-AG
                ADUM=0.5D0*(SST1(1)-SST1(2))*DCOS(2.D0*DTH)
      BDUM=(EPSN(1)-EPSN(2))*DCOS(2.D0*DTH)+EPSN(3)*DSIN(2.D0*DTH)
                  D(3,3)=ADUM/BDUM
            ENDIF
      ENDIF
      IF(D(3,3).LE.0.D0) THEN
      D(3,3)=YOC/1.D3
      ENDIF
      IF(D(3,3).GT.YOC*1.D2) THEN
      D(3,3)=YOC*1.D2
      ENDIF
C
C.....TRANSFORMATION MATRIX
C
   20 TP(1,1)=DCOS(AG)*DCOS(AG)
      TP(1,2)=DSIN(AG)*DSIN(AG)
      TP(1,3)=DSIN(AG)*DCOS(AG)
      TP(2,1)=TP(1,2)
      TP(2,2)=TP(1,1)
      TP(2,3)=-TP(1,3)
      TP(3,1)=-2.D0*TP(1,3)
      TP(3,2)=2.D0*TP(1,3)
      TP(3,3)=TP(1,1)-TP(1,2)
      DO 100 II=1,3
      DO 100 JJ=1,3
      TT(II,JJ)=0.D0
      DO 100 IJ=1,3
  100 TT(II,JJ)=TT(II,JJ)+D(II,IJ)*TP(IJ,JJ)
      DO 200 II=1,3
      DO 200 JJ=1,3
      D(II,JJ)=0.D0
      DO 200 IJ=1,3
  200 D(II,JJ)=D(II,JJ)+TP(IJ,II)*TT(IJ,JJ)
      RETURN
      END

      SUBROUTINE DSTRESS(EP,EPSN2,USC,YOC,YSC,YFC,UFC,
     *        D1,D2,ISYM,UST,YOT,YST,YFT,UFT,US,YS,RS,AS,DS,
     *        AG,SST1,CMST,TMST,RMST,HS,AGP,CRRN,AGE,REP,IREN)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /ITRN/ JST,IST
      COMMON /CL/ ISOL,ISP
      DIMENSION US(6),YS(6),RS(6),AS(6),DS(6)
      DIMENSION PMAX(6),PMAX2(6),ES(30),STR(30),STIF(30)
      DIMENSION TMST(8),RMST(8),HS(4),CRRN(4)
      DIMENSION REP(6),IREN(6)
      PAI=2.D0*DASIN(1.D0)
      IF(AG.GT.0.D0) AG1=AG-PAI/2.D0
      IF(AG.LE.0.D0) AG1=AG+PAI/2.D0
      ESC=DABS(USC/YSC)
      EST=DABS(UST/YST)
      CALL AGPICK(AG,KK,TH)
      CALL AGPICK(AG1,KK1,TH1)
      CALL REFPICK(AG,CMST,RMST,ESC,KK,TH,ECT)
```

```
 CALL REFPICK(AG1,CMST,RMST,ESC,KK1,TH1,ECT1)
 CALL FINMAX(AG,TMST,TM,KK,TH)
 CALL FINMAX(AG1,TMST,TM1,KK1,TH1)
 CM=CMST
 IF((EPSN2-ECT1).GT.TM1) THEN
 EPSM=(EPSN2-ECT1)
 ELSE
 EPSM=TM1
 ENDIF
 IF((EPSN2-ECT1).LT.0.D0) EPSM=TM1
       DUM=DABS(CM/ESC)
       IF(DUM.LE.3.D0) THEN
       ESP=-ESC*(.145D0*DUM*DUM+.13D0*DUM)
       ELSE
       ESP=CM+(3.D0*ESC-1.695D0*ESC)
       ENDIF
       DUM=TM/0.9D0/EST
       IF(DUM.LE.1.D0) THEN
       REFS=0.D0
       ELSE
       REFS=-UFC*(DUM-1.D0)/2.D0/2.D0
       ENDIF
       IF(REFS.LT.-UFC/2.D0) REFS=-UFC/2.D0
       REFS=0.D0
 IF(ISOL.EQ.1) THEN
 IF(EP.GE.ECT) THEN
       IF((EP-ECT).GE.TM) THEN
 CALL DTENS((EP-ECT),UST,YOT,YST,YFT,UFT,AG,D1,ISYM,
*     SST1,CRRN,US,YS,RS,AS,DS,TM,AGE,REP,IREN)
       ELSE
 CALL DTENS(TM,UST,YOT,YST,YFT,UFT,AG,PD1,ISYM,PSST1,
*           CRRN,US,YS,RS,AS,DS,TM,AGE,REP,IREN)
       D1=(PSST1-REFS)/TM
       SST1=PSST1-D1*(TM-EP+ECT)
       ENDIF
 ELSEIF(EP.LT.ECT) THEN
       IF(EP.LE.CM) THEN
 CALL DCOMP(EP,EPSM,USC,YOC,YSC,YFC,UFC,
*           D1,D2,ISYM,SST1)
       ELSE
 CALL DCOMP(CM,EPSM,USC,YOC,YSC,YFC,UFC,
*           PD1,D2,ISYM, PSST1)
       UFCC=PSST1/5.D0
       D1=PSST1/(CM-ESP)
       REFE1=(UFCC-PSST1)/D1+CM
             IF(EP.GT.CM.AND.EP.LE.REFE1) THEN
             SST1=PSST1-D1*(CM-EP)
             ELSEIF(EP.GT.REFE1.AND.EP.LE.ECT) THEN
             D1=(UFCC-REFS)/(REFE1-ECT)
             SST1=REFS+D1*(EP-ECT)
             ENDIF
       IF(SST1.GT.REFS) THEN
       SST1=REFS
       D1=0.D0
       ENDIF
       ENDIF
 ENDIF
 ELSE
 CALL STPOS(EP,EPSN2,EPSM,USC,YOC,YSC,YFC,UFC,ISYM,
```

```
*       UST,YOT,YST,YFT,UFT,US,YS,RS,AS,DS,AG,ECT,
*       IMODE,ES,STR,STIF,ESP,CM,TM,HS,AGP,REFS,CRRN,
*       AGE,REP,IREN)
 IF(ES(1).EQ.0.D0.AND.ES(15).EQ.0.D0) THEN
       IF(EP.GE.0.D0) THEN
 CALL DTENS(EP,UST,YOT,YST,YFT,UFT,AG,D1,ISYM,SST1,
*             CRRN,US,YS,RS,AS,DS,TM,AGE,REP,IREN)
       ELSE
 CALL DCOMP(EP,EPSM,USC,YOC,YSC,YFC,UFC,
*            D1,D2,ISYM,SST1)
       ENDIF
 RETURN
 ENDIF
 IF(EP.GE.ES(15)) THEN
 CALL DTENS((EP-ECT),UST,YOT,YST,YFT,UFT,AG,D1,ISYM,
*            SST1,CRRN,US,YS,RS,AS,DS,TM,AGE,REP,IREN)
 ELSEIF(EP.LT.ES(15).AND.EP.GE.ES(6)) THEN
       JMODE=2
 CALL CMODE(EP,D1,SST1,ES,STR,STIF,JMODE)
 ELSEIF(EP.LT.ES(6).AND.EP.GT.ES(2)) THEN
       IF(EP.GE.ES(10)) THEN
       JMODE=2
 CALL CMODE(EP,D1,SST1,ES,STR,STIF,JMODE)
       ELSEIF(EP.LE.ES(9)) THEN
       JMODE=1
 CALL CMODE(EP,D1,SST1,ES,STR,STIF,JMODE)
       ELSEIF(EP.GT.ES(9).AND.EP.LT.ES(10)) THEN
 CALL TLINE(EP,ES(10),ES(9),STR(10),STR(9),D1,SST1)
       ENDIF
 ELSEIF(EP.LE.ES(2).AND.EP.GT.ES(1)) THEN
       JMODE=1
 CALL CMODE(EP,D1,SST1,ES,STR,STIF,JMODE)
 ELSEIF(EP.LE.ES(1)) THEN
 CALL DCOMP(EP,EPSM,USC,YOC,YSC,YFC,UFC,
*            D1,D2,ISYM,SST1)
 ENDIF
       IF(EP.LT.ECT.AND.SST1.GT.REFS) THEN
       SST1=REFS
       D1=0.D0
       ENDIF
 ENDIF
 RETURN
 END


 SUBROUTINE CMODE(EP,D1,SST1,ES,STR,STIF,JMODE)
 IMPLICIT REAL*8(A-H,O-Z)
 COMMON /CNTL2/ EEFT,FFT,TOL
 DIMENSION ES(30),STR(30),STIF(30)
 IF(JMODE.EQ.1) THEN
       IF(EP.LE.ES(15).AND.EP.GT.ES(6)) THEN
 CALL TLINE(EP,ES(15),ES(6),STR(15),STR(6),D1,SST1)
       ELSEIF(EP.LE.ES(6).AND.EP.GT.ES(5)) THEN
 CALL TLINE(EP,ES(6),ES(5),STR(6),STR(5),D1,SST1)
       ELSEIF(EP.LE.ES(5).AND.EP.GT.ES(2)) THEN
 CALL TLINE(EP,ES(5),ES(2),STR(5),STR(2),D1,SST1)
       ELSEIF(EP.LE.ES(2).AND.EP.GE.ES(1)) THEN
 CALL TLINE(EP,ES(2),ES(1),STR(2),STR(1),D1,SST1)
       ENDIF
```

```
 ELSEIF(JMODE.EQ.2) THEN
        IF(EP.LE.ES(15).AND.EP.GT.ES(6)) THEN
 CALL TLINE(EP,ES(15),ES(6),STR(15),STR(6),D1,SST1)
        ELSEIF(EP.LE.ES(6).AND.EP.GT.ES(5)) THEN
 CALL TLINE(EP,ES(6),ES(5),STR(6),STR(5),D1,SST1)
        ELSEIF(EP.LE.ES(5).AND.EP.GT.ES(4)) THEN
 CALL TLINE(EP,ES(5),ES(4),STR(5),STR(4),D1,SST1)
        ELSEIF(EP.LE.ES(4).AND.EP.GT.ES(3)) THEN
 CALL TLINE(EP,ES(4),ES(3),STR(4),STR(3),D1,SST1)
        ELSEIF(EP.LE.ES(3).AND.EP.GT.ES(2)) THEN
 CALL TLINE(EP,ES(3),ES(2),STR(3),STR(2),D1,SST1)
        ELSEIF(EP.LE.ES(2).AND.EP.GE.ES(1)) THEN
 CALL TLINE(EP,ES(2),ES(1),STR(2),STR(1),D1,SST1)
        ENDIF
 ENDIF
 RETURN
 END

 SUBROUTINE STPOS(P,P2,EPSM,USC,YOC,YSC,YFC,UFC,
*          ISYM,
*        UST,YOT,YST,YFT,UFT,US,YS,RS,AS,DS,
*     AG,ECT,IMODE,ES,STR,STIF,ESP,CM,TM,HS,AGP,REFS,CRRN,
*     AGE,REP,IREN)
 IMPLICIT REAL*8 (A-H,O-Z)
 COMMON /CNTL2/ EEFT,FFT,TOL
 DIMENSION US(6),YS(6),RS(6),AS(6),DS(6)
 DIMENSION ES(30),STR(30),STIF(30)
 DIMENSION HS(4),CRRN(4),REP(6),IREN(6)
 ESC=DABS(USC/YSC)
 EST=DABS(UST/YST)
 EFT=EEFT
 EFC=DABS(UFC/YFC)
 IF(DABS(TM).LE.0.9D0*EST.AND.DABS(CM).LE.0.9D0*EST) THEN
 ES(1)=0.D0
 ES(15)=0.D0
 ENDIF
 IF(DABS(CM).GE.0.9D0*EST) THEN
        IF(CM.LT.P2) THEN
        ES(1)=CM
        ELSE
        ES(1)=P2
        ENDIF
 CALL DCOMP(ES(1),EPSM,USC,YOC,YSC,YFC,UFC,
*             STIF(1),D2,ISYM,STR(1))
        STR(2)=0.85D0*STR(1)
        ES(2)=(STR(2)-STR(1))/YOC+ES(1)
        STR(3)=0.5D0*STR(1)
        ES(3)=(STR(3)-STR(1))/YOC+ES(1)
        STR(5)=0.D0
        ES(5)=ESP
        STR(4)=0.D0
        ES(4)=(-STR(3))*(ES(2)-ESP)/STR(2)+ES(3)
        ES(6)=ECT
        STR(6)=0.D0
 ELSE
        STR(6)=0.D0
        ES(6)=0.D0
 DO 100 I=1,5
 STR(I)=STR(6)
```

```
100 ES(I)=ES(6)
    ENDIF
    IF(HS(1).NE.0.D0) THEN
          ES(9)=HS(1)
          ES(10)=ES(9)+ESC/8.D0
    ELSEIF(HS(2).NE.0.D0) THEN
          ES(10)=HS(2)
          ES(9)=ES(10)-ESC/8.D0
    ENDIF
    IF(ES(9).LT.ES(2)) ES(9)=ES(2)
    IF(ES(10).GT.ES(5)) ES(10)=ES(5)
          JMODE=1
    CALL CMODE(ES(9),STIF(9),STR(9),ES,STR,STIF,JMODE)
          JMODE=2
    CALL CMODE(ES(10),STIF(10),STR(10),ES,STR,STIF,JMODE)
    IF(DABS(TM).GE.0.9D0*EST) THEN
          ES(15)=TM+ECT
    CALL DTENS((ES(15)-ECT),UST,YOT,YST,YFT,UFT,
   *      AG,STIF(15),ISYM,STR(15),CRRN,US,YS,RS,AS,DS,TM,AGE,
   *    REP,IREN)
          STR(14)=0.D0
          ES(14)=ECT
    ELSE
    ES(6)=ES(5)
    ES(14)=ES(5)
    ES(15)=ES(5)
    STR(6)=STR(5)
    STR(14)=STR(5)
    STR(15)=STR(5)
    ENDIF
    RETURN
    END

    SUBROUTINE DCOMP(EPS1,EPS2,USC,YOC,YSC,YFC,UFC,
   *           D1,D2,ISYM,ST1)
    IMPLICIT REAL*8 (A-H,O-Z)
    COMMON /ITRN/ JST,IST
    STE=1.D0
    DUM=DABS(EPS1)
    ESC=DABS(USC/YSC)
    EFC=DABS(UFC/YFC)
    IF(EPS2.GT.0.D0) THEN
    STE=.8D0+.34D0*EPS2/ESC
    IF(STE.LT.1.D0) STE=1.D0
    IF(STE.GT.USC/UFC) STE=USC/UFC
    ENDIF
    USCC=USC/STE
    YSCC=USCC/ESC
    CALL TENFF(DUM,USCC,YOC,YSCC,YFC,UFC,ESC,EFC,D1,ST1)
    ST1=-ST1
    RETURN
    END

    SUBROUTINE TLINE(EP,ES1,ES2,STR1,STR2,D1,SST1)
    IMPLICIT REAL*8 (A-H,O-Z)
    IF(ES2.EQ.ES1) THEN
    D1=0.D0
    SST1=STR1
    RETURN
```

```
      ENDIF
      D1=(STR2-STR1)/(ES2-ES1)
      SST1=D1*(EP-ES1)+STR1
      RETURN
      END

      SUBROUTINE DTENS(EPS1,UST,YOT,YST,YFT,UFT,AG,D1,ISYM,
     *             ST1,CRRN,US,YS,RS,AS,DS,TM,AGE,REP,IREN)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION CRRN(4),REP(6),IREN(6)
      DIMENSION US(6),YS(6),RS(6),AS(6),DS(6)
      COMMON /CNTL2/ EEFT,FFT,TOL
      COMMON /DIMS/ IIDUM(14),IARC,IBAU,ISTL
      PAI=2.D0*DASIN(1.D0)
      ALAM=0.01D0
      EFT=EEFT
      UFT=ALAM*UST
      EST=(UST/YST)
      CALL TENFF(EPS1,UST,YOT,YST,YFT,UFT,EST,EFT,D1,ST1)
      IF(EPS1.LT.EST) RETURN
      DO 300 I=1,6
      IF(IREN(I).NE.0.AND.REP(I).GT.0.D0) THEN
      EFTE=US(I)/YS(I)
      CALL TENFF(REP(I),UST,YOT,YST,YFT,UFT,EST,EFTE,D1E,ST1E)
      DAGM=DABS(AG-AS(I))
      IF(DAGM.GT.PAI/2.D0) DAGM=PAI-DAGM
      ST1E=ST1E*(DCOS(DAGM)**.5)
      D1E=D1E*(DCOS(DAGM)**.5)
      ENDIF
      IF(ST1.LT.ST1E) THEN
      ST1=ST1E
      D1=D1E
      ENDIF
  300 CONTINUE
      RETURN
      END

      SUBROUTINE DMATS1(D,REP,EMAX,CONSTM,NCM,NMAT,IREN,RST,
     *                   UST,YST,IBAU,EPS,CMST,USC,YSC,UFC,YFC)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /ITRN/ JST,IST
      COMMON /CNTL2/ EEFT,FFT,TOL
      DIMENSION CONSTM(NCM),EMAX(6),REPP(3),EPS(3)
      DIMENSION D(3,3),TP(3,3),TT(3,3),SG(4),CL(4)
      ESC=USC/YSC
      EFC=UFC/YFC
      IF(IREN.NE.0) THEN
      US=DABS(CONSTM(1))
      YS=DABS(CONSTM(2))
      RS=DABS(CONSTM(3))
      AS=(CONSTM(4))
      ALFA=DABS(CONSTM(5))
      ESH=DABS(CONSTM(7))
      EUT=DABS(CONSTM(8))
      SUT=DABS(CONSTM(9))
      ELSE
      US=0.D0
      YS=0.D0
      RS=0.D0
```

```
      AS=0.D0
      ALFA=0.D0
      ESH=0.D0
      EUT=0.D0
      SUT=0.D0
      ENDIF
      PAI=2.D0*DASIN(1.D0)
      AS=AS*PAI/180.D0
      DO 10 I=1,3
      DO 10 J=1,3
   10 D(I,J)=0.D0
      IF(IREN.LT.1) RETURN
      IF(IBAU.EQ.1) THEN
      CALL DDMAT1(REP,US,YS,AS,DD,RRT,EMAX)
      ELSEIF(IBAU.EQ.2) THEN
      CALL DDMAT2(REP,US,YS,AS,DD,RRT,EMAX,ESH,EUT,SUT)
      ENDIF
      D(1,1)=DD*RS
      TP(1,1)=DCOS(AS)*DCOS(AS)
      TP(1,2)=DSIN(AS)*DSIN(AS)
      TP(1,3)=DSIN(AS)*DCOS(AS)
      TP(2,1)=TP(1,2)
      TP(2,2)=TP(1,1)
      TP(2,3)=-TP(1,3)
      TP(3,1)=-2.D0*TP(1,3)
      TP(3,2)=2.D0*TP(1,3)
      TP(3,3)=TP(1,1)-TP(1,2)
      DO 100 II=1,3
      DO 100 JJ=1,3
      TT(II,JJ)=0.D0
      DO 100 IJ=1,3
  100 TT(II,JJ)=TT(II,JJ)+D(II,IJ)*TP(IJ,JJ)
      DO 200 II=1,3
      DO 200 JJ=1,3
      D(II,JJ)=0.D0
      DO 200 IJ=1,3
  200 D(II,JJ)=D(II,JJ)+TP(IJ,II)*TT(IJ,JJ)
      RETURN
      END

      SUBROUTINE DDMAT1(REP,US,YS,AS,DD,RRT,EMAX)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CNTL2/ EEFT,FFT,TOL
      COMMON /ITRN/ JST,IST
      DIMENSION EMAX(6),ES(6),STR(6),STIF(6)
      EY=US/YS
      IF(JST.EQ.0) THEN
      DD=YS
      RRT=YS*REP
      RETURN
      ENDIF
      CALL SSTPOS(EMAX,ES,STR,STIF,US,YS)
      IF(REP.GE.EMAX(1)) THEN
      DD=YS/FFT
      RRT=DD*(REP-EY)+US
      ELSEIF(REP.LE.EMAX(6)) THEN
      DD=YS/FFT
      RRT=DD*(REP+EY)-US
      ELSEIF(REP.LT.EMAX(1).AND.REP.GT.EMAX(6)) THEN
```

```
      CALL TLINE(REP,ES(1),ES(6),STR(1),STR(6),DD,RRT)
      ENDIF
      RETURN
      END

      SUBROUTINE DDMAT2(REP,US,YS,AS,DD,RRT,EMAX,ESH,EUT,SUT)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CNTL2/ EEFT,FFT,TOL
      COMMON /ITRN/ JST,IST
      DIMENSION EMAX(6)
      EY=US/YS
      IF(JST.EQ.0) THEN
      DD=YS
      RRT=YS*REP
      RETURN
      ENDIF
      IF(EMAX(1).EQ.0.D0.AND.EMAX(6).EQ.0.D0) THEN
             IF(DABS(REP).LE.EY) THEN
             DD=YS
             RRT=YS*DABS(REP)
             ELSEIF(DABS(REP).GT.EY.AND.DABS(REP).LE.ESH) THEN
             RRT=US
             DD=YS/1.D3
             ELSEIF(DABS(REP).GT.ESH) THEN
                    ESS=DABS(REP)-ESH
      ADUM=(60.D0*ESS+2.D0)
      BDUM=(112.D0*ESS+2.D0)
      RRT=BDUM/ADUM+ESS*(SUT/US-1.7D0)/(EUT-ESH)
      RRT=RRT*US
      DD=(112.D0*ADUM-60.D0*BDUM)/ADUM/ADUM
      DD=DD+(SUT/US-1.7D0)/(EUT-ESH)
      DD=DD*US
             ENDIF
             IF(REP.LT.0.D0) RRT=-RRT
      RETURN
      ENDIF
      IF(EMAX(1).NE.0.D0)  THEN
      EM=EMAX(1)
      EP1=EMAX(4)
      EP2=EMAX(2)
      EP3=EMAX(5)
      EPC=(EM-EMAX(5))*0.8D0+EMAX(5)
      IF(EPC.LT.EP2) EPC=EP2
      CALL CUVSOL(EM,US,YS,DD,RRT1,ESH,EUT,SUT,
     *              EP1,EP2)
      DDUM=DABS(RRT1/(EM-EPC))
      IF(DDUM.GT.YS) EPC=EM-RRT1/YS
      ELSEIF(EMAX(6).NE.0.D0)  THEN
      EM=EMAX(6)
      EP1=EMAX(5)
      EP2=EMAX(2)
      EP3=EMAX(4)
      EPC=(EM-EMAX(4))*0.8D0+EMAX(4)
      IF(EPC.GT.EP2) EPC=EP2
      CALL CUVSOL(EM,US,YS,DD,RRT1,ESH,EUT,SUT,
     *              EP1,EP2)
      DDUM=DABS(RRT1/(EM-EPC))
      IF(DDUM.GT.YS) EPC=EM+RRT1/YS
      ENDIF
```

```
      IF(DABS(REP-EPC).GT.DABS(EM-EPC).AND.
     *        (REP-EPC)*(EM-EPC).GT.0.D0) THEN

      CALL CUVSOL(REP,US,YS,DD,RRT,ESH,EUT,SUT,
     *             EP1,EP2)
      ELSEIF(DABS(REP-EPC).LT.DABS(EM-EPC).AND.
     *           (REP-EPC)*(EM-EPC).GT.0.D0) THEN
      CALL CUVSOL(EM,US,YS,DD,RRT1,ESH,EUT,SUT,
     *             EP1,EP2)
      DD=DABS(RRT1/(EM-EPC))
      RRT=RRT1-DABS(DD*(EM-REP))
      ELSE
      CALL CUVSOL(REP,US,YS,DD,RRT,ESH,EUT,SUT,
     *             EP3,EPC)
      ENDIF
      IF(REP.LT.EPC) RRT=-RRT
      RETURN
      END

      SUBROUTINE CUVSOL(REP,US,YS,DD,RRT,ESH,EUT,SUT,
     *      EP1,EP2)
      IMPLICIT REAL*8(A-H,O-Z)
      EY=US/YS
      IF(EP1.EQ.0.D0.AND.EP2.EQ.0.D0) THEN
               IF(DABS(REP).LT.ESH) THEN
               RRT=US
               DD=YS/1.D3
               ELSE
               ESS=DABS(REP)-ESH
      ADUM=(60.D0*ESS+2.D0)
      BDUM=(112.D0*ESS+2.D0)
      RRT=BDUM/ADUM+ESS*(SUT/US-1.7D0)/(EUT-ESH)
      RRT=RRT*US
      DD=(112.D0*ADUM-60.D0*BDUM)/ADUM/ADUM
      DD=DD+(SUT/US-1.7D0)/(EUT-ESH)
      DD=DD*US
               ENDIF
      ELSE
      EIP=DABS(EP2-EP1)
      ESHD=ESH*DLOG(0.5d0*EIP/EY)/1.38D0
      IF(ESHD.LT.0.3D0*ESH) ESHD=ESH*0.3D0
      ESD=DABS(REP-EP2)

      RRT=US*(1.D0-DEXP(-2.05D0*ESD/ESHD)+0.129D0*ESD/ESHD)
      DD=US*(2.05D0*DEXP(-2.05D0*ESD/ESHD)/ESHD+0.129D0/ESHD)
      IF(RRT.LT.US) RETURN
      EM1=0.D0
      EM2=ESD
  100 EMM=(EM1+EM2)/2.D0
      RRT=US*(1.D0-DEXP(-2.05D0*EMM/ESHD)+0.129D0*EMM/ESHD)
      IF(DABS(1.D0-RRT/US).GT.1.D-2) THEN
      IF(RRT.GT.US) THEN
      EM2=EMM
      ELSE
      EM1=EMM
      ENDIF
      GOTO 100
      ENDIF
      ESS=ESD-EMM
```

```
      ADUM=(60.D0*ESS+2.D0)
      BDUM=(112.D0*ESS+2.D0)
      RRT=BDUM/ADUM+ESS*(SUT/US-1.7D0)/(EUT-ESH)
      RRT=RRT*US
      DD=(112.D0*ADUM-60.D0*BDUM)/ADUM/ADUM
      DD=DD+(SUT/US-1.7D0)/(EUT-ESH)
      DD=DD*US
      ENDIF
      RETURN
      END

      SUBROUTINE SSTPOS(EMAX,ES,STR,STIF,US,YS)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CNTL2/ EEFT,FFT,TOL
      DIMENSION EMAX(6),ES(6),STR(6),STIF(6)
      EY=US/YS
      IF(EMAX(1).EQ.0.D0.AND.EMAX(6).EQ.0.D0) THEN
      ES(2)=EY
      EMAX(1)=EY
      ES(1)=EMAX(1)
      ES(3)=EY
      ES(5)=-EY
      EMAX(6)=-EY
      ES(6)=EMAX(6)
      ES(4)=-EY
      ELSE
      ES(1)=EMAX(1)
      ES(2)=ES(1)
      ES(3)=ES(2)
      ES(6)=EMAX(6)
      ES(5)=ES(6)
      ES(4)=ES(5)
      ENDIF
      STR(1)=YS/FFT*(ES(1)-EY)+US
      STIF(1)=YS/FFT
      STR(2)=YS/FFT*(ES(2)-EY)+US
      STIF(2)=YS/FFT
      STR(6)=YS/FFT*(ES(6)+EY)-US
      STIF(6)=YS/FFT
      STR(5)=YS/FFT*(ES(5)+EY)-US
      STIF(5)=YS/FFT
      RETURN
      END

      SUBROUTINE STMAT(ST1,AG,EPSN,CONSTM1,NCM,P,PMAX,NMAT1,
     *                 NMAT2,IELM,CONSTM2,IE,KK,AGP,AGE,REP)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CL/ ISOL,ISP
      COMMON /CNTL/ ISYM,IDUM(28)
      COMMON /ITRN/ JST,IST
      DIMENSION CONSTM1(NCM,NMAT1),CONSTM2(NCM,NMAT2)
      DIMENSION ST1(3),SST1(3),EPSN(3),TP(3,3),TT(3,3),P(2)
      DIMENSION US(6),YS(6),RS(6),AS(6),DS(6),D(3,3)
      DIMENSION PMAX(28),IREN(6),REP(6)
      PAI=2.D0*DASIN(1.D0)
      CALL SCONS(CONSTM1,CONSTM2,YM,PR,THIC,NINT,UWT,
     *     USC,YOC,YSC,YFC,UFC,UST,YOT,YST,YFT,UFT,
     *     IREN,US,YS,RS,AS,DS,NCM,IELM,NMAT1,NMAT2)
      IF(AG.GT.0.D0) AG1=AG-PAI/2.D0
```

```
      IF(AG.LE.0.D0) AG1=AG+PAI/2.D0
      IF(AGE.GT.0.D0) AGE1=AGE-PAI/2.D0
      IF(AGE.LE.0.D0) AGE1=AGE+PAI/2.D0
      DO 20 I=1,3
      ST1(I)=0.D0
      SST1(I)=0.D0
   20 CONTINUE
      IF(JST.EQ.0) THEN
      DO 10 I=1,3
      DO 10 J=1,3
   10 D(I,J)=0.D0
C
C.....INITIAL STIFFNESS
C
      DUM=YOC/(1.D0-PR*PR)
      D(1,1)=DUM
      D(1,2)=DUM*PR
      D(2,1)=DUM*PR
      D(2,2)=DUM
      D(3,3)=YOC/2.D0/(1.D0+PR)
      DO 15 I=1,3
      DO 15 J=1,3
      SST1(I)=SST1(I)+D(I,J)*EPSN(J)
   15 CONTINUE
      GOTO 200
      ENDIF
C
C....PRINCIPAL AXIS 1
C
      CALL DSTRESS(EPSN(1),EPSN(2),USC,YOC,YSC,YFC,UFC,
     *       D(1,1),D(1,2),ISYM,UST,YOT,YST,YFT,UFT,US,YS,RS,
     *       AS,DS,AG,SST1(1),PMAX(1),PMAX(9),PMAX(17),
     *       PMAX(25),AGP,PMAX(2),AGE,REP,IREN)
C
C....PRINCIPAL AXIS 2
C
      CALL DSTRESS(EPSN(2),EPSN(1),USC,YOC,YSC,YFC,UFC,
     *       D(2,2),D(2,1),ISYM,UST,YOT,YST,YFT,UFT,US,YS,RS,
     *       AS,DS,AG1,SST1(2),PMAX(1),PMAX(9),PMAX(17),
     *       PMAX(25),AGP,PMAX(2),AGE1,REP,IREN)
C
C.....TRANSFORMATION MATRIX
C
  200 TP(1,1)=DCOS(AG)*DCOS(AG)
      TP(1,2)=DSIN(AG)*DSIN(AG)
      TP(1,3)=-2.D0*DSIN(AG)*DCOS(AG)
      TP(2,1)=TP(1,2)
      TP(2,2)=TP(1,1)
      TP(2,3)=-TP(1,3)
      TP(3,1)=DSIN(AG)*DCOS(AG)
      TP(3,2)=-TP(3,1)
      TP(3,3)=TP(1,1)-TP(1,2)
      DO 100 II=1,3
      DO 100 JJ=1,3
      ST1(II)=ST1(II)+TP(II,JJ)*SST1(JJ)
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TENFF(EPS1,UST,YOT,YST,YFT,UFT,EST,EFT,
     *           D1,ST1)
      IMPLICIT REAL*8 (A-H,O-Z)
      ESFT=EST+(EFT-EST)/5.D0
      IF(EPS1.LE.EST) THEN
             DUM=EPS1/EST
             ST1=UST*(2.D0*DUM-DUM*DUM)
             D1=UST*(2.D0/EST-2.D0*DUM/EST)
      ELSEIF(EPS1.GT.EST.AND.EPS1.LE.EFT) THEN
             D1=(UFT-UST)/(EFT-EST)
             ST1=D1*(EPS1-EST)+UST
      ELSEIF(EPS1.GT.EFT) THEN
             D1=0.D0
             ST1=UFT
      ENDIF
      RETURN
      END

      SUBROUTINE STMATS1(RRST,REP,EMAX,CONSTM,NCM,
     *   RST,NMAT,IREN,UST,YST,IBAU,EPS,CMST,USC,YSC,UFC,YFC)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /ITRN/ JST,IST
      COMMON /CNTL2/ EEFT,FFT,TOL
      DIMENSION CONSTM(NCM),EMAX(6),EPS(3)
      DIMENSION RRST(3),TP(3,3),TT(3,3),SG(4),CL(4)
      ESC=USC/YSC
      EFC=UFC/YFC
      IF(IREN.NE.0) THEN
      US=DABS(CONSTM(1))
      YS=DABS(CONSTM(2))
      RS=DABS(CONSTM(3))
      AS=(CONSTM(4))
      ALFA=DABS(CONSTM(5))
      ESH=DABS(CONSTM(7))
      EUT=DABS(CONSTM(8))
      SUT=DABS(CONSTM(9))
      ELSE
      US=0.D0
      YS=0.D0
      RS=0.D0
      AS=0.D0
      ALFA=0.D0
      ESH=0.D0
      EUT=0.D0
      SUT=0.D0
      ENDIF
      PAI=2.D0*DASIN(1.D0)
      AS=AS*PAI/180.D0
      DO 10 I=1,3
      RRST(I)=0.D0
   10 CONTINUE
      IF(IREN.LT.1) RETURN
      IF(IBAU.EQ.1) THEN
      CALL DDMAT1(REP,US,YS,AS,DD,RRT,EMAX)
      ELSEIF(IBAU.EQ.2) THEN
      CALL DDMAT2(REP,US,YS,AS,DD,RRT,EMAX,ESH,EUT,SUT)
      ENDIF
      RST=RRT
      RRT=RRT*RS
```

```
   20 TP(1,1)=DCOS(AS)*DCOS(AS)
      TP(1,2)=DSIN(AS)*DSIN(AS)
      TP(1,3)=-2.D0*DSIN(AS)*DCOS(AS)
      TP(2,1)=TP(1,2)
      TP(2,2)=TP(1,1)
      TP(2,3)=-TP(1,3)
      TP(3,1)=DSIN(AS)*DCOS(AS)
      TP(3,2)=-TP(3,1)
      TP(3,3)=TP(1,1)-TP(1,2)
      DO 100 II=1,3
      RRST(II)=RRST(II)+TP(II,1)*RRT
  100 CONTINUE
      RETURN
      END

      SUBROUTINE UPQD4(XX,CONSTM1,ST1,AGP,EMAX,RST1,
     *                      RST2,NCM,EU,IE,EEP,EEU,PMAX,
     *           NMAT1,NMAT2,IELM,ICOMP,ELRHS,
     *           CONSTM2,NGAU,MNDOFN,MNNE,MNDOFE,NDIM,
     *    SHST,IARC,IBAU,ISTL)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTL/ ISYM,IIDUM(28)
      COMMON /ITRN/ JST,IST
      COMMON /XGWGT/ XG(4,4),WGT(4,4)
      DIMENSION CONSTM1(NCM,NMAT1),EU(MNDOFE),EEP(MNDOFE)
      DIMENSION EEU(MNDOFE),ELRHS(MNDOFE)
      DIMENSION CONSTM2(NCM,NMAT2),DB(4)
      DIMENSION D(4,4),B(4,16),XX(NDIM,MNNE),S(16,16)
      DIMENSION EPSN(3),SIGM(3),ST1(3,NGAU*NGAU)
      DIMENSION D1(3,3),D2(3,3),AGP(NGAU*NGAU)
      DIMENSION EMAX(2*6,NGAU*NGAU)
      DIMENSION RRST1(3),RRST2(3),EPS(3)
      DIMENSION RST1(NGAU*NGAU),RST2(NGAU*NGAU)
      DIMENSION P(2),TST1(3),DST1(3)
      DIMENSION PMAX(28,NGAU*NGAU),IREN(6)
      DIMENSION DD(9,30),H(8)
      DIMENSION US(6),YS(6),RS(6),AS(6),DS(6)
      DIMENSION REP(6),REPP(6)
      PAI=2.D0*DASIN(1.D0)
      CALL SCONS(CONSTM1,CONSTM2,YM,PR,THIC,NINT,UWT,
     *       USC,YOC,YSC,YFC,UFC,UST,YOT,YST,YFT,UFT,
     *       IREN,US,YS,RS,AS,DS,NCM,IELM,NMAT1,NMAT2)
      NINT=NGAU
      ITYPE=2
      SHST=0.D0
      DO 500 I=1,MNDOFE
      EEP(I)=0.D0
      ELRHS(I)=0.D0
  500 CONTINUE
      KK=0
      DO 1830 II=1,NINT
      RI=XG(II,NINT)
      DO 1830 IJ=1,NINT
      KK=KK+1
      SI=XG(IJ,NINT)
      DO 1810 J=1,3
      SIGM(J)=0.0D0
      DST1(J)=0.0D0
 1810 EPSN(J)=0.0D0
```

```
      IF(ICOMP.EQ.2) THEN
      CALL STDM8(XX,B,H,DET,RI,SI,XBAR,NEL,ITYPE,NDIM,MNNE)
      ELSE
      CALL STDM4(XX,B,H,DET,RI,SI,XBAR,NEL,ITYPE,NDIM,MNNE)
      ENDIF
      IF(ITYPE.GT.0) XBAR=THIC
      WT=WGT(II,NINT)*WGT(IJ,NINT)*XBAR*DET
      DO 1815 J=2,MNDOFE,2
      JJ=J-1
      EPSN(1)=EPSN(1)+B(1,JJ)*EU(JJ)
      EPSN(2)=EPSN(2)+B(2,J )*EU(J )
      EPSN(3)=EPSN(3)+B(3,JJ)*EU(JJ)+B(3,J)*EU(J)
 1815 CONTINUE
      SHST=SHST-(EPSN(1)-EPSN(2))
C
C.....PRINCIPAL STRAIN DIRECTION
C
      CC=(EPSN(1)+EPSN(2))*0.5D0
      BB=(EPSN(1)-EPSN(2))*0.5D0
      DUM=AGP(KK)
      EPSN(3)=EPSN(3)/2.D0
      CALL PRINCIPAL(EPSN,P,AG,DUM)
      EPSN(3)=EPSN(3)*2.D0
      EPSN(1)=P(1)
      EPSN(2)=P(2)
      AGS=AG
      EPS(1)=EPSN(1)
      EPS(2)=EPSN(2)
      DO 955 I=1,6
      IF(IREN(I).NE.0) THEN
      REP(I)=CC+BB*DCOS(2.D0*AS(I))+EPSN(3)*DSIN(2.D0*AS(I))/2.D0
      REPP(I)=REP(I)
      CALL REPST(REP(I),UST,YST,USC,YSC,
     *  AS(I),PMAX(1,KK),PMAX(9,KK),PMAX(17,KK))
      ENDIF
  955 CONTINUE
      CALL STMATS1(RRST1,REPP(1),EMAX(1,KK),
     *            CONSTM1(1,IREN(1)),NCM,RST1(KK),
     *  NMAT1,IREN(1),UST,YST,IBAU,EPS,PMAX(1,KK),USC,YSC,UFC,YFC)
      CALL STMATS1(RRST2,REPP(2),EMAX(7,KK),
     *            CONSTM1(1,IREN(2)),NCM,RST2(KK),
     *  NMAT1,IREN(2),UST,YST,IBAU,EPS,PMAX(1,KK),USC,YSC,UFC,YFC)

      CALL STMAT(ST1(1,KK),AGS,EPS,CONSTM1,NCM,P,PMAX(1,KK),
     *          NMAT1,NMAT2,IELM,
     *          CONSTM2,IE,KK,AGP(KK),AG,REP)
      DO 900 I=1,MNDOFE
      DO 910 J=1,3
      DUM=ST1(J,KK)+RRST1(J)
      DUM=DUM+RRST2(J)
  910 EEP(I)=EEP(I)+B(J,I)*DUM*WT
  900 CONTINUE
      KA=0
      DO 960 I=2,MNDOFE,2
      KA=KA+1
      EEP(I)=EEP(I)-H(KA)*UWT*WT
  960 CONTINUE
 1830 CONTINUE
      TKK=KK
```

```
      SHST=-SHST/TKK
      RETURN
      END

      SUBROUTINE SCONS(CONSTM1,CONSTM2,YM,PR,THIC,NINT,UWT,
     *        USC,YOC,YSC,YFC,UFC,UST,YOT,YST,YFT,UFT,
     *        IREN,US,YS,RS,AS,DS,NCM,IELM,NMAT1,NMAT2)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION CONSTM1(NCM,NMAT1),CONSTM2(NCM,NMAT2)
      DIMENSION US(6),YS(6),RS(6),AS(6),DS(6),IREN(6)
      PAI=2.D0*DASIN(1.D0)
      YM=DABS(CONSTM2(1,IELM))
      PR=DABS(CONSTM2(2,IELM))
      THIC=DABS(CONSTM2(3,IELM))
      NINT=DABS(CONSTM2(4,IELM))
      UWT=DABS(CONSTM2(5,IELM))
      USC=DABS(CONSTM2(6,IELM))
      YOC=DABS(CONSTM2(7,IELM))
      YSC=DABS(CONSTM2(8,IELM))
      YFC=DABS(CONSTM2(9,IELM))
      UFC=DABS(CONSTM2(10,IELM))
      UST=DABS(CONSTM2(11,IELM))
      YOT=DABS(CONSTM2(12,IELM))
      YST=DABS(CONSTM2(13,IELM))
      YFT=DABS(CONSTM2(14,IELM))
      UFT=DABS(CONSTM2(15,IELM))
      IREN(1)=DABS(CONSTM2(16,IELM))
      IREN(2)=DABS(CONSTM2(17,IELM))
      IREN(3)=DABS(CONSTM2(18,IELM))
      IREN(4)=DABS(CONSTM2(19,IELM))
      IREN(5)=DABS(CONSTM2(20,IELM))
      IREN(6)=DABS(CONSTM2(21,IELM))
      DO 5 I=1,6
      IF(IREN(I).NE.0) THEN
      US(I)=DABS(CONSTM1(1,IREN(I)))
      YS(I)=DABS(CONSTM1(2,IREN(I)))
      RS(I)=DABS(CONSTM1(3,IREN(I)))
      AS(I)=(CONSTM1(4,IREN(I)))
      DS(I)=DABS(CONSTM1(5,IREN(I)))
      AS(I)=AS(I)*PAI/180.D0
      ELSE
      US(I)=0.D0
      YS(I)=0.D0
      RS(I)=0.D0
      AS(I)=0.D0
      DS(I)=0.D0
      ENDIF
    5 CONTINUE
      RETURN
      END

      SUBROUTINE EFQD4(XX,CONSTM1,ST1,AGP,EMAX,RST1,RST2,
     *            NCM,EU,IE,EEP,ASTRESS,ASTRAIN,
     *            PMAX,NMAT1,NMAT2,IELM,ICOMP,
     *            CONSTM2,NGAU,MNDOFN,MNNE,MNDOFE,NDIM,
     *            SHDUM,NEL,IARC,IBAU,ISTL)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CL/ ISOL,ISP
      COMMON /CNTL/ IDUM(29)
```

```
      COMMON /ITRN/ JST,IST
      COMMON /CNTL1/ TAB
      COMMON /CNTL2/ EEFT,FFT,TOL
      COMMON /XGWGT/ XG(4,4),WGT(4,4)
      DIMENSION CONSTM1(NCM,NMAT1),EU(MNDOFE),EEP(MNDOFN,MNNE)
      DIMENSION CONSTM2(NCM,NMAT2),DB(4)
      DIMENSION D(4,4),B(4,16),XX(NDIM,MNNE)
      DIMENSION EPSN(3),SIGM(4),ST1(3,NGAU*NGAU),EPS(3)
      DIMENSION AGP(NGAU*NGAU),EMAX(2*6,NGAU*NGAU)
      DIMENSION RST1(NGAU*NGAU),RST2(NGAU*NGAU)
      DIMENSION ASTRESS(3),ASTRAIN(3),P(2),P1(2)
      DIMENSION PMAX(28,NGAU*NGAU),IREN(6),OUP(12)
      DIMENSION US(6),YS(6),RS(6),AS(6),DS(6)
      DIMENSION REP(3),RES(2)
      DIMENSION HSS(4)
      PAI=2.D0*DASIN(1.D0)
      CALL SCONS(CONSTM1,CONSTM2,YM,PR,THIC,NINT,UWT,
     *      USC,YOC,YSC,YFC,UFC,UST,YOT,YST,YFT,UFT,
     *      IREN,US,YS,RS,AS,DS,NCM,IELM,NMAT1,NMAT2)
      ESC=DABS(USC/YSC)
      EST=DABS(UST/YST)
      NINT=NGAU
      ITYPE=2
      DO 2500 I=1,12
      OUP(I)=0.D0
 2500 CONTINUE
      SHDUM=0.D0
      KK=0
      DO 830 II=1,NINT
      DO 830 IJ=1,NINT
      KK=KK+1
      CALL BMAT4(IE,ITYPE,NINT,XX,B,II,IJ,NDIM,MNNE,
     *        ICOMP)
      DO 810 J=1,4
      SIGM(J)=0.0D0
  810 EPSN(J)=0.0D0
      DO 815 J=2,MNDOFE,2
      JJ=J-1
      EPSN(1)=EPSN(1)+B(1,JJ)*EU(JJ)
      EPSN(2)=EPSN(2)+B(2,J )*EU(J )
      EPSN(3)=EPSN(3)+B(3,JJ)*EU(JJ)+B(3,J)*EU(J)
      IF(ITYPE.GT.0) GOTO 815
      EPSN(4)=EPSN(4)+B(4,JJ)*EU(JJ)
  815 CONTINUE
  845 CONTINUE
C
C.....CALCULATE PRINCIPAL STRAINS $ DIRECTION
C
      CC=(EPSN(1)+EPSN(2))*0.5D0
      BB=(EPSN(1)-EPSN(2))*0.5D0
      DUM=AGP(KK)
      EPSN(3)=EPSN(3)/2.D0
      CALL PRINCIPAL(EPSN,P1,AG,DUM)
      EPSN(3)=EPSN(3)*2.D0
      EPSN(1)=P1(1)
      EPSN(2)=P1(2)
      AGS=AG
      EPS(1)=EPSN(1)
      EPS(2)=EPSN(2)
```

```
C
C     CALCULATE PRINCIPAL STRESSES & DIRECTIONS
C
      CC1=(ST1(1,KK)+ST1(2,KK))*0.5D0
      BB1=(ST1(1,KK)-ST1(2,KK))*0.5D0
      P(1)=CC1+BB1*DCOS(2.D0*AGS)+ST1(3,KK)*DSIN(2.D0*AGS)
      P(2)=2.D0*CC1-P(1)
      AGG=AGS*180.D0/PAI
      SHDUM=P(1)-P(2)
      IF(AG.GT.0.D0) AG1=AG-PAI/2.D0
      IF(AG.LE.0.D0) AG1=AG+PAI/2.D0
      IF(AGS.GT.0.D0) AGS1=AGS-PAI/2.D0
      IF(AGS.LE.0.D0) AGS1=AGS+PAI/2.D0
      DDUM1=0.D0
      DDUM2=0.D0
C
C.....HISTORY OF CONCRETE
C
      IF(JST.NE.0) THEN
      CALL STRM(EPS(1),EST, ESC,USC,YOC,YSC,
     *      YFC,UFC,UST,YOT,YST,YFT,UFT,DDUM1,
     *      PMAX(1,KK),PMAX(9,KK),PMAX(17,KK),NEL,KK,AGS,
     *      PMAX(2,KK),AG)
      AGP(KK)=AG
      IF(ISOL.NE.1) THEN
      HSS(1)=PMAX(25,KK)
      HSS(2)=PMAX(26,KK)
      HSS(3)=PMAX(27,KK)
      HSS(4)=PMAX(28,KK)
      IF(EPS(1).LT.EPS(2)) THEN
      CALL STRMAX(EPS(1),EPS(2),USC,YOC,YSC,YFC,UFC,
     *      ISYM,UST,YOT,YST,YFT,UFT,US,YS,RS,AS,DS,AGS,
     *      PMAX(1,KK),PMAX(9,KK),PMAX(17,KK),PMAX(25,KK),
     *      HSS(1),AGP(KK),PMAX(2,KK),AG)
      ELSEIF(EPS(2).LT.EPS(1)) THEN
      CALL STRMAX(EPS(2),EPS(1),USC,YOC,YSC,YFC,UFC,
     *      ISYM,UST,YOT,YST,YFT,UFT,US,YS,RS,AS,DS,AGS1,
     *      PMAX(1,KK),PMAX(9,KK),PMAX(17,KK),PMAX(25,KK),
     *      HSS(1),AGP(KK),PMAX(2,KK),AG1)
      ENDIF
      PMAX(25,KK)=HSS(1)
      PMAX(26,KK)=HSS(2)
      PMAX(27,KK)=HSS(3)
      PMAX(28,KK)=HSS(4)
      ENDIF
      ENDIF
      AG=AG*180.D0/PAI
C
C.....HISTORY OF SMEARED REINFORCING STEEL
C
      REP(1)=CC+BB*DCOS(2.D0*AS(1))+EPSN(3)*DSIN(2.D0*AS(1))/2.D0
      REP(2)=CC+BB*DCOS(2.D0*AS(2))+EPSN(3)*DSIN(2.D0*AS(2))/2.D0
      DDUM1=0.D0
      DDUM2=0.D0
      IF(JST.NE.0) THEN
      IF(IREN(1).NE.0.D0) THEN
      CALL STRAINLT(EMAX(1,KK),REP(1),DDUM1,IBAU,
     *     CONSTM1(1,IREN(1)),NCM,RST1(KK))
      ENDIF
```

```
      IF(IREN(2).NE.0.D0) THEN
      CALL STRAINLT(EMAX(7,KK),REP(2),DDUM2,IBAU,
     *      CONSTM1(1,IREN(2)),NCM,RST2(KK))
      ENDIF
      ENDIF
      RES(1)=RST1(KK)
      IF(RS(1).NE.0.D0) RES(1)=RES(1)/RS(1)
      RES(2)=RST2(KK)
      IF(RS(2).NE.0.D0) RES(2)=RES(1)/RS(2)
      WRITE(42) P1(1),P1(2),P(1),P(2),AGG,
     *       EPSN(1),RES(1),EPSN(2),RES(2)
 830  CONTINUE
      RETURN
      END

      SUBROUTINE DAMSUR(AG,P,TMST,KK,TH,KK1)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION TMST(8)
      PAI=2.D0*DASIN(1.D0)
      CALL FINMAX(AG,TMST,TM,KK,TH)
      IF(DABS(TM).LT.DABS(P)) FAS=AG
      AK=0.D0
      DO 300 I=1,8
      TTH=-PAI/2.D0+(AK+1.D0)*PAI/8.D0
      DAG=DABS(TTH-AG)
      IF(DAG.GE.PAI*5.D0/6.D0) DAG=PAI-DAG
      IF(DAG.LE.PAI/6.D0) THEN
      PM=P*DCOS(3.D0*DAG)
      IF(DABS(PM).GT.DABS(TMST(I))) TMST(I)=PM
      ENDIF
      AK=AK+1.D0
 300  CONTINUE
      RETURN
      END

      SUBROUTINE SETCR(AG,P,TMST,KK,TH,KK1,CRRN,EST)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION TMST(8),CRRN(4)
      PAI=2.D0*DASIN(1.D0)
      CALL FINMAX(AG,TMST,TM,KK,TH)
      IF(TM.LT.EST.AND.P.GE.EST) THEN
      DO 300 I=1,4
      IK=I
      IF(CRRN(I).EQ.0.D0) GOTO 500
      DAG=DABS(CRRN(I)-AG)
      IF(DAG.LE.PAI/6.D0.OR.DAG.GE.PAI*5.D0/6.D0) RETURN
 300  CONTINUE
 500  CRRN(IK)=AG
      ENDIF
      RETURN
      END

      SUBROUTINE REFSUR(AG,TMST,RMST,CMST,CMST1,ESC,EST,
     *       KK)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION TMST(8),RMST(8)
      PAI=2.D0*DASIN(1.D0)
      K1=KK-1
      IF(K1.EQ.0) K1=8
```

```
      AK=0.D0
      DO 300 I=1,8
      IF(I.NE.KK.AND.I.NE.K1) THEN
      IF(TMST(I).LT.0.9D0*EST) THEN
      TTH=-PAI/2.D0+(AK+1.D0)*PAI/8.D0
      DAG=DABS(TTH-AG)
      IF(DAG.GT.PAI/2.D0) DAG=PAI-DAG
      RMST1=(CMST+CMST1)/2.D0+(CMST-CMST1)*DCOS(2.D0*DAG)/2.D0
      IF(RMST(I).GT.RMST1) RMST(I)=RMST1
      ENDIF
      ENDIF
      AK=AK+1.D0
300   CONTINUE
      RETURN
      END

      SUBROUTINE REFPICK(AG,CMST,RMST,ESC,KK,TH,ECTT)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION RMST(8)
      PAI=2.D0*DASIN(1.D0)
      PTH=TH-PAI/8.D0
      K1=KK-1
      IF(K1.EQ.0) K1=8
      RMST1=RMST(K1)
      RMST2=RMST(KK)
      RM=RMST1+(AG-PTH)*(RMST2-RMST1)/(TH-PTH)
            DUM=DABS(RM/ESC)
            IF(DUM.LE.3.D0) THEN
            ESP=-ESC*(.145D0*DUM*DUM+.13D0*DUM)
            ELSE
            ESP=RM+(3.D0*ESC-1.695D0*ESC)
            ENDIF
            ECTT=ESP
      RETURN
      END

      SUBROUTINE FINMAX(AG,TMST,TM,KK,TH)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION TMST(8)
      PAI=2.D0*DASIN(1.D0)
      PTH=TH-PAI/8.D0
      K1=KK-1
      IF(K1.EQ.0) K1=8
      TM=TMST(K1)+(AG-PTH)*(TMST(KK)-TMST(K1))/(TH-PTH)
      RETURN
      END

      SUBROUTINE AGPICK(AG,KK,TH)
      IMPLICIT REAL*8 (A-H,O-Z)
      PAI=2.D0*DASIN(1.D0)
      AK=0.D0
      DO 100 I=1,8
      PTH=-PAI/2.D0+AK*PAI/8.D0
      TH=-PAI/2.D0+(AK+1.D0)*PAI/8.D0
      IF(AG.GT.PTH.AND.AG.LE.TH) THEN
      KK=I
      RETURN
      ENDIF
      AK=AK+1.D0
```

```
100 CONTINUE
    RETURN
    END

    SUBROUTINE STRM(P,EST,ESC,USC,YOC,YSC,
   *       YFC,UFC,UST,YOT,YST,YFT,UFT,DUM,
   *       CMST,TMST,RMST,NEL,KJL,AG,CRRN,AGE)
    IMPLICIT REAL*8 (A-H,O-Z)
    COMMON /ITRN/ JST,IST
    DIMENSION TMST(8),RMST(8),CRRN(4)
    DIMENSION P(2)
    PAI=2.D0*DASIN(1.D0)
    IF(AG.GT.0.D0) AG1=AG-PAI/2.D0
    IF(AG.LE.0.D0) AG1=AG+PAI/2.D0
    CALL AGPICK(AG,KK,TH)
    CALL AGPICK(AG1,KK1,TH1)
    CALL REFPICK(AG,CMST,RMST,ESC,KK,TH,ECT)
    CALL REFPICK(AG1,CMST,RMST,ESC,KK1,TH1,ECT1)
    IF((P(1)-ECT).GE.0.D0.AND.(P(2)-ECT1).GE.0.D0) THEN
    CALL SETCR(AG,(P(1)-ECT),TMST,KK,TH,KK1,CRRN,EST)
    CALL SETCR(AG1,(P(2)-ECT1),TMST,KK1,TH1,KK,CRRN,EST)
          CALL DAMSUR(AG,(P(1)-ECT),TMST,KK,TH,KK1)
          CALL DAMSUR(AG1,(P(2)-ECT1),TMST,KK1,TH1,KK)
    ELSEIF((P(1)-ECT).GE.0.D0.AND.(P(2)-ECT1).LT.0.D0) THEN
    CALL SETCR(AG,(P(1)-ECT),TMST,KK,TH,KK1,CRRN,EST)
          CALL DAMSUR(AG,(P(1)-ECT),TMST,KK,TH,KK1)
          IF(CMST.GT.P(2)) THEN
          CMST=P(2)
          DUM=0.D0
    CALL REFSUR(AG1,TMST,RMST,CMST,DUM,ESC,EST,KK)
          ENDIF
    ELSEIF((P(1)-ECT).LT.0.D0.AND.(P(2)-ECT1).GE.0.D0) THEN
    CALL SETCR(AG1,(P(2)-ECT1),TMST,KK1,TH1,KK,CRRN,EST)
          CALL DAMSUR(AG1,(P(2)-ECT1),TMST,KK1,TH1,KK)
          IF(CMST.GT.P(1)) THEN
          CMST=P(1)
          DUM=0.D0
    CALL REFSUR(AG,TMST,RMST,CMST,DUM,ESC,EST,KK1)
          ENDIF
    ELSEIF((P(1)-ECT).LT.0.D0.AND.(P(2)-ECT1).LT.0.D0) THEN
          IF(CMST.GT.P(1).AND.P(2).GT.P(1)) THEN
          CMST=P(1)
    CALL REFSUR(AG,TMST,RMST,CMST,P(2),ESC,EST,KK1)
          ELSEIF(CMST.GT.P(2).AND.P(1).GT.P(2)) THEN
          CMST=P(2)
    CALL REFSUR(AG1,TMST,RMST,CMST,P(1),ESC,EST,KK)
          ENDIF
    ENDIF
    RETURN
    END

    SUBROUTINE STRMAX(EP,EPSN2,USC,YOC,YSC,YFC,UFC,
   *        ISYM,UST,YOT,YST,YFT,UFT,US,YS,RS,AS,DS,
   *        AG,CMST,TMST,RMST,HS,HSS,AGP,CRRN,AGE)
    IMPLICIT REAL*8(A-H,O-Z)
    COMMON /ITRN/ JST,IST
    COMMON /CL/ ISOL,ISP
    DIMENSION US(6),YS(6),RS(6),AS(6),DS(6)
    DIMENSION ES(30),STR(30),STIF(30)
```

```
      DIMENSION TMST(8),RMST(8),HS(4),HSS(2),CRRN(4)
      PAI=2.D0*DASIN(1.D0)
      IF(AG.GT.0.D0) AG1=AG-PAI/2.D0
      IF(AG.LE.0.D0) AG1=AG+PAI/2.D0
      ESC=DABS(USC/YSC)
      EST=DABS(UST/YST)
      CALL AGPICK(AG,KK,TH)
      CALL AGPICK(AG1,KK1,TH1)
      CALL REFPICK(AG,CMST,RMST,ESC,KK,TH,ECT)
      CALL REFPICK(AG1,CMST,RMST,ESC,KK1,TH1,ECT1)
      CALL FINMAX(AG,TMST,TM,KK,TH)
      CALL FINMAX(AG1,TMST,TM1,KK1,TH1)
      CM=CMST
      IF((EPSN2-ECT1).GT.TM1) THEN
      EPSM=(EPSN2-ECT1)
      ELSE
      EPSM=TM1
      ENDIF
      IF((EPSN2-ECT1).LT.0.D0) EPSM=TM1
            DUM=DABS(CM/ESC)
            IF(DUM.LE.3.D0) THEN
            ESP=-ESC*(.145D0*DUM*DUM+.13D0*DUM)
            ELSE
            ESP=CM+(3.D0*ESC-1.695D0*ESC)
            ENDIF
            DUM=TM/0.9D0/EST
            IF(DUM.LE.1.D0) THEN
            REFS=0.D0
            ELSE
            REFS=-UFC*(DUM-1.D0)/2.D0/2.D0
            ENDIF
            IF(REFS.LT.-UFC/2.D0) REFS=-UFC/2.D0
            REFS=0.D0
      CALL STPOS(EP,EPSN2,EPSM,USC,YOC,YSC,YFC,UFC,
     *         ISYM,UST,YOT,YST,YFT,UFT,US,YS,RS,AS,DS,
     *         AG,ECT,IMODE,ES,STR,STIF,ESP,CM,TM,HS,AGP,
     *         REFS,CRRN,AGE,REP,IREN)
      IF(EP.EQ.ES(1)) THEN
            HSS(1)=ES(2)
            HSS(2)=0.D0
      ELSE
      IF(EP.LT.ES(9)) THEN
            HSS(1)=EP
            IF(HSS(1).LT.ES(2)) HSS(1)=ES(2)
            HSS(2)=0.D0
      ELSEIF(EP.GT.ES(10)) THEN
            HSS(2)=EP
            IF(HSS(2).GT.ES(5)) HSS(2)=ES(5)
            HSS(1)=0.D0
      ENDIF
      ENDIF
      RETURN
      END

      SUBROUTINE STRAINLT(EMAX,EP,DUM,IBAU,CONSTM,NCM,RST)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION CONSTM(NCM),EMAX(6)
      US=DABS(CONSTM(1))
      YS=DABS(CONSTM(2))
```

```
      RS=DABS(CONSTM(3))
      AS=(CONSTM(4))
      ALFA=DABS(CONSTM(5))
      ESH=DABS(CONSTM(7))
      EUT=DABS(CONSTM(8))
      SUT=DABS(CONSTM(9))
      EY=US/YS
      IF(IBAU.EQ.1) THEN
      IF(EMAX(1).LT.EP) THEN
      EMAX(1)=EP
      EMAX(6)=EMAX(1)-EY*2.D0
      ELSEIF(EMAX(6).GT.EP) THEN
      EMAX(6)=EP
      EMAX(1)=EMAX(6)+EY*2.D0
      ENDIF
      IF(EP.GE.EMAX(1)) THEN
      DUM=1.D0
      ELSEIF(EP.LE.EMAX(6)) THEN
      DUM=-1.D0
      ENDIF
      ELSE
      IF(EMAX(1).EQ.0.D0.AND.EMAX(6).EQ.0.D0.AND.
     *       DABS(EP).LT.EY) RETURN
      IF(EMAX(1).EQ.0.D0.AND.EMAX(6).EQ.0.D0) THEN
             IF(EP.GT.0.D0) EMAX(1)=EP
             IF(EP.LT.0.D0) EMAX(6)=EP
      ELSEIF(EMAX(1).NE.0.D0.AND.EMAX(6).EQ.0.D0) THEN
      EM=EMAX(1)
      EP1=EMAX(4)
      EP2=EMAX(2)
      EPC=(EMAX(1)-EMAX(5))*0.8D0+EMAX(5)
      IF(EPC.LT.EP2) EPC=EP2
      CALL CUVSOL(EM,US,YS,DD,RRT1,ESH,EUT,SUT,
     *             EP1,EP2)
      DDUM=DABS(RRT1/(EM-EPC))
      IF(DDUM.GT.YS) EPC=EM-RRT1/YS
             IF(EP.GT.EMAX(1)) THEN
                    EMAX(1)=EP
                    EMAX(6)=0.D0
             ELSEIF(EP.LT.EPC.AND.RST.LE.-US/3.D0) THEN
                    IF(EPC.GT.EMAX(4)) EMAX(4)=EPC
                    EMAX(6)=EP
                    EMAX(1)=0.D0
                    EMAX(2)=EPC
             ENDIF
      ELSEIF(EMAX(1).EQ.0.D0.AND.EMAX(6).NE.0.D0) THEN
      EM=EMAX(6)
      EP1=EMAX(5)
      EP2=EMAX(2)
      EPC=(EMAX(6)-EMAX(4))*0.8D0+EMAX(4)
      IF(EPC.GT.EP2) EPC=EP2
      CALL CUVSOL(EM,US,YS,DD,RRT1,ESH,EUT,SUT,
     *             EP1,EP2)
      DDUM=DABS(RRT1/(EM-EPC))
      IF(DDUM.GT.YS) EPC=EM+RRT1/YS
             IF(EP.LT.EMAX(6)) THEN
                    EMAX(6)=EP
                    EMAX(1)=0.D0
             ELSEIF(EP.GT.EPC.AND.RST.GE.US/3.D0) THEN
```

```
            IF(EPC.LT.EMAX(5)) EMAX(5)=EPC
            EMAX(1)=EP
            EMAX(6)=0.D0
            EMAX(2)=EPC
      ENDIF
ENDIF
ENDIF
RETURN
END

SUBROUTINE PRINCIPAL(S,P,AG,DUM)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /ITRN/ JST,IST
DIMENSION S(3)
DIMENSION P(2)
PAI=2.D0*DASIN(1.D0)
CC=(S(1)+S(2))*0.5D0
BB=(S(1)-S(2))*0.5D0
CR=DSQRT(BB*BB+S(3)*S(3))
IF(DABS(BB).GT.1.0D-40) THEN
AG=DATAN(S(3)/BB)/2.D0
ELSE
AG=DUM
ENDIF
IF(AG.GT.0.D0) THEN
APP=AG
AMM=AG-PAI/2.D0
ENDIF
IF(AG.LE.0.D0) THEN
APP=AG+PAI/2.D0
AMM=AG
ENDIF
IF(DUM.LT.AMM) THEN
RA1=DABS(AMM-DUM)
RA2=DABS(APP-DUM-PAI)
      IF(RA1.LT.RA2) THEN
      AG=AMM
      ELSE
      AG=APP
      ENDIF
ELSEIF(DUM.GE.AMM.AND.DUM.LT.APP) THEN
RA1=DABS(AMM-DUM)
RA2=DABS(APP-DUM)
      IF(RA1.LT.RA2) THEN
      AG=AMM
      ELSE
      AG=APP
      ENDIF
ELSEIF(DUM.GE.APP) THEN
RA1=DABS(DUM-AMM-PAI)
RA2=DABS(APP-DUM)
      IF(RA1.LT.RA2) THEN
      AG=AMM
      ELSE
      AG=APP
      ENDIF
ENDIF
P(1)=CC+BB*DCOS(2.D0*AG)+S(3)*DSIN(2.D0*AG)
P(2)=2.D0*CC-P(1)
```

```
      IF(JST.EQ.0) THEN
      IF(P(2).GT.P(1)) THEN
      DDUM=P(2)
      P(2)=P(1)
      P(1)=DDUM
      IF(AG.GT.0.D0) THEN
      AG=AG-PAI/2.D0
      ELSE
      AG=AG+PAI/2.D0
      ENDIF
      ENDIF
      ENDIF
      RETURN
      END

      SUBROUTINE BMAT4 (NEL,ITYPE,NINT,XX,B,I,J,NDIM,MNNE,
     *                  ICOMP)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /XGWGT/ XG(4,4),WGT(4,4)
      DIMENSION B(4,16),XX(NDIM,MNNE),H(8)
      RI=XG(I,NINT)
      SI=XG(J,NINT)
      IF(ICOMP.EQ.2) THEN
      CALL STDM8(XX,B,H,DET,RI,SI,XBAR,NEL,ITYPE,NDIM,MNNE)
      ELSE
      CALL STDM4(XX,B,H,DET,RI,SI,XBAR,NEL,ITYPE,NDIM,MNNE)
      ENDIF
      RETURN
      END

      SUBROUTINE SF1(XX,CONSTM,S,EMAX,NCM,NEL,EU,ELRHS,RST,
     *               ICOMP,NGAU,MNDOFN,MNNE,MNDOFE,NDIM,IBAU)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CNTL/ISYM,NUMEL,IRESOL,IIDUM(26)
      COMMON /ITRN/ JST,IST
      COMMON /CONSTS/ ZERO,ONE,TWO
      COMMON /XGWGT/ XG(4,4),WGT(4,4)
      DIMENSION B(3),XX(NDIM,MNNE),S(MNDOFE,MNDOFE)
      DIMENSION CONSTM(NCM)
      DIMENSION EMAX(6,NGAU)
      DIMENSION EU(MNDOFN,MNNE),ELRHS(MNDOFE)
      DIMENSION RST(NGAU),UV(3),UL(3),XL(3),SL(3,3)
      US=DABS(CONSTM(1))
      YS=DABS(CONSTM(2))
      ALFA=DABS(CONSTM(5))
      AS=DABS(CONSTM(6))
      ESH=DABS(CONSTM(7))
      EUT=DABS(CONSTM(8))
      SUT=DABS(CONSTM(9))
      NINT=NGAU
      CALL CLEAR(ELRHS,MNDOFE)
      IF(IRESOL.EQ.1) RETURN
      DO 30 I=1,MNDOFE
      DO 30 J=1,MNDOFE
   30 S(I,J)=0.D0
      BL=ZERO
      DO 10 I=1,NDIM
      BL=BL+(XX(I,2)-XX(I,1))*(XX(I,2)-XX(I,1))
   10 CONTINUE
```

```
      BL=DSQRT(BL)
      DO 20 I=1,NDIM
      UV(I)=(XX(I,2)-XX(I,1))/BL
   20 CONTINUE
C
C.....TWO NODE LINE ELEMENT
C
      IF(ICOMP.EQ.0) THEN
      REP=(EU(1,2)-EU(1,1))*UV(1)
      REP=REP+(EU(2,2)-EU(2,1))*UV(2)
      REP=REP/BL
      IF(IBAU.EQ.1) THEN
      CALL DDMAT1(REP,US,YS,AS,DD,RRT,EMAX(1,1))
      ELSEIF(IBAU.EQ.2) THEN
      CALL DDMAT2(REP,US,YS,AS,DD,RRT,EMAX(1,1),ESH,EUT,SUT)
      ENDIF
      ASTF=DD*AS/BL
      DO 40 I=1,NDIM
      DO 40 J=I,NDIM
      S(I,J)=ASTF*UV(I)*UV(J)
      S(I,J+NDIM)=-S(I,J)
      S(I+NDIM,J)=-S(I,J)
      S(I+NDIM,J+NDIM)=S(I,J)
   40 CONTINUE
      RETURN
      ELSEIF(ICOMP.EQ.2) THEN
C
C.....THREE NODE LINE ELEMENT
C
      DO 55 I=1,3
      DO 55 J=1,3
   55 SL(I,J)=0.D0
      DO 50 I=1,3
   50 UL(I)=EU(1,I)*UV(1)+EU(2,I)*UV(2)
      XL(1)=0.D0
      XL(2)=BL
      BL=ZERO
      DO 70 I=1,NDIM
      BL=BL+(XX(I,3)-XX(I,2))*(XX(I,3)-XX(I,2))
   70 CONTINUE
      BL=DSQRT(BL)
      XL(3)=XL(2)+BL
      KK=0
      DO 80 LX=1,NINT
      RI=XG(LX,NINT)
      KK=KK+1
      CALL TRSTDM8(XL,B,DET,RI,NDIM,MNNE,NEL)
      WT=WGT(LX,NINT)*AS*DET
      EPSN=0.0D0
      DO 815 J=1,3
      EPSN=EPSN+B(J)*UL(J)
  815 CONTINUE
      IF(IBAU.EQ.1) THEN
      CALL DDMAT1(EPSN,US,YS,AS,DD,RRT,EMAX(1,KK))
      ELSEIF(IBAU.EQ.2) THEN
      CALL DDMAT2(EPSN,US,YS,AS,DD,RRT,EMAX(1,KK),ESH,EUT,SUT)
      ENDIF
      DO 370 J=1,3
      DO 360 I=1,3
```

```
 360 SL(I,J)=SL(I,J)+B(I)*DD*B(J)*WT
 370 CONTINUE
  80 CONTINUE
     S(1,1)=SL(1,1)*UV(1)*UV(1)
     S(1,2)=SL(1,1)*UV(1)*UV(2)
     S(1,3)=SL(1,2)*UV(1)*UV(1)
     S(1,4)=SL(1,2)*UV(1)*UV(2)
     S(1,5)=SL(1,3)*UV(1)*UV(1)
     S(1,6)=SL(1,3)*UV(1)*UV(2)
     S(2,2)=SL(1,1)*UV(2)*UV(2)
     S(2,3)=SL(1,2)*UV(2)*UV(1)
     S(2,4)=SL(1,2)*UV(2)*UV(2)
     S(2,5)=SL(1,3)*UV(2)*UV(1)
     S(2,6)=SL(1,3)*UV(2)*UV(2)
     S(3,3)=SL(2,2)*UV(1)*UV(1)
     S(3,4)=SL(2,2)*UV(1)*UV(2)
     S(3,5)=SL(2,3)*UV(1)*UV(1)
     S(3,6)=SL(2,3)*UV(1)*UV(2)
     S(4,4)=SL(2,2)*UV(2)*UV(2)
     S(4,5)=SL(2,3)*UV(2)*UV(1)
     S(4,6)=SL(2,3)*UV(2)*UV(2)
     S(5,5)=SL(3,3)*UV(1)*UV(1)
     S(5,6)=SL(3,3)*UV(1)*UV(2)
     S(6,6)=SL(3,3)*UV(2)*UV(2)
     DO 400 I=1,6
     DO 400 J=I,6
     S(J,I)=S(I,J)
 400 CONTINUE
     ENDIF
     RETURN
     END

     SUBROUTINE TRSTDM8(XX,B,DET,R,NDIM,MNNE,NEL)
     IMPLICIT REAL*8(A-H,O-Z)
     DIMENSION XX(3),B(3),H(3)
     H(1) = -(1.D0-2.D0*R)/2.D0
     H(2) = -2.D0*R
     H(3) = (1.D0+2.D0*R)/2.D0
     DET=0.0D0
     DO 20 K=1,3
  20 DET=DET+H(K)*XX(K)
     IF(DET.GT.0.00000001D0) GO TO 40
     WRITE(50,2000) NEL
     STOP
  40 DUM=1.0D0/DET
     DO 60 K=1,3
     B(K)=H(K)*DUM
  60 CONTINUE
     RETURN
2000 FORMAT (10H0*** ERROR,
   1        52H ZERO OR NEGATIVE JACOBIAN DETERMINANT FOR ELEMENT (,I4,
   2        1H)    )
     END

     SUBROUTINE UPSS1(XX,CONSTM,EMAX,NCM,NEL,EU,ELRHS,RST,
    *              ICOMP,NGAU,MNDOFN,MNNE,MNDOFE,NDIM,EEP,IBAU)

     IMPLICIT REAL*8(A-H,O-Z)
     COMMON /CNTL/ ISYM,IIDUM(28)
```

```
      COMMON /ITRN/ JST,IST
      COMMON /CONSTS/ ZERO,ONE,TWO
      COMMON /XGWGT/ XG(4,4),WGT(4,4)
      DIMENSION B(3),XX(NDIM,MNNE)
      DIMENSION EPP(3)
      DIMENSION CONSTM(NCM),EEP(MNDOFN,MNNE)
      DIMENSION EMAX(6,NGAU)
      DIMENSION EU(MNDOFN,MNNE),ELRHS(MNDOFE),P(2),H(8)
      DIMENSION RST(NGAU),UV(3),UL(3),XL(3),SL(3,3)
      US=DABS(CONSTM(1))
      YS=DABS(CONSTM(2))
      ALFA=DABS(CONSTM(5))
      AS=DABS(CONSTM(6))
      ESH=DABS(CONSTM(7))
      EUT=DABS(CONSTM(8))
      SUT=DABS(CONSTM(9))
      NINT=NGAU
      CALL CLEAR(ELRHS,MNDOFE)
      DO 5 I=1,MNNE
      DO 5 J=1,MNDOFN
      EEP(J,I)=0.D0
    5 CONTINUE
      DO 15 I=1,3
      EPP(I)=0.D0
   15 CONTINUE
      BL=ZERO
      DO 10 I=1,NDIM
      BL=BL+(XX(I,2)-XX(I,1))*(XX(I,2)-XX(I,1))
   10 CONTINUE
      BL=DSQRT(BL)
      DO 20 I=1,NDIM
      UV(I)=(XX(I,2)-XX(I,1))/BL
   20 CONTINUE
      IF(ICOMP.EQ.0) THEN
      REP=(EU(1,2)-EU(1,1))*UV(1)
      REP=REP+(EU(2,2)-EU(2,1))*UV(2)
      REP=REP/BL
      IF(IBAU.EQ.1) THEN
      CALL DDMAT1(REP,US,YS,AS,DD,RRT,EMAX(1,1))
      ELSEIF(IBAU.EQ.2) THEN
      CALL DDMAT2(REP,US,YS,AS,DD,RRT,EMAX(1,1),ESH,EUT,SUT)
      ENDIF
      RST(1)=RRT
      EEP(1,1)=-RST(1)*UV(1)*AS
      EEP(2,1)=-RST(1)*UV(2)*AS
      EEP(1,2)=RST(1)*UV(1)*AS
      EEP(2,2)=RST(1)*UV(2)*AS
      RETURN
      ELSEIF(ICOMP.EQ.2) THEN
      DO 50 I=1,3
   50 UL(I)=EU(1,I)*UV(1)+EU(2,I)*UV(2)
      XL(1)=0.D0
      XL(2)=BL
      BL=ZERO
      DO 70 I=1,NDIM
      BL=BL+(XX(I,3)-XX(I,2))*(XX(I,3)-XX(I,2))
   70 CONTINUE
      BL=DSQRT(BL)
      XL(3)=XL(2)+BL
```

```
      KK=0
      DO 80 LX=1,NINT
      RI=XG(LX,NINT)
      KK=KK+1
      CALL TRSTDM8(XL,B,DET,RI,NDIM,MNNE,NEL)
      WT=WGT(LX,NINT)*AS*DET
      EPSN=0.0D0
      DO 815 J=1,3
      EPSN=EPSN+B(J)*UL(J)
  815 CONTINUE
      IF(IBAU.EQ.1) THEN
      CALL DDMAT1(EPSN,US,YS,AS,DD,RRT,EMAX(1,KK))
      ELSEIF(IBAU.EQ.2) THEN
      CALL DDMAT2(EPSN,US,YS,AS,DD,RRT,EMAX(1,KK),ESH,EUT,SUT)
      ENDIF
      RST(KK)=RRT
      DO 900 I=1,3
  900 EPP(I)=EPP(I)+B(I)*RST(KK)*WT
   80 CONTINUE
      DO 1000 I=1,3
      DO 1000 J=1,2
      EEP(J,I)=EPP(I)*UV(J)
 1000 CONTINUE
      ENDIF
      RETURN
      END

      SUBROUTINE SS1(XX,CONSTM,EMAX,NCM,NEL,EU,RST,
     *           ICOMP,NGAU,MNDOFN,MNNE,MNDOFE,NDIM,IBAU)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CNTL/ ISYM,IIDUM(28)
      COMMON /ITRN/ JST,IST
      COMMON /CONSTS/ ZERO,ONE,TWO
      COMMON /CNTL1/ TAB
      COMMON /XGWGT/ XG(4,4),WGT(4,4)
      DIMENSION B(3),XX(NDIM,MNNE)
      DIMENSION CONSTM(NCM)
      DIMENSION EMAX(6,NGAU)
      DIMENSION EU(MNDOFN,MNNE)
      DIMENSION RST(NGAU),UV(3),UL(3),XL(3)
      US=DABS(CONSTM(1))
      YS=DABS(CONSTM(2))
      ALFA=DABS(CONSTM(5))
      AS=DABS(CONSTM(6))
      ESH=DABS(CONSTM(7))
      EUT=DABS(CONSTM(8))
      SUT=DABS(CONSTM(9))
      NINT=NGAU
      BL=ZERO
      DO 10 I=1,NDIM
      BL=BL+(XX(I,2)-XX(I,1))*(XX(I,2)-XX(I,1))
   10 CONTINUE
      BL=DSQRT(BL)
      DO 20 I=1,NDIM
      UV(I)=(XX(I,2)-XX(I,1))/BL
   20 CONTINUE
      IF(ICOMP.EQ.0) THEN
      REP=(EU(1,2)-EU(1,1))*UV(1)
      REP=REP+(EU(2,2)-EU(2,1))*UV(2)
```

```
      REP=REP/BL
      DDUM=0.D0
      IF(JST.NE.0) THEN
      CALL STRAINLT(EMAX(1,1),REP,DDUM,IBAU,CONSTM,NCM,RST(1))
      ENDIF
      WRITE(42) REP,RST(1)
      RETURN
      ELSEIF(ICOMP.EQ.2) THEN
      DO 50 I=1,3
   50 UL(I)=EU(1,I)*UV(1)+EU(2,I)*UV(2)
      XL(1)=0.D0
      XL(2)=BL
      BL=ZERO
      DO 70 I=1,NDIM
      BL=BL+(XX(I,3)-XX(I,2))*(XX(I,3)-XX(I,2))
   70 CONTINUE
      BL=DSQRT(BL)
      XL(3)=XL(2)+BL
      KK=0
      DO 80 LX=1,NINT
      RI=XG(LX,NINT)
      KK=KK+1
      CALL TRSTDM8(XL,B,DET,RI,NDIM,MNNE,NEL)
      WT=WGT(LX,NINT)*AS*DET
      EPSN=0.0D0
      DO 815 J=1,3
      EPSN=EPSN+B(J)*UL(J)
  815 CONTINUE
      DDUM=0.D0
      IF(JST.NE.0) THEN
      CALL STRAINLT(EMAX(1,KK),EPSN,DDUM,IBAU,CONSTM,NCM,RST(KK))
      ENDIF
      WRITE(42) EPSN,RST(KK)
   80 CONTINUE
      ENDIF
      RETURN
      END

      SUBROUTINE SFBOND(XX,CONSTM,S,EMAX,NCM,NEL,EU,ELRHS,RST,
     *                  ICOMP,NGAU,MNDOFN,MNNE,MNDOFE,NDIM)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CNTL/ ISYM,NUMEL,IRESOL,IIDUM(26)
      COMMON /ITRN/ JST,IST
      COMMON /CONSTS/ ZERO,ONE,TWO
      COMMON /XGWGT/ XG(4,4),WGT(4,4)
      DIMENSION B(3),XX(NDIM,MNNE),S(MNDOFE,MNDOFE)
      DIMENSION CONSTM(NCM)
      DIMENSION EMAX(11,NGAU)
      DIMENSION EU(MNDOFN,MNNE),ELRHS(MNDOFE)
      DIMENSION RST(NGAU),UV(3),UL(3),XL(3),SL(6,6)
      DIMENSION ULL(6),TAU(10),SE(10)
      PAI=2.D0*DASIN(1.D0)
      US=DABS(CONSTM(1))
      YS=DABS(CONSTM(2))
      DS=DABS(CONSTM(5))
      AS=DABS(CONSTM(6))
      TAU(1)=DABS(CONSTM(10))
      TAU(3)=DABS(CONSTM(11))
      SE(1)=DABS(CONSTM(12))
```

```
      SE(2)=DABS(CONSTM(13))
      SE(3)=DABS(CONSTM(14))
      CS=DS*PAI
      NINT=NGAU
      CALL CLEAR(ELRHS,MNDOFE)
      IF(IRESOL.EQ.1) RETURN
      DO 30 I=1,MNDOFE
      DO 30 J=1,MNDOFE
   30 S(I,J)=0.D0
      BL=ZERO
      DO 10 I=1,NDIM
      BL=BL+(XX(I,2)-XX(I,1))*(XX(I,2)-XX(I,1))
   10 CONTINUE
      BL=DSQRT(BL)
      DO 20 I=1,NDIM
      UV(I)=(XX(I,2)-XX(I,1))/BL
   20 CONTINUE
      DO 1055 I=1,6
      DO 1055 J=1,6
 1055 SL(I,J)=0.D0
C
C.....4-NODE OUT OF PLANE ELEMENT
C
      IF(ICOMP.EQ.0) THEN
      DO 1050 I=1,4
 1050 ULL(I)=EU(1,I)*UV(1)+EU(2,I)*UV(2)
      UL(1)=ULL(1)-ULL(3)
      UL(2)=ULL(2)-ULL(4)
      XL(1)=0.D0
      XL(2)=BL
      KK=0
      DO 1080 LX=1,NINT
      RI=XG(LX,NINT)
      KK=KK+1
      CALL BOND2(XL,B,DET,RI,NDIM,MNNE,NEL)
      WT=WGT(LX,NINT)*CS*DET
      EPSN=0.0D0
      DO 1815 J=1,2
      EPSN=EPSN+B(J)*UL(J)
 1815 CONTINUE
      CALL DMATBOND(EPSN,DD,RST(KK),EMAX(1,KK),NEL,KK,TAU,SE,QST)
      DO 1370 J=1,2
      DO 1360 I=1,2
 1360 SL(I,J)=SL(I,J)+B(I)*DD*B(J)*WT
 1370 CONTINUE
 1080 CONTINUE
      DO 1620 J=1,2
      DO 1620 I=1,2
      SL(I+2,J)=-SL(I,J)
      SL(I,J+2)=-SL(I,J)
      SL(I+2,J+2)=SL(I,J)
 1620 CONTINUE
      DO 1640 J=1,4
      DO 1640 I=1,4
      II=I*2-1
      JJ=J*2-1
      S(II,JJ)=SL(I,J)*UV(1)*UV(1)
      S(II+1,JJ)=SL(I,J)*UV(2)*UV(1)
      S(II,JJ+1)=SL(I,J)*UV(1)*UV(2)
```

```
      S(II+1,JJ+1)=SL(I,J)*UV(2)*UV(2)
 1640 CONTINUE
      DO 1650 I=5,8
      IF(S(I,I).EQ.0.D0) S(I,I)=1.D0
 1650 CONTINUE
      RETURN
C
C.....6-NODE OUT OF PLANE ELEMENT
C
      ELSEIF(ICOMP.EQ.2) THEN
      DO 50 I=1,6
   50 ULL(I)=EU(1,I)*UV(1)+EU(2,I)*UV(2)
      UL(1)=ULL(1)-ULL(4)
      UL(2)=ULL(2)-ULL(5)
      UL(3)=ULL(3)-ULL(6)
      XL(1)=0.D0
      XL(2)=BL
      BL=ZERO
      DO 70 I=1,NDIM
      BL=BL+(XX(I,3)-XX(I,2))*(XX(I,3)-XX(I,2))
   70 CONTINUE
      BL=DSQRT(BL)
      XL(3)=XL(2)+BL
      KK=0
      DO 80 LX=1,NINT
      RI=XG(LX,NINT)
      KK=KK+1
      CALL BOND3(XL,B,DET,RI,NDIM,MNNE,NEL)
      WT=WGT(LX,NINT)*CS*DET
      EPSN=0.0D0
      DO 815 J=1,3
      EPSN=EPSN+B(J)*UL(J)
  815 CONTINUE
      CALL DMATBOND(EPSN,DD,RST(KK),EMAX(1,KK),NEL,KK,TAU,SE,QST)
      DO 370 J=1,3
      DO 360 I=1,3
  360 SL(I,J)=SL(I,J)+B(I)*DD*B(J)*WT
  370 CONTINUE
   80 CONTINUE
      DO 620 J=1,3
      DO 620 I=1,3
      SL(I+3,J)=-SL(I,J)
      SL(I,J+3)=-SL(I,J)
      SL(I+3,J+3)=SL(I,J)
  620 CONTINUE
      DO 640 J=1,6
      DO 640 I=1,6
      II=I*2-1
      JJ=J*2-1
      S(II,JJ)=SL(I,J)*UV(1)*UV(1)
      S(II+1,JJ)=SL(I,J)*UV(2)*UV(1)
      S(II,JJ+1)=SL(I,J)*UV(1)*UV(2)
      S(II+1,JJ+1)=SL(I,J)*UV(2)*UV(2)
  640 CONTINUE
      DO 650 I=7,12
      IF(S(I,I).EQ.0.D0) S(I,I)=1.D0
  650 CONTINUE
      ENDIF
      RETURN
```

```
      END

      SUBROUTINE DMATBOND(EP,DD,RRT,EMAX,NEL,KK,TAU,SE,QST)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /ITRN/ JST,IST
      DIMENSION EMAX(11),ES(15),STR(15),SE(10),TA(10),TAU(10)
      CALL STPBOND(EMAX,ES,STR,SE,TA,TAU,QST,ALP,EFGI,NEL,KK)
      IF(EMAX(1).EQ.0.D0.AND.EMAX(4).EQ.0.D0) THEN
      IF(EP.GE.0.D0) THEN
      CALL ENVEL(EP,SE,TAU,QST,ALP,RRT,DD)
      ELSE
      CALL ENVEL(EP,SE(6),TAU(6),QST,ALP,RRT,DD)
      ENDIF
      RETURN
      ENDIF
      IF(EP.GE.ES(1)) THEN
      CALL ENVEL(EP,SE,TA,QST,ALP,RRT,DD)
      ELSEIF(EP.GE.ES(2).AND.EP.LT.ES(1)) THEN
      CALL TLINE(EP,ES(2),ES(1),STR(2),STR(1),DD,RRT)
      ELSEIF(EP.GE.ES(9).AND.EP.LT.ES(2)) THEN
      CALL TLINE(EP,ES(9),ES(2),STR(9),STR(2),DD,RRT)
      ELSEIF(EP.GT.ES(10).AND.EP.LT.ES(9)) THEN
      CALL TLINE(EP,ES(10),ES(9),STR(10),STR(9),DD,RRT)
      ELSEIF(EP.GT.ES(14).AND.EP.LE.ES(10)) THEN
      CALL TLINE(EP,ES(14),ES(10),STR(14),STR(10),DD,RRT)
      ELSEIF(EP.GT.ES(15).AND.EP.LE.ES(14)) THEN
      CALL TLINE(EP,ES(15),ES(14),STR(15),STR(14),DD,RRT)
      ELSEIF(EP.LE.ES(15)) THEN
      CALL ENVEL(EP,SE(6),TA(6),QST,ALP,RRT,DD)
      ENDIF
      RETURN
      END

      SUBROUTINE SMATBOND(EP,DD,RRT,EMAX,NEL,KK,TAU,SE)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /ITRN/ JST,IST
      DIMENSION EMAX(11),ES(15),STR(15),SE(10),TA(10),TAU(10)
      CALL STPBOND(EMAX,ES,STR,SE,TA,TAU,QST,ALP,EFGI,NEL,KK)
      IF(EMAX(1).EQ.0.D0.AND.EMAX(4).EQ.0.D0) THEN
      IF(EP.GT.SE(4)) THEN
      EMAX(1)=EP
      EMAX(3)=EP
      EMAX(2)=0.D0
      CALL ENERGY(SE,TAU,QST,ALP,SE(4),EP,TAU(4),RRT,EMAX(5))
      EMAX(7)=EMAX(5)
      EMAX(11)=DABS((9.D0*EP/SE(3)/5.D0+0.1D0)*TAU(3))
      ELSEIF(EP.LT.SE(9)) THEN
      EMAX(4)=EP
      EMAX(2)=EP
      EMAX(3)=0.D0
      CALL ENERGY(SE(6),TAU(6),QST,ALP,SE(9),EP,TAU(9),
     *            RRT,EMAX(5))
      EMAX(6)=EMAX(5)
      EMAX(11)=DABS((9.D0*EP/SE(8)/5.D0+0.1D0)*TAU(8))
      ENDIF
      RETURN
      ENDIF
      IF(EP.GT.ES(1)) THEN
      EMAX(1)=EP
```

```
 EMAX(3)=EP
 EMAX(2)=0.D0
 EMAX(5)=EMAX(5)+TA(5)*(ES(2)-ES(9))
 CALL ENERGY(SE,TA,QST,ALP,ES(1),EP,STR(1),RRT,EDUM)
 EMAX(5)=EMAX(5)+EDUM
 EMAX(7)=EMAX(5)
 EMAX(8)=EMAX(8)+TA(5)*(ES(2)-ES(9))
 EMAX(10)=EMAX(8)
 DUM=-1.2D0*((EMAX(10)/EFGI)**0.67)
 DAM=1.D0-EXP(DUM)
 TA(10)=-(1.D0-DAM)*EMAX(11)
 EMAX(11)=DABS(TA(10))+9.D0*(EP-ES(1))*TAU(3)/SE(3)/5.D0
 EMAX(8)=0.D0
 EMAX(9)=0.D0
 EMAX(10)=0.D0
 ELSEIF(EP.LE.ES(1).AND.EP.GE.ES(15)) THEN
       IF(EP.GE.ES(9)) THEN
              EMAX(2)=EP
              EMAX(3)=0.D0
              IF(EP.GT.ES(2)) THEN
              EMAX(5)=EMAX(5)+TA(5)*(ES(2)-ES(9))
              EMAX(8)=EMAX(8)+TA(5)*(ES(2)-ES(9))
              ELSEIF(EP.LE.ES(2)) THEN
              EMAX(5)=EMAX(5)+TA(5)*(EP-ES(9))
              EMAX(8)=EMAX(8)+TA(5)*(EP-ES(9))
              ENDIF
              EMAX(7)=EMAX(5)
              EMAX(10)=EMAX(8)
       ELSEIF(EP.LE.ES(10)) THEN
              EMAX(3)=EP
              EMAX(2)=0.D0
              IF(EP.LT.ES(14)) THEN
              EMAX(5)=EMAX(5)+TA(10)*(ES(14)-ES(10))
              EMAX(8)=EMAX(8)+TA(10)*(ES(14)-ES(10))
              ELSEIF(EP.GE.ES(14)) THEN
              EMAX(5)=EMAX(5)+TA(10)*(EP-ES(10))
              EMAX(8)=EMAX(8)+TA(10)*(EP-ES(10))
              ENDIF
              EMAX(6)=EMAX(5)
              EMAX(9)=EMAX(8)
       ENDIF
ELSEIF(EP.LT.ES(15)) THEN
EMAX(4)=EP
EMAX(2)=EP
EMAX(3)=0.D0
EMAX(5)=EMAX(5)+TA(10)*(ES(14)-ES(10))
CALL ENERGY(SE(6),TA(6),QST,ALP,ES(15),EP,STR(15),RRT,EDUM)
EMAX(5)=EMAX(5)+EDUM
EMAX(6)=EMAX(5)
EMAX(8)=EMAX(8)+TA(10)*(ES(14)-ES(10))
EMAX(9)=EMAX(8)
DUM=-1.2D0*((EMAX(9)/EFGI)**0.67)
DAM=1.D0-EXP(DUM)
TA(5)=(1.D0-DAM)*EMAX(11)
EMAX(11)=TA(5)+9.D0*DABS((EP-ES(15)))*TAU(3)/SE(3)/5.D0
EMAX(8)=0.D0
EMAX(9)=0.D0
EMAX(10)=0.D0
ENDIF
```

```
      RETURN
      END

      SUBROUTINE STPBOND(EMAX,ES,STR,SE,TA,TAU,QST,ALP,EFGI,NEL,KK)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /ITRN/ JST,IST
      DIMENSION EMAX(11),ES(15),STR(15),SE(10),TAU(10)
      DIMENSION TA(10)
      ALP=0.4
      QST=15.D0
      SE(4)=(TAU(1)/QST/(SE(1)**ALP))**(1./(1.-ALP))
      TAU(4)=QST*SE(4)
      SE(6)=-SE(1)
      SE(7)=-SE(2)
      SE(8)=-SE(3)
      TAU(6)=-TAU(1)
      TAU(8)=-TAU(3)
      SE(9)=-SE(4)
      TAU(9)=-TAU(4)
      EFGI=SE(3)*TAU(3)
      CALL ENERGY(SE,TAU,QST,ALP,SE(4),SE(3),TAU(4),TAU(3),ENGI)
      IF(EMAX(1).EQ.0.D0.AND.EMAX(4).EQ.0.D0) RETURN
      DUM=-1.2D0*((EMAX(6)/ENGI)**1.1)
      DAM=1.D0-EXP(DUM)
      TA(1)=(1.D0-DAM)*TAU(1)
      TA(3)=(1.D0-DAM/(2.D0-DAM))*TAU(3)
      DUM=-1.2D0*((EMAX(7)/ENGI)**1.1)
      DAM=1.D0-EXP(DUM)
      TA(6)=(1.D0-DAM)*TAU(6)
      TA(8)=(1.D0-DAM/(2.D0-DAM))*TAU(8)
      SE(4)=(TA(1)/QST/(SE(1)**ALP))**(1./(1.-ALP))
      TA(4)=QST*SE(4)
      SE(9)=(DABS(TA(6))/QST/(DABS(SE(6))**ALP))**(1./(1.-ALP))
      SE(9)=-SE(9)
      TA(9)=QST*SE(9)
      DUM=-1.2D0*((EMAX(9)/EFGI)**0.67)
      DAM=1.D0-EXP(DUM)
      TA(5)=(1.D0-DAM)*EMAX(11)
      DUM=-1.2D0*((EMAX(10)/EFGI)**0.67)
      DAM=1.D0-EXP(DUM)
      TA(10)=-(1.D0-DAM)*EMAX(11)
      IF(EMAX(1).NE.0.D0) THEN
      ES(1)=EMAX(1)
      CALL ENVEL(ES(1),SE,TA,QST,ALP,STR(1),DD)
      STR(2)=TA(5)
      ES(2)=(STR(2)-STR(1))/QST+ES(1)
      ELSE
      STR(2)=TA(5)
            IF(TA(5).GT.TA(4)) THEN
            ES(2)=(TA(5)/TA(1))**(1./ALP)
            ES(2)=ES(2)*SE(1)
            ELSE
            ES(2)=TA(5)/QST
            ENDIF
      STR(1)=STR(2)
      ES(1)=ES(2)
      ENDIF
      STR(3)=0.D0
      ES(3)=(-STR(1))/QST+ES(1)
```

```
STR(4)=TA(10)
ES(4)=(STR(4)-STR(1))/QST+ES(1)
IF(EMAX(4).NE.0.D0) THEN
ES(15)=EMAX(4)
CALL ENVEL(ES(15),SE(6),TA(6),QST,ALP,STR(15),DD)
STR(14)=TA(10)
ES(14)=(STR(14)-STR(15))/QST+ES(15)
ELSE
STR(14)=TA(10)
      IF(TA(10).LT.TA(9)) THEN
      ES(14)=(DABS(TA(10)/TA(6)))**(1./ALP)
      ES(14)=ES(14)*SE(6)
      ELSE
      ES(14)=TA(10)/QST
      ENDIF
STR(15)=STR(14)
ES(15)=ES(14)
ENDIF
STR(13)=0.D0
ES(13)=(-STR(15))/QST+ES(15)
STR(12)=TA(5)
ES(12)=(STR(12)-STR(15))/QST+ES(15)
STR(9)=TA(5)
STR(10)=TA(10)
IF(EMAX(2).NE.0.D0) THEN
      IF(EMAX(2).LT.ES(2).AND.EMAX(2).GT.ES(12)) THEN
            ES(9)=EMAX(2)
            ES(10)=(STR(10)-STR(9))/QST+ES(9)
      ELSEIF(EMAX(2).GE.ES(2)) THEN
            ES(9)=ES(2)
            ES(10)=ES(4)
      ELSEIF(EMAX(2).LE.ES(12)) THEN
            ES(9)=ES(12)
            ES(10)=ES(14)
      ENDIF
ENDIF
IF(EMAX(3).NE.0.D0) THEN
      IF(EMAX(3).LT.ES(4).AND.EMAX(3).GT.ES(14)) THEN
            ES(10)=EMAX(3)
            ES(9)=(STR(9)-STR(10))/QST+ES(10)
      ELSEIF(EMAX(3).GE.ES(4)) THEN
            ES(9)=ES(2)
            ES(10)=ES(4)
      ELSEIF(EMAX(3).LE.ES(14)) THEN
            ES(9)=ES(12)
            ES(10)=ES(14)
      ENDIF
ENDIF
RETURN
END

SUBROUTINE ENVEL(ESS,PSE,PTA,QST,ALP,STR,DD)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /ITRN/ JST,IST
DIMENSION SE(4),TA(4),PSE(4),PTA(4)
ES=DABS(ESS)
DO 100 I=1,4
SE(I)=DABS(PSE(I))
TA(I)=DABS(PTA(I))
```

```
 100 CONTINUE
     IF(ES.LE.SE(4)) THEN
           STR=QST*ES
           DD=QST
     ELSEIF(ES.GT.SE(4).AND.ES.LE.SE(1)) THEN
           STR=TA(1)*((ES/SE(1))**ALP)
           DD=(ES/SE(1))**ALP
           DD=DD*TA(1)*ALP/ES
     ELSEIF(ES.GT.SE(1).AND.ES.LE.SE(2)) THEN
           STR=TA(1)
           DD=0.D0
     ELSEIF(ES.GT.SE(2).AND.ES.LE.SE(3)) THEN
           STR=(ES-SE(2))*(TA(3)-TA(1))/(SE(3)-SE(2))+TA(1)
           DD=(TA(3)-TA(1))/(SE(3)-SE(2))
     ELSEIF(ES.GT.SE(3)) THEN
           STR=TA(3)
           DD=0.D0
     ENDIF
     IF(ESS.LT.0.D0) STR=-STR
     RETURN
     END

     SUBROUTINE ENERGY(PSE,PTAU,QST,ALP,PEP1,PEP2,
    *                  PSG1,PSG2,ENG)
     IMPLICIT REAL*8(A-H,O-Z)
     COMMON /ITRN/ JST,IST
     DIMENSION PSE(4),PTAU(4)
     DIMENSION SE(4),TAU(4)
     ALP1=ALP+1.0
     EP1=DABS(PEP1)
     EP2=DABS(PEP2)
     SG1=DABS(PSG1)
     SG2=DABS(PSG2)
     DO 100 I=1,4
     SE(I)=DABS(PSE(I))
     TAU(I)=DABS(PTAU(I))
 100 CONTINUE
     IF(EP1.LE.SE(1)) THEN
           IF(EP2.LE.SE(1)) THEN
           ENG=TAU(1)*(EP2**ALP1-EP1**ALP1)/ALP1/(SE(1)**ALP)
                 ENG=ENG+SG1*SG1/2.D0/QST-SG2*SG2/2.D0/QST
           ELSEIF(EP2.GT.SE(1).AND.EP2.LE.SE(2)) THEN
           ENG=TAU(1)*(SE(1)**ALP1-EP1**ALP1)/ALP1/(SE(1)**ALP)
                 ENG=ENG+SG1*SG1/2.D0/QST-TAU(1)*TAU(1)/2.D0/QST
                 ENG=ENG+(EP2-SE(1))*TAU(1)
           ELSEIF(EP2.GT.SE(2).AND.EP2.LE.SE(3)) THEN
           ENG=TAU(1)*(SE(1)**ALP1-EP1**ALP1)/ALP1/(SE(1)**ALP)
                 ENG=ENG+SG1*SG1/2.D0/QST-SG2*SG2/2.D0/QST
                 ENG=ENG+(SE(2)-SE(1))*TAU(1)
           ENG=ENG+(TAU(1)-SG2)*(EP2-SE(2))/2.D0+SG2*(EP2-SE(2))
           ELSEIF(EP2.GT.SE(3)) THEN
           ENG=TAU(1)*(SE(1)**ALP1-EP1**ALP1)/ALP1/(SE(1)**ALP)
                 ENG=ENG+SG1*SG1/2.D0/QST-SG2*SG2/2.D0/QST
                 ENG=ENG+(SE(2)-SE(1))*TAU(1)
     ENG=ENG+(TAU(1)-TAU(3))*(SE(3)-SE(2))/2.D0+SG2*(EP2-SE(2))
           ENDIF
     ELSEIF(EP1.GT.SE(1).AND.EP1.LE.SE(2)) THEN
           IF(EP2.GT.SE(1).AND.EP2.LE.SE(2)) THEN
                 ENG=(EP2-EP1)*TAU(1)
```

```
        ELSEIF(EP2.GT.SE(2).AND.EP2.LE.SE(3)) THEN
              ENG=SG1*SG1/2.D0/QST-SG2*SG2/2.D0/QST
              ENG=ENG+(SE(2)-EP1)*TAU(1)
        ENG=ENG+(TAU(1)-SG2)*(EP2-SE(2))/2.D0+SG2*(EP2-SE(2))
        ELSEIF(EP2.GT.SE(3)) THEN
              ENG=SG1*SG1/2.D0/QST-SG2*SG2/2.D0/QST
              ENG=ENG+(SE(2)-EP1)*TAU(1)
  ENG=ENG+(TAU(1)-TAU(3))*(SE(3)-SE(2))/2.D0+SG2*(EP2-SE(2))
        ENDIF
  ELSEIF(EP1.GT.SE(2).AND.EP1.LE.SE(3)) THEN

        IF(EP2.GT.SE(2).AND.EP2.LE.SE(3)) THEN
              ENG=SG1*SG1/2.D0/QST-SG2*SG2/2.D0/QST
        ENG=ENG+(SG1-SG2)*(EP2-EP1)/2.D0+SG2*(EP2-EP1)
        ELSEIF(EP2.GT.SE(3)) THEN
              ENG=SG1*SG1/2.D0/QST-SG2*SG2/2.D0/QST
        ENG=ENG+(SG1-TAU(3))*(SE(3)-EP1)/2.D0+SG2*(EP2-EP1)
        ENDIF
  ELSEIF(EP1.GT.SE(3)) THEN
        ENG=SG2*(EP2-EP1)
  ENDIF
  RETURN
  END

  SUBROUTINE BOND3(XX,B,DET,R,NDIM,MNNE,NEL)
  IMPLICIT REAL*8(A-H,O-Z)
  DIMENSION XX(3),B(3),H(3)
  H(1) = -(1.D0-2.D0*R)/2.D0
  H(2) = -2.D0*R
  H(3) = (1.D0+2.D0*R)/2.D0
  B(1)=R*(R-1.D0)/2.D0
  B(2)=-(R-1.D0)*(R+1.D0)
  B(3)=R*(R+1.D0)/2.D0
  DET=0.0D0
  DO 20 K=1,3
20 DET=DET+H(K)*XX(K)
  IF(DET.GT.0.00000001D0) RETURN
  WRITE(50,2000) NEL
  STOP
2000 FORMAT (10H0*** ERROR,
 1       52H ZERO OR NEGATIVE JACOBIAN DETERMINANT FOR ELEMENT (,I4,
 2       1H)    )
  END

  SUBROUTINE BOND2(XX,B,DET,R,NDIM,MNNE,NEL)
  IMPLICIT REAL*8(A-H,O-Z)
  DIMENSION XX(2),B(2),H(2)
  H(1) = -1.D0/2.D0
  H(2) = 1.D0/2.D0
  B(1)=-(R-1.D0)/2.D0
  B(2)=(R+1.D0)/2.D0
  DET=0.0D0
  DO 20 K=1,2
20 DET=DET+H(K)*XX(K)
  IF(DET.GT.0.00000001D0) RETURN
  WRITE(50,2000) NEL
  STOP
2000 FORMAT (10H0*** ERROR,
 1       52H ZERO OR NEGATIVE JACOBIAN DETERMINANT FOR ELEMENT (,I4,
```

```
2        1H)    )
      END

      SUBROUTINE UPBOND(XX,CONSTM,EMAX,NCM,NEL,EU,ELRHS,RST,
     *               ICOMP,NGAU,MNDOFN,MNNE,MNDOFE,NDIM,EEP)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CNTL/ ISYM,IIDUM(28)
      COMMON /ITRN/ JST,IST
      COMMON /CONSTS/ ZERO,ONE,TWO
      COMMON /XGWGT/ XG(4,4),WGT(4,4)
      DIMENSION B(3),XX(NDIM,MNNE)
      DIMENSION EPP(3)
      DIMENSION CONSTM(NCM),EEP(MNDOFN,MNNE)
      DIMENSION EMAX(11,NGAU)
      DIMENSION EU(MNDOFN,MNNE),ELRHS(MNDOFE),P(2)
      DIMENSION RST(NGAU),UV(3),UL(3),XL(3)
      DIMENSION ULL(6),TAU(10),SE(10)
      PAI=2.D0*DASIN(1.D0)
      US=DABS(CONSTM(1))
      YS=DABS(CONSTM(2))
      DS=DABS(CONSTM(5))
      AS=DABS(CONSTM(6))
      TAU(1)=DABS(CONSTM(10))
      TAU(3)=DABS(CONSTM(11))
      SE(1)=DABS(CONSTM(12))
      SE(2)=DABS(CONSTM(13))
      SE(3)=DABS(CONSTM(14))
      CS=DS*PAI
      NINT=NGAU
      CALL CLEAR(ELRHS,MNDOFE)
      DO 5 I=1,MNNE
      DO 5 J=1,MNDOFN
      EEP(J,I)=0.D0
    5 CONTINUE
      DO 15 I=1,3
      EPP(I)=0.D0
   15 CONTINUE
      BL=ZERO
      DO 10 I=1,NDIM
      BL=BL+(XX(I,2)-XX(I,1))*(XX(I,2)-XX(I,1))
   10 CONTINUE
      BL=DSQRT(BL)
      DO 20 I=1,NDIM
      UV(I)=(XX(I,2)-XX(I,1))/BL
   20 CONTINUE
      IF(ICOMP.EQ.0) THEN
      DO 1050 I=1,4
 1050 ULL(I)=EU(1,I)*UV(1)+EU(2,I)*UV(2)
      UL(1)=ULL(1)-ULL(3)
      UL(2)=ULL(2)-ULL(4)
      XL(1)=0.D0
      XL(2)=BL
      KK=0
      DO 1080 LX=1,NINT
      RI=XG(LX,NINT)
      KK=KK+1
      CALL BOND2(XL,B,DET,RI,NDIM,MNNE,NEL)
      WT=WGT(LX,NINT)*CS*DET
      EPSN=0.0D0
```

```
      DO 1815 J=1,2
      EPSN=EPSN+B(J)*UL(J)
1815  CONTINUE
      CALL DMATBOND(EPSN,DD,RRT,EMAX(1,KK),NEL,KK,TAU,SE,QST)
      RST(KK)=RRT
      DO 1900 I=1,2
1900  EPP(I)=EPP(I)+B(I)*RST(KK)*WT
1080  CONTINUE
      DO 1950 I=1,2
      DO 1950 J=1,2
      EEP(J,I)=EPP(I)*UV(J)
1950  EEP(J,I+2)=-EEP(J,I)
      RETURN
      ELSEIF(ICOMP.EQ.2) THEN
      DO 50 I=1,6
   50 ULL(I)=EU(1,I)*UV(1)+EU(2,I)*UV(2)
      UL(1)=ULL(1)-ULL(4)
      UL(2)=ULL(2)-ULL(5)
      UL(3)=ULL(3)-ULL(6)
      XL(1)=0.D0
      XL(2)=BL
      BL=ZERO
      DO 70 I=1,NDIM
      BL=BL+(XX(I,3)-XX(I,2))*(XX(I,3)-XX(I,2))
   70 CONTINUE
      BL=DSQRT(BL)
      XL(3)=XL(2)+BL
      KK=0
      DO 80 LX=1,NINT
      RI=XG(LX,NINT)
      KK=KK+1
      CALL BOND3(XL,B,DET,RI,NDIM,MNNE,NEL)
      WT=WGT(LX,NINT)*CS*DET
      EPSN=0.0D0
      DO 815 J=1,3
      EPSN=EPSN+B(J)*UL(J)
  815 CONTINUE
      CALL DMATBOND(EPSN,DD,RRT,EMAX(1,KK),NEL,KK,TAU,SE,QST)
      RST(KK)=RRT
      DO 900 I=1,3
  900 EPP(I)=EPP(I)+B(I)*RST(KK)*WT
   80 CONTINUE
      DO 950 I=1,3
      DO 950 J=1,2
      EEP(J,I)=EPP(I)*UV(J)
  950 EEP(J,I+3)=-EEP(J,I)
      ENDIF
      RETURN
      END

      SUBROUTINE SSBOND(XX,CONSTM,EMAX,NCM,NEL,EU,RST,
     *          ICOMP,NGAU,MNDOFN,MNNE,MNDOFE,NDIM)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CNTL/ ISYM,IIDUM(28)
      COMMON /ITRN/ JST,IST
      COMMON /CONSTS/ ZERO,ONE,TWO
      COMMON /CNTL1/ TAB
      COMMON /XGWGT/ XG(4,4),WGT(4,4)
      DIMENSION B(3),XX(NDIM,MNNE)
```

```
      DIMENSION CONSTM(NCM)
      DIMENSION EMAX(11,NGAU)
      DIMENSION EU(MNDOFN,MNNE)
      DIMENSION RST(NGAU),UV(3),UL(3),XL(3)
      DIMENSION ULL(6),TAU(10),SE(10)
      PAI=2.D0*DASIN(1.D0)
      US=DABS(CONSTM(1))
      YS=DABS(CONSTM(2))
      DS=DABS(CONSTM(5))
      AS=DABS(CONSTM(6))
      TAU(1)=DABS(CONSTM(10))
      TAU(3)=DABS(CONSTM(11))
      SE(1)=DABS(CONSTM(12))
      SE(2)=DABS(CONSTM(13))
      SE(3)=DABS(CONSTM(14))
      CS=DS*PAI
      NINT=NGAU
      BL=ZERO
      DO 10 I=1,NDIM
      BL=BL+(XX(I,2)-XX(I,1))*(XX(I,2)-XX(I,1))
   10 CONTINUE
      BL=DSQRT(BL)
      DO 20 I=1,NDIM
      UV(I)=(XX(I,2)-XX(I,1))/BL
   20 CONTINUE
      IF(ICOMP.EQ.0) THEN
      DO 1050 I=1,4
 1050 ULL(I)=EU(1,I)*UV(1)+EU(2,I)*UV(2)
      UL(1)=ULL(1)-ULL(3)
      UL(2)=ULL(2)-ULL(4)
      XL(1)=0.D0
      XL(2)=BL
      KK=0
      DO 1080 LX=1,NINT
      RI=XG(LX,NINT)
      KK=KK+1
      CALL BOND2(XL,B,DET,RI,NDIM,MNNE,NEL)
      WT=WGT(LX,NINT)*CS*DET
      EPSN=0.0D0
      DO 1815 J=1,2
      EPSN=EPSN+B(J)*UL(J)
 1815 CONTINUE
      IF(JST.NE.0) THEN
      CALL SMATBOND(EPSN,DD,RST(KK),EMAX(1,KK),NEL,KK,TAU,SE)
      ENDIF
      DDUM=0.D0
      WRITE(42) EPSN,RST(KK)
 1080 CONTINUE
      RETURN
      ELSEIF(ICOMP.EQ.2) THEN
      DO 50 I=1,6
   50 ULL(I)=EU(1,I)*UV(1)+EU(2,I)*UV(2)
      UL(1)=ULL(1)-ULL(4)
      UL(2)=ULL(2)-ULL(5)
      UL(3)=ULL(3)-ULL(6)
      XL(1)=0.D0
      XL(2)=BL
      BL=ZERO
      DO 70 I=1,NDIM
```

```
      BL=BL+(XX(I,3)-XX(I,2))*(XX(I,3)-XX(I,2))
   70 CONTINUE
      BL=DSQRT(BL)
      XL(3)=XL(2)+BL
      KK=0
      DO 80 LX=1,NINT
      RI=XG(LX,NINT)
      KK=KK+1
      CALL BOND3(XL,B,DET,RI,NDIM,MNNE,NEL)
      WT=WGT(LX,NINT)*CS*DET
      EPSN=0.0D0
      DO 815 J=1,3
      EPSN=EPSN+B(J)*UL(J)
  815 CONTINUE
      IF(JST.NE.0) THEN
      CALL SMATBOND(EPSN,DD,RST(KK),EMAX(1,KK),NEL,KK,TAU,SE)
      ENDIF
      DDUM=0.D0
      WRITE(42) EPSN,RST(KK)
   80 CONTINUE
      ENDIF
      RETURN
      END


      SUBROUTINE  SOLVE(REALA,INTA,A)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTL/ ISYM,NUMEL,IRESOL,JDUM(8),NNEGP,NPOSP,NRHSF,
     *                       IDUM(15)
      DIMENSION A(1)
      NNEGP = 0
      NPOSP = 0
      IF(IRESOL .EQ. 0.OR.IRESOL.EQ.2) CALL COMPLT(REALA,INTA,A)
      IF(IRESOL .EQ. 1) CALL RESOL(REALA,INTA,A)
      RETURN
      END

      SUBROUTINE COMPLT(REALA,INTA,A)
C
C     INITIATE FORWARD ELIMINATION OF LHS AND RHS
C     FOLLOWED BY BACKSUBSTITUTION
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CNTL/ISYM,NUMEL,IRESOL,NRHS,NTAPEB,NTAPEU,NTAPEL,
     *                MA,IWRT,IPRINT,IERR,NNEGP,NPOSP,NRHSF,
     *                IB,IU,IL,IFB,IFU,IFL,MBUF,MW,MKF,
     *                MELEM,MFWR,MB,MDOF,MFW,MLDEST
      DIMENSION A(1)
C     CALL SECOND(T0)
      IERR = 1
      N = NUMEL+MLDEST+2*MDOF
      IF(ISYM .GT. 1) GO TO 10
      MELEM = (MDOF*(MDOF+1))/2+MDOF*NRHS
      MKF = (MFW*(MFW+1))/2
      GO TO 20
   10 MELEM = MDOF*(MDOF+NRHS)
      MKF = MFW*MFW
   20 MFWR = MKF+MFW*NRHS
      MW = MELEM+MFWR
```

```
      MBUF = MA-MW-N
      IF(MBUF .LT. MFW+NRHS) GO TO 70
      IAL = 1+NUMEL
      IAM = IAL+MLDEST
      IAN = IAM+MDOF
      IAE = IAN+MDOF
      IAF = IAE+MELEM
      IAB = IAF+MFWR
      CALL FRWCP(REALA,INTA,
     *           A(1),A(IAL),A(IAM),A(IAN),A(IAE),A(IAF),A(IAB))
      IF(IRESOL.EQ.2) RETURN
C     CALL SECOND(TF)
      DT = TF-T0
      IF(IERR .NE. 1) RETURN
      IF(NRHS .EQ. 0) GO TO 60
      CALL BCKWRD(REALA,INTA,
     *            A(1),A(IAL),A(IAM),A(IAN),A(IAE),A(IAF),A(IAB),
     *              A(IAB))
C     CALL SECOND(TB)
      DT = TB-TF
   60 RETURN
   70 IERR = 6
 1000 FORMAT(2(/), 5X,29HSYMMETRIC FORWARD ELIMINATION        ,/)
 1010 FORMAT(2(/), 5X,31HUNSYMMETRIC FORWARD ELIMINATION  ,/)
 1020 FORMAT(        5X,22HRESOLUTION INACTIVATED        ,/)
 1030 FORMAT(        4X,21H      INTEGER ARRAY:          ,I7,/,
     *               4X,21H     REAL ARRAY:              ,I7,/,
     *               4X,21H        ELEMENT:              ,I7,/,
     *               4X,21H          FRONT:              ,I7,/,
     *               4X,21H         BUFFER:              ,I7,/,
     *               4X,21H      TOTAL STORAGE:          ,I7)
 1040 FORMAT(        10X,29HTIME IN FORWARD ELIMINATION:        ,D9.2)
 1043 FORMAT(        10X,18HWRITES TO NTAPEU:            ,I4)
 1045 FORMAT(        10X,18HWRITES TO NTAPEL:            ,I4)
 1060 FORMAT(2(/), 5X,32HERROR: NOT ENOUGH ROOM IN BUFFER    )
      RETURN
      END
      SUBROUTINE RESOL(REALA,INTA,A)
C
C     INITIATE FORWARD ELIMINATION OF RHS ONLY
C     FOLLOWED BY BACKSUBSTITUTION
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CNTL/ISYM,NUMEL,IRESOL,NRHS,NTAPEB,NTAPEU,NTAPEL,
     *            MA,IWRT,IPRINT,IERR,NNEGP,NPOSP,NRHSF,
     *            IB,IU,IL,IFB,IFU,IFL,MBUF,MW,MKF,
     *            MELEM,MFWR,MB,MDOF,MFW,MLDEST
      DIMENSION A(1)
C     CALL SECOND(T0)
      REWIND NTAPEB
      REWIND NTAPEU
      IF(ISYM .EQ. 3) REWIND NTAPEL
      IF(ISYM .EQ. 2) GO TO 30
      IF(NRHS .EQ. 0) GO TO 40
      IERR = 1
      IFB = 0
      N = NUMEL+MLDEST+2*MDOF
      MELEM = MDOF*NRHS
      MFWR = MFW*NRHS
```

```
      MB = MW-MELEM-MFWR
      IF(MB .LT. 1) GO TO 45
      IAL = 1+NUMEL
      IAM = IAL+MLDEST
      IAN = IAM+MDOF
      IAE = IAN+MDOF
      IAF = IAE+MELEM
      IABR = IAF+MFWR
      IABF = IABR+MB
      CALL FRWRS(REALA,INTA,
     *           A(1),A(IAL),A(IAM),A(IAN),A(IAE),A(IAF),A(IABR),
     *             A(IABF))
C     CALL SECOND(TF)
      DT = TF-T0
      CALL BCKWRD(REALA,INTA,
     *            A(1),A(IAL),A(IAM),A(IAN),A(IAE),A(IAF),A(IABR),
     *              A(IABF))
C     CALL SECOND(TB)
      DT = TB-T0
      GO TO 50
   30 IERR = 3
      RETURN
   40 IERR = 4
      RETURN
   45 IERR = 5
   50 RETURN
 1000 FORMAT(2(/), 5X,26HSYMMETRIC RESOLUTION WITH   ,I3,4H RHS,/)
 1010 FORMAT(2(/), 5X,28HUNSYMMETRIC RESOLUTION WITH   ,I3,4H RHS,/)
 1030 FORMAT(        4X,21H      INTEGER ARRAY:      ,I7,/,
     *               4X,21H     REAL ARRAY:          ,I7,/,
     *               4X,21H        ELEMENT:          ,I7,/,
     *               4X,21H          FRONT:          ,I7,/,
     *               4X,21H     RHS BUFFER:          ,I7,/,
     *               4X,21H     LHS BUFFER:          ,I7,/,
     *               4X,21H      TOTAL STORAGE:      ,I7)
 1040 FORMAT(       10X,29HTIME IN FORWARD ELIMINATION:      ,D9.2)
 1045 FORMAT(       10X,18HWRITES TO NTAPEB:  ,I4)
 1060 FORMAT(2(/), 5X,41HERROR: UNSYMMETRIC RESOLUTION INACTIVATED  )
 1070 FORMAT(2(/), 5X,28HERROR: RESOLUTION WITH 0 RHS )
 1080 FORMAT(2(/),5X,19HERROR: TOO MANY RHS   )
      END

      SUBROUTINE FRWCP(REALA,INTA,
     *                 LELM,LDEST,MDEST,NDEST,ELEM,FRNT,BUF)
C
C     FORWARD ELIMINATION OF BOTH LHS AND RHS
C     CALLS SOLIN FOR DEST. VECTORS, ELEMENT LHS AND RHS'S
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CNTL/ISYM,NUMEL,IRESOL,NRHS,NTAPEB,NTAPEU,NTAPEL,
     *            MA,IWRT,IPRINT,IERR,NNEGP,NPOSP,NRHSF,
     *              IB,IU,IL,IFB,IFU,IFL,MBUF,MW,MKF,
     *              MELEM,MFWR,MB,MDOF,MFW,MLDEST
      DIMENSION LDEST(1),MDEST(1),NDEST(1),ELEM(1),FRNT(1),BUF(1),
     *          LELM(1)
      REWIND NTAPEU
      IF(ISYM .EQ. 3) REWIND NTAPEL
      IFU = 0
      IFL = 0
```

```
      NRHSF = NRHS
      IU = 1
      IL = MBUF
      NFW = 0
      LFW = 0
      DO 200 IEL=1,NUMEL
      CALL SOLIN(REALA,INTA,IEL,3,NRHS,NUMDES,LDEST,ELEM)
      CALL DEST(NUMDES,LDEST,NFW,NDOF,NE,MDEST,NDEST)
      IF(LFW .GT. NFW) NFW = LFW
      IF(ISYM .EQ. 1) CALL SYMASM(NDOF,LFW,NFW,MDEST,ELEM,FRNT)
      IF(ISYM .GT. 1) CALL UNSASM(NDOF,LFW,NFW,MDEST,ELEM,FRNT)
      KFW = NFW
      IF(NRHS .EQ. 0) GO TO 30
      IF(ISYM .GT. 1) GO TO 10
      MKE = (NDOF*(NDOF+1))/2
      GO TO 20
   10 MKE = NDOF*NDOF
   20 CALL SEMRHS(LFW,NFW,NDOF,NRHS,MFW,MDEST,ELEM(MKE+1),FRNT(MKF+1))
   30 IF(NE .EQ. 0) GO TO 155
      DO 150 IE=1,NE
      N = IU+NFW+NRHS-1
      IF(N .LE. IL) GO TO 40
      CALL TOUT(1,IU,IFU,NTAPEU,BUF)
      IU = 1
   40 M = IU
      IF(ISYM .EQ. 3) GO TO 50
      IF(ISYM .EQ. 2) CALL UNSELM(IEL,KFW,NFW,NDEST(IE),FRNT,BUF(IU))
      IF(ISYM .EQ. 1) CALL SYMELM(IEL,NFW,NDEST(IE),FRNT,BUF(IU))
      IU = IU+NRHS+NFW
      GO TO 70
   50 N = IU+NFW+NRHS-1
      IF(N .LE. IL) GO TO 60
      CALL TOUT(IL,MBUF,IFL,NTAPEL,BUF(IL+1))
      IL = MBUF
   60 CALL UNSELM(IEL,KFW,NFW,NDEST(IE),FRNT,BUF(IU))
      IU = IU+NRHS+NFW
   70 IF(IERR .EQ. 1) GO TO 75
      PRINT 1000,IEL
      RETURN
   75 IF(NRHS .EQ. 0) GO TO 90
      IF(ISYM .GT.1) GO TO 80
      CALL ELMRHS(NFW,MFW,NRHS,NDEST(IE),1,FRNT(MKF+1),BUF(M),
     *               BUF(M+NFW))
      GO TO 120
   80 CALL ELMRHS(NFW,MFW,NRHS,NDEST(IE),KFW,FRNT(MKF+1),FRNT(NFW),
     *               BUF(M+NFW))
      IF(ISYM .EQ. 2) GO TO 120
   90 IF(ISYM .NE. 3) GO TO 120
      IF(IL-NFW+1 .GE. N) GO TO 100
      CALL TOUT(1,IU,IFU,NTAPEU,BUF)
      IU = 1
  100 M = NFW
      N = NFW-1
      DO 110 J=1,N
      BUF(IL) = FRNT(M)
      IL = IL-1
  110 M = M+KFW
  120 CONTINUE
  150 NFW = NFW-1
```

```
  155 CONTINUE
      LFW = NFW
      LELM(IEL) = LFW
      IF(ISYM .EQ. 1  .OR. NE .EQ. 0) GO TO 200
      N = KFW
      M = NFW+1
      DO 170 I=2,NFW
      DO 160 J=1,NFW
      FRNT(M) = FRNT(N+J)
  160 M = M+1
  170 N = N+KFW
  200 CONTINUE
      IB = IU
      IF(IWRT .EQ. 0  .AND.  IFU .EQ. 0) GO TO 210
      CALL TOUT(1,IU,IFU,NTAPEU,BUF)
      BACKSPACE NTAPEU
  210 IF(ISYM .NE. 3) RETURN
      IF(IWRT .EQ. 0  .AND.  IFL .EQ. 0) RETURN
      CALL TOUT(IL,MBUF,IFL,NTAPEL,BUF(IL+1))
 1000 FORMAT(//,5X,'ERROR: ZERO PIVOT IN ELEMENT:',
     *                       I5)
      RETURN
      END

      SUBROUTINE FRWRS(REALA,INTA,
     *                    LELM,LDEST,MDEST,NDEST,ELEM,FRNT,B,BUF)
C
C     FORWARD ELIMINATION OF RHS'S ONLY
C     CALLS SOLIN FOR DEST. VECTORS AND ELEMENT RHS'S
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CNTL/ISYM,NUMEL,IRESOL,NRHS,NTAPEB,NTAPEU,NTAPEL,
     *            MA,IWRT,IPRINT,IERR,NNEGP,NPOSP,NRHSF,
     *              IB,IU,IL,IFB,IFU,IFL,MBUF,MW,MKF,
     *              MELEM,MFWR,MB,MDOF,MFW,MLDEST
      DIMENSION LDEST(1),MDEST(1),NDEST(1),ELEM(1),FRNT(1),B(1),
     *            BUF(1),LELM(1)
      IB = 1
      NFW = 0
      IF(ISYM .EQ. 3) GO TO 10
      INC = 1
      NT = NTAPEU
      IFG = IFU
      IS = 1
      ILL = IU-1
      IX = 0
      GO TO 20
   10 INC = -1
      NT = NTAPEL
      IFG = IFL
      IS = MBUF
      ILL = IL+1
      IX = 1
   20 CONTINUE
      IF(IFG .EQ. 0) GO TO 30
      CALL TIN(IX,IS,ILL,NT,BUF)
   30 LFW = 0
      DO 100 IEL =1,NUMEL
      CALL SOLIN(REALA,INTA,IEL,2,NRHS,NUMDES,LDEST,ELEM)
```

```
      CALL DEST(NUMDES,LDEST,NFW,NDOF,NE,MDEST,NDEST)
      IF(LFW .GT. NFW) NFW = LFW
      CALL SEMRHS(LFW,NFW,NDOF,NRHS,MFW,MDEST,ELEM,FRNT)
      IF(NE .EQ. 0) GO TO 90
      DO 70 IE=1,NE
      N = IB+NRHS-1
      IF(N .LE. MB) GO TO 40
      CALL TOUT(1,IB,IFB,NTAPEB,B)
      IB = 1
   40 IF(ISYM .GT. 1) GO TO 50
      IF(IS .LE.ILL) GO TO 60
C
C      IF(IEL.EQ.NUMEL.AND.IE.EQ.NE.AND.(ILL-IS).EQ.1) GOTO 60
C
      CALL TIN(IX,IS,ILL,NT,BUF)
      GO TO 60
   50 IF(IS .GE. ILL) GO TO 60
      IF(IEL.EQ.NUMEL.AND.IE.EQ.NE.AND.(ILL-IS).EQ.1) GOTO 60
      CALL TIN(IX,IS,ILL,NT,BUF)
   60 CONTINUE
      CALL ELMRHS(NFW,MFW,NRHS,NDEST(IE),INC,FRNT,BUF(IS),B(IB))
      IB = IB+NRHS
      IF(ISYM .EQ. 1) IS = IS+NFW+NRHSF
      IF(ISYM .EQ. 3) IS = IS-NFW+1
      NFW = NFW-1
   70 CONTINUE
   90 CONTINUE
      LFW = NFW
      LELM(IEL) = LFW
  100 CONTINUE
      IF(IFU .EQ. 0) RETURN
      REWIND NTAPEU
      DO 110 I=1,IFU
  110 READ(NTAPEU) IU,(BUF(II),II=1,IU)
      IU = 1
      RETURN
      END

      SUBROUTINE BCKWRD(REALA,INTA,
     *                  LELM,LDEST,MDEST,NDEST,ELEM,FRNT,B,U)
C
C     BACKSUBSTITUTION
C     CALLS SOLIN FOR DEST. VECTORS
C     PASSES ELEMENTAL SOLUTIONS TO SOLOUT
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTL/ ISYM,NUMEL,IRESOL,NRHS,NTAPEB,NTAPEU,NTAPEL,
     *              MA,IWRT,IPRINT,IERR,NNEGP,NPOSP,NRHSF,
     *                IB,IUU,IL,IFB,IFU,IFL,MBUF,MW,MKF,
     *                MELEM,MFWR,MB,MDOF,MFW,MLDEST
      DIMENSION LDEST(1),MDEST(1),NDEST(1),ELEM(1),FRNT(1),B(1),U(1)
     *             ,LELM(1)
      IU = IUU
      JEL = NUMEL+1
      IB = IB-NRHS
      DO 100 IEL=1,NUMEL
      JEL = JEL-1
      CALL SOLIN(REALA,INTA,JEL,1,NRHS,NUMDES,LDEST,ELEM)
      CALL DEST(NUMDES,LDEST,NFW,NDOF,NE,MDEST,NDEST)
```

```
      IF(JEL .EQ. 1) GO TO 7
      LFW = LELM(JEL-1)
      IF(LFW .GT. NFW) NFW = LFW
    7 CONTINUE
      NFW = NFW-NE+1
      IF(NE .EQ. 0) GO TO 35
      J = NE+1
      DO 30 I=1,NE
      J = J-1
      IF(IU .GT. 1) GO TO 10
      BACKSPACE NTAPEU
      READ(NTAPEU) IU,(U(II),II=1,IU)
      BACKSPACE NTAPEU
      IU = IU+1
   10 IU = IU-NFW-NRHSF
      IF(IRESOL .EQ. 1) GO TO 20
      N = IU+NFW
      CALL ELMSOL(NFW,MFW,NRHS,NDEST(J),U(IU),U(N),FRNT(1))
      GO TO 30
   20 IF(IB .GE. 1) GO TO 25
      BACKSPACE NTAPEB
      READ(NTAPEB) IB,(B(II),II=1,IB)
      BACKSPACE NTAPEB
      IB = IB-NRHS+1
   25 CONTINUE
      CALL ELMSOL(NFW,MFW,NRHS,NDEST(J),U(IU),B(IB),FRNT(1))
      IB = IB-NRHS
   30 NFW = NFW+1
   35 DO 40 I=1,NDOF
      K = 0
      L = 0
      M = MDEST(I)
      DO 40 J=1,NRHS
      ELEM(K+I) = FRNT(L+M)
      K = K+NDOF
   40 L = L+MFW
      CALL SOLOUT(REALA,INTA,JEL,NDOF,NRHS,ELEM)
  100 CONTINUE
      RETURN
      END

      SUBROUTINE SYMASM(NDOF,LFWX,NFWX,MDEST,ELLHS,FLHS)
C
C     ASSEMBLES THE LHS FOR SYMMETRIC MATRICIES
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION MDEST(1),ELLHS(1),FLHS(1)
      LFW = LFWX
      NFW = NFWX
      IF(NFW .EQ. LFW) GO TO 20
      MI = (LFW*(LFW+1))/2+1
      MJ = (NFW*(NFW+1))/2
      DO 10 I=MI,MJ
   10 FLHS(I) = 0.D0
   20 N = 1
      DO 50 I=1,NDOF
      MI = MDEST(I)
      DO 50 J=1,I
      MJ = MDEST(J)
```

```
      MK = MAX0(MI,MJ)
      MJ = MIN0(MI,MJ)
      MK = (MK*(MK-1))/2+MJ
      FLHS(MK) = FLHS(MK)+ELLHS(N)
   50 N = N+1
      RETURN
      END

      SUBROUTINE UNSASM(NDOF,LFWX,NFWX,MDEST,ELLHS,FLHS)
C
C     ASSEMBLES LHS FOR UNSYMMETRIC MATRICIES
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION MDEST(1),ELLHS(1),FLHS(1)
      LFW = LFWX
      NFW = NFWX
      IF(NFW .EQ. LFW) GO TO 40
      MI = LFW*NFW+1
      MJ = NFW*NFW
      MK = LFW*LFW+1
      DO 10 I=MI,MJ
   10 FLHS(I) = 0.D0
      IF(LFW .EQ. 0) GO TO 40
      MJ = NFW-LFW
      DO 30 I=1,LFW
      DO 20 J=1,MJ
      MI = MI-1
   20 FLHS(MI) = 0.D0
      DO 30 J=1,LFW
      MI = MI-1
      MK = MK-1
   30 FLHS(MI) = FLHS(MK)
      MI = NFW*NFW
   40 N = 1
      DO 50 I=1,NDOF
      MI = MDEST(I)
      MK = (MI-1)*NFW
      DO 50 J=1,NDOF
      MJ = MDEST(J)
      ML = MK+MJ
      FLHS(ML) = FLHS(ML)+ELLHS(N)
   50 N = N+1
      RETURN
      END

      SUBROUTINE SEMRHS(LFW,NFW,NDOF,NRHS,MFW,MDEST,ELRHS,FRHS)
C
C     ASSEMBLES RHS'S
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION MDEST(1),ELRHS(1),FRHS(1)
      N = 1
      DO 70 IN=1,NRHS
      IA = (IN-1)*MFW
      IF(NFW .EQ. LFW) GO TO 15
      M = LFW+1
      DO 13 I=M,NFW
   13 FRHS(IA+I) = 0.D0
   15 CONTINUE
```

```
      DO 50 I=1,NDOF
      J = IA+MDEST(I)
      FRHS(J) = FRHS(J)+ELRHS(N)
   50 N = N+1
   70 CONTINUE
      RETURN
      END

      SUBROUTINE SYMELM(IEL,NFWX,IDX,FLHS,U)
C
C      ELIMINATION OF ONE EQUATION (ID) FOR SYMMETRIC MATRICIES
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTL/ IDUM(9),IPRINT,IERR,NNEGP,NPOSP,IIDUM(16)
      DIMENSION FLHS(1),U(1)
      ID = IDX
      NFW = NFWX
      MP=(ID*(ID+1))/2
      IDM = ID-1
      IDP = ID+1
      M = MP-ID+1
      K = 1
      PIVOT = FLHS(MP)
C     IF(IPRINT .GE. 2) PRINT 200,IEL,NFW,ID,PIVOT
  200 FORMAT(5X,17HIEL,NFW,ID,PIVOT    ,3I5,D13.4)
      U(ID) = PIVOT
      IF(ABS(PIVOT) .LE. 1.D-30) GO TO 90
      IF(PIVOT .LT. 0.D0) NNEGP = NNEGP+1
      IF(PIVOT .GT. 0.D0) NPOSP = NPOSP+1
      IF(IDM .EQ. 0) GO TO 30
      DO 20 I=1,IDM
      S = FLHS(M)
      U(I) = S/PIVOT
      DO 10 J=1,I
      FLHS(K) = FLHS(K)-S*U(J)
   10 K=K+1
   20 M=M+1
   30 M=MP
      K = 0
      IF(IDP .GT. NFW) GO TO 100
      DO 60 I=IDP,NFW
      NN = M-ID
      M = M+ID+K
      N = M-ID
      S = FLHS(M)
      U(I) = S/PIVOT
      IF(IDM .EQ. 0) GO TO 50
      DO 40 J=1,IDM
   40 FLHS(NN+J) = FLHS(N+J)-S*U(J)
   50 NN = NN-1
      DO 55 J=IDP,I
   55 FLHS(NN+J) = FLHS(N+J)-S*U(J)
   60 K=K+1
      GO TO 100
   90 IERR = 2
  100 RETURN
      END

      SUBROUTINE UNSELM(IEL,KFWX,NFWX,IDX,FLHS,U)
```

```
C
C     ELIMINATION OF ONE EQUATION (ID) FOR UNSYMMETRIC MATRICIES
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTL/ IDUM(9),IPRINT,IERR,NNEGP,NPOSP,IIDUM(16)
      DIMENSION FLHS(1),U(1)
      ID = IDX
      KFW = KFWX
      NFW = NFWX
      IDM = ID-1
      IDP = ID+1
      K = IDM*KFW
      MP = K+ID
      PIVOT = FLHS(MP)
C     IF(IPRINT .GE. 2) PRINT 200, IEL,NFW,ID,PIVOT
  200 FORMAT(5X,17HIEL,NFW,ID,PIVOT    ,3I5,D13.4)
      IF(ABS(PIVOT) .LE. 1.D-30) GO TO 90
      IF(PIVOT .LT. 0.D0) NNEGP = NNEGP+1
      IF(PIVOT .GT. 0.D0) NPOSP = NPOSP+1
      DO 5 I=1,NFW
    5 U(I) = FLHS(K+I)
      K = 0
      KK = 0
      IF(IDM .EQ. 0) GO TO 40
      DO 30 I=1,IDM
      S = FLHS(ID+K)/PIVOT
      DO 10 J=1,IDM
      M = J+K
   10 FLHS(M) = FLHS(M)-S*U(J)
      M = K-1
      IF(IDP .GT. NFW) GO TO 25
      DO 20 J=IDP,NFW
   20 FLHS(J+M) = FLHS(J+K)-S*U(J)
   25 K = K+KFW
   30 FLHS(K-KFW+NFW) = S
   40 K=K+KFW
      IF(IDP .GT. NFW) GO TO 100
      DO 70 I=IDP,NFW
      S = FLHS(ID+K)/PIVOT
      M = K-KFW
      IF(IDM .EQ. 0) GO TO 55
      DO 50 J=1,IDM
   50 FLHS(J+M) = FLHS(K+J)-S*U(J)
   55 M=M-1
      DO 60 J=IDP,NFW
   60 FLHS(M+J) = FLHS(K+J)-S*U(J)
      FLHS(K-KFW+NFW) = S
   70 K = K+KFW
      GO TO 100
   90 IERR = 2
  100 CONTINUE
      RETURN
      END

      SUBROUTINE ELMRHS(NFW,MFW,NRHS,ID,INC,FRHS,U,B)
C
C     ELIMINATION OF RHS'S FOR EQUATION (ID)
C
      IMPLICIT REAL*8 (A-H,O-Z)
```

```
      COMMON /CNTL/ ISYM,IIDUM(28)
      DIMENSION FRHS(1),U(1),B(1)
      IDM = ID-1
      IDP = ID+1
      IM = 0
      DO 50 IN = 1,NRHS
      IU = 1
      S = FRHS(IM+ID)
      B(IN) = S
      IF(IDM .EQ. 0) GO TO 25
      DO 20 I=1,IDM
      II = IM+I
      FRHS(II) = FRHS(II)-S*U(IU)
   20 IU=IU+INC
   25 IF(ISYM .EQ. 1) IU = IU+1
      IF(IDP .GT. NFW) GO TO 50
      DO 30 I=IDP,NFW
      II = IM+I
      FRHS(II-1) = FRHS(II)-S*U(IU)
   30 IU = IU+INC
   50 IM = IM+MFW
      RETURN
      END

      SUBROUTINE ELMSOL(NFW,MFW,NRHS,IDX,U,B,X)
C
C     CALCULATES SOLUTION FOR ONE DOF SPECIFIED BY (ID)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTL/ ISYM,IIDUM(28)
      DIMENSION U(1),B(1),X(1)
      ID = IDX
      IDM = ID-1
      IDP = ID+1
      IF(ISYM .GT. 1) GO TO 5
      F1 = U(ID)
      F2 = 1.D0
      GO TO 7
    5 F1 = 1.
      F2 = U(ID)
    7 CONTINUE
      DO 40 IN=1,NRHS
      IU = NFW
      JA = (IN-1)*MFW
      IA = JA+NFW-1
      S = B(IN)/F1
      IF(IDP .GT. NFW) GO TO 20
      DO 10 I=IDP,NFW
      X(IA+1) = X(IA)
      S = S-U(IU)*X(IA)
      IA = IA-1
   10 IU = IU-1
   20 IU = IU-1
      IF(IDM .LT. 1) GO TO 40
      DO 30 I=1,IDM
      S = S-U(IU)*X(IA)
      IA = IA-1
   30 IU = IU-1
   40 X(JA+ID) = S/F2
```

```
      RETURN
      END

      SUBROUTINE DEST(ND,LDEST,NFW,NDOF,NEE,MDEST,NDEST)
C
C     CONVERTS NODAL DEST. VECTORS TO DOF DEST. VECTORS
C     EQUATIONS TO BE ELIMINATED ARE WRITTEN TO NDEST
C     GIVING CURRENT LOCATION IN FRONT
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION LDEST(1),MDEST(1),NDEST(1)
      COMMON /CNTL/ IDUM(9),IPRINT,IIDUM(19)
      MODR(I,J) = I-I/J*J
      NFW = 0
      KM = 1
      KN = 1
      NDOF = 0
      NE = 0
      DO 50 I=1,ND
      M = MODR(LDEST(I),10)
      N = MODR(LDEST(I),100)/10
      NDOF = NDOF+N
      IF(M .GE. 1) NE = NE+N
      L = LDEST(I)/100-1
      DO 10 J=1,N
      MDEST(KM) = L+J
      IF(M .EQ. 0) GO TO 10
      NDEST(KN) = L+J
      KN = KN+1
   10 KM = KM+1
      L = MDEST(KM-1)
      IF(L .GT. NFW) NFW = L
   50 CONTINUE
      IF(NE .EQ. 0) GO TO 80
      DO 70 I=1,NE
      J = I+1
      DO 70 L=J,NE
      IF(NDEST(I) .LT. NDEST(L)) NDEST(L) = NDEST(L)-1
   70 CONTINUE
   80 NEE = NE
      IF(IPRINT .LE. 2) RETURN
 1000 FORMAT(/1X,'IN DEST: NODAL DESTINATION VECTORS',
     *             10I7,10(/,35X,10I7))
 1010 FORMAT(11X,'DOF DESTINATION VECTORS',10I7,10(/,35X,10I7))
 1020 FORMAT(9X,'ELIM. DESTINATION VECTORS',10I7,10(/,35X,10I7))
      RETURN
      END

      SUBROUTINE TIN(L,I,J,NT,B)
C
C     READS RHS BUFFER TAPE
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION B(1)
      READ(NT) K,(B(II),II=1,K)
      IF(L .GT. 0) GO TO 5
      I = 1
      J = K
      RETURN
```

```
    5 I = K
      J = 1
      RETURN
      END
      SUBROUTINE TOUT(I,J,IF,NT,B)
C
C     WRITES ALL BUFFERS TO TAPE
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION B(1)
      IF(J .EQ. I) RETURN
      K = J-I
      IF = IF+1
      WRITE(NT) K,(B(II),II=1,K)
      RETURN
      END

      SUBROUTINE PREFNT(INTA,IN,IA,MS,MU,MR)
C
C     INITIATE PREFRONT
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTL/ISYM,NUMEL,IDUM(24),MDOF,MFW,MLDEST
      DIMENSION IN(1),IA(1)
C     CALL SECOND(T0)
      NFN = 0
      MLDEST = 0
      DO 10 I=1,NUMEL
      IF(IN(I) .GT. MLDEST) MLDEST = IN(I)
   10 NFN = NFN+IN(I)
      CALL DESVEC(INTA,NFN,IN,IA,IA(NFN+1),IA(2*NFN+1))
      MR = MDOF+MFW+1
      MS = NUMEL+MLDEST+2*MDOF+(MDOF*(MDOF+1))/2+(MFW*(MFW+1))/2+MFW
      MU = NUMEL+MLDEST+2*MDOF+MDOF*MDOF+MFW*MFW+MFW
C     CALL SECOND(T1)
      DT = T1-T0
      RETURN
      END

      SUBROUTINE DESVEC(INTA,NFNX,IN,IA,IB,IC)
C
C     CALCULATION OF DESTINATION VECTORS FROM NICKNAMES
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CNTL/ ISYM,NUMEL,IDUM(24),MDOF,MFW,MLDEST
      DIMENSION IN(1),IA(1),IB(1),IC(1)
      MODR(I,J) = I-I/J*J
      NFN = NFNX
      MDOF = 0
      MFW = 0
      IDES = 1
      IP = 0
      JDN = 0
      DO 10 I=1,NFN
   10 IB(I) = 0
      DO 100 IEL=1,NUMEL
      N = IN(IEL)
      NT = 0
      IPS = IP
```

```
      IPC = 1
      NE = 0
      NTT = 0
      DO 60 ID=1,N
      IP = IP+1
      INIC = IA(IP)
      NDOF = MODR(INIC,10)
      NT = NT+NDOF
      IF(IB(IP) .GT. 0) GO TO 20
      JDES = IDES
      IB(IP) = IDES*100+NDOF*10
      IDES = IDES+NDOF
      IF(IDES-1 .GT. MFW) MFW = IDES-1
      GO TO 30
   20 JDES = IB(IP)
      IB(IP) = IB(IP)*100+NDOF*10
   30 JP = IPS+N+1
      IF(JP .GT. NFN) GO TO 45
      DO 40 JD=JP,NFN
      IF(INIC .EQ. IA(JD)) GO TO 50
   40 CONTINUE
   45 IB(IP) = IB(IP)+1
      IC(IPC) = JDES
      IC(IPC+1) = NDOF
      IPC = IPC+2
      NE = NE+1
      NTT = NTT+NDOF
      GO TO 60
   50 IB(JD) = JDES
      IF(JD .GT. JDN) JDN=JD
   60 CONTINUE
      IF(NT .GT. MDOF) MDOF = NT
      IF(IEL .EQ. NUMEL  .OR. NE .EQ. 0) GO TO 90
      IDES = IDES-NTT
      JP = IPS+N+1
      IF(JP .GT. JDN) GO TO 90
      DO 80 JD=JP,JDN
      IF(IB(JD) .EQ. 0) GO TO 80
      IPC = 1
      NT = 0
      DO 70 I=1,NE
      IF(IB(JD) .LT. IC(IPC)) GO TO 70
      NT = NT+IC(IPC+1)
   70 IPC = IPC+2
      IB(JD) = IB(JD)-NT
   80 CONTINUE
   90 CALL PREOUT(INTA,IEL,N,IA(IPS+1),IB(IPS+1))
  100 CONTINUE
      RETURN
      END

      SUBROUTINE SECOND(T)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER IH,IM,IS,IHS
C     CALL GETTIM(IH,IM,IS,IHS)
C     T=3600.D0*IH+60.D0*IM+IS+1.D0-2*IHS
      T=0.D0
      RETURN
      END
```

REFERENCES

1. American Society of Civil Engineers, *Finite Element Analysis of Reinforced Concrete Structures*, 1982.

2. Balakrishnan, S. and Murray, D. W., "Concrete Constitutive Model for NLFE Analysis of Structures," *Journal of the Structural Division*, ASCE, Vol. 114, No. 7, July 1988, pp. 1449-1466.

3. Barzegar, F., "Analysis of R/C Membrane Elements with Anisotropic Reinforcement," *Journal of the Structural Division*, ASCE, Vol. 115, No. 3, Mar. 1989, pp. 647-665.

4. Bazant, Z. P. and Oh, B. H., "Deformation of Cracked Net-Reinforced Concrete Walls," *Journal of the Structural Division*, ASCE, Vol. 109, No. 1, Jan. 1983, pp. 93-108.

5. Bazant, Z. P., "Comment on Orthotropic Models for Concrete and Geomaterials," *Journal of the Engineering Mechanics Division*, ASCE, Vol. 109, No. EM3, June 1983, pp. 849-865.

6. Bazant, Z. P. and Bhat, P. D., "Prediction of Hysteresis of Reinforced Concrete Members," *Journal of the Structural Division*, ASCE, Vol. 103, No. ST1, Jan. 1977, pp. 153-167.

7. Bhide, S., "Reinforced Concrete Elements in Shear and Tension" thesis presented to University of Toronto, in 1986, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

8. Bresler, B. and Scordelis, A. C., "Shear Strength of Reinforced Concrete Beams," *American Concrete Institute Journal, Proceedings*, Vol. 60, Jan. 1963, pp. 51-72.

9. Brown, R. H. and Jirsa, J. O., "Reinforced Concrete Beams under Load Reversals," *American Concrete Institute Journal, Proceedings*, Vol. 68, May 1971, pp. 380-390.

10. Buyukozturk, O. and Shareef, S. S., "Constitutive Modeling of Concrete in Finite Element Analysis," *Computers & Structures*, Vol. 21, No. 3, 1985, pp. 581-610.

11. Cervenka, V. and Gerstle, K. H., "Inelastic Analysis of Reinforced Concrete Panels: Theory," *Publications* of the IABSE, Vol. 31, (11), 1971, pp. 31-45.

12. Ciampi, V., Eligehausen, R., Bertero, V. V. and Popov, E. P., "Analytical Model for Concrete Anchorage of Reinforcing Bars under Generalized Excitations," Earthquake Engineering Research Center, *Report No. UCB/EERC-82/83*, University of California, Berkeley, 1982.

13. Collins, M. P., Vecchio, F. J. and Mehlhorn, G., "An International Competition to Predict the Response of Reinforced Concrete Panels," *Canadian Journal of Civil Engineering,* Vol. 12, No. 3, Sept. 1985, pp. 624-644.

14. Crisfield, M. A., "Local Instabilities in the Non-Linear Analysis of Reinforced Concrete Beams and Slabs," *Proc. Instn Civ. Engrs*, Part 2, 1982, 73, Mar., pp. 135-145.

15. Darwin, D. and Pecknold, D. A., "Analysis of Cyclic Loading of Plane R/C Structures," *Computers and Structures*, Vol. 7, Pergamon Press, Inc., New York, N.Y., 1977, pp. 137-147.

16. Darwin, D. and Pecknold, D. A., "Nonlinear Biaxial Law for Concrete," *Journal of the Engineering Mechanics Division*, ASCE, Vol. 103, No. EM2, Proc. Paper 12839, Apr. 1977, pp. 229-241.

17. Darwin, D. and Pecknold, D. A., "Analysis of R/C Shear Panels under Cyclic Loading," *Journal of the Structural Division*, ASCE, Vol. 102, No. ST2, Feb. 1976, pp. 355-369.

18. Eligehausen, R., Popov, E. P. and Bertero, V. V., "Local Bond Stress-Slip Relationships of Deformed Bars under Generalized Excitations," Earthquake Engineering Research Center, *Report No. UCB/EERC-83/23,* University of California, Berkeley, 1983.

19. Elwi, A. A. and Murray, D. W., "A 3D Hypoelastic Concrete Constitutive Relationship," *Journal of the Engineering Mechanics Division*, ASCE, Vol. 105, No. EM4, Aug. 1979, pp. 623-641.

20. Gupta, A. K. and Akbar, H., "Cracking in Reinforced Concrete Analysis," *Journal of the Structural Division*, ASCE, Vol. 110, No. 8, Aug. 1984, pp. 1735-1746.

21. Hu, H. and Schnobrich, W. C., "Nonlinear Analysis of Cracked Reinforced Concrete," *American Concrete Institute Journal, Proceedings*, Vol. 87, No. 2, Mar., 1990, pp. 199-207.

22. Karsan, I. D. and Jirsa, J. O., "Behavior of Concrete under Compressive Loadings," *Journal of the Structural Division*, ASCE, Vol. 95, No. 12, Dec. 1969, pp. 2543-2563.

23. Kupfer, H. B. and Gerstle, K. H., "Behavior of Concrete under Biaxial Stresses," *Journal of the Engineering Mechanics Division*, ASCE, Vol. 99, No. EM4, Aug. 1973, pp. 853-866.

24. Kupfer, H. B., Hilsdorf, H. K. and Rusch, H., "Behavior of Concrete under Biaxial Stresses," *American Concrete Institute Journal, Proceedings*, Vol. 66, No. 8, Aug. 1969, pp. 656-666.

25. Leibengood, L. D., Darwin, D. and Dodds, R. H., "Parameters Affecting FE Analysis of Concrete Structures," *Journal of the Structural Division*, ASCE, Vol. 112, No. 2, Feb. 1986, pp. 326-341.

26. Liu, T. C. Y., Nilson, A. H. and Slate, F. O., "Biaxial Stress-Strain Relations for Concrete," *Journal of the Structural Division*, ASCE, Vol. 98, No. 5, May. 1972, pp. 1025-1034.

27. Liu, T. C. Y., Nilson, A. H. and Slate, F. O., "Stress-Strain Response and Fracture of Concrete in Uniaxial and Biaxial Compression," *American Concrete Institute Journal, Proceedings*, Vol. , No. , May. 1972, pp. 291-295.

28. Massicotte, B., Elwi, A. E. and MacGregor, J. G., "Tension-Stiffening Model for Planar Reinforced Concrete Members," *Journal of the Structural Division*, ASCE, Vol. 116, No. 11, Nov. 1990, pp. 3039-3058.

29. Oluokun, F. A., "Prediction of Concrete Tensile Strength from its Compressive Strength: Evaluation of Existing Relations for Normal Weight Concrete," *American Concrete Institute Material Journal, Proceedings*, Vol. 88, No. 3, May. 1991, pp. 302-308.

30. Ramm, E., "Strategies for Tracing the Nonlinear Response Near Limit Points," *Nonlinear Finite Element Analysis in Structural Mechanics*, edited by Wundelich, W., Stein, E. and Bathe, K. J., Springer-Verlag, Berlin, 1981.

31. de la Rovere, H. L., "Nonlinear Analysis of Reinforced Concrete Masonry Walls under Simulated Seismic Loadings," thesis presented to the University of California at San Diego, in partial fulfillment of the requirements for the degree of Doctor of Philosophy, 1990.

32. Saenz, L. P., discussion of "Equation for the Stress-Strain Curve of Concrete," by Desayi, P. and Krishnan, S., *American Concrete Institute Journal, Proceedings*, Vol. 61, No. 9, Sept. 1964, pp. 1229-1235.

33. Stevens, N. J., Uzumeri, S. M., Collins, M. P. and Will, G. T., " Constitutive Model for Reinforced Concrete Finite Element Analysis," *American Concrete Institute Journal, Proceedings*, Vol. 88, No. 1, Jan. 1991, pp. 49-59.

34. Stevens, N. J., Uzumeri, S. M. and Collins, M. P., "Reinforced Concrete Subjected to Reversed Cyclic Shear-Experiments and Constitutive Model," *American Concrete Institute Journal, Proceedings*, Vol. 88, No. 2, Mar. 1991, pp. 135-146.

35. Taylor, R. L., Beresford, P. J. and Wilson, E. L., "A Non-Conforming Element for Stress Analysis," *International Journal for Numerical Methods in Engineering*, Vol. 10, 1976, pp. 1211-1219.

36. Vecchio, F. J. and Collins, M. P., "The Modified Compression-Field Theory for Reinforced Concrete Elements Subjected to Shear," *American Concrete Institute Journal, Proceedings*, Vol. 86, No. 2, Mar. 1986, pp. 219-231.

37. Vecchio, F. J., "Nonlinear Finite Element Analysis of Reinforced Concrete Membranes," *American Concrete Institute Journal, Proceedings*, Vol. 86, No. 1, Jan. 1989, pp. 26-35.

38. Vecchio, F. J., "The Response of Reinforced Concrete to In-Plane Shear and Normal Stresses," thesis presented to University of Toronto, in partial fulfillment of the requirements for the degree of Doctor of Philosophy, 1981.

39. Yankelevsky, D. Z. and Reinhardt, H. W., "Uniaxial Behavior of Concrete in Cyclic Tension," *Journal of the Structural Division*, ASCE, Vol. 115, No. 1, an. 1989, pp. 166-182.

40. Wilson, E. L., Taylor, R. L., Doherty, W. P. and Ghaboussi, J., "Incompatible displacement models," *Numerical and Computer Methods in Structural Mechanics,* Academic Press, New York, 1973.

41. ACI Committee 318, "Building Code Requirements for Reinforced Concrete (ACI-318-89)," *American Concrete Institute*, Detroit, 1989.

42. Orangun, C. O., Jirsa, J. O., and Breen, J. E., "A Reevaluation of Test Data on Development Length and Splices," *American Concrete Institute Journal, Proceedings*, Vol. 74, No. 11, Mar. 1977, pp. 114.

VITA

Honggun Park was born in Seoul, Korea, on October 14, 1962, the son of Jintae Park and Bongsun Kim. After completing his work at Korea High School, Seoul, Korea, in 1981, he entered Seoul National University. After studying architectural design and structural engineering in the Department of Architecture, he received the degree of Bachelor of Science from Seoul National University, and entered the Graduate School there in 1985. After studying structural engineering in the Department of Architecture, he received the degree of Master of Science from Seoul National University in 1987. After finishing his Army service, he joined Jeon & Associates as a structural engineer. In July 1990, he was married to Meelae Kim. In September 1990, he entered the Graduate School of The University of Texas.

Permanent address: 391-13 Yangjae-dong, Seocho-gu, Seoul, Korea

This dissertation was typed by the author.