This is to certify that the

thesis entitled

DEVELOPMENT OF A MODIFIED FIBER-BASED BEAM-
COLUMN ELEMENT FOR DRAIN2DX

presented by

Antonio Cordero-Domenech

has been accepted towards fulfillment
of the requirements for

MS degree in Civil and Environmental Engineering

Amit Varma

Major professor

Date 07/22/02

DEVELOPMENT OF A MODIFIED FIBER-BASED BEAM-COLUMN ELEMENT FOR DRAIN2DX

By

Antonio Cordero-Domenech

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Civil and Environmental Engineering

2002

ABSTRACT

DEVELOPMENT OF A MODIFIED FIBER-BASED BEAM-COLUMN ELEMENT FOR DRAIN2DX

By

Antonio Cordero-Domenech

This work presents the development and validation of a modified fiber-based beam-column element for the nonlinear structural analysis program Drain2DX (Dynamic Response Analysis of INelastic 2-Dimensional Structures, an eXtended version). A major drawback of the existing fiber-based element is that it cannot model the effects of local buckling and/or biaxial stresses in the steel elements of a member cross-section. A modified S-type steel stress-strain curve that account for the effects of local buckling and/or biaxial stresses on the monotonic and *cyclic* behavior was developed.

For implementation in Drain2DX, the modified S-type steel cyclic stress-strain curve was discretized into linear segment *branches*. The changes in stiffness associated with the fiber response moving from one branch to the other under cyclic loading were identified using unique event numbers. The event calculation and response state determination algorithm for the fiber were developed and implemented in Drain2DX through subroutines SMEV15 and SMSD15.

A major focus of this report is to demonstrate the capability of the modified fiber-based element to model the static (monotonic and cyclic) behavior of composite concrete-steel elements, such as concrete filled steel tube (CFT) members. The results indicate that the modified fiber-based element is well suited for modeling the behavior of CFT beam-columns.

ACKNOWLEDGMENTS

I would like to thank God for everything that He has given to me and for providing me with strength and wisdom to achieve this goal. Additionally I am grateful to my advisors, professor Amit H. Varma and professor Ronald S. Harichandran for all their help, support, advice and encouragement throughout my graduate studies at Michigan State University. I am also indebted to them and many others from my graduate and undergraduate years. Thanks also to my college and school friends for their support. Finally I would never have got this far, in life or academia, without the support of my close family – Sonia and Bolivar – thanks for everything.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1 GENERAL

This report presents the development and validation of a *modified* fiber-based beam-column finite element for the nonlinear structural analysis program Drain2DX (**D**ynamic **R**esponse **A**nalysis of **IN**elastic **2**-**D**imensional Structures, an e**X**tended version, Prakash et al. 1993). Chapter 1 identifies the major drawback of the existing fiber-based element in Drain2DX. Chapter 2 presents the development and implementation of a modified fiber-based element that addresses this drawback. The validation of the modified fiber-based element for static and cyclic loading is presented in Chapter 3. Finally, Chapter 4 provides recommendations for using the modified fiber-based element effectively. Appendix A provides a listing of all the files and subroutines that were modified in the source code of Drain2DX.

1.2 FIBER-BASED BEAM-COLUMN ELEMENT IN DRAIN2DX

Drain2DX is a general-purpose computer program for conducting nonlinear static and dynamic analysis of two-dimensional (2D) structures. This nonlinear structural analysis program uses an event-to-event based solution algorithm (Prakash et al. 1993). Drain2DX provides several beam, beam-column, and connection/joint element libraries. One of these is the fiber-based beam-column element (Element Type 15).

The fiber-based beam-column element is a distributed plasticity type finite element with a flexibility-based formulation. The formulation of the fiber-based element is presented in detail in Kurama (1997). The assumptions involved in the formulation of the fiber-based beam-column element are as follows:

(1) Plane sections remain plane and perpendicular to the neutral axis before and after bending.

(2) Relative motion (slip) does not occur between the materials in the cross-section.

(3) Inelastic shear deformations are negligible.

(4) The materials in the cross-section are subjected to *uniaxial* stress-strain states.

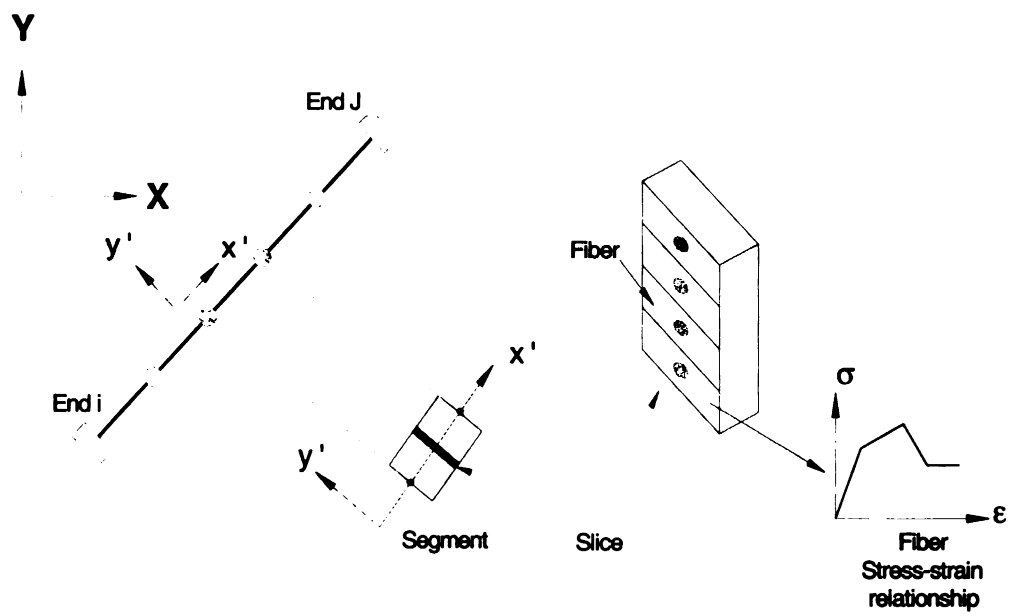


Figure 1.1 Fiber-based beam-column element

A schematic of the fiber-based model is presented in Figure 1.1. As shown in Figure 1.1, the fiber-based element (defined by the end nodes I and J) can be used to model the deformable length of a prismatic member. The member length is divided into *segments*, where the user controls the number and length of each segment. At the mid-length of each segment is a *slice*, which models the member cross-section using *fibers*. Each *fiber* has an associated area, distance from the cross-section centroid, and material stress-strain curve, all of which are specified by the user. The fiber stress-strain states are integrated over the cross-section to obtain the slice force-deformation response, i.e., the section axial force – axial strain – moment – curvature ($P - \varepsilon - M - \phi$) response. Force equilibrium is *enforced* at the slice locations. The slice force-deformation responses are assumed to remain constant over the length of the corresponding segments and are integrated along the length of the member to obtain the element force-displacement response.

The force-displacement response of the fiber-based beam-column element depends on:

(1) The number and location of segments along the length of the member;

(2) The number and distribution of fibers used to model the member cross-section; and

(3) The fiber stress-strain relationships.

All of these can be specified by the user for each individual element or model.

Currently, Drain2DX includes two types of stress-strain models that can be used for the cross-section fibers. These are:

(1) The C-type concrete stress-strain relationship, and

(2) The S-type steel stress-strain relationship.

The monotonic stress-strain relationship, for the C and S-type fibers are shown in Figures 1.2 (a) and (b), respectively. Since Drain2DX uses an event-to-event solution strategy, both stress-strain relationships must be multi-linear, i.e., consisting of linear segments, where each linear segment is defined by a pair of stress-strain points as shown in Figure 1.2.

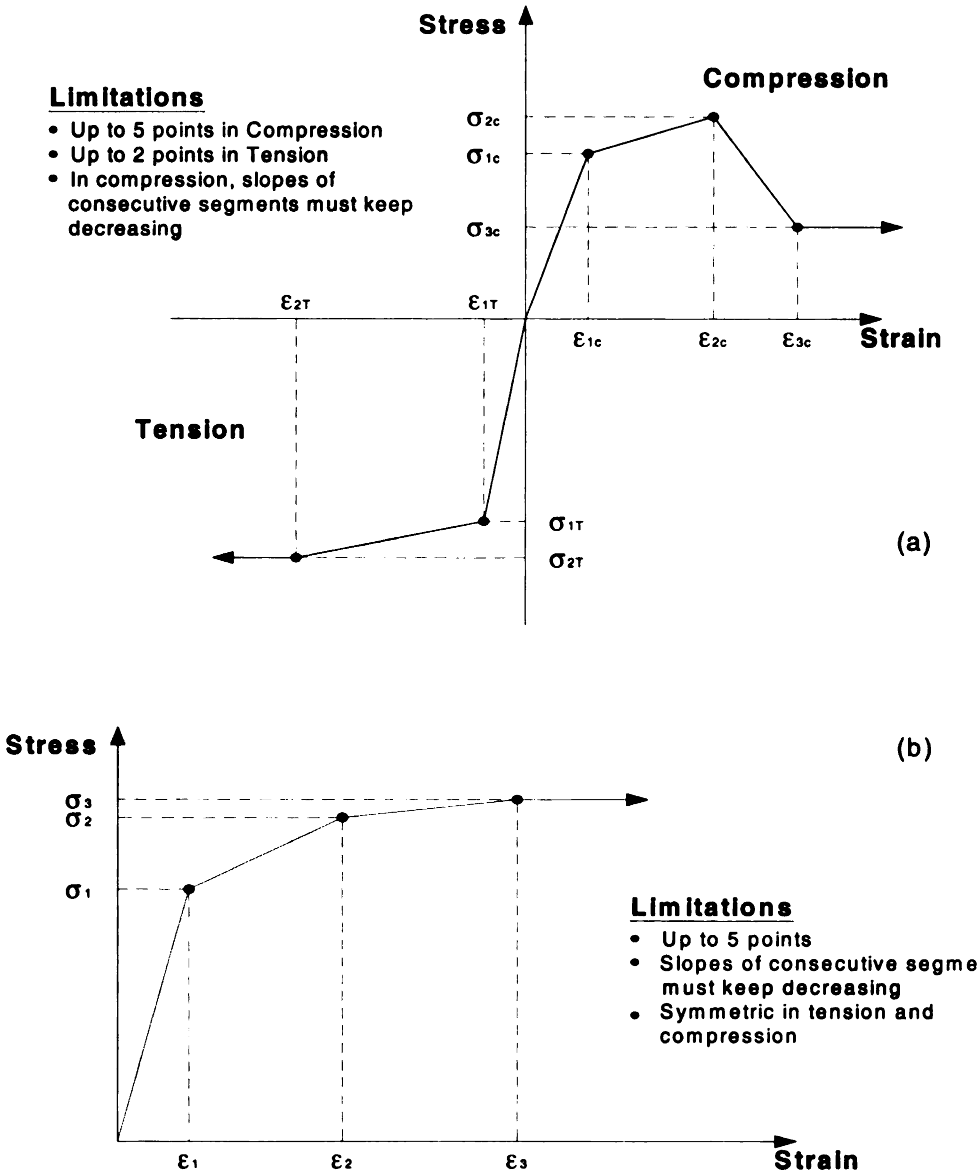
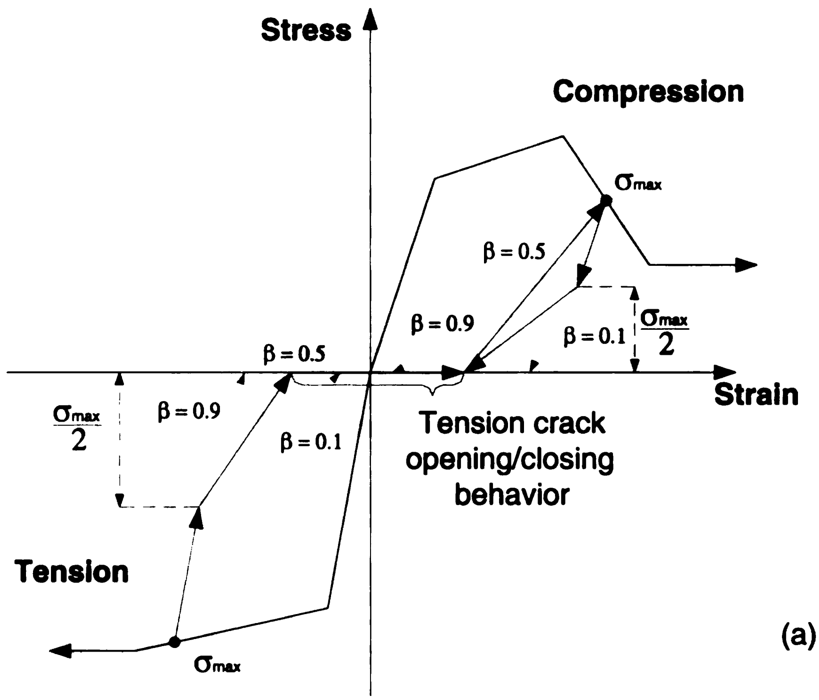


Figure 1.2 Monotonic stress-strain behavior for fibers: (a) C-type; and (b) S-type

Drain2DX also imposes the following limitations on the C and S-type stress-strain curves:

- The C-type stress-strain curve can be defined with a maximum number of five stress-strain points in compression and a maximum number of two stress-strain points in tension.
- The stress is assumed to remain constant for strains numerically greater than the last input points in both compression and tension.
- The slopes of consecutive straight-line segments must be decreasing.
- The S-type stress-strain curve is assumed to be *symmetric* in tension and compression.
- The S-type stress-strain curve can be defined with a maximum number of 5 stress-strain points.
- The S-type stress-strain curve cannot undergo strain softening, i.e., the slope of any linear segment cannot be less than zero.

The cyclic behavior of the C-type and S-type stress-strain curves is controlled by pre-assigned hysteresis rules and the monotonic stress-strain curves (Figure 1.2 (a) and (b)), which act as the envelopes for the cyclic stress-strain curves. The cyclic behavior of the C-type concrete stress-strain curve is shown in Figure 1.3 (a). It can account for tension crack opening and closing behavior and stiffness degradation under cyclic loading. The stiffness degradation is controlled by an unloading factor (β). As shown in Figure 1.3 (a), a value of $\beta = 0.1$ causes almost elastic unloading, and a value of $\beta = 0.9$ causes severe stiffness degradation. Figure 1.3 (b) shows the cyclic behavior of S-type steel stress-strain curve, which undergoes kinematic hardening behavior under cyclic loading.



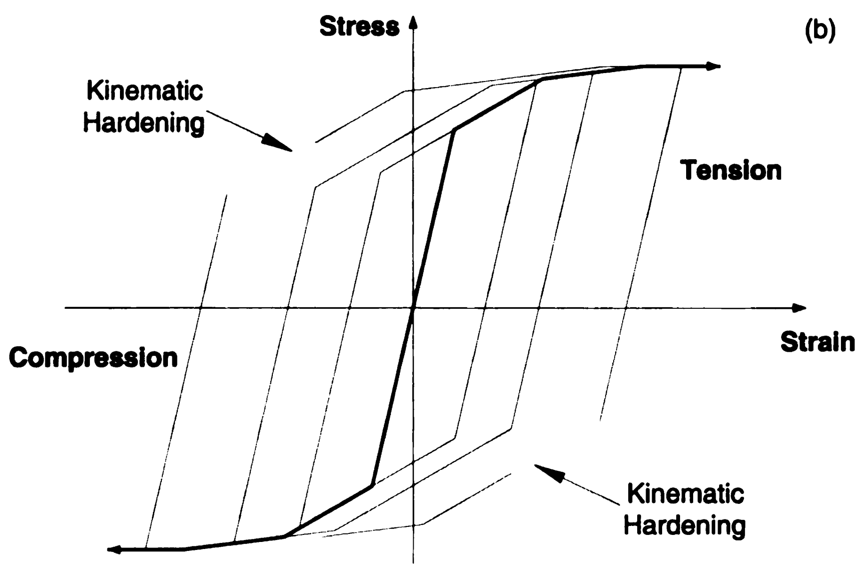


Figure 1.3 Cyclic stress-strain behavior for fibers: (a) C-type; (b) S-type

1.3 DRAWBACK OF THE EXISTING FIBER-BASED ELEMENT

The existing fiber-based beam-column element in Drain2DX is very useful for modeling steel, reinforced concrete, and composite steel-concrete members. This beam-column element has been used by various researchers for investigating the force-deformation behavior of member cross-sections, the force-displacement behavior of structural members, and also for modeling beams and beam-columns while investigating the static or dynamic behavior of complete 2D structural frames. For example, El-Sheikh et al. (1997), Kurama et al. (1997), Ricles et al. (2001), and Shen and Kurama (2002).

More recently, Varma et al. (2001) tried to use the existing fiber-based element for modeling composite concrete filled steel tube (CFT) beam-columns. These researchers had developed three-dimensional (3D) finite element model based on effective uniaxial stress-strain curves for the steel and concrete fibers of the CFT cross-section. As a result, the effective stress-strain curves for the steel fibers implicitly accounted for the effects of local buckling and biaxial stresses in the steel tube. Figure 1.4 shows a typical steel cyclic stress-strain curve that Varma et al. (2001) wanted to use with the fiber-based beam-column element in Drain2DX. However, neither the C nor the S-type stress-strain curve could be used to model this stress-strain behavior (compare Figures 1.3 and 1.4).

Thus, Varma et al. (2001) identified a *major drawback* of the existing fiber-based beam-column element in Drain2DX. The existing S-type stress-strain curve cannot model the effects of local buckling or biaxial stresses in the steel plate elements of a member cross-section for the following reasons:

(1) It is assumed to be symmetric in tension and compression.

(2) It cannot account for strain softening in compression due to local buckling or biaxial stress effects.

(3) Its pre-assigned hysteresis rule (Figure 1.3 (b)) cannot model the desired effects of local buckling or biaxial stresses on the hysteretic behavior of plate elements (Figure 1.4).

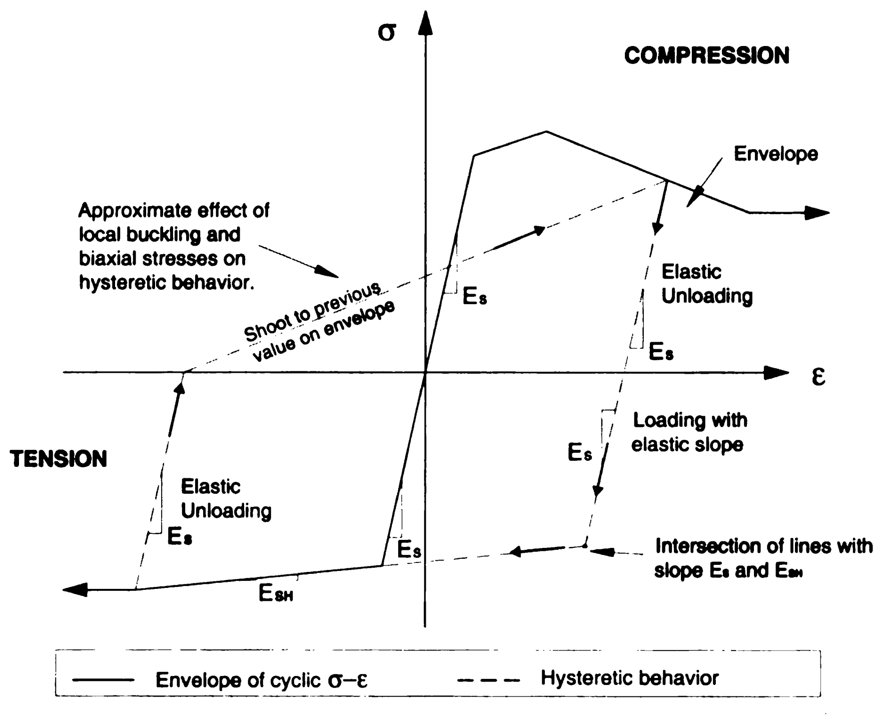


Figure 1.4 Cyclic stress-strain behavior for steel fibers of CFT columns (Varma et al. 2001)

1.4 PROJECT OBJECTIVES AND SCOPE

The objective of this research project is to develop, implement, and verify a *modified* S-type stress-strain model for the fiber-based beam-column element in Drain2DX (Prakash et al. 1993). The modified S-type stress-strain curve will be based on the recommendations of Varma et al. (2001), and it will be able to model the effects of local buckling and biaxial stresses under static or cyclic loading.

1.5 SUMMARY

The fiber-based beam-column element is a simple yet versatile tool used for modeling the nonlinear inelastic behavior of structural members. A fiber-based beam-column element has been previously implemented in the nonlinear structural analysis program Drain2DX (Prakash et al. 1993, Kurama 1997). It has been successfully used by several researchers to investigate the force-deformation behavior of cross-sections, members, and frames. A major drawback of the existing fiber-based element in Drain2DX is that it cannot model the effects of local buckling and/or biaxial stresses on the force-deformation responses of steel or steel-concrete composite members. This research directly addresses this drawback and focuses on the development, implementation, and verification of a modified fiber-based element in Drain2DX.

CHAPTER 2. DEVELOPMENT AND IMPLEMENTATION OF THE MODIFIED FIBER-BASED ELEMENT

The development and implementation of the modified fiber-based beam-column element in Drain2DX is presented in this chapter. No changes were done to the overall formulation of the fiber-based element in Drain2DX. The C-type concrete stress-strain curve was retained as it is and only the S-type steel stress-strain curve was modified. Section 2.1 presents the modified S-type steel stress-strain curve. Section 2.2 presents the implementation of the modified S-type stress-strain curve along with the event calculation and state determination algorithms.

2.1 MODIFIED S-TYPE STEEL STRESS-STRAIN CURVE

A modified S-type steel stress-strain curve was developed for the fiber-based beam-column element in Drain2DX. The modified S-type stress-strain curve is *partially* based on the recommendations of Varma et al. (2001). It is more general in application and has the following features:

(1) It can model asymmetric behavior in tension and compression;

(2) It can model strain hardening or softening behavior in compression;

(3) It can model kinematic strain hardening behavior under cyclic loading; and

(4) It can also model the effects of local buckling and biaxial stresses on the hysteretic behavior in a flexible manner.

Figure 2.1 shows the monotonic behavior of the modified S-type steel stress-strain curve. In Figure 2.1, σ_{yc}, σ_{bc}, σ_{uc}, σ_{yt}, and σ_{sh} are the yield stress in compression, the buckling stress in compression, the ultimate compressive stress, the yield stress in tension and the ultimate stress in tension, respectively. E_{ini}, E_{inel}, E_{int} and E_{sh} are the initial stiffness, the inelastic stiffness (strain-hardening in compression), the softening stiffness in compression and the stain-hardening stiffness in tension, respectively. Finally, ε_{yc}, ε_{bc}, ε_{uc}, ε_{yt}, and ε_{sh} are the strains corresponding to σ_{yc}, σ_{bc}, σ_{uc}, σ_{yt}, and σ_{sh}, respectively.

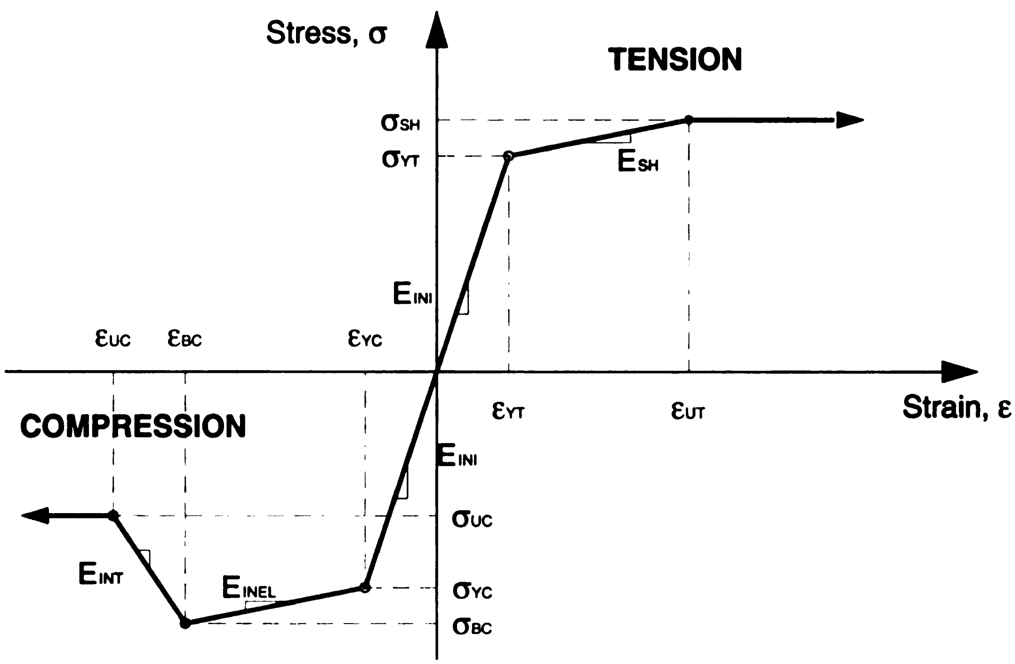


Figure 2.1 Monotonic behavior of modified S-type stress-strain curve

The proposed stress-strain curve has the following limitations:

(1) Three stress-strain points *have to be specified* in compression;

(2) Two stress-strain points *have to be specified* in tension;

(3) The tangent stiffness of consecutive linear segments *must decrease*.

The monotonic stress-strain curve shown in Figure 2.1 serves as the envelope for the stress-strain behavior under cyclic loading. The hysteresis rules under cyclic loading depend on whether the fiber has undergone strain softening (due to buckling) in compression during its cyclic response history. Figure 2.2 (a) shows the cyclic stress-strain behavior and the hysteresis rules for the modified S-type stress-strain curve when it has not undergone compression strain softening (local buckling) during its cyclic response history.

As shown in Figure 2.2 (a), kinematic hardening occurs under cyclic loading as long as the fiber does not undergo compression strain softening during its cyclic response history. Figure 2.2 (b) shows the cyclic stress-strain behavior and the hysteresis rules for the modified S-type stress-strain curve after compression strain softening (due to local buckling and/or biaxial stresses) has occurred during the cyclic response history. As shown in Figure 2.2 (b), the hysteretic behavior is controlled by the unloading parameter α. α equal to zero corresponds to a case of no stiffness degradation, i.e., biaxial stress effects dominate according to Varma et al. (2001), and α equal to one corresponds to a case of significant stiffness degradation, i.e., local buckling effects dominate. The user can specify any value of α between zero and one depending on the application and model. Thus, the modified S-type steel stress-strain curve also accounts for the effects of local buckling and biaxial stresses on the hysteretic behavior.

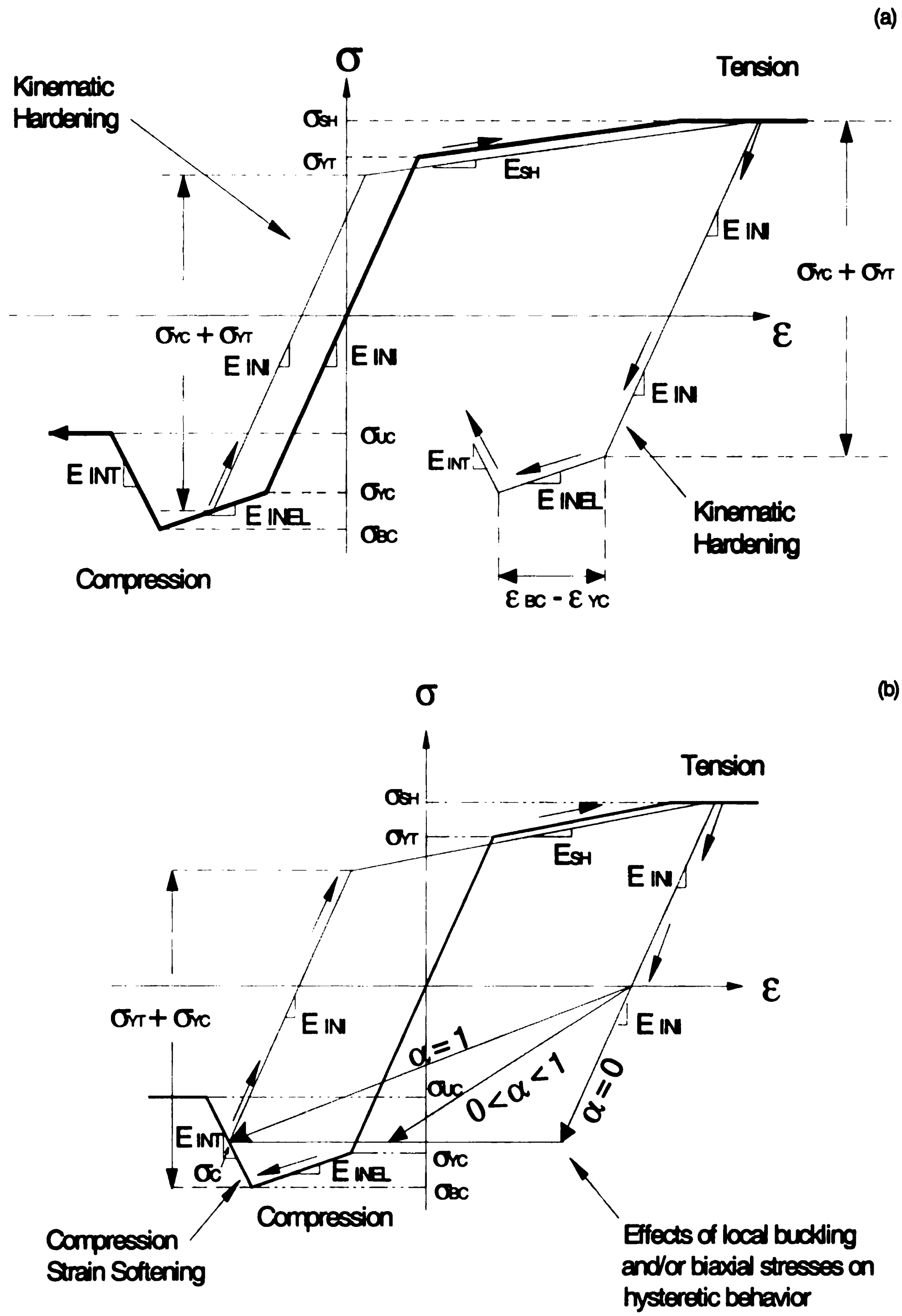


Figure 2.2 **Cyclic behavior of modified S-type stress-strain curve: (a) before and, (b) after compression strain softening**

2.2 IMPLEMENTATION OF MODIFIED S-TYPE STEEL STRESS-STRAIN CURVE

As mentioned previously, Drain2DX (Prakash et al. 1993) is a general-purpose nonlinear structural analysis computer program for static and dynamic analysis of two-dimensional (2D) structures. Static nonlinear analyses are performed based on an event-to-event solution strategy, where each event corresponds to a significant change in stiffness.

Therefore, for implementation into Drain2DX, the linear segments of the picewise linear modified S-type cyclic stress-strain curve were identified using unique *branch numbers*. The changes in stiffnesses associated with the response (stress-strain) states moving from one branch to the other under cyclic loading were identified using *event numbers*.

The modified S-type steel stress-strain behavior was implemented in Drain2DX using two major subroutines SMEV15 and SMSD15, which are the event factor calculation and the state determination subroutines, respectively. Since Drain2DX uses a displacement control analysis, several events can occur within a strain increment. The SMEV15 subroutine monitors the magnitude of the strain increment and determines the occurrence of events using the current state (stress, strain and tangent stiffness) and the stress-strain curve input by the user.

The SMSD15 subroutine determines the state of the fiber following the applied strain increment. If an event occurs (according to the SMEV15 subroutine), the strain increment is scaled down to the value causing the event and the state (stress, strain, and tangent stiffness) of the fiber is updated. The remaining portion of the strain increment is applied incrementally using the recalculated state, and so on.

Section 2.2.1 presents the discretization of the modified S-type stress-strain behavior into branch numbers. Section 2.2.2 identifies the events that can occur during cyclic loading. Section 2.2.3 presents relevant portions of the algorithm that has been used to implement the modified S-type stress-strain behavior in the SMEV15 and SMSD15 subroutines. The source code for the subroutines is provided in Appendix A.

2.2.1 Branch Numbers

The modified S-type cyclic stress-strain behavior was discretized into ten unique branches, which are identified in Figure 2.3 and described in Table 2.1.

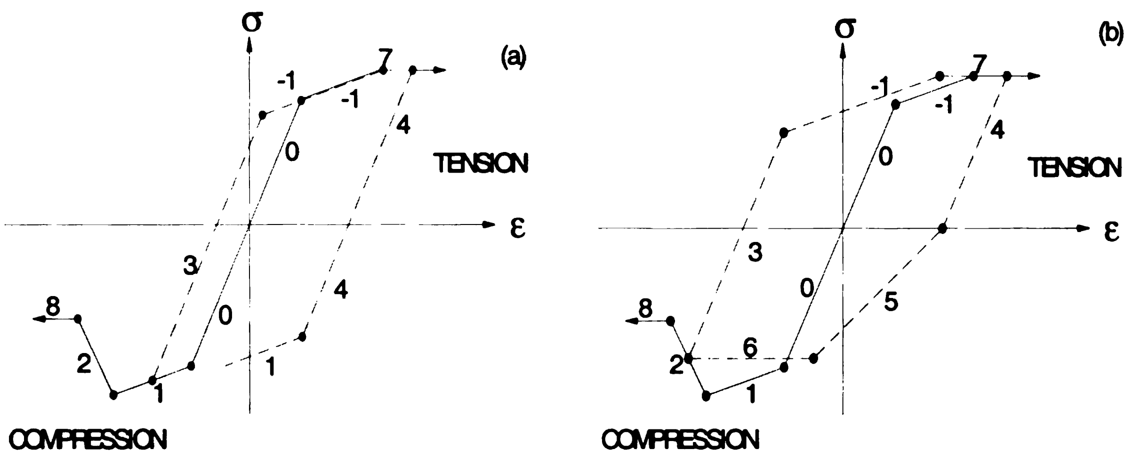


Figure 2.3 Discretization of the modified S-type cyclic stress-strain curve into branches for fibers: (a) before; and (b) after compression strain softening

Table 2.1 Description of branch numbers

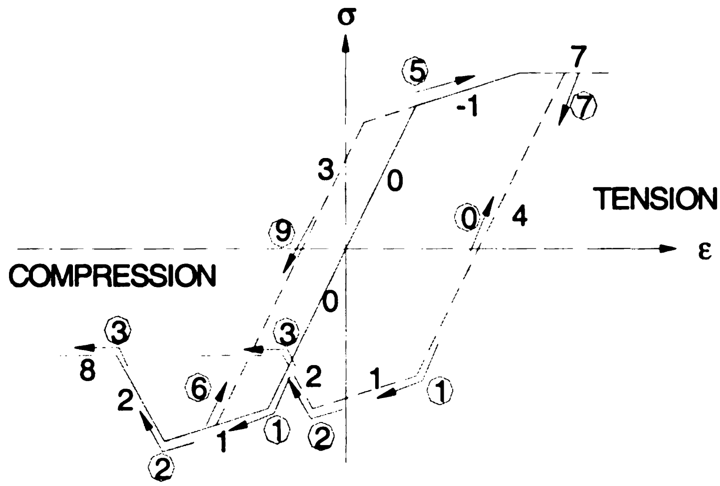
Branch Number	Description
0	Elastic branch
1	Compression strain-hardening branch
-1	Tension strain-hardening branch
2	Compression softening branch
3	Unloading/reloading from/to compression branch
4	Unloading/reloading from/to tension branch
5	First shooting branch
6	Second shooting branch
7	Ultimate tensile branch
8	Ultimate compressive branch

2.2.2 Events Numbers

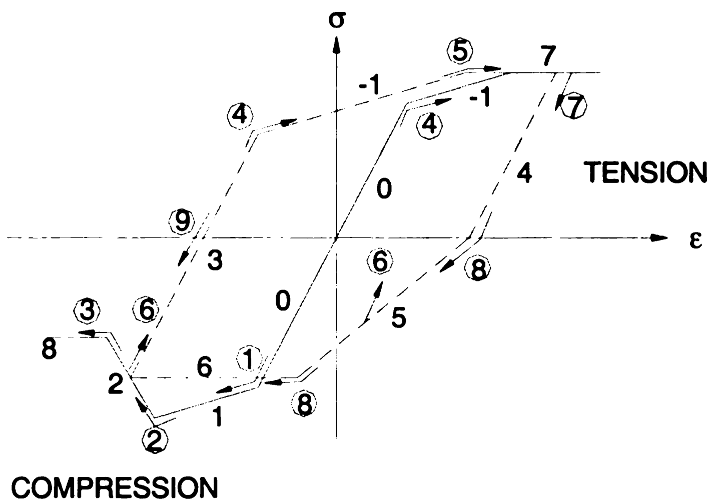
The changes in stiffnesses associated with the stress-strain response states moving from one branch to the other under cyclic loading are identified using event numbers. These event numbers are shown in Figure 2.4, where the arrows indicate the direction of loading and the event numbers are circled. The branch numbers are also indicated in Figure 2.4. A brief description of each event is given in Table 2.2 below.

Table 2.2 Description of events

Event No.	Description	Path		Strain Increment	Special Comments
		From branch	To branch		
1	Yielding in compression	0	1	Compressive	
2	Compression Strain softening	1	2	Compressive	
3	Compression Ultimate (Post-buckling)	2	8	Compressive	
4	Yielding in tension	0	-1	Tensile	
5	Tension ultimate	-1	7	Tensile	
6	Unloading from compression	1,2,5,6,8	3	Tensile	Stress reversal from compression
7	Unloading from tension envelope towards compression	-1,7	4	Compressive	Stress reversal from tension
8	Shooting from tension envelope	4 5	5 6	Compressive	Occurs only after compression strain softening
9	Reloading onto compression envelope	3 6	1,2,8 2,8	Compressive	
0	Reloading onto tension envelope	4	-1,7	Tensile	Occurs due to strain reversal on unloading/reloading branch



(a) Before compression strain softening



(b) After compression strain softening

Figure 2.4 Events for the modified S-type cyclic stress-strain curve (a) before; and (b) after compression strain softening

2.2.3 Event Calculation and State Determination Algorithm

The algorithm for determining the response state (stress σ, strain ε, tangent stiffness E_{mod}, and branch number) of the fiber subjected to strain increments ($d\varepsilon$) is presented in this section. The response state of the fiber will depend on its initial (current) state, cyclic loading history, magnitude and direction of $d\varepsilon$, and the occurrence of events.

If an event occurs, the strain increment $d\varepsilon$ is scaled down using a *calculated* event factor (FACT) to the value (FACT•$d\varepsilon$), and the event, response state and branch number of the fiber after the event are determined. The remaining portion of the strain increment (1-FACT) •$d\varepsilon$ is applied incrementally using the calculated response state as the initial state, and so on. If no events occur, then the event factor (FACT) is equal to 1.0 and the complete $d\varepsilon$ can be applied without change in stiffness and the response state can be calculated using the initial state (stress, strain, tangent stiffness) alone.

The algorithm for determining the response state (stress, strain, tangent stiffness, and branch number) of the fiber subjected to *compressive and tensile* strain increments is presented in the following two sub-sections. The state determination algorithm is presented with reference to the initial state (stress, strain, tangent stiffness, and branch number) of the fiber and its partial cyclic response history (required for branch number 2, 3, 4, 5, and 6). The partial cyclic response history terms that are monitored and updated include:

- Lastt – last tensile branch before strain reversal
- Lastc – last compression branch before strain reversal

- σ_c – compressive stress when the last strain reversal occurred (always on lastc)
- σ_T – tensile stress when the last strain reversal occurred (always on lastt)
- ε_{min} – strain corresponding to σ_c
- ε_{max} – strain corresponding to σ_T

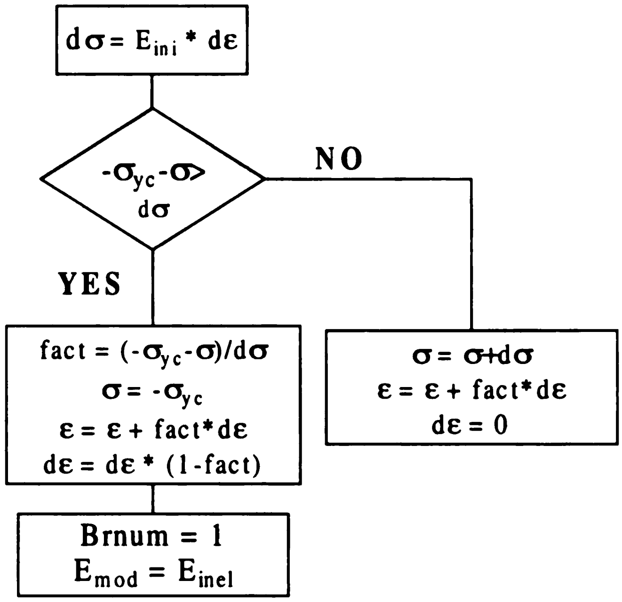
The algorithm is illustrated for each initial branch case using flowcharts and loading path diagrams.

2.2.3.1 Compressive strain increments

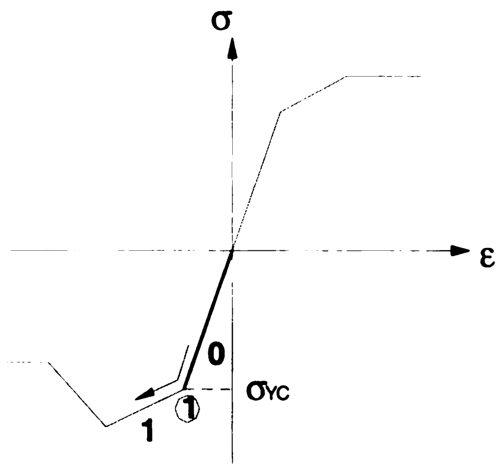
The compressive strain increment has negative sign and the behavior on the various initial branches is as follows:

- Initial branch = Branch number 0

  The fiber will behave elastically with tangent stiffness equal to E_{ini} until compression yielding (event no. 1) occurs. As shown in Figure 2.5, if the strain increment $d\varepsilon$ causes event no. 1, then the stresses and strains are updated, and the branch number and tangent stiffness are set equal to 1 and E_{inel}, respectively.



(a) Flowchart



(b) Loading path diagram

Figure 2.5 Behavior of branch number 0 for compressive strain increments

- Initial branch = Branch number 1

The fiber will have a tangent stiffness equal to E_{inel} on this compression strain-hardening branch until strain softening (event no. 2) occurs. As shown in Figures 2.6 and 2.7, if the strain increment dε causes event no. 2, then the stresses and strains are updated, and the branch number and tangent stiffness are set equal to 2 and E_{int}, respectively. The occurrence of event no. 2 depends on whether the fiber has undergone tension yielding during its cyclic response history, which is established using the cyclic response variable LASTT as shown in Figure 2.7.

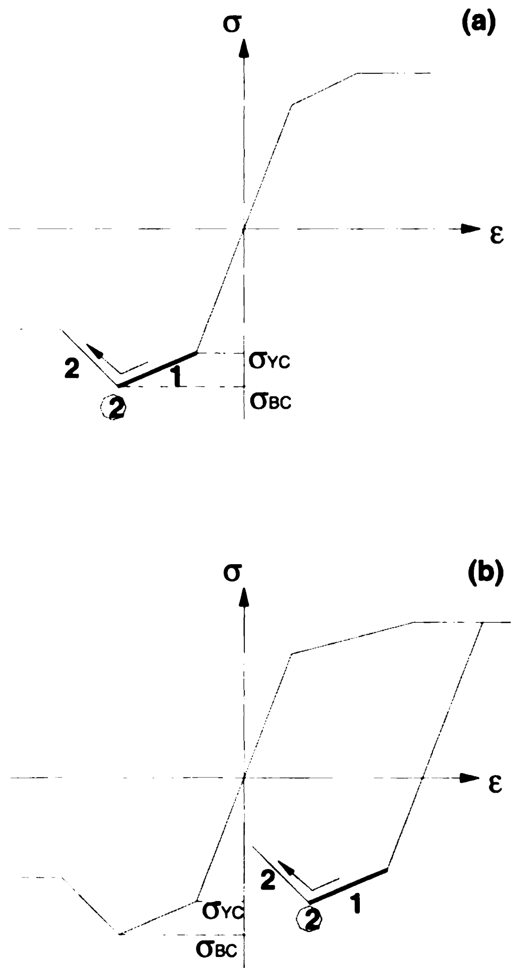


Figure 2.6 Behavior of branch number 1 for compressive strain increments: (a) starting in compression; and (b) starting in tension

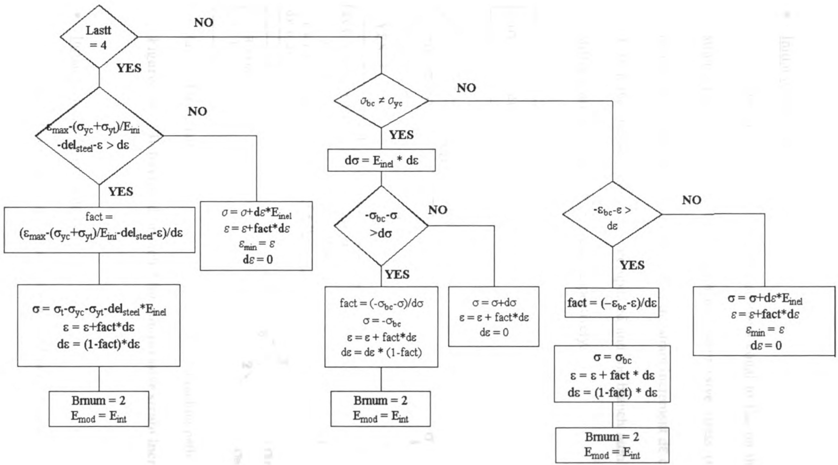
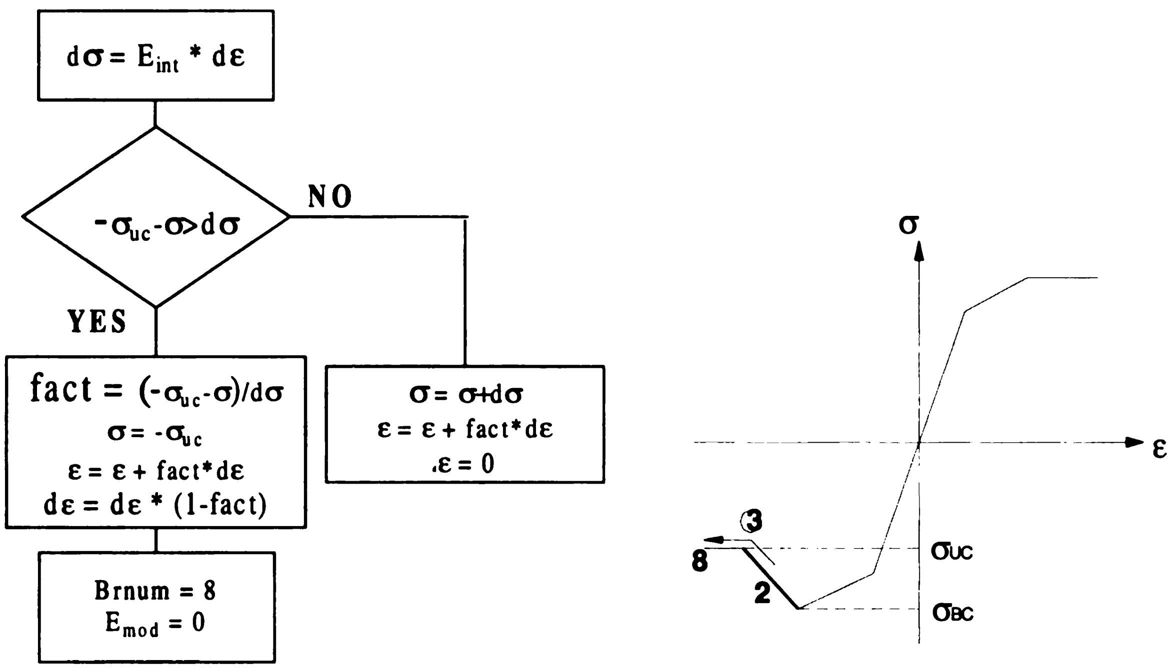


Figure 2.7 **Flowchart for the behavior of branch no. 1 for compressive strain increments**

- <u>Initial branch = Branch number 2</u>

The fiber will have a tangent stiffness equal to E_{int} on this compression strain-softening branch until the ultimate compressive stress (σ_{uc}) is reached (event no. 3). As shown in Figure 2.8, if the strain increment dε causes event no. 3, then the stresses and strains are updated and the branch number and tangent stiffness are set equal to 8 and zero, respectively.



(a) Flowchart (b) Loading path diagram

Figure 2.8 Behavior of branch 2 for compressive strain increments

- <u>Initial branch = Branch number 3</u>

The fiber will have a tangent stiffness equal to E_{ini} on this unloading/reloading branch until the compressive stress (σ_c) is reached (event no.

9). σ_c is the compressive stress where the fiber initially went into strain reversal (branch number 3). As shown in Figure 2.9, σ_c can be on branch numbers 1, 2, 5, 6, or 8. If σ_c is on branch 1 and $d\varepsilon$ causes event 9, then the stress-state is updated and the branch number and the tangent stiffness are set equal to 1 and E_{inel}, respectively. As shown in Figure 2.10, corresponding behavior is enforced for σ_c located on branches 2, 8, 5, or 6. In Figure 2.10, the terms $\varepsilon_{\sigma o}$ and E_{ab} correspond to behavior on branch 5 and are explained later.

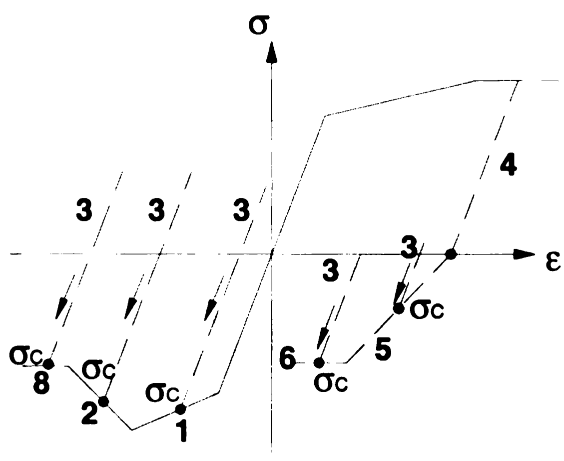


Figure 2.9 Behavior on branch 3 for compressive strain increment

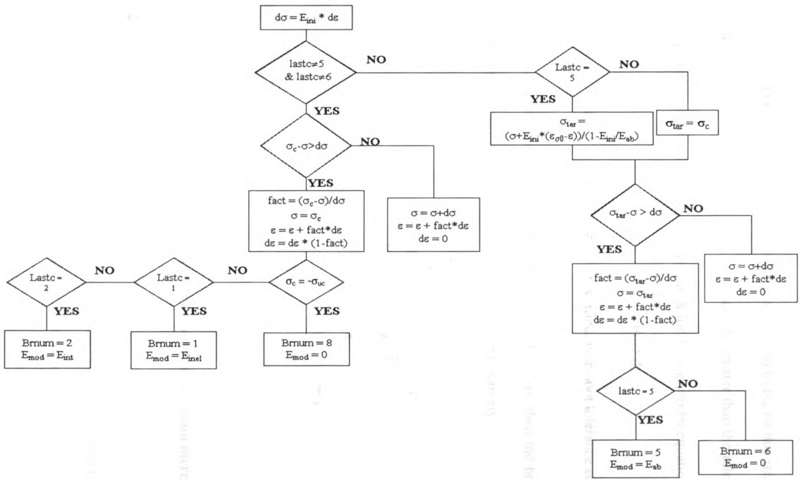


Figure 2.10 Flowchart of behavior on branch 3 for compressive strain increments

- <u>Initial branch = Branch number –1</u>

The fiber will have a tangent stiffness equal to E_{sh} on this strain-hardening branch until a compressive strain increment $d\varepsilon$ greater than the tolerance limit is applied (event no. 7). The tolerance check for $d\varepsilon$ is given by equation (1).

$$\sigma_{tol} \leq E_{ini} \cdot d\varepsilon \quad (1)$$

where: σ_{tol} = specified overshoot tolerance stress.

As shown in Figure 2.11, if event no. 7 occurs, then the branch number and tangent stiffness are set equal to 4 and E_{ini}, respectively.

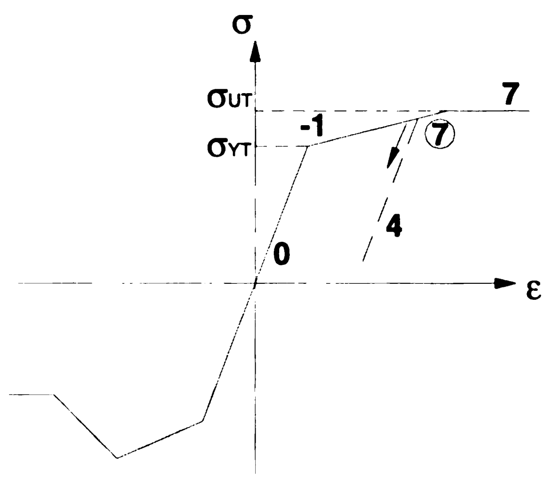


Figure 2.11 Behavior on branch –1 for compressive strain increments

- <u>Initial branch = Branch number 4</u>

The behavior of the fiber on this unloading/reloading branch from tension will depend on whether compression strain softening has occurred during its cyclic loading history. If compression strain softening has not occurred, then the response will remain on branch number 4 until compression yielding (event no. 1) occurs. If compression strain softening has occurred during its cyclic loading history, then the response will remain on branch number 4 until the zero stress state (event no. 8) is reached.

As shown in Figure 2.12 (a) and 2.13, if event no. 1 occurs, the branch number and tangent stiffness are equal to 1 and E_{inel}, respectively. As shown in Figure 2.12 (b) and 2.13, if the event no. 8 occurs, the branch number and the tangent stiffness are equal to 5 and E_{ab}, respectively.

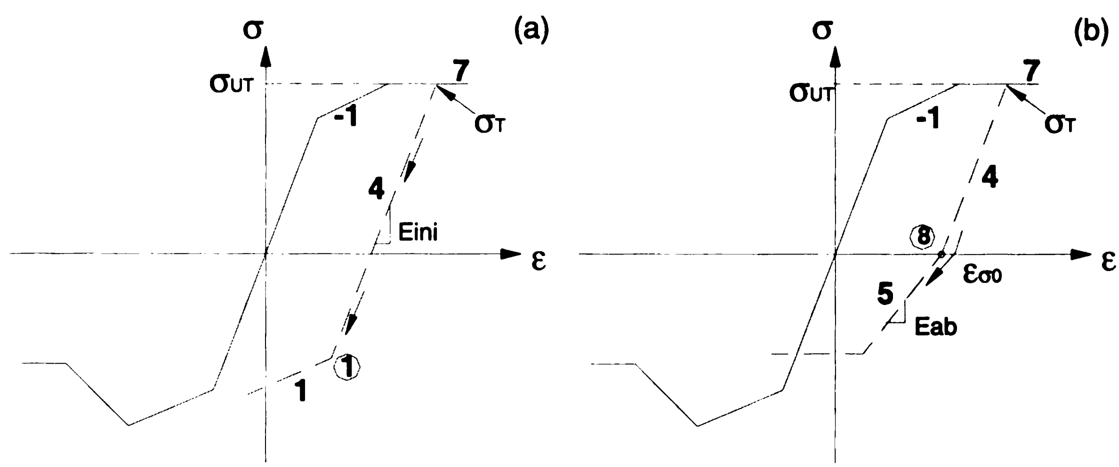


Figure 2.12 Behavior on branch 4 for compressive strain increment if: (a) before compression strain softening; and (b) after compression softening

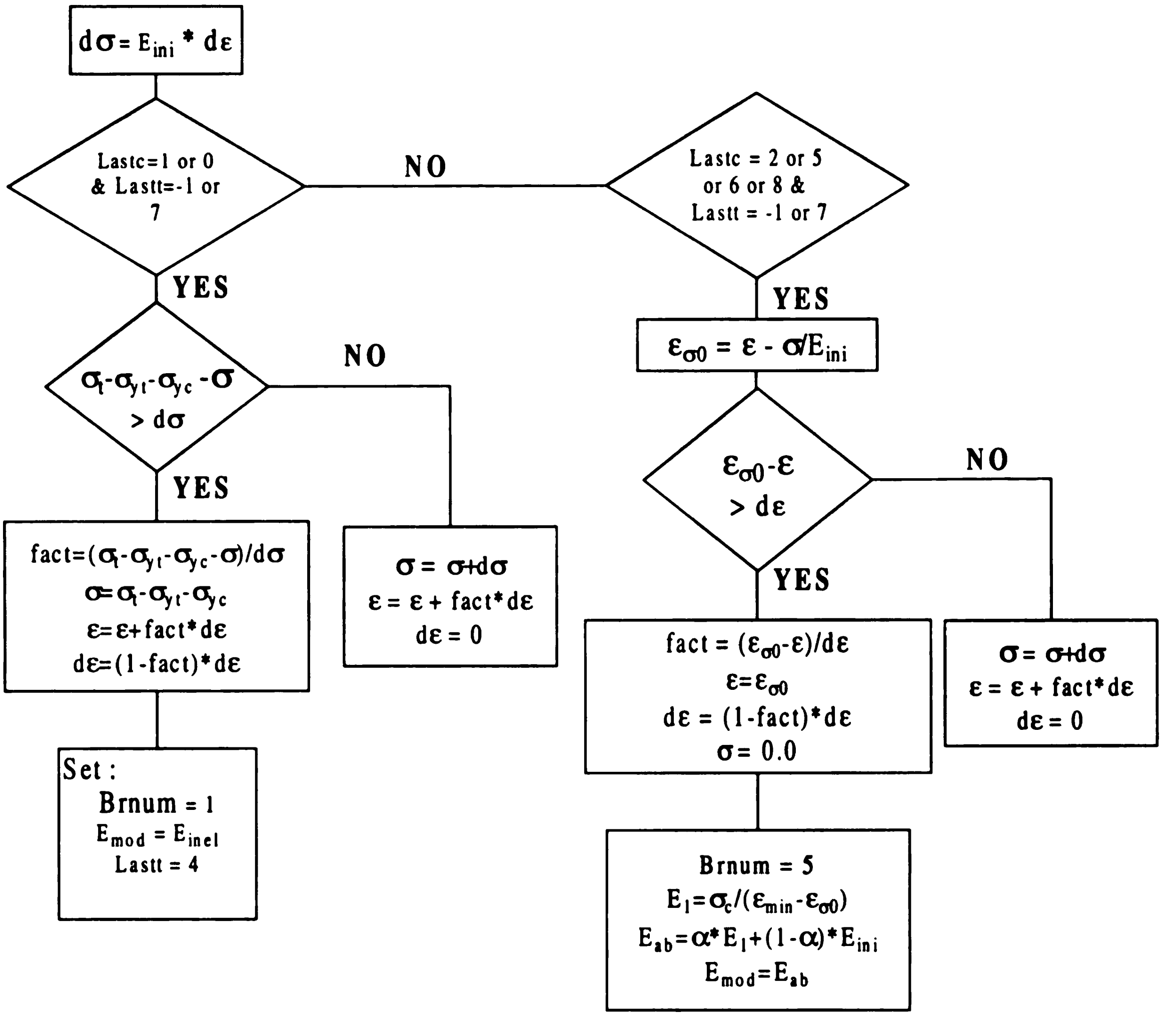


Figure 2.13 Flowchart of the behavior on branch 4 for compressive strain increments

- <u>Initial branch = Branch number 5 (first shooting branch)</u>

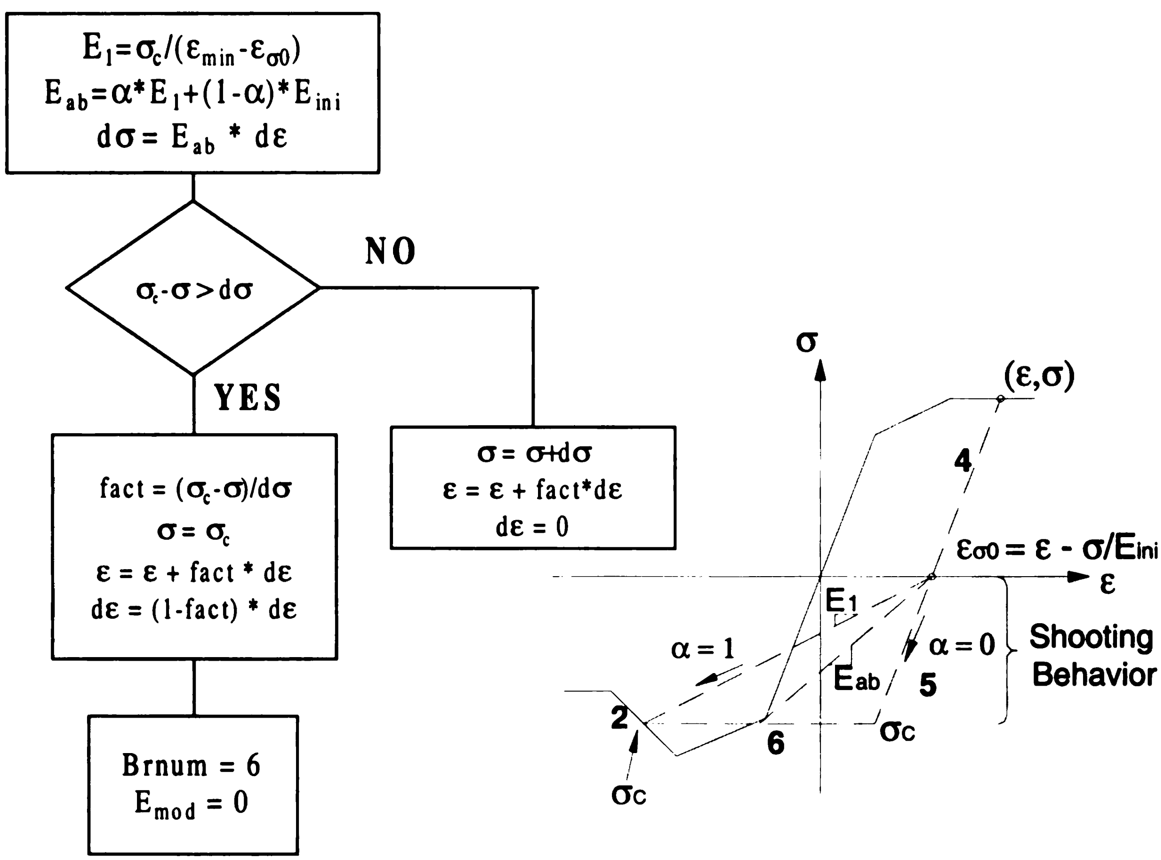
The fiber will have a tangent stiffness equal to E_{ab} on this shooting branch until the compressive stress (σ_c) at the last strain reversal is reached. E_{ab} is given by equation (2) and it is a function of α, E_{ini} and E_1, where α is specified by the user and E_1 is calculated as shown in Figure 2.14 (a). For α equal to zero and one, E_{ab} will be equal to E_{ini} and E_1, respectively. For values of $0 < \alpha < 1$, $E_1 <$

$E_{ab} < E_{ini}$. As shown in Figure 2.14, if σ_c is reached due to $d\varepsilon$, the branch number and tangent stiffness will be set equal to 6 and zero, respectively.

$$E_{ab} = \alpha \cdot E_l + (1-\alpha) \cdot E_{ini} \tag{2}$$

where:

E_{ini}	=	initial material stiffness
E_l	=	shooting stiffness from $(\varepsilon_{\sigma o}, 0)$ to $(\varepsilon_{min}, \sigma_c)$
E_{ab}	=	shooting stiffness



(a) Flowchart (b) Loading path diagram

Figure 2.14 Behavior on branch 5 for compressive strain increments

- <u>Initial branch = Branch number 6 (second shooting branch)</u>

The fiber will have a tangent stiffness equal to zero on this second shooting branch until the compressive strain at last strain reversal (ε_{min}) is reached (event no. 9). As shown in Figures 2.15 (a) and (b), $\varepsilon\varepsilon$ is the length of branch no. 6 and σ_c can be on branch number 2 or 8. If σ_c is on branch 2 and $d\varepsilon$ causes event no. 9, then the branch number and the tangent stiffness are set equal to 2 and E_{int}, respectively. Otherwise, as shown in Figure 2.15 (b), the branch number will be set to 8 and the tangent stiffness will remain equal to zero.

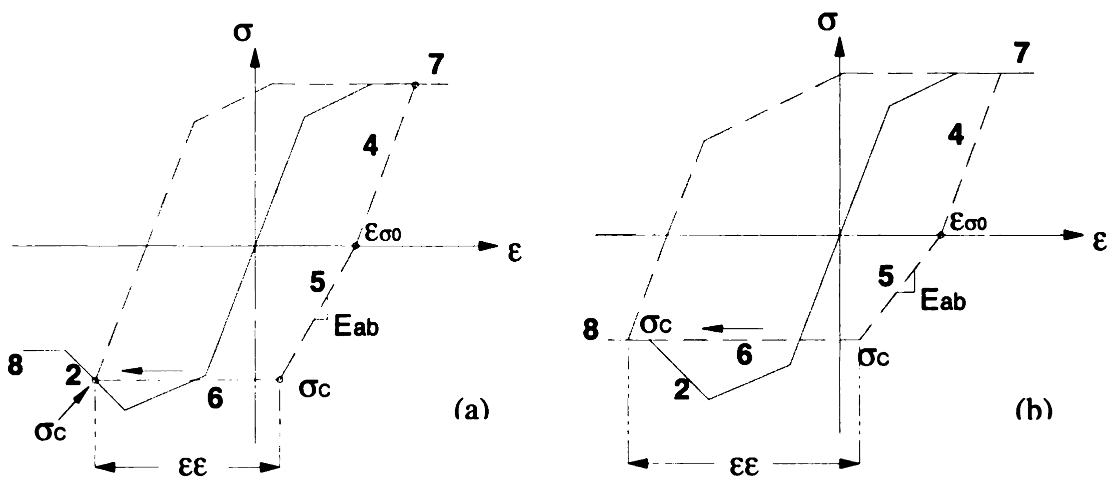


Figure 2.15 Behavior on branch 6 for compressive strain increments and strain reversal from: (a) branch 2; and (b) branch 8

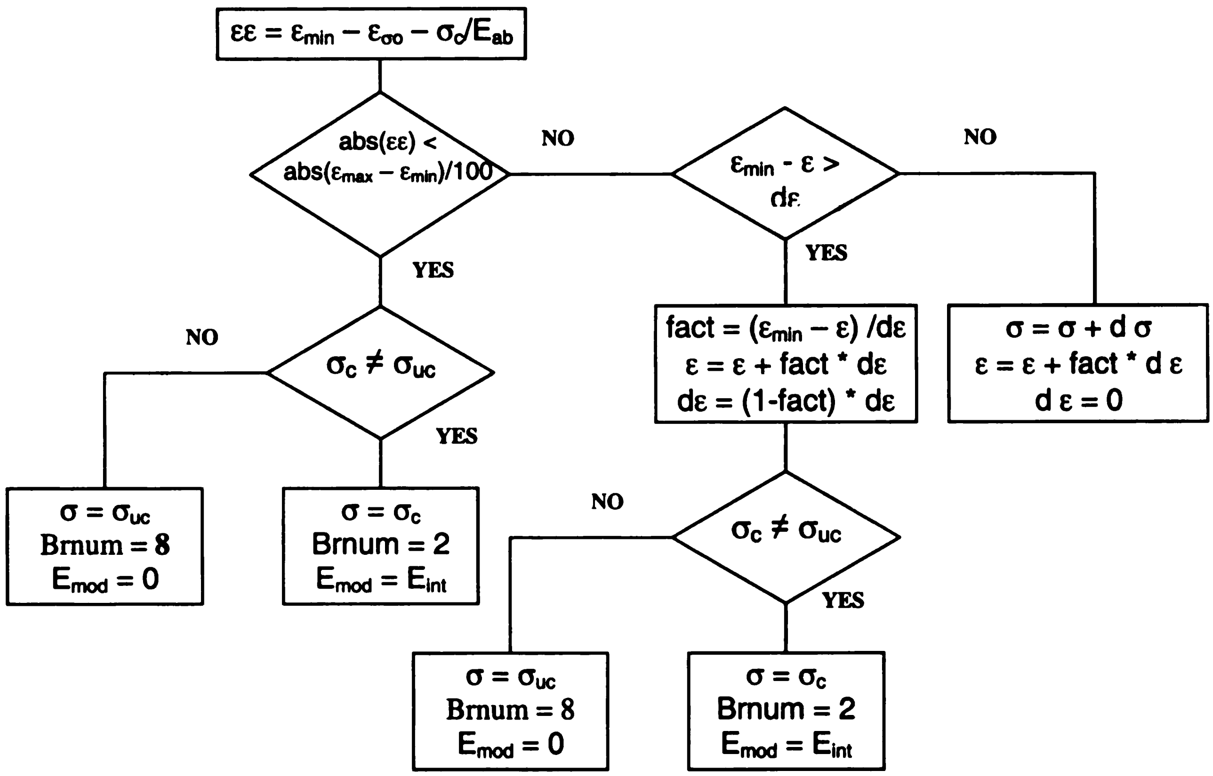


Figure 2.16 Flowchart of the behavior on branch 6 for compression strain increments

- Initial branch = Branch number 7

The fiber will have a tangent stiffness equal to zero on this ultimate tensile branch until a compressive strain increment $d\varepsilon$ greater than the tolerance limit is applied (event no. 7). The tolerance check was presented in equation (1). As shown in Figure 2.17, if event no. 7 occurs, the branch number and the tangent stiffness are set equal to 4 and E_{ini}, respectively.

- Initial branch = Branch number 8

The fiber will have a tangent stiffness equal to zero on this ultimate compressive branch. No events are associated on this branch while subjected to additional compressive strain increments since there is no change in stiffness (see Figure 2.17).

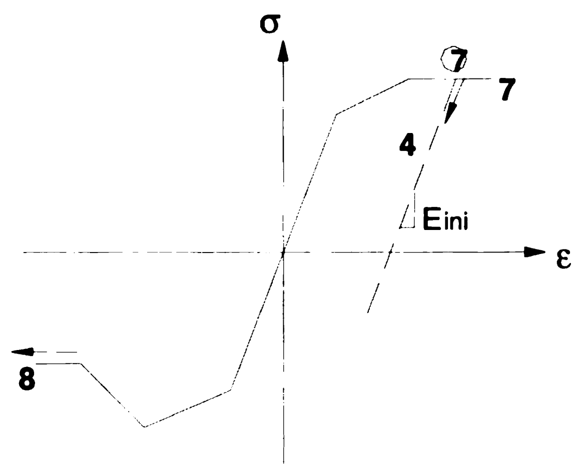


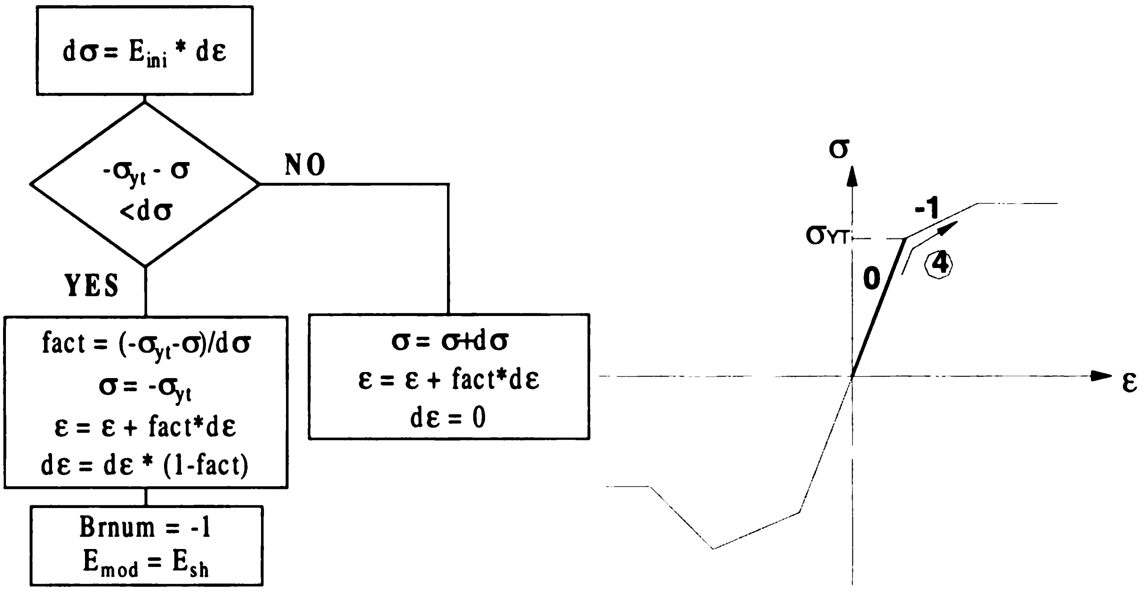
Figure 2.17 Behavior of branches 7 and 8 for compression strain increments

2.2.3.2 Tensile strain increments

On the other hand, the tensile strain increment has positive sign and the behavior on the various initial branches is as follows:

- <u>Initial branch = Branch number 0</u>

  The fiber will behave elastically with tangent stiffness equal to E_{ini} until tension yielding (event no. 4) occurs. As shown in Figure 2.18, if the strain increment $d\varepsilon$ causes event no. 4, then the stresses and strains are updated, and the branch number and tangent stiffness are set equal to -1 and E_{sh}, respectively.

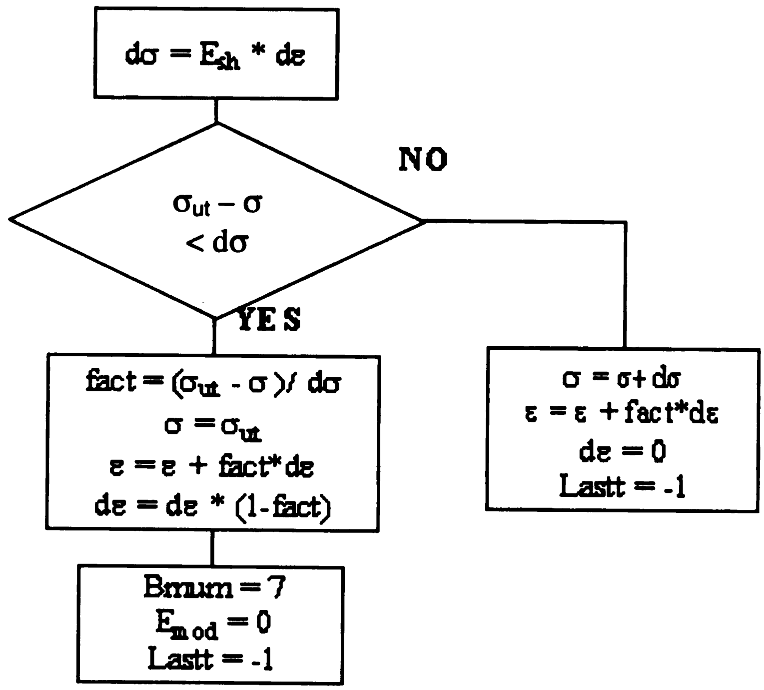


(a) Flowchart (b) Loading path diagram

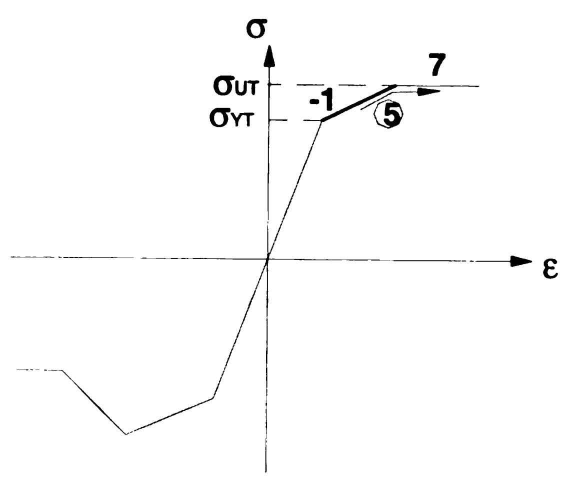
Figure 2.18 Behavior of branch number 0 for tensile strain increments

- <u>Initial branch = Branch number -1</u>

The fiber will have a tangent stiffness equal to E_{sh} on this tension strain-hardening branch until tension ultimate (event no. 5) occurs. As shown in Figure 2.19, if the strain increment $d\varepsilon$ causes event no. 5, then the stresses and strains are updated, and the branch number and tangent stiffness are set equal to 7 and zero, respectively.



(a) Flowchart



(b) Loading path diagram

Figure 2.19 Behavior on branch –1 for tensile strain increments

- Initial branch = Branch number 1

The fiber will have a tangent stiffness equal to E_{inel} on this compression strain-hardening branch until a tensile strain increment $d\varepsilon$ greater than the tolerance limit is applied (event no. 6). The tolerance check for $d\varepsilon$ is given by equation (1). As shown in Figure 2.20, if event no. 6 occurs, then the branch number and tangent stiffness are set equal to 3 and E_{ini}, respectively.

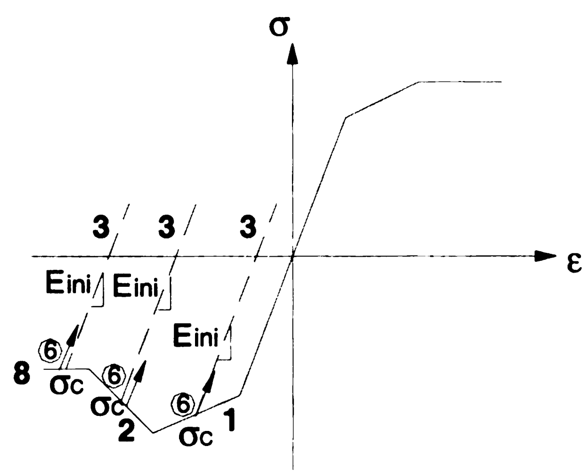


Figure 2.20 Behavior of branches 1, 2, and 8 for tensile strain increments

- Initial branch = Branch number 2

The fiber will have a tangent stiffness equal to E_{int} on this compression strain-softening branch until a tensile strain increment $d\varepsilon$ greater than the tolerance limit is applied (event no. 6). The tolerance check for $d\varepsilon$ is given by equation (1). As shown in Figure 2.20, if event no. 6 occurs, then the branch number and tangent stiffness are set equal to 3 and E_{ini}, respectively.

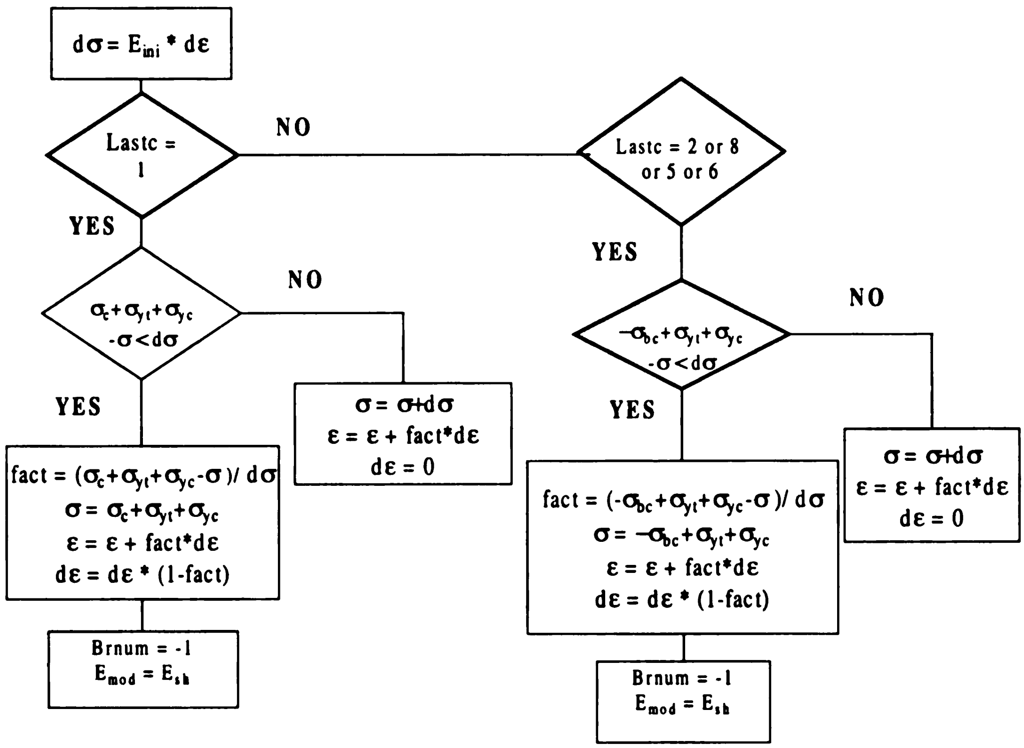
- Initial branch = Branch number 8

The fiber will have a tangent stiffness equal to zero on this ultimate compressive branch until a tensile strain increment $d\varepsilon$ greater than the tolerance limit is applied (event no. 6). The tolerance check for $d\varepsilon$ is given by equation (1). As shown in Figure 2.20, if event no. 6 occurs, then the branch number and tangent stiffness are set equal to 3 and E_{ini}, respectively.

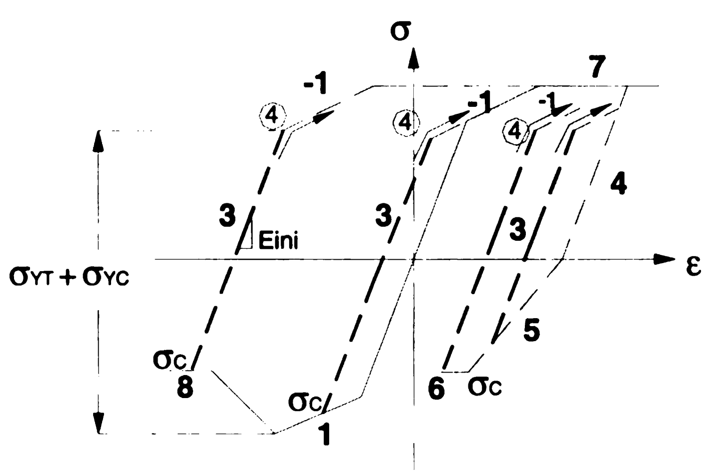
- Initial Branch = Branch number 3

The fiber will have a tangent stiffness equal to E_{ini} on this unloading/reloading from compression branch until tension yielding (event no. 4) occurs. As shown in Figure 2.21, the occurrence of tension yielding will depend on whether compression strain softening has occurred. If compression strain softening has not occurred (lastc = 1), the fiber will follow kinematic hardening rules with respect to its current stress. If compression strain softening has occurred (lastc $\neq$ 1), the fiber will follow kinematic hardening with respect to the stress at compression strain softening (σ_{bc}).

As shown in Figure 2.21, if event no. 4 occurs, the branch number and tangent stiffness are equal to -1 and E_{sh}, respectively.



(a) Flowchart



(b) Loading path diagram

Figure 2.21 Behavior on branch 3 for tensile strain increments

- <u>Initial branch = Branch number 4</u>

The fiber will have a tangent stiffness equal to E_{ini} on this unloading/reloading tensile branch until the stress at last tensile strain reversal (σ_T) is reached (event no. 0). As shown in Figure 2.22, σ_T can be on branch numbers −1 or 7. If σ_T is on branch −1 and $d\varepsilon$ causes event 0, then the stress-state is updated and the branch number and the tangent stiffness are set equal to -1 and E_{sh}, respectively. The same behavior is enforced for σ_T located on branch 7. In this case, the branch number and the tangent stiffness will be set to 7 and zero, respectively.

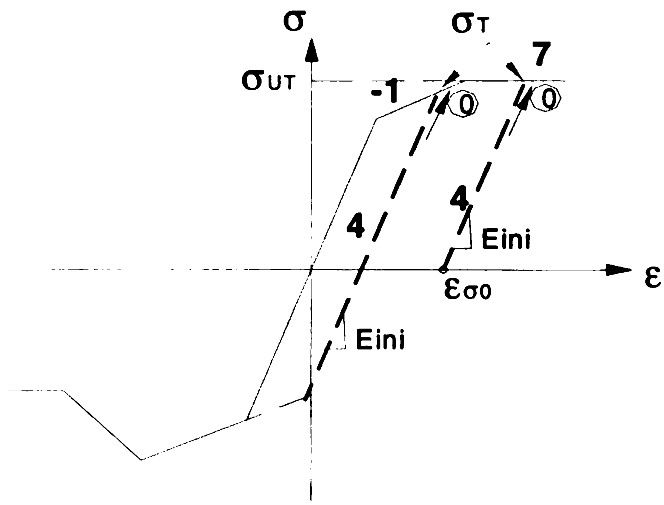


Figure 2.22 Loading path diagram for branch 4 for tensile strain increments

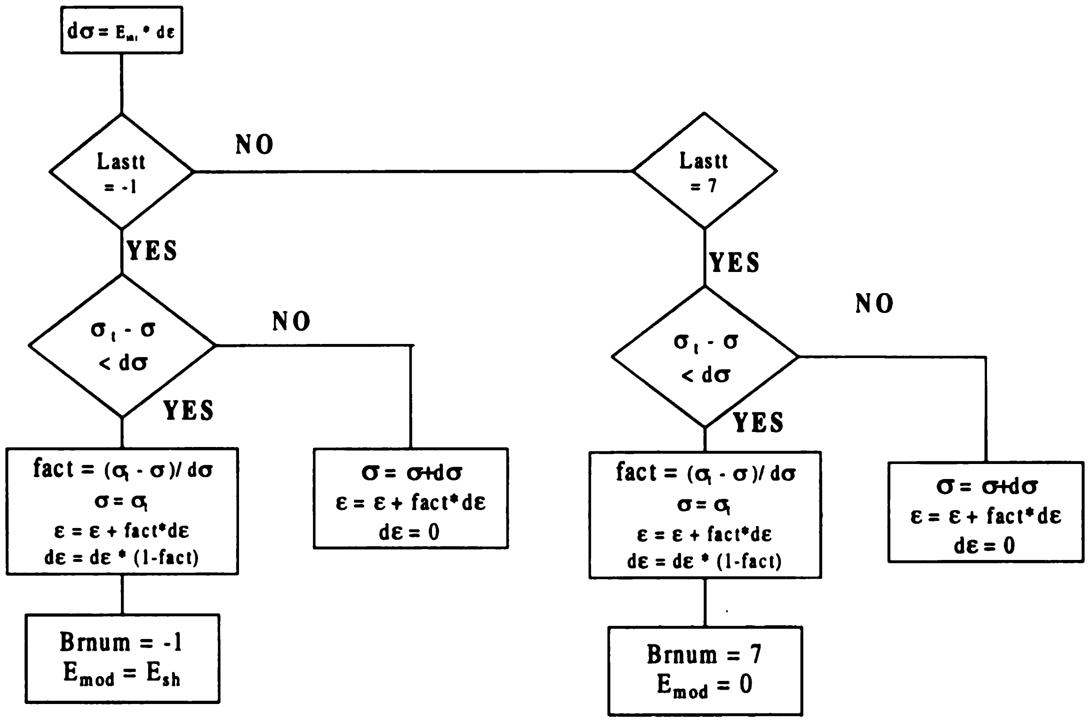


Figure 2.23 Flowchart of behavior on branch 4 for tensile strain increments

- Initial branch = Branch number 5 (first shooting branch)

The fiber will have a tangent stiffness equal to E_{ab} on this shooting branch until a tensile strain increment dε greater than the tolerance limit is applied (event no. 6). The tolerance check for dε is given by equation (1). As shown in Figure 2.24, if event no. 6 occurs, then the branch number and tangent stiffness are set equal to 3 and E_{ini}, respectively.

- Initial branch = Branch number 6 (second shooting branch)

The fiber will have a tangent stiffness equal to zero on this second shooting branch until a tensile strain increment dε greater than the tolerance limit is applied (event no. 6). The tolerance check for dε is given by equation (1). As shown in Figure 2.24, if event no. 6 occurs, then the branch number and tangent stiffness are set equal to 3 and E_{ini}, respectively.

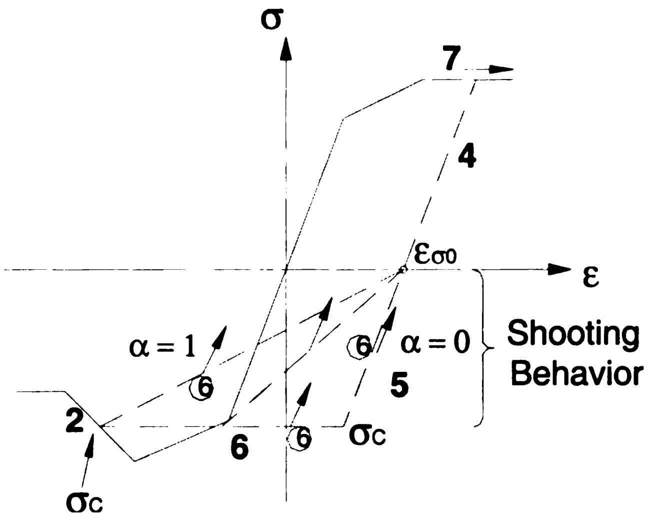


Figure 2.24 Reloading from shooting branches

- <u>Initial branch = Branch 7</u>

The fiber will have a tangent stiffness equal to zero on this ultimate tensile branch. No events occur on this branch under additional tensile strain increments since there is no change in stiffness (see Figure 2.24).

2.3 SUMMARY

A modified S-type steel stress-strain curve was developed for the fiber-based beam-column element in Drain2DX. This stress-strain curve can model various aspects of steel behavior such as compression and tension strain hardening, compression strain softening due to local buckling or biaxial stresses, kinematic hardening under cyclic loadings, and the effects of local buckling and biaxial stresses on the hysteretic behavior.

This is a significant improvement over the S-type steel stress-strain curve previously available in Drain2DX.

Since Drain2DX uses an event-to-event based solution algorithm, the cyclic stress-strain curve was discretized into ten unique branch numbers and ten event numbers corresponding to changes in stiffness due to compressive and tensile strain increments. The response state depends on the initial (current) state, partial cyclic loading history, strain magnitude and direction, and the occurrence of events. The event calculation and state determination algorithm were developed accordingly and implemented in the corresponding SMEV15 and SMSD15 event calculation and state determination subroutines in Drain2DX.

CHAPTER 3. VERIFICATION OF MODIFIED FIBER-BASED ELEMENT

3.1 GENERAL

The implementation of the modified S-type steel stress-strain curve in Drain2DX was verified using three analytical models with increasing levels of complexity. Section 3.2 presents the results of monotonic and cyclic analysis of a two-fiber steel cross-section subjected to axial loading. This simple analysis was conducted to specifically verify the implemented modified S-type stress-strain curve. Section 3.3 presents the results of monotonic and cyclic analysis of a steel column subjected to lateral loading. Section 3.4 presents the results of monotonic and cyclic analysis of high strength CFT beam-columns tested by Varma et al. (2001). The results of the fiber analysis are also compared with the experimental results of Varma et al. (2001). Section 3.5 presents a summary of chapter 3. Section 3.6 presents recommendations for using the modified fiber-based element in Drain2DX.

3.2 FIBER ANALYSIS OF SIMPLE CROSS-SECTION

In order to verify the implementation of the modified S-type steel stress-strain curve in Drain2DX, a simple analytical model of a steel bar was developed and analyzed for monotonic and cyclic axial loading. Figure 3.1 shows a schematic of the analytical model. As shown in Figure 3.1, the fiber-based element had a length of 1.0 in. The bottom node (1) of the element was fixed and the top node (2) was free. The element had only one segment, i.e., only one slice located at mid-length. Figure 3.2 shows the discretization of the cross-section (Area = 1.0 in^2) into two fibers. The fiber stress-strain curve is shown in Figure 3.3 and the numerical values are provided in Table 3.1.

As shown in Figure 3.1, the analytical model was analyzed for three loading conditions: (1) monotonic tension, (2) monotonic compression; and (3) cyclic loading. The external load (P) is equal to the stress in the section fiber (area = 1.0 in^2) and the displacement (Δ) of node 2 (length = 1.0 in) is equal to the strain in the fibers. Thus the load-displacement from the fiber analysis is also the stress-strain response of the cross-section fibers. Figures 3.3 and 3.4 show the results from the fiber analysis for the monotonic tension and compression loadings, respectively. As shown in Figure 3.3 and 3.4, the results from the fiber analysis follow the monotonic stress-strain curve input to the program perfectly, thus validating the implementation of the modified S-type steel stress-strain curve in Drain2DX.

Table 3.1 Fiber stress-strain curve for steel bar

	Stress (ksi)	Strain (in/in)
Compression	90	0.003
	100	0.005
	70	0.008
Tension	90	0.003
	100	0.010

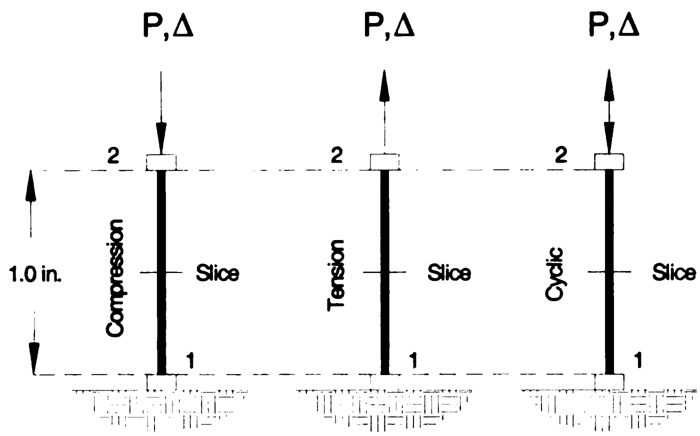


Figure 3.1 Schematic of fiber element for steel bar

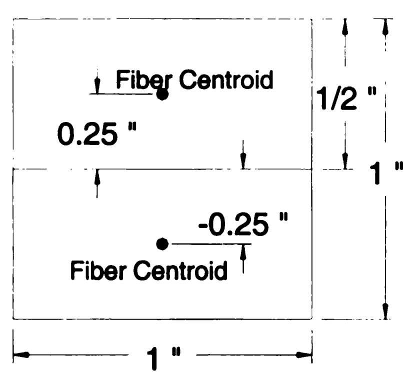


Figure 3.2 Fiber discretization of steel bar cross-section

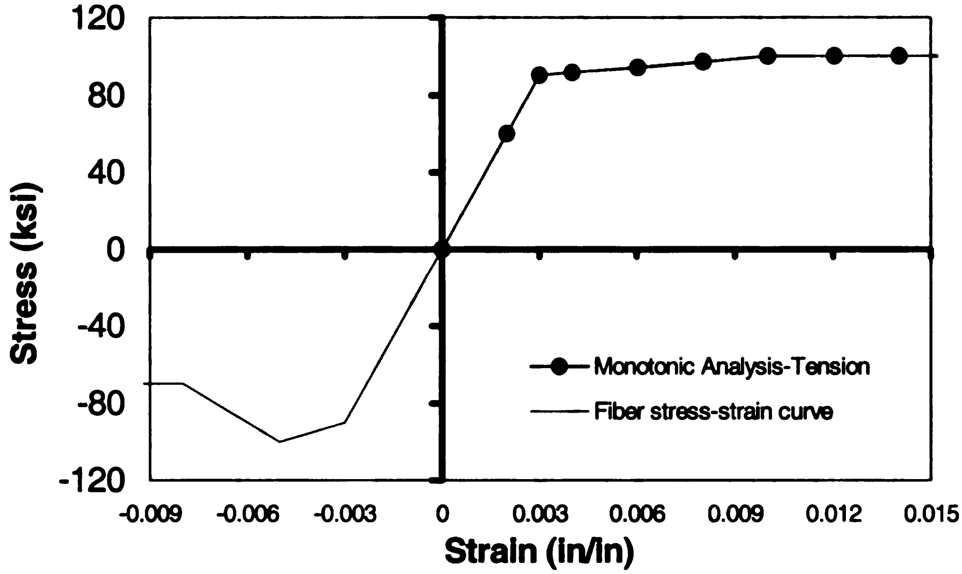


Figure 3.3 Axial load - tensile monotonic analysis

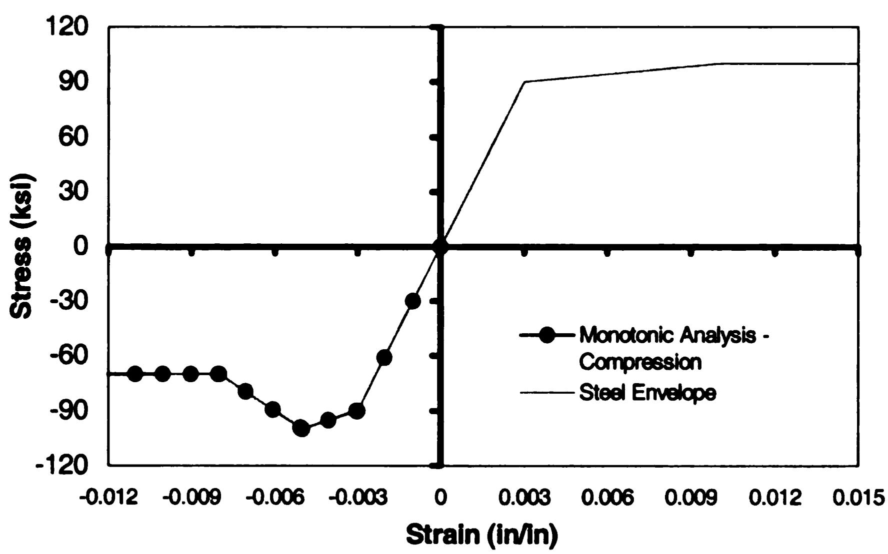


Figure 3.4 Axial load – compressive monotonic analysis

The analytical model was subjected to cyclic loading history so that all branches and events identified in Section 2.2 and 2.3 would occur. The cyclic loading history included cycles at 0.75 ε_y, 2.0 ε_y, and 4.0 ε_y. The fiber analyses were conducted for values of the unloading parameter α equal to 0.0, 0.5 and 1.0. The results from the fiber analysis for cyclic loading are shown in Figure 3.5.

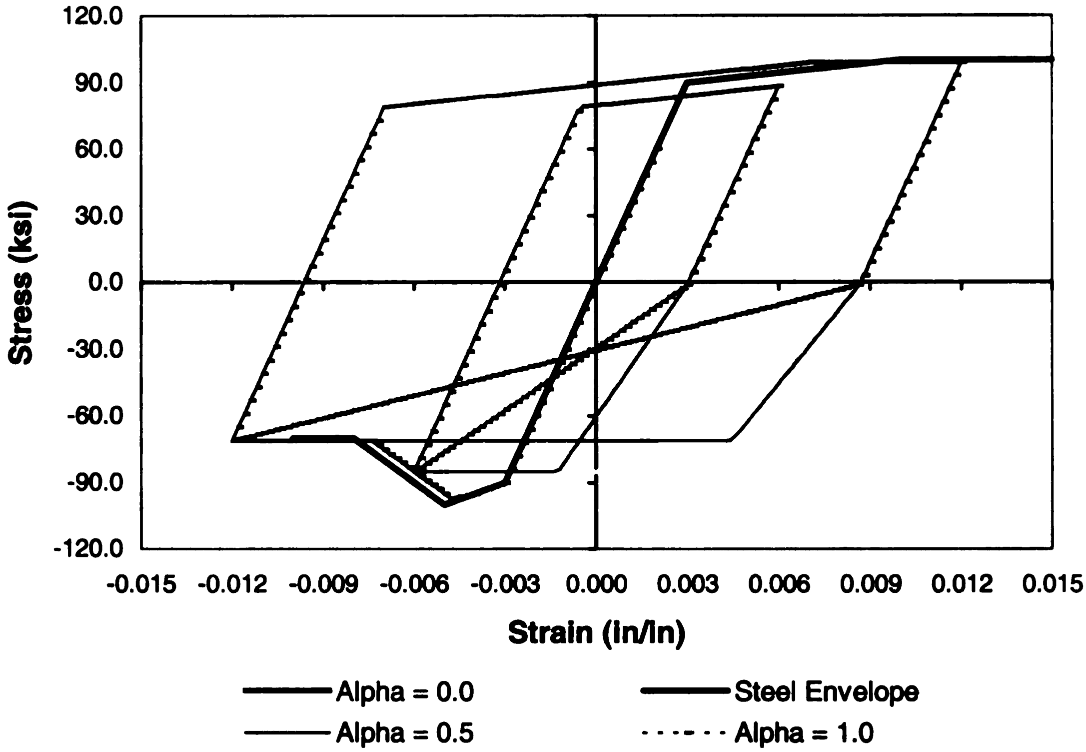


Figure 3.5 Axial load – cyclic analysis

As shown in Figure 3.5, the results from the fiber analysis follows the implemented modified S-type steel stress-strain behavior (Figure 2.3 and 2.4) closely.

3.3 FIBER ANALYSIS OF A STEEL COLUMN

In order to verify the modified fiber-based element in Drain2DX, a simple analytical model of a 10 feet long W14x82 cantilever steel beam-column was developed and analyzed for monotonic and cyclic lateral loading. Figure 3.6 shows a schematic of the analytical model. As shown in Figure 3.6, the fiber-based element was divided into five equal segments. The element was fixed at one end (node 1) and free at the top (node 2) end. Figure 3.7 shows the discretization of the cross-section into six fibers. The fiber stress-strain curve was the same used for the steel bar analysis in the previous section. It is shown in Figure 3.3 and listed in Table 3.1.

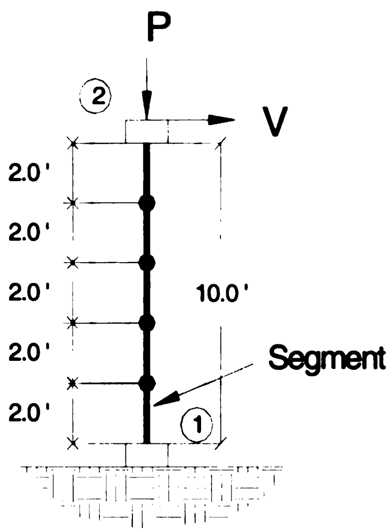


Figure 3.6 Pushover analysis load schematic

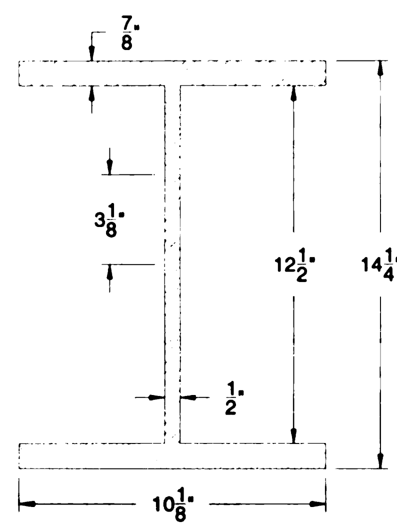


Figure 3.7 W14x82 cross -section discretization

The analytical model was analyzed for two loading conditions: (1) constant axial load (P) and monotonically increasing lateral loading (V), and (2) constant axial load (P) and cyclically varying lateral load (V). The constant axial load was equal to 50 kips, which is 11.1% of the axial load capacity ($\phi_c P_n$) (AISC 1999). Figure 3.8 shows the lateral load-displacement response from the monotonic fiber analysis. Under this analysis, the column reaches it lateral load capacity of 120 kips at a lateral load displacement of 3.3 in. The column lateral load resistance decreases due to the column strain softening in compression (local buckling). With increasing lateral deformations, the lateral load reaches a plateau of 96 kips. The same model was analyzed using the same geometric properties and stress-strain curve but using the original fiber-based element[1] with C-type fibers in Drain2DX. As shown in Figure 3.8, the results were identical, thus validating the results using the modified fiber-based element.

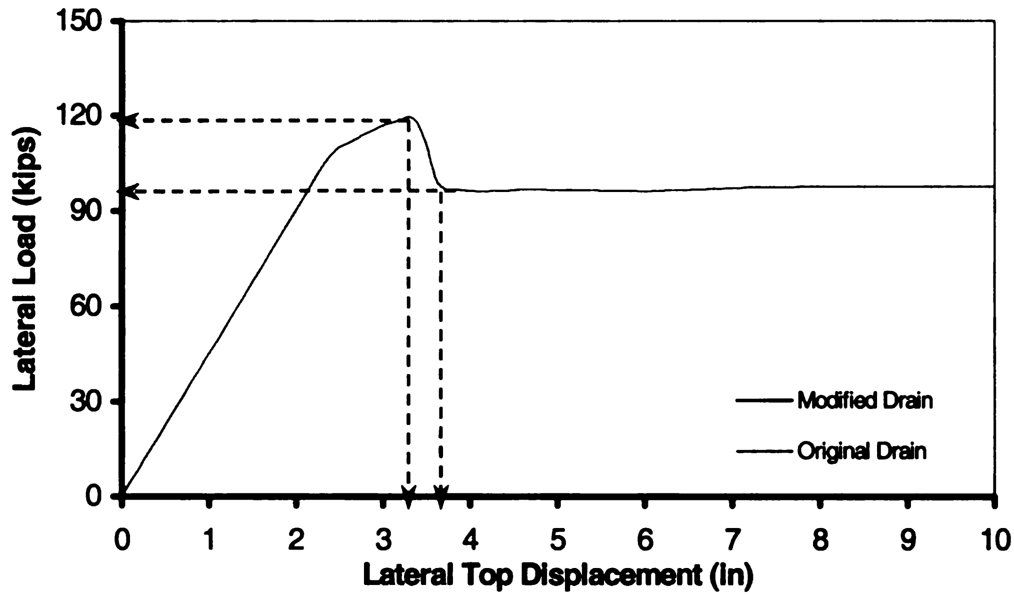


Figure 3.8 Lateral load – monotonic analysis

Figure 3.9 shows the moment-curvature responses for the five segments of the column. As shown in Figure 3.9, inelastic deformations occur in the segment at the base of the column, where the plastic hinge forms, while the remaining segments unload elastically.

[1] Since the stress-strain curve used for this analysis was asymmetric, to compare the results with the unmodified version of Drain2DX, the material was analyzed as concrete (only in the unmodified version).

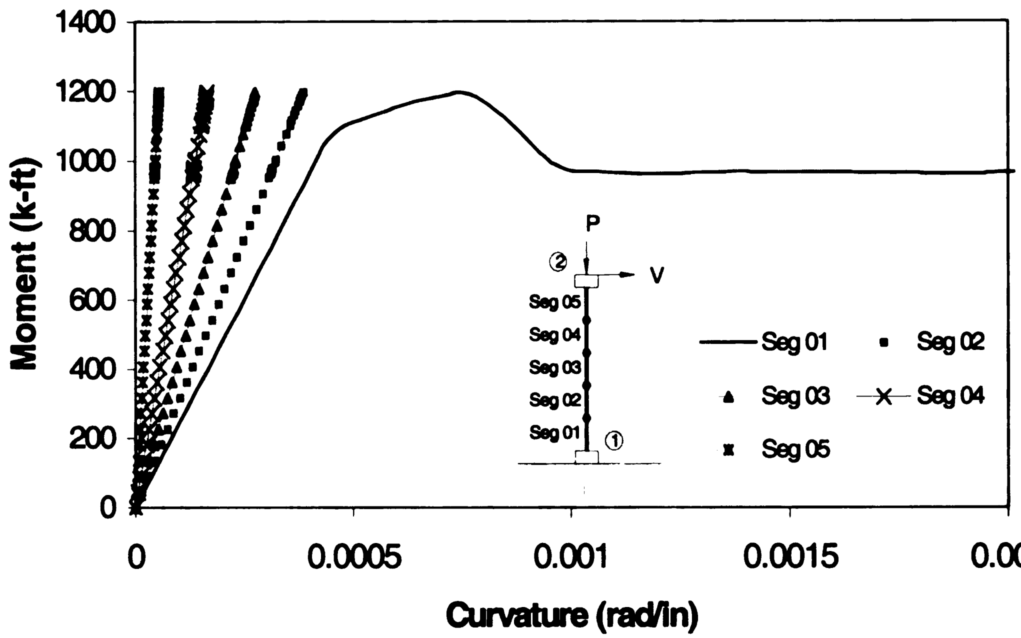


Figure 3.9 Moment-curvature analysis

The cyclic analysis was performed using the model shown in Figure 3.10. The same geometry and material properties provided previously were used to perform the analysis. The lateral load required, to yield the most extreme fiber in the cross-section, was calculated by assuming a linear elastic behavior on the element. The following equation was used:

$$\sigma = \frac{V * L * c}{I} \tag{3}$$

where,

σ	=	Yield stress, ksi
V	=	Lateral load, kips
L	=	Length of the member, in
c	=	Distance from centroid to the extreme fiber in the cross-section, in
I	=	Moment of inertia, in^4

A lateral load of 93.0 kips will cause initial yielding. Only 4.0 % of the calculated axial load capacity was applied in compression. Since Drain2DX works under displacement control, only lateral displacements need to be input. The lateral displacement at which the fiber yields was calculated based on Equation (4).

$$\Delta_y = \frac{V * L^3}{3 * E * I} \tag{4}$$

where, Δ_y = displacement when yielding takes places, in

E = Young's modulus, ksi

The cyclic loading history consisted of cycles at lateral displacements equal to Δ_y, $2\Delta_y$, $3\Delta_y$, $5\Delta_y$ and $7\Delta_y$, where Δ_y was equal to 2.0 inches.

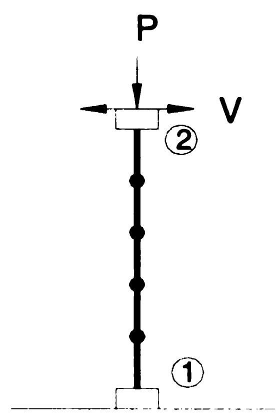


Figure 3.10 Cyclic loading schematic

The results of the analysis are shown in Figures 3.11 and 3.12. The influence of the unloading factor (α) is included in both figures. The response is almost identical for α = 0 and 0.5. Meanwhile, for α = 1.0, which represents the most extreme shooting behavior, the response deviates from the other cases. The effects of stiffness degradation due to local buckling on the behavior of the steel beam-column is apparent in Figures

3.11 and 3.12. α =1 has a tendency to make the element modeled in Drain2DX unstable. A value of α < 0.9 provides a more stable response.

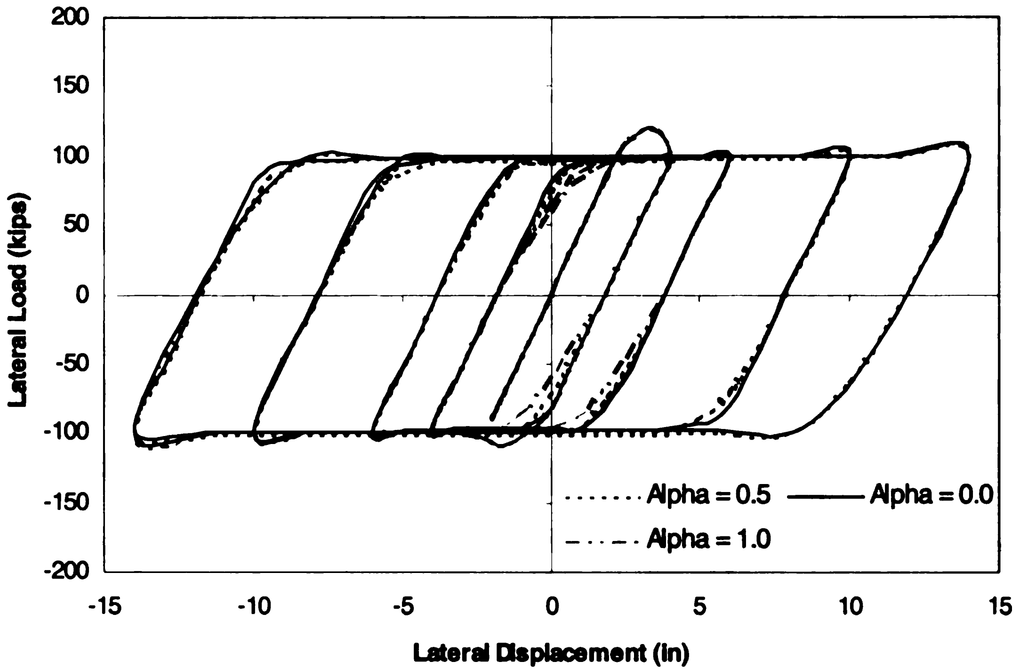


Figure 3.11 Lateral load – lateral displacement response for steel column

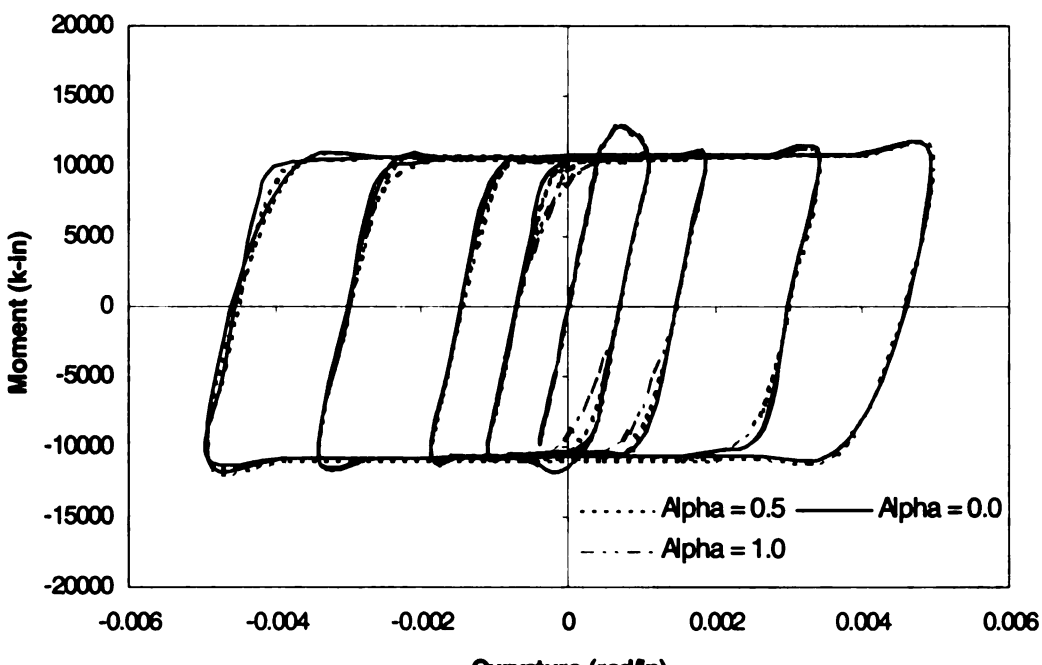


Figure 3.12 Moment-curvature response at failure segment for steel column

3.4 FIBER ANALYSIS OF CFT BEAM-COLUMNS

Fiber-based models were developed for the high strength square CFT beam-column specimens tested by Varma et al. (2001). The analyses were conducted to validate the modified fiber-based element for modeling CFT beam-columns, while accounting for local buckling and biaxial stress effects.

The CFT specimens tested by Varma et al. (2001) were square tubes with b/t ratios of 32 or 48, made from conventional (A500 Gr. 46) or high strength (A500 Gr. 80) steel and filled with high strength (16.0 ksi) concrete. Eight monotonic CFT beam-column specimens were tested under constant axial load (20% or 40% of axial load capacity) and monotonic flexural loading. Eight cyclic beam-column specimens were tested under constant axial load (10% and 20% of axial load capacity) and cyclic flexural loading. Varma et al. (2001) also developed fiber-based models for the monotonic and cyclic beam-column specimens using Drain2DX. The steel and concrete fiber stress-strain curves were derived from three-dimensional (3D) nonlinear finite element analysis of the CFTs. However, they could not use the previously existing fiber-based beam-column element in Drain2DX directly to conduct the fiber analyses. They were forced to develop alternate analytical models using Drain2DX to analyze the specimens.

The modified fiber-based beam-column element developed in this research could be used directly to analyze the high strength CFT specimens. Section 3.4.1 presents the fiber analysis of the monotonic CFT beam-column specimens and section 3.4.2 presents the fiber analysis of the cyclic CFT beam-column specimens of Varma et al. (2001).

3.4.1 Monotonic analysis of CFT beam-columns

The monotonic CFT beam-column specimens were tested under constant axial load (20% or 40% P_o) and monotonic flexural loading (Varma et al. 2001). The specimen test lengths (60 in.) were subjected to uniform primary bending moment. Each specimen failed with the formation of a 12 in. long inelastic failure segment close to the mid-length, while the remaining portion of the column unloaded elastically. The experimental results were reported in the form of the fundamental moment-curvature (M-ϕ) response of the 12 in. failure segment.

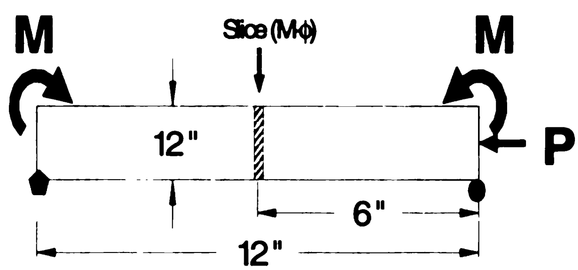


Figure 3.13 Schematic for monotonic analysis of CFT beam-columns

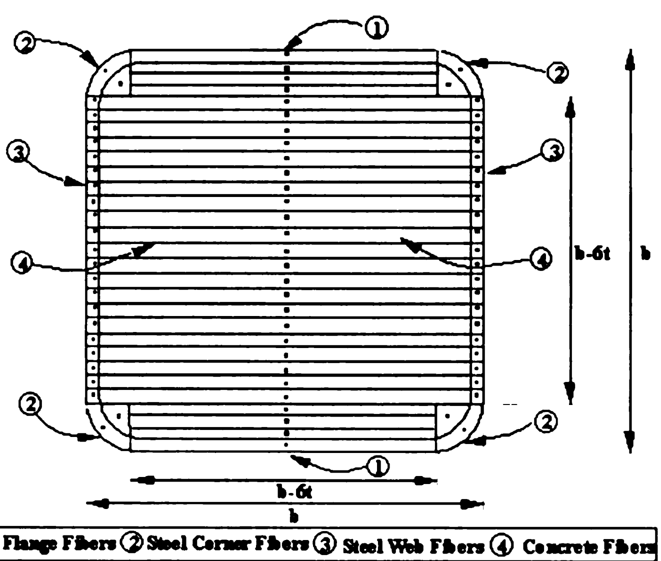


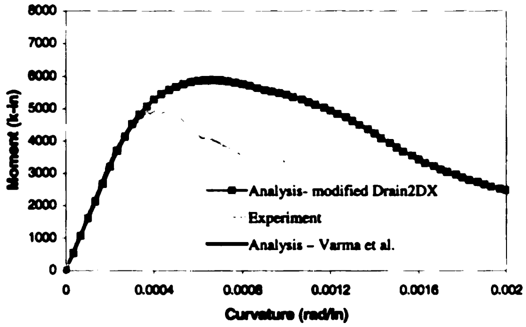
Figure 3.14 Fiber discretization of the slice (cross-section)

Figure 3.13 shows the fiber-based model for the monotonic beam-column specimens. Figure 3.14 shows the fiber discretization of the slice (cross-section). Table 3.2 summarizes the 3D finite element model (FEM)-based steel and concrete fiber stress-strain curve recommended by Varma et al. (2001) for conducting the fiber analyses. The fiber-based models were analyzed for the same loading condition as the test specimens, i.e., constant axial load and monotonically increasing flexural loading. The moment-curvature (M-ϕ) responses of the slices are shown in Figures 3.15 (a) to (h) for the eight monotonic beam-column specimens. In Figures 3.15 (a) to (h), the nomenclature used to identify the specimens (e.g., BC 32-80-20) consists of the specimen type (BC stands for monotonic beam-columns), the nominal b/t ratio, the nominal yield stress in ksi, and the nominal axial load level (P/P_o). The experimental moment-curvature responses of the corresponding specimens are also shown in Figures 3.15 (a) to (h). Additionally, the moment-curvature response from the analyses conducted by Varma et al. (2001) using the same stress-strain curves shown in Table 3.2 and *mathematically* identical analytical models in Drain2DX are also shown in Figures 3.15 (a) to (h).

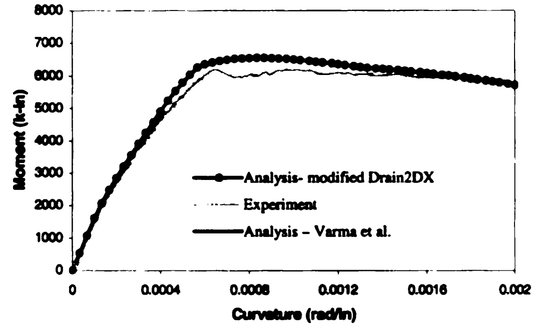
Figures 3.15 (a) to (h) clearly indicate that the results from the fiber analyses using the modified fiber-based element follow closely the analytical results of Varma et al. (2001), thus validating the implementation of the modified fiber-based element. Additionally, both analytical models predict the experimental results with reasonable accuracy, which is due to the 3D FEM-based fiber stress-strain curves recommended by Varma et al. (2001).

Table 3.2 Stress-strain points defining the stress strain curves for steel and concrete fibers (Varma et al. (2001))

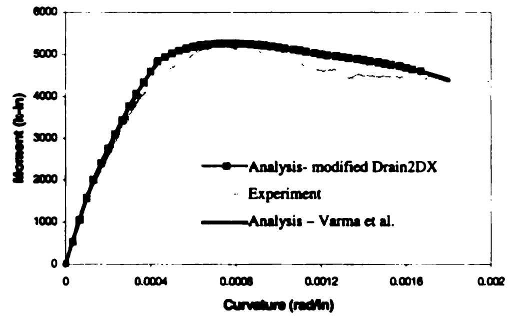
CFT Specimen type	Steel tension fibers		Steel compression fibers				Concrete compression fibers	
			Flange & web		Corner			
	stress	strain	stress	strain	stress	strain	stress	strain
b/t = 32 Gr. = 80	0	0	0	0	0	0	0	0
	87.0	0.003	83.1	0.0029	85.6	0.003	16.0	0.0027
	97.7	0.124	83.1	0.0033	85.6	0.0032	16.3	0.0033
			55.0	0.0076	55.0	0.0076	11.5	0.01
b/t = 48 Gr. = 80	0	0	0	0	0	0	0	0
	95.8	0.0033	80.2	0.0028	86.3	0.003	16.0	0.0027
	107.0	0.12	80.2	0.0034	86.3	0.0036	16.5	0.0032
			54.0	0.0076	54.0	0.006	9.4	0.01
b/t = 32 Gr. = 46	0	0	0	0	0	0	0	0
	39.0	0.0134	39.0	0.0013	39.0	0.0013	16.0	0.0027
	40.5	0.0525	40.5	0.0031	40.5	0.0031	16.0	0.003
	62.0	0.2277	21.0	0.0057	21.0	0.0057	6.3	0.01
b/t = 48 Gr. = 46	0	0	0	0	0	0	0	0
	68.3	0.0024	61.0	0.0021	66.6	0.0023	16.0	0.0027
	77.5	0.148	61.0	0.0028	66.6	0.0029	16.5	0.0032
			54.1	0.0039	39.0	0.0051	6.4	0.01
			35.0	0.0063				



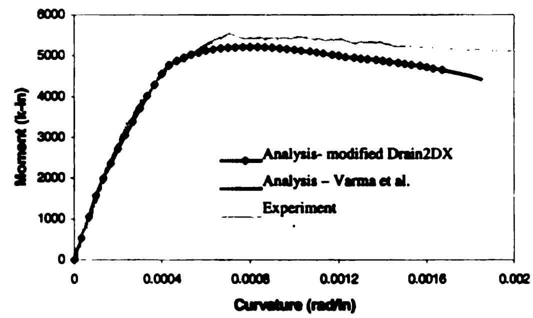
(a) BC 48-80-40



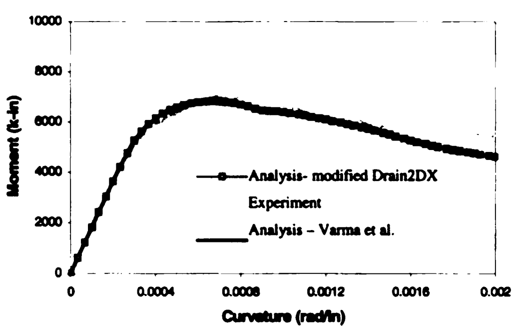
(b) BC 48-80-20



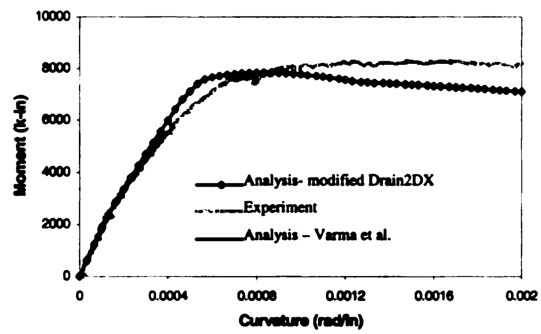
(c) BC 48-46-22



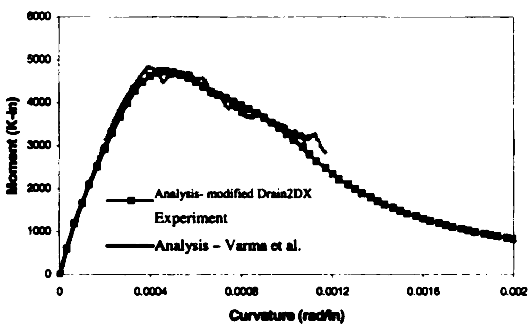
(d) BC 48-46-20



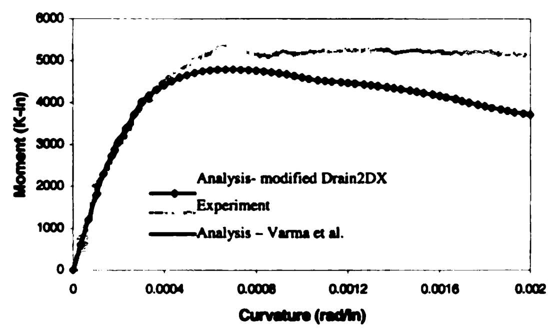
(e) BC 32-80-40



(f) BC 32-80-20



(g) BC 32-46-40



(h) BC 32-46-20

Figure 3.15 Moment-curvature response of the failure segment for CFT beam-columns

3.4.2 Cyclic Analysis of CFT beam-columns

The test-length (58.5 in.) of the cyclic beam-column specimens was fixed at the bottom and subjected to a constant axial load (10% or 20% of P_o) and cyclic lateral loading at the top. Thus, the specimen test-length was subjected to a linearly varying bending moment. The cyclic loading history is presented in detail by Varma et al. (2001). Each specimen failed with the formation of a 12 in. long inelastic failure segment at the base of the column. The experimental results were reported in the form of the lateral load-displacement response of the beam-columns and the moment-curvature response of the 12 in. long inelastic failure segment at the base.

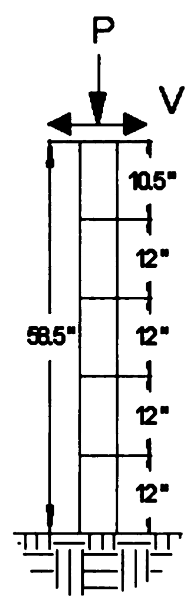


Figure 3.16 Schematic for CFT beam-columns

Figure 3.16 shows the fiber-based model for the cyclic beam-column specimens. As shown in Figure 3.16 the fiber-based element was divided into five segments. The second-order P-Δ effects were not included in the analysis because they were not presented in the experiments. The fiber-based models were analyzed for the same loading conditions as the test specimens, i.e., constant axial load and cyclically varying lateral loading. The lateral loading history was similar to the experimental loading history. Lateral loading cycles were conducted under displacement control in the analyses at displacement levels of 0.25 Δ_y, 0.5 Δ_y, 0.75 Δ_y, 1.0 Δ_y, 1.5 Δ_y, 2.0 Δ_y, 3.0 Δ_y, 5.0 Δ_y, and 7.0 Δ_y, where Δ_y is the displacement at which the steel yields. Two cycles were conducted at each displacement level. Table 3.3 shows the yield displacement (Δ_y) used for each specimen. These values were also reported by Varma et al. (2001) based on the experimental results.

Table 3.3 Yield displacements of CFT beam-columns

Specimen	Yield Displacement, Δ_y (inch)
CBC 32-80-20	0.57
CBC 32-80-10	0.60
CBC 48-80-20	0.46
CBC 48-80-10	0.55
CBC 32-46-20	0.39
CBC 32-46-10	0.32
CBC 48-46-20	0.41
CBC 48-46-10	0.42

The lateral load-displacement responses and the moment-curvature responses of the failure segments from the fiber analyses are shown in Figures 3.17 to 3.24 for the eight cyclic beam-column specimen. In Figures 3.17 to 3.24, the specimen nomenclature consists of the specimen type (CBC stands for cyclic beam column). The experimental lateral load-displacement and moment-curvature responses of the corresponding specimens are also shown in Figures 3.17 to 3.24. Additionally, Varma et al. (2001) conducted fiber analyses of the failure segment *only* using a complex and indirect but mathematically accurate model in Drain2DX. The cyclic moment-curvature responses from the fiber analyses of Varma et al. (2001) were also shown in Figures 3.17 to 3.24. However, Varma et al. (2001) could not obtain cyclic lateral load-displacement responses using their model.

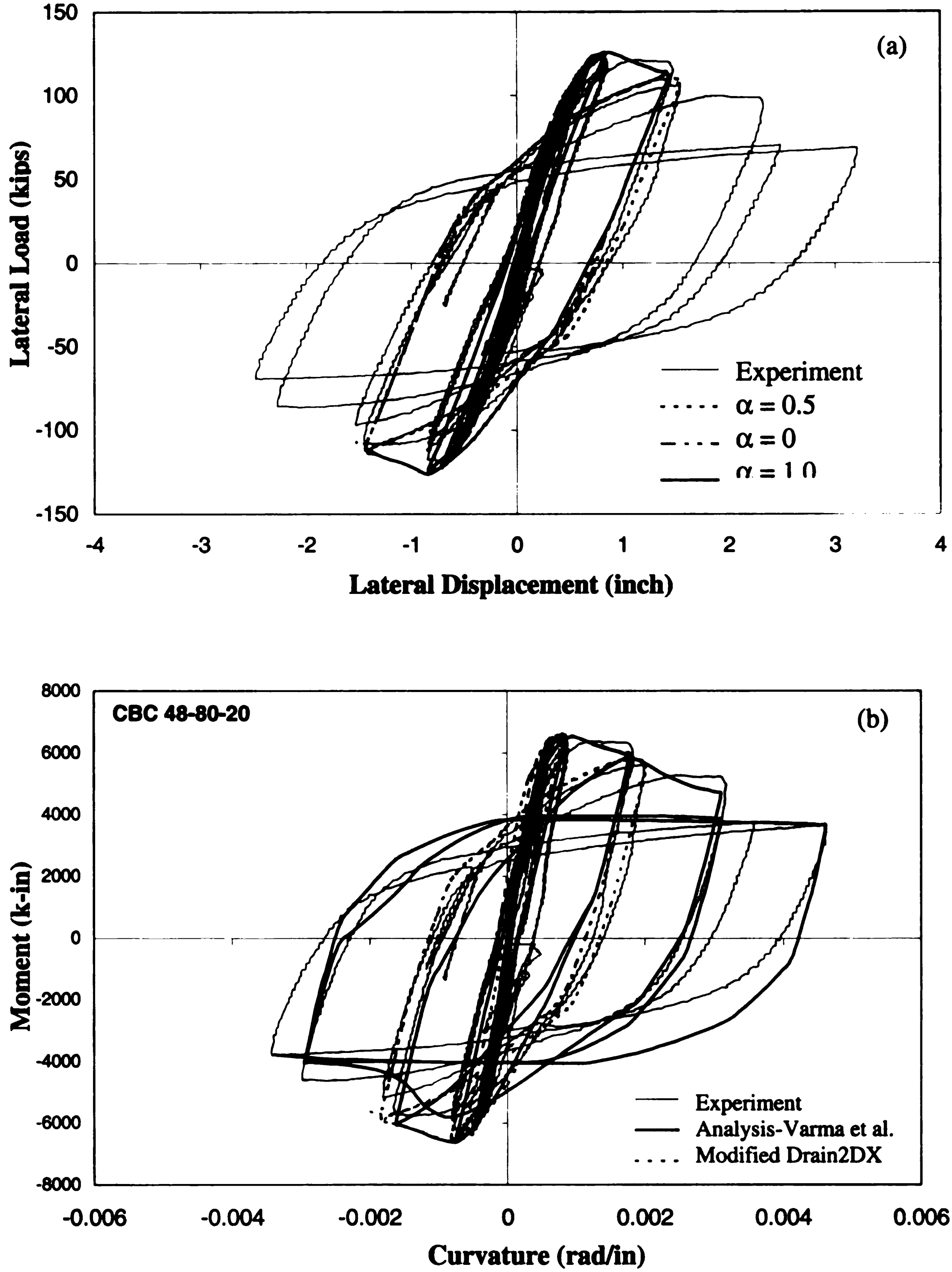


Figure 3.17 Fiber analysis of CFT beam-column specimen CBC 48-80-20: (a) lateral load-displacement; and (b) moment-curvature of the failure segment

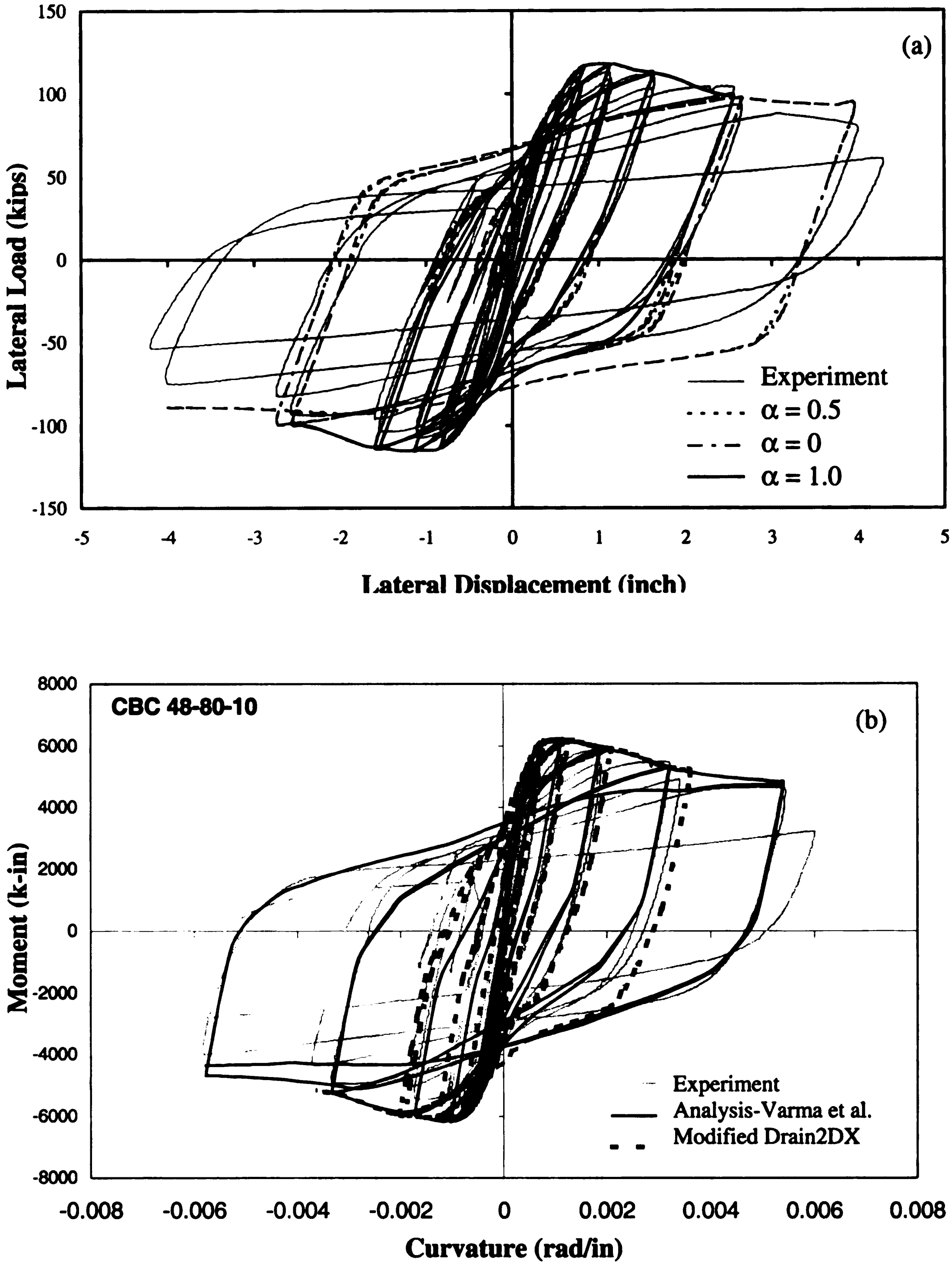


Figure 3.18 Fiber analysis of CFT beam-column specimen CBC 48-80-10: (a) lateral load-displacement; and (b) moment-curvature of the failure segment

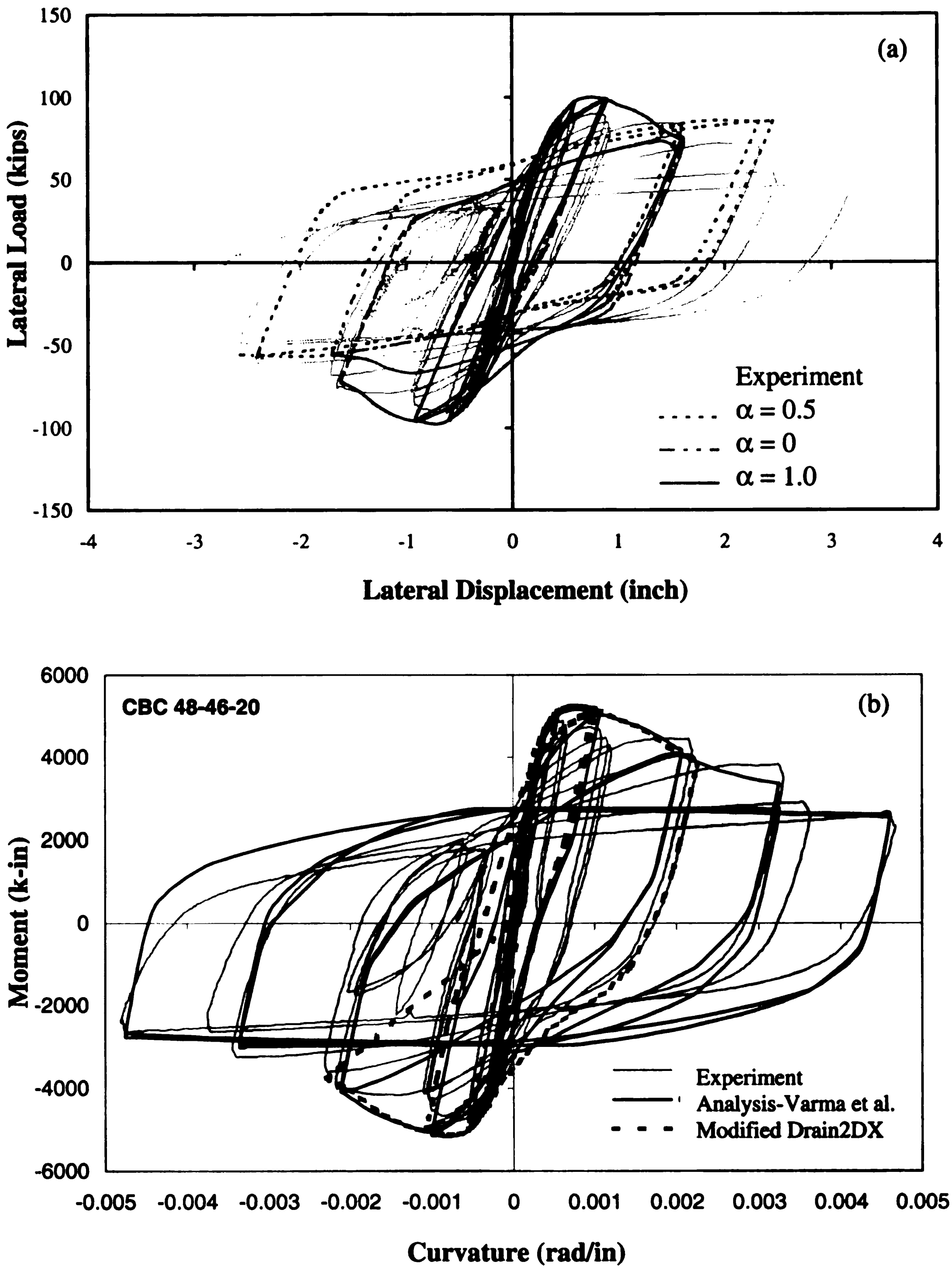


Figure 3.19 Fiber analysis of CFT beam-column specimen CBC 48-46-20: (a) lateral load-displacement; and (b) moment-curvature of the failure segment

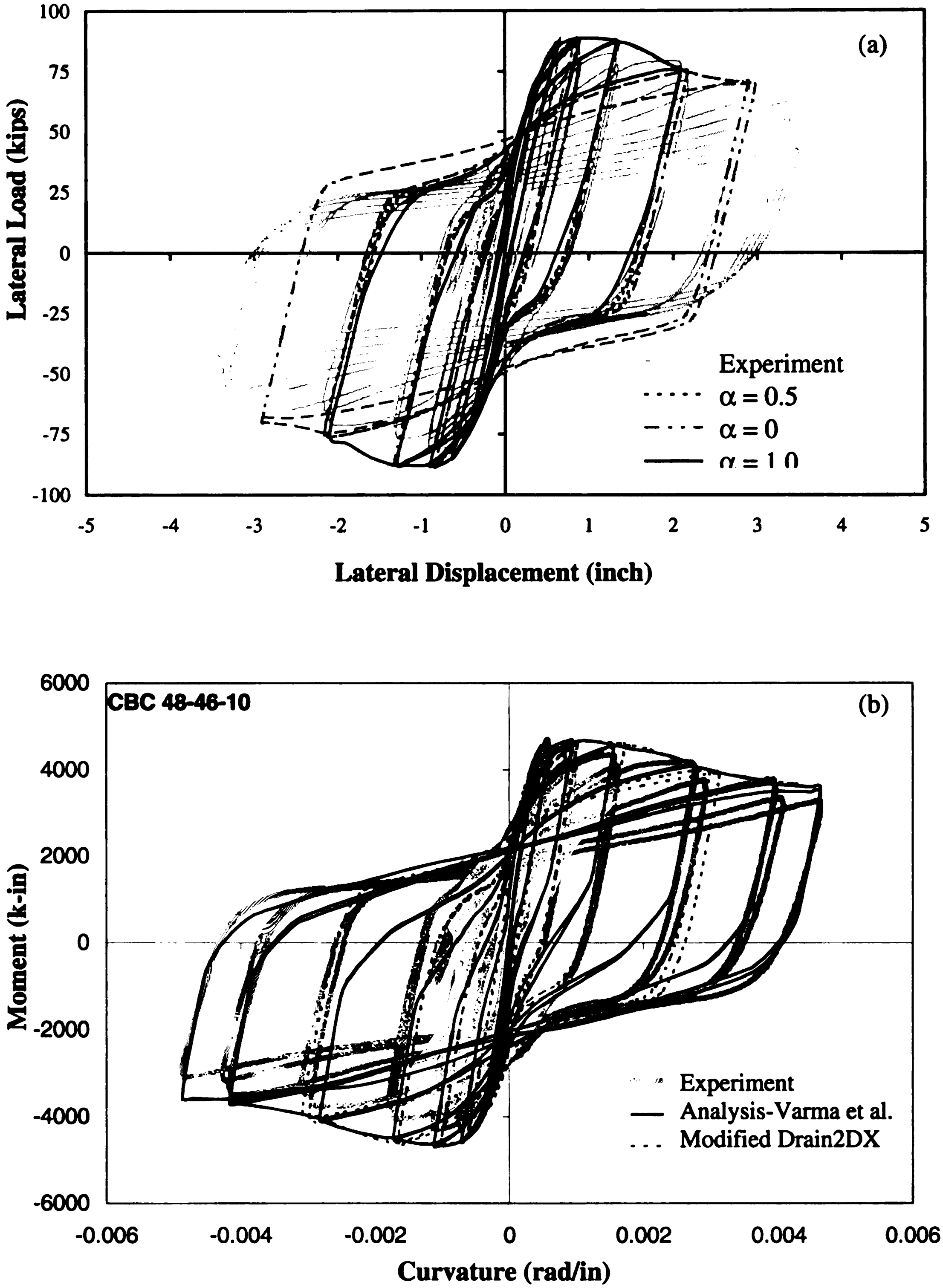


Figure 3.20 Fiber analysis of CFT beam-column specimen CBC 48-46-10: (a) lateral load-displacement; and (b) moment-curvature of the failure segment

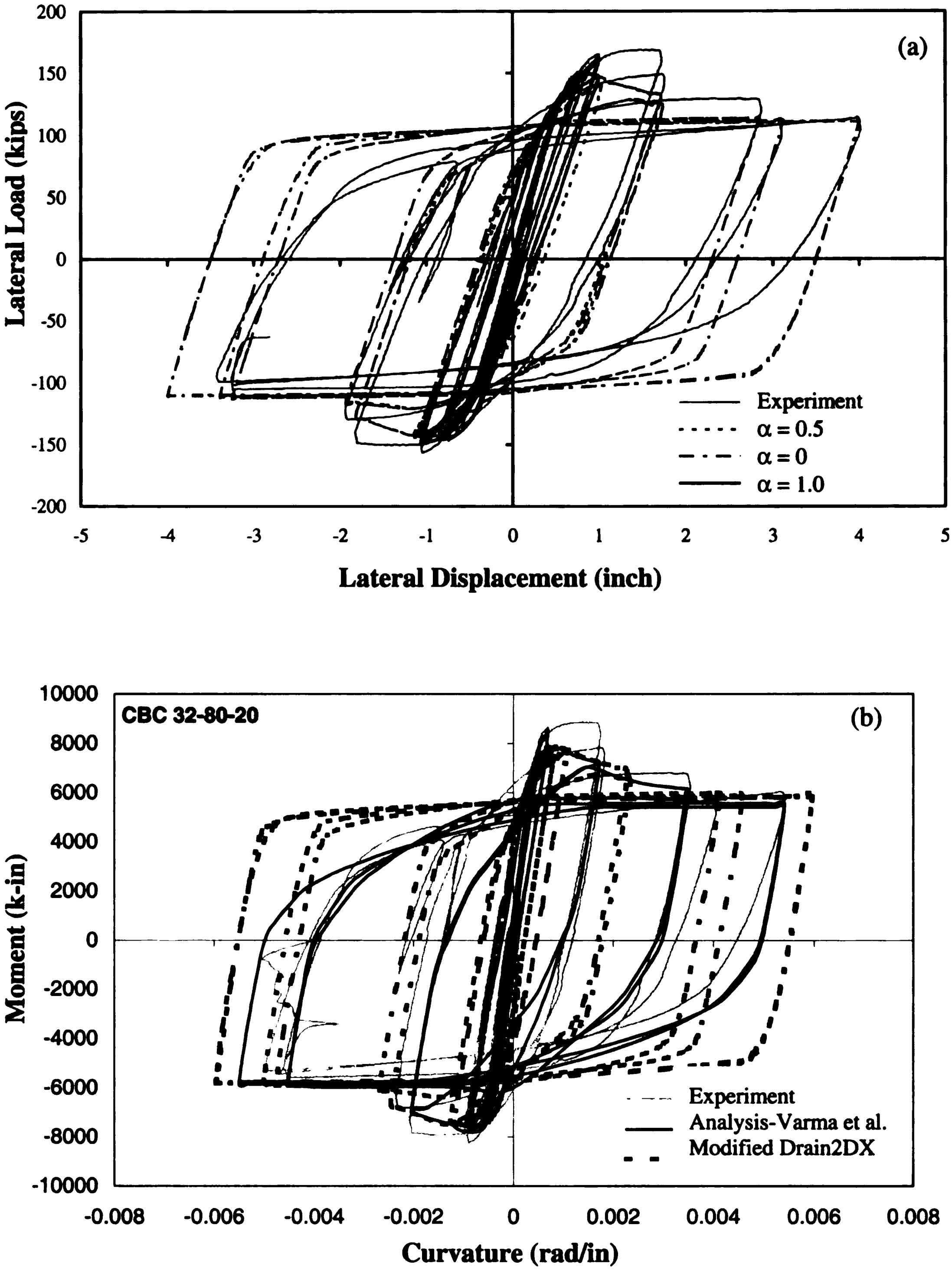


Figure 3.21 Fiber analysis of CFT beam-column specimen CBC 32-80-20: (a) lateral load-displacement; and (b) moment-curvature of the failure segment

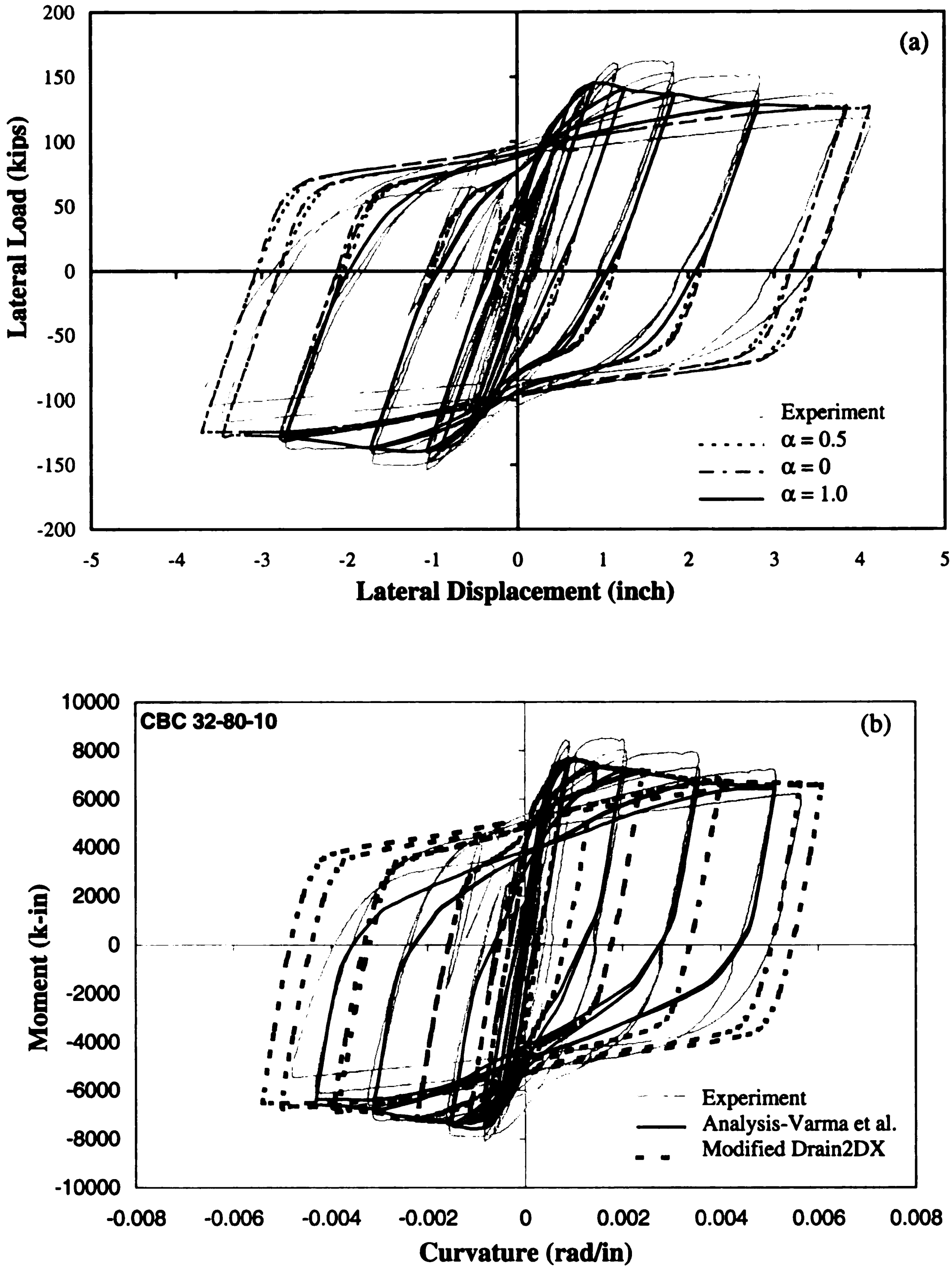


Figure 3.22 Fiber analysis of CFT beam-column specimen CBC 32-80-10: (a) lateral load-displacement; and (b) moment-curvature of the failure segment

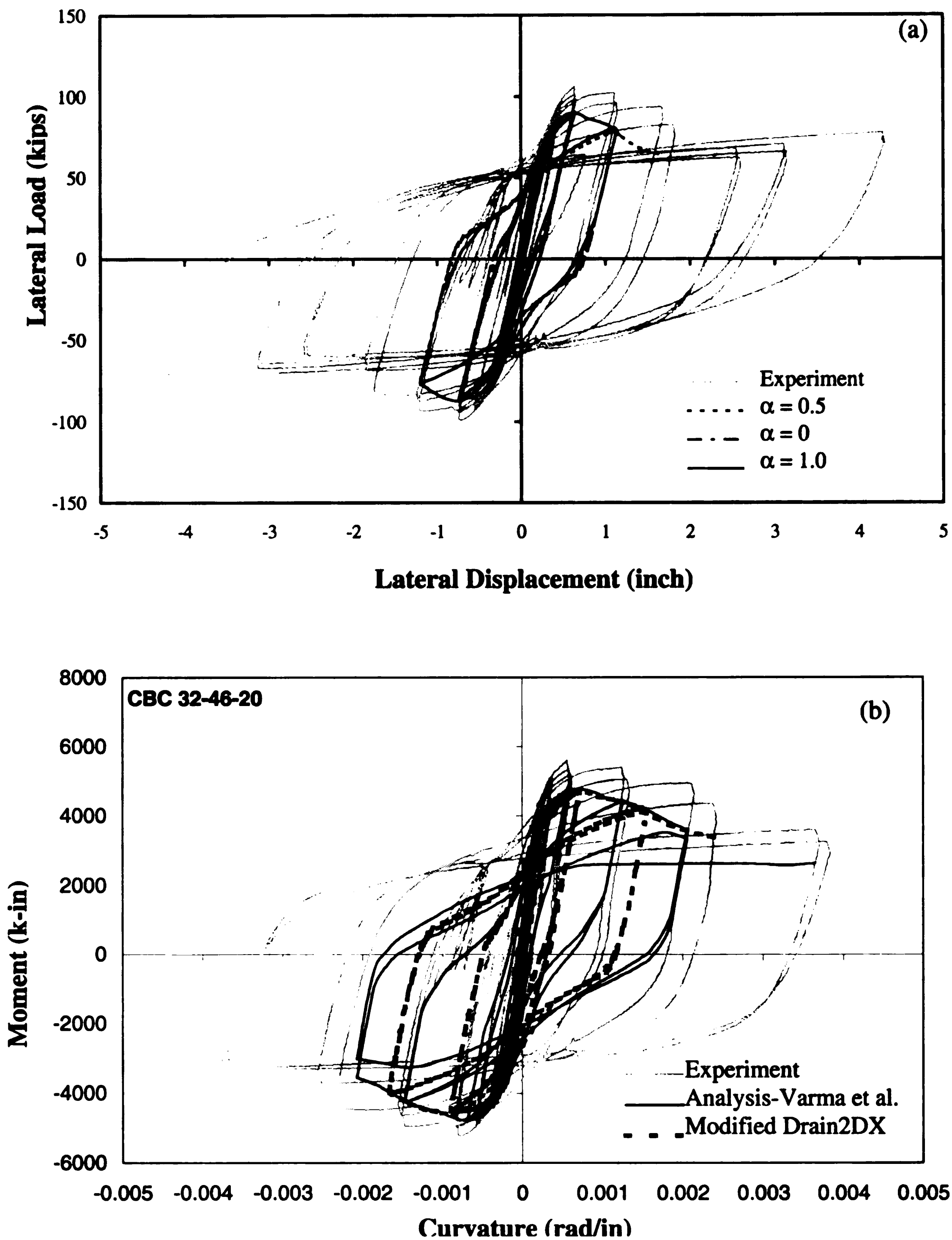


Figure 3.23 Fiber analysis of CFT beam-column specimen CBC 32-46-20: (a) lateral load-displacement; and (b) moment-curvature of the failure segment

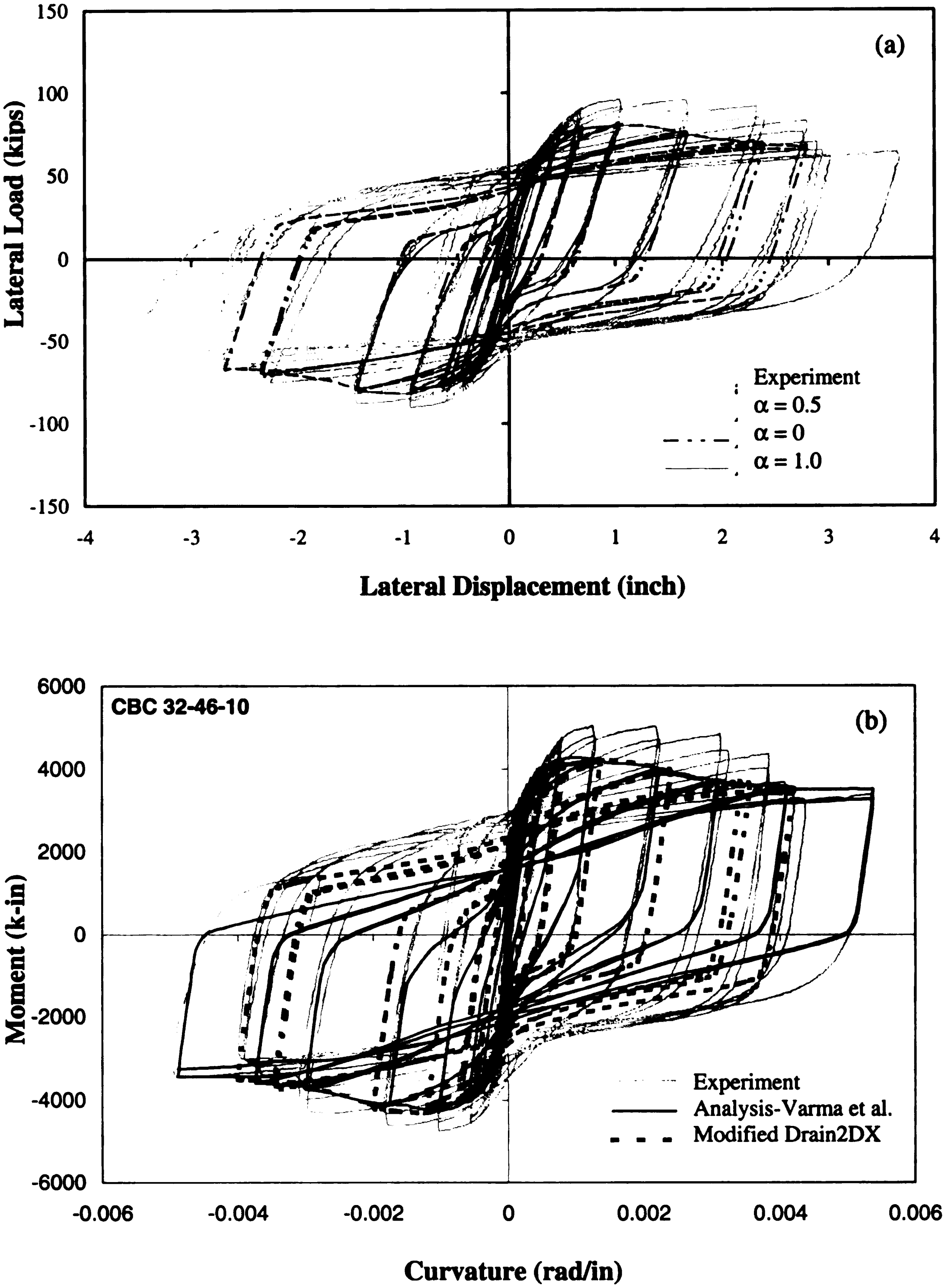


Figure 3.24 Fiber analysis of CFT beam-column specimen CBC 32-46-10: (a) lateral load-displacement; and (b) moment-curvature of the failure segment

Figures 3.17 to 3.24 indicates that:

- Cyclic analysis using the modified fiber-based element could be completed for the full loading history for six specimens.
- The cyclic loading history could not be completed for specimens CBC 48-80-20 and CBC 32-46-20 due to convergence problems.
- The cyclic analyses using the modified fiber-based element were most stable (converged and completed the loading history) for the case of α equal to zero. It was least stable for the case of α equal to one.
- The results from the analyses using the modified fiber-based beam-column element compared well with the experimental results. For specimens CBC 48-80-10, CBC 48-46-20, CBC 48-46-10, CBC 32-80-20, CBC 32-46-20 α equal to zero provided the best comparison. For specimen CBC 48-80-20, CBC 32-80-10 and CBC 32-46-10 for α equal to one provided the best comparison.
- The moment-curvature response from the analyses using the modified fiber-based element compared favorably with the analytical results from Varma et al. (2001). They had used the steel fiber cyclic stress-strain behavior shown in Figure 1.4, which corresponds closely to the case with α equal to one for the modified fiber-based element.
- It is important to note that the analytical model of Varma et al. (2001) was different than the model shown in Figure 3.16. As a result, they used a slightly different cyclic loading history, which accounts for the discrepancies between the analytical moment-curvature responses shown in Figures 3.17 to 3.24.

3.5 SUMMARY

The implementation of the modified S-type steel stress-strain curve in Drain2DX was verified using three analytical models with increasing levels of complexity; (1) monotonic and cyclic axial load analyses of a two-fiber steel cross-section, (2) monotonic and cyclic analysis of a steel column subjected to lateral loading, and (3) monotonic and cyclic analysis of high strength CFT beam-columns tested by Varma et al. (2001). Additionally, the results from the CFT fiber analysis were compared with the experimental and analytical results of Varma et al. (2001).

The results from the fiber analyses of the axially loaded steel bar validated the implementation of the S-type stress-strain curve in Drain2DX. The results from the fiber analyses of the W14x82 steel beam-column subjected to constant axial load and monotonic or cyclic lateral loading validated the implementation of the modified fiber-based element in Drain2DX. The results from the fiber analyses of the CFT beam-column specimens subjected to constant axial load and monotonic or cyclic flexural loading validated the modified fiber-based element for modeling members with local buckling and biaxial stress effects dominating the fundamental force-deformation response.

3.6 RECOMMENDATIONS FOR USING THE MODIFIED FIBER-BASED ELEMENT

Some recommendations for using the modified fiber-based element effectively are as follows:

1. Use a reasonable stress-strain curve. The S-type stress-strain curve can be asymmetric in tension and compression. However, the yield ratios (ratios of ultimate stress to yield stress, σ_u/σ_y) must not be too high. Usually, this will not be a problem for steel materials.
2. The negative slope of the descending compression strain softening branch must not be too large (steep). This can lead to convergence problems.
3. Choose the value of α for each application/analytical model carefully. α equal to zero corresponds to elastic-plastic behavior and α equal to one corresponds to shooting behavior as shown in Figure 2.2 (b). α equal to zero was found to be more stable for the CFT beam-columns and for large strain applications. However, α equal to one may be more appropriate for situations were local buckling dominates.

CHAPTER 4. SUMMARY AND CONCLUSIONS

4.1 SUMMARY

The existing fiber-based beam-column element in Drain2DX has a major drawback. It cannot be used to model the effects of local buckling and/or biaxial stresses in the steel elements of the member cross-section. As a result, it cannot account for their influence on the force-deformation behaviors of members. Therefore, a modified fiber-based beam-column element with a modified S-type steel stress-strain curve was developed in Drain2DX. The modified S-type steel stress-strain curve can be asymmetric in tension and compression and can model tension and compression strain hardening, compression strain softening, kinematic hardening under cyclic loading, and the effects of local buckling and biaxial stresses on the hysteretic behavior.

Drain2DX uses an event-to-event based solution algorithm. Therefore, the multi–linear S-type stress-strain curve was discretized into ten unique branches. The change in stiffness associated with the fiber response moving from one branch to the other was identified using unique event numbers. The response state of the fiber depends on its current state, cyclic response history, magnitude and direction of the strain increment, and the occurrence of events. The event calculation and state determination algorithms were developed and implemented in SMEV15 and SMSD15 subroutines in Drain2DX.

The implementation of the modified fiber-based beam-column element in Drain2DX was validated using three analytical models with increasing levels of complexity. These are: (1) monotonic and cyclic axial load analyses of a simple two fiber steel cross-section, (2) monotonic and cyclic lateral load analyses of a steel column, and (3) monotonic and cyclic analyses of high strength CFT beam-columns tested by Varma et al. (2001). The results from analyses of the axially loaded steel bar fiber analyses validated the implementation of the S-type stress-strain curve in Drain2DX. The results from the fiber analyses of the W14x82 steel beam-column subjected to constant axial load and monotonic or cyclic lateral loading validated the implementation of the modified fiber-based element in Drain2DX.

The modified fiber-based element was also used to model and analyze the monotonic and cyclic high strength square CFT beam-column specimens tested by Varma et al. (2001). These researchers had reported the specimen geometric and material properties, the relevant experimental results, and the 3D FEM-based steel and concrete fiber stress-strain curves to be used with the fiber-based models. The results from the monotonic and cyclic fiber analyses compared well with the experimental and analytical results reported by Varma et al. (2001). Some convergence problems were encountered during the cyclic fiber analyses.

4.2 CONCLUSIONS

1. The modified fiber-based beam-column element developed in this research can be used to model the effects of local buckling and/or biaxial stresses on the behavior of the steel and steel-concrete composite members.

2. The modified fiber-based element can model the effects of local buckling and/or biaxial stresses on the hysteretic behavior through the use of parameter α.

   - α equal to zero corresponds to elastic-plastic behavior upon strain reversal, which is appropriate for the case where biaxial stresses dominate.
   - α equal to one corresponds to the shooting behavior upon strain reversal, which is appropriate for the case where local buckling dominates.
   - A value of α between zero and one models both effects.

3. Cyclic analyses using the modified fiber-based element are most stable for the case of α equal to zero. They are least stable for the case of α equal to one combined with large strains.

4. Some convergence problems may occur while using this element if:

   - The yield ratios (ratio of ultimate stress to yield stress, σ_u/σ_y) for the S-type stress-strain curve are too high.
   - The negative slope of the descending compression strain softening branch is too large (steep).
   - For the case of α equal to one and large strains, because the shooting slope (stiffness) may be too small.

APPENDIX A
MODIFIED SUBROUTINES IN DRAIN2DX

APPENDIX A-1
STEEL MATERIAL FIBER PROPERTIES

```
c ********************************************************************
c  INFPR15.H
c  3D FIBER FLEXIBILITY BEAM COLUMN ELEMENT
c  PROPERTY COMMON BLOCKS

c  GEPROP = properties for a geometry type
c  FSPROP = properties for a fiber section type
c  ESPROP = properties for an elastic section type
c  CMPROP = properties for a concrete material type
c  SMPROP = properties for a steel material type
c  CHPROP = properties for a fiber hinge type
c  BMPROP = properties for a pullout fiber type
c  GMPROP = properties for a gap fiber type
c ********************************************************************

c ********************************************************************
c  SMPROP
c-----------------------------------------STEEL MATERIAL FIBER PROPERTIES
c--Series model.
c--Set up when steel properties are read.
c--Recall whenever a steel fiber is processed.
c--Data does not change.
c--OVERALL DATA (lprsm=5)
      real*4 smot      ! overshoot tolerance
      real*4 smel      ! fully yielded modulus (very small)
      real*4 smrf      ! reloading factor
      integer*2 lppksm ! length of fiber property packet
      integer*2 dure   ! Dummy
      integer*4 kppksm ! pointer to first fiber property packet
      integer*4 ltotsm ! total state data length for this material

c********NOT USED**********
c--SUBFIBER PACKET (lppksm=3)
c       real*4 smsy    ! subfiber yield stress
c      real*4 smm     ! subfiber elastic modulus
c       real*4 smey    ! subfiber yield strain
c       real*4 smeu    ! subfiber ultimate strain
c      real*4 syrem   ! total elastic modulus of all remaining subfibers
c***************************

c--FIBER PACKET (lppksm=10)
      real*4 smcom    ! stress-strain matrix - Comp.
      real*4 smten    ! stress-strain matrix - Tension
c--NOT CURRENTLY SAVED
      integer*2 iduml  ! dummy
      integer*2 nsmcom ! no. of points on stress-strain curve (Comp.)
      integer*2 nsmten ! no. of points on stress-strain curve (tension)

      common /smpr15/ smcom(2,3), smten(2,2)     ! current fiber props.
     1             ,smot, smel, smrf, lppksm, kppksm ,ltotsm
     2             ,nsmcom, nsmten       ! overall fiber properties
     3             ,iduml, dure          ! input stress-strain curve
      real*4 prsm(6)          ! array to transfer basic data
      equivalence (prsm,smot)
      real*4 ppksm(10)          ! array to transfer subfiber packet
      equivalence (ppksm,smcom)
c ********************************************************************
```

APPENDIX A-2
STEEL MATERIAL FIBER STATE

```
c *****************************************************************
c  INFST15.H
c  3D FIBER FLEXIBILITY BEAM COLUMN ELEMENT
c  STATE COMMON BLOCKS

c  FSSTAT = state for a fiber section
c  ESSTAT = state for an elastic section
c  CMSTAT = state for a concrete material fiber
c  SMSTAT = state for a steel material fiber
c  CHSTAT = state for a fiber hinge
c  BMSTAT = state for a pullout fiber
c  GMSTAT = state for a gap fiber
c  NVSTAT = envelope values
c *****************************************************************

c *****************************************************************
c  SMSTAT
c--------------------------------------------STEEL MATERIAL FIBER STATE
c--Initialized when element is created.
c--Recall whenever a steel fiber is processed.
c--Data changes in each state determination.

c--OVERALL STATE (lstsm=2)
      integer*4 kspksm ! pointer to fiber state packet
      integer*2 lspksm ! length of fiber state packet
      integer*2 iismdm ! dummy

c--FIBER PACKET (lspksm=12)
c      real*4 sms        ! subfiber stress
c      integer*4 iysm    ! fiber branch number
c                       ! 0 = elastic
c                       ! 1 = yielded positive
c                       !-1 = yielded negative


      real*4 ssig        ! current stress
      real*4 setot       ! current strain
      real*4 semax       ! maximum strain
      real*4 semin       ! minimum strain
      integer*4 sbrnum   ! branch number
      integer*4 slastt   ! last tension branch
      integer*4 slastc   ! last compression branch
      real*4 ssc         ! compression stress at unloading
      real*4 sst         ! tension stress at reloading
      real*4 smod        ! current modulus
      integer*4 piti     ! dummy
      real*4 yab

c--COMMON BLOCK
      common /smst15/ ssig,setot,semax,semin,sbrnum,slastt,slastc
     1               ,ssc,sst,smod,piti,yab    ! current fiber state
     2               ,kspksm,lspksm,iismdm  ! overall material state

      real*4 stsm(2)            ! array to transfer overall state
      equivalence (stsm,kspksm)

      real*4 spksm(12)          ! array to transfer fiber packet
      equivalence (spksm,ssig)
c *****************************************************************
```

APPENDIX A-3
Steel material data input

```
c ******************************************************************
      SUBROUTINE  SMIN15(ntyp,smot,smel,nsmcom,nsmten,smcom,smten,MXSMC
     1                 ,MXSMT,kdata,smrf)
c -----------------------------------------------------------------
c  DRAIN FIBER ELEMENTS
c     Steel material data input. Parallel model.
c -----------------------------------------------------------------
c  DOUBLE PRECISION / LARGE
      include 'double.h'
c ------------------------------------------------------------------
c  CALLED FROM : inel15
c  FILE I/O    : read data from inp.
c                write data to iou.
c ------------------------------------------------------------------
c  ARGUMENTS
c    INPUT:
      integer*4 ntyp        ! type no.
c    OUTPUT:
      real*4 smot           ! overshoot tolerance
      real*4 smel           ! modulus of elastic subfiber (very small)
      integer*2 nsmcom ! no. of points on compression stress-strain curve
      integer*2 nsmten ! no. of points on tension stress-strain curve
      real*4 smcom(2,MXSMC) ! Compression stress-strain points
      real*4 smten(2,MXSMT) ! Tension stress-strain points
      real*4 smrf           ! steel material reloading factor
c    MODIFY:
      integer*4 kdata       ! error counter, add 1 for each error
c ------------------------------------------------------------------
c  LABELLED COMMONS
      include 'cline.h'
      include 'ptop.h'
      common/tapes/inp,iou
c ------------------------------------------------------------------
c  LOCAL VARIABLES
      real*4 small  ! small number
      real*4 sig    ! stress
      real*4 eps    ! strain
      real*4 ymd    ! modulus
      real*4 yyp    ! previous modulus
      real*4 yye    ! initial elastic modulus
      real*4 ss     ! stress
      real*4 ee     ! strain
      real*4 yy     ! modulus
c ------------------------------------------------------------------
c ------------------------------------------------------------CONSTANTS
      small=1.e-6
c --------------------------------------------------------------HEADING
      if(ntyp.eq.1) write(iou,10)
   10 format (/' Steel Material Types'//
     1         '    Type',2x,'Comprn',2x,' Tensn',
     2         2x,' Unloading',2x,' Overshoot',
     3         2x,'    Stress',2x,'    Strain',2x,'Stress',
     4         2x,'   Tangent'/
     5         '     No.',2x,'Points',2x,'Points',
     6         2x,'   Factor ',2x,' Tolerance',
     7         2x,'     Value',2x,'     Value',2x,'  Type',
     8         2x,'   Modulus')
c-------------------------------------------READ AND ECHO CONTROL VALUES
      call getlin
      read(xxline,'(2i5,2f10.0)') nsmcom,nsmten,smrf,smot
      write(iou,'(i8,2i8,1pe12.3,1pe12.3)') ntyp,nsmcom,nsmten,smrf,smot
      if(nsmcom.gt.MXSMC .or. nsmten.gt.MXSMT) then
        write(iou,*) ' ***ERROR - too many points'
        kdata=kdata+1
      end if
      if(smot.lt.small) then
        write(iou,*) ' ***ERROR - overshoot tol too small'
        kdata=kdata+1
      end if
      smel=small
c---------------------------------------READ AND ECHO COMPRESSION VALUES
       ss=0
       ee=0
       yyp=1.e30
      do 21 i=1,nsmcom
c--read
        call getlin
```

```
        read(xxline,'(2f10.0)') sig,eps
        if(sig.le.0. .or. eps.le.0.) then
          write(iou,*) ' ***ERROR - values must be > 0'
          kdata=kdata+1
          sig=abs(sig)+1.e-10
          eps=abs(eps)+1.e-20
        end if
        if(eps.le.ee+small) then
          write(iou,*) ' ***ERROR - strains must increase'
          ee=eps-small
          kdata=kdata+1
        end if
c--tangent modulus
        smcom(1,i)=sig
        smcom(2,i)=eps
        yy=(smcom(1,i)-ss)/(smcom(2,i)-ee)
c--elastic modulus
        if(i.eq.1) yye=yy
c--echo
        write(iou,2000) sig,eps,'  comprn',yy
 2000   format(50x,1p2e12.4,a8,1pe12.4)
        if(yy.ge.yyp) then
          write(iou,*) ' ***ERROR - modulus must decrease'
          kdata=kdata+1
        end if
        yyp=yy
        if(yy.lt.0. .and. abs(yy).ge.0.5*yye) then
          write(iou,*) ' ***WARNING - large negative modulus'
        end if
c--save
        ss=sig
        ee=eps
 21    continue
c-------------------------------------READ AND ECHO TENSION VALUES
      ss=0.
      ee=0.
      yy=1.e30
      do 22 i=1,nsmten
c--read
        call getlin
        read(xxline,'(2f10.0)') sig,eps
        if(sig.le.0. .or. eps.le.0.) then
          write(iou,*) ' ***WARNING - values must be > 0'
          sig=abs(sig)+1.e-10
          eps=abs(eps)+1.e-20
        end if
        smten(1,i)=sig
        smten(2,i)=eps
        if(sig.le.ss+small) then
          write(iou,*) '***ERROR - stresses must increase'
          kdata=kdata+1
        end if
        if(eps.le.ee+small) then
          write(iou,*) '***ERROR - strains must increase'
          kdata=kdata+1
          ee=ee-small
        end if
c--tangent modulus
        ymd=(smten(1,i)-ss)/(smten(2,i)-ee)
c--echo
        write(iou,2000) sig,eps,'  tension',ymd
        if(ymd.gt.yy-small) then
          write(iou,*) '***ERROR - modulus must decrease'
          kdata=kdata+1
        end if
        ss=sig
        ee=eps
        yy=ymd
c--save
   22 continue
c----------------------------------------------------------------------
      RETURN
      END
c **********************************************************************
```

APPENDIX A-4
Steel material event determination

```
c *******************************************************************
      SUBROUTINE SMEV15(smcom, smten, deps, fact, ifact, stol, sig,
     1              etot, emax, emin, brnum, lastc, lastt, ssc,
     2              sst, smrf, piti)
C     MODIFIED BY THE MSU CFT RESEACH TEAM FOR STEEL MODEL, 02/20/02
c-------------------------------------------------------------------
c  DRAIN-3DX FIBER ELEMENTS
c-------------------------------------------------------------------
c  DOUBLE PRECISION / LARGE
      include 'double.h'
c-------------------------------------------------------------------
c  CALLED FROM : slev15
c -------------------------------------------------------------------
c INPUT:
      real*4 smcom(2,3)     ! stress-strain compressive matrix
      real*4 smten(2,2)     ! stress-strain tensile matrix
      real*8 deps  ! strain increment
      real*8 stol  ! stress overshoot tolerance
      real*4 sig   ! current stress
      real*4 etot  ! current stress
      real*4 emin  ! minimum strain
      real*4 emax  ! maximum strain
      real*4 epsu   ! strain at zero stress
      real*4 ds     ! stress increment
      integer*4 brnum ! branch number
                     !-1 = strain hardening branch
                     ! 0 = elastic branch
                     ! 1 = inelastic branch
                     ! 2 = intermediate branch
                     ! 3 = unloading/reloading branch - Tension
                     ! 4 = unloading/reloading branch - Compression
                     ! 5 = shooting branch
                     ! 6 = stress unloading platteu branch
                     ! 7 = yield platteu branch
                     ! 8 = buckled branch
      integer*4 lastc ! last branch number (compression)
      integer*4 lastt ! last branch number (tension)
c    MODIFY
      real*8 fact ! event factor
      integer*4 ifact ! event type
                      ! 1 = compression yielding
                      ! 2 = compression buckling
                      ! 3 = compression post-buckling
                      ! 4 = tension yielding
                      ! 5 = tension ultimate
                      ! 6 = unloading from compression envelope
                      ! 7 = unloading from tension envelope
                      ! 8 = shooting from tension to compression
                      ! 9 = loading onto compression envelope
                      ! 0 = loading onto tension envelope
c -------------------------------------------------------------------
c  LOCAL VARIABLES
      real*4 ssc    ! stress at unloading
      real*4 sst    ! stress at reloading
      real*4 ezsig  ! strain at zero stress
      real*4 syc    ! yield stress (compression)
      real*4 sbc    ! buckling stress (compression)
      real*4 suc    ! ultimate compressive stress
      real*4 syt    ! yield stress (tension)
      real*4 sut    ! ultimate tensile stress
      real*4 eyc    ! strain at yielding in compression
      real*4 delsteel  ! difference in strains between suc and sbc
      real*4 yini   ! fiber initial modulus
      real*4 yinel  ! fiber inelastic modulus
      real*4 yint   ! fiber intermediate modulus
      real*4 ysh    ! fiber strain-hardening modulus
      real*4 smrf   ! reloading factor
      real*4 sl     !
      real*4 yab    ! shooting modulus
      real*4 ee     ! difference in strain at branch #6
      real*4 star   ! target stress
      save yab
c -------------------------------------------------------------------

c --declaration of variables
      syc=smcom(1,1)
      sbc=smcom(1,2)
```

```
      suc=smcom(1,3)
      syt=smten(1,1)
      sut=smten(1,2)
      epsu=smcom(2,3)
      eyc=smcom(2,1)
      yini=syc/eyc
      delsteel=smcom(2,2)-eyc
      yinel=(sbc-syc)/delsteel
      yint=(suc-sbc)/(epsu-smcom(2,2))
      ysh=(sut-syt)/(smten(2,2)-smten(2,1))

C ----------------------------------------START EVENT FACTOR CALCULATION

C ----------------------------------START NEGATIVE DISPLACEMENT INCREMENT
      if(deps.lt.0.) then
C ----------------------------------------------------START ELASTIC FIBER
        if(brnum.eq.0) then
c --Branch 0-                                             Fifure 2.5
           ds=yini*deps
           if(-syc-stol-sig.ge.ds) then
              fact=(-syc-stol-sig)/ds
              ifact=1
           end if
C ---------------------------------------------------END ELASTIC FIBER

C -----------------------------------------------START INELASTIC FIBER
        else if(brnum.eq.1) then
c --Branch 1-                                             Figure 2.6
            if(lastt.eq.4) then
              if(emax-(syc+syt)/yini-delsteel-etot-stol/yinel.ge.de)then
                  fact=(emax-(syc+syt)/yini-delsteel-etot-stol/yinel)/de
                  ifact=2
               end if
            else
               if(sbc.ne.syc) then
                  ds=yinel*deps
                  if(-sbc-stol-sig.ge.ds) then
                     fact=(-sbc-stol-sig)/ds
                     ifact=2
                  end if
               else
                   if(emax-(syc+syt)/yini-delsteel-etot
     1                                 -stol/yinel.ge.de) then
                      fact=(emax-(syc+syt)/yini-delsteel
     2                      -etot-stol/yinel)/de
                      ifact=2
                   end if
               end if
            end if
            piti=316
c---------------------------------

        else if(brnum.eq.2) then
c --Branch 2-                                             Figure 2.8
            ds=yint*deps
            if(-suc+stol-sig.le.ds) then
                fact=(-suc+stol-sig)/ds
                ifact=3
            end if
c---------------------------------

        else if(brnum.eq.3) then
c --Branch 3-                                             Figure 2.9
            ds=yini*deps
            if(lastc.ne.5 .and. lastc.ne.6) then
               if(ssc-sig-stol.ge.ds) then
                   fact=(ssc-sig-stol)/(ds)
                   ifact=9
               end if
            else
               if(lastc.eq.5) then
                   star=(sig+yini*(ezsig-etot))/(1-yini/yab)
               else
                   star=ssc
               end if
               if (sig.ge.0) then
                    if(star-sig-stol.ge.ds) then
                         fact=(star-sig-stol)/ds
                         ifact=9
                   end if
               else
```

```
                  if(star-sig-stol.ge.ds) then
                       fact=(star-sig-stol)/ds
                       ifact=9
                  end if
              end if
           end if
c---------------------------------

        else if(brnum.eq.-1) then
c --Branch -1 -                                        Figure 2.11
           ds=yini*deps
           if(-stol.ge.ds) then
               fact=-stol/ds
               ifact=7
           end if
c---------------------------------

        else if(brnum.eq.4) then
c --Branch 4-                                          Figure 2.12
           ds=yini*deps
           if(lastc.eq.1 .and.(lastt.eq.-1 .or.lastt.eq.7)) then
              if(sst-syt-syc-stol-sig.ge.ds) then
                 fact=(sst-syt-syc-stol-sig)/ds
                 ifact=1
              end if
           else if((lastc.eq.2 .and. (lastt.eq.-1 .or. lastt.eq.7))
     1     .or. (lastc.eq.8 .and. (lastt.eq.-1 .or. lastt.eq.7))) then
              ezsig=etot-sig/yini
              if(ezsig-etot.ge.deps) then
                 fact=(ezsig-etot)/deps
                 ifact=8
              end if
           else if((lastt.eq.-1 .or. lastt.eq.7).and.piti.ne.316) then
              if(sst-syt-syc-stol-sig.ge.ds) then
                 fact=(sst-syt-syc-stol-sig)/ds
                 ifact=1
              end if
           end if
c---------------------------------

        else if(brnum.eq.5) then
c --Branch 5-                                          Figure 2.14
           s1=ssc/(emin-ezsig)
           yab=smrf*s1+(1-smrf)*yini
           ds=yab*deps
           if(ssc-sig-stol.ge.ds) then
              fact=(ssc-sig-stol)/ds
              ifact=8
           end if
c---------------------------------

        else if(brnum.eq.6) then
c --Branch 6-                                          Figure 2.15
           ee=emin-ezsig-ssc/yab
           s1=ssc/(emin-ezsig)
           yab=smrf*s1+(1-smrf)*yini
           if(emin-etot-stol/yab.ge.deps) then
              fact=(emin-etot-stol/yab)/deps
           else if(ee.eq.0) then
              fact=0
           end if
           ifact=9
c---------------------------------

        else if(brnum.eq.7) then
c --Branch 7-                                          Figure 2.17
           ds=yini*deps
           if(-stol.ge.ds) then
              fact=-stol/ds
              ifact=7
           end if
c---------------------------------

        else if(brnum.eq.8) then
c --Branch 8-                                          Figure 2.17
c --                                                       no event

C -----------------------------------------------END INELASTIC FIBER
        end if
C ----------------------------------END NEGATIVE DISPLACEMENT INCREMENT
      end if
```

```
C ---------------------------------START POSITIVE DISPLACEMENT INCREMENT
      if (deps.gt.0.) then
C --------------------------------------------------START ELASTIC FIBER
        if(brnum.eq.0) then
c -- Branch 0-                                       Figure 2.18
           ds=yini*deps
           if(syt+stol-sig.le.ds) then
              fact=(syt+stol-sig)/ds
              ifact=4
           end if
C -------------------------------------------------END ELASTIC FIBER

C ---------------------------------------------START INELASTIC FIBER
        else if(brnum.eq.1) then
c --Branch 1-                                        Figure 2.20
            ds=yini*deps
            if(stol.le.ds) then
                fact=stol/ds
                ifact=6
            end if
c---------------------------------

        else if(brnum.eq.2) then
c --Branch 2-                                        Figure 2.20
            ds=yini*deps
            if(stol.le.ds) then
                fact=stol/ds
                ifact=6
            end if
c---------------------------------

        else if(brnum.eq.3) then
c --Branch 3-                                        Figure 2.21
            ds=yini*deps
            if(lastc.eq.1) then
                if(ssc+syt+syc+stol-sig.le.ds) then
                   fact=(ssc+syt+syc+stol-sig)/(ds)
                   ifact=4
                end if
            else if((lastc.eq.2 .or. lastc.eq.8).or.(lastc.eq.5
     1              .or. lastc.eq.6)) then
                if(-sbc+syt+syc+stol-sig.le.ds) then
                   fact=(-sbc+syt+syc+stol-sig)/(ds)
                   ifact=4
                end if
            end if
c---------------------------------

        else if(brnum.eq.-1) then
c --Branch -1 -                                      Figure 2.19
            ds=ysh*deps
            if(sut-sig+stol.le.ds) then
                fact=(sut-sig+stol)/ds
                ifact=5
            end if
c---------------------------------

        else if(brnum.eq.4) then
c --Branch 4-                                        Figure 2.22
            ds=yini*deps
            if(lastt.eq.-1) then
               if(sst+stol-sig.le.ds) then
                  fact=(sst+stol-sig)/ds
                  ifact=0
               end if
            else if(lastt.eq.7) then
               if(sut-sig+stol.le.ds) then
                  fact=(sut-sig+stol)/ds
                  ifact=0
               end if
            end if
c---------------------------------

        else if(brnum.eq.5) then
c --Branch 5-                                        Figure 2.24
            ds=yini*deps
            if(stol.le.ds) then
               fact=stol/ds
               ifact=6
            end if
```

```
c---------------------------------

       else if(brnum.eq.6) then
c --Branch 6-                                              Figure 2.24
           ds=yini*deps
           if(stol.le.ds) then
             fact=stol/ds
              ifact=6
           end if
c---------------------------------

       else if(brnum.eq.7) then
c --Branch 7-                                              Figure 2.24
c --                                                          no event
c---------------------------------

       else if(brnum.eq.8) then
c --Branch 8-                                              Figure 2.20
           ds=yini*deps
           if(stol.le.ds) then
              fact=stol/ds
              ifact=6
           end if
C -----------------------------------------------END INELASTIC FIBER
       end if
C ----------------------------------END POSITIVE DISPLACEMENT INCREMENT
     end if
c----------------------------------------------------------------------
     RETURN
     END
c ********************************************************************
```

APPENDIX A-5
Steel material State determination

```
c ******************************************************************
      SUBROUTINE SMSD15(smcom, smten, deps, fact, ifact, stol, sig,
     1                  etot, emax, emin, brnum, lastc, lastt, ssc,
     2                  sst, smod, smrf, piti)
C     MODIFIED BY THE MSU CFT RESEACH TEAM FOR STEEL MODEL, 02/20/02
c-------------------------------------------------------------------
c  DRAIN-3DX FIBER ELEMENTS
c-------------------------------------------------------------------
c  DOUBLE PRECISION / LARGE
      include 'double.h'
c-------------------------------------------------------------------
c  CALLED FROM : slev15
c -------------------------------------------------------------------
c INPUT:
      real*4 smcom(2,3)    ! stress-strain compressive matrix
      real*4 smten(2,2)    ! stress-strain tensile matrix
      real*8 deps     ! strain increment
      real*4 sig      ! current stress
      real*4 etot     ! current strain
      real*4 smod     ! current modulus
      real*4 emin     ! minimum strain (Compression)
      real*4 emax     ! maximum strain (Tension)
      real*4 epsu     ! strain at zero stress
      real*4 ds       ! stress increment
      integer*4 brnum ! branch number
                      !-1 = strain hardening branch
                      ! 0 = elastic branch
                      ! 1 = inelastic branch
                      ! 2 = intermediate branch
                      ! 3 = unloading/reloading branch - Tension
                      ! 4 = unloading/reloading branch - Compression
                      ! 5 = shooting branch
                      ! 6 = stress unloading platteu branch
                      ! 7 = yield platteu branch
                      ! 8 = buckled branch
       integer*4 lastc ! last branch number (compression)
       integer*4 lastt ! last branch number (tension)

c    MODIFY
      real*8 fact ! event factor
c -------------------------------------------------------------------
c  LOCAL VARIABLES
      real*4 ssc    ! stress at unloading
      real*4 sst    ! stress at reloading
      real*4 ezsig  ! strain at zero stress
      real*4 syc    ! yield stress (compression)
      real*4 sbc    ! buckling stress (compression)
      real*4 suc    ! ultimate compressive stress
      real*4 syt    ! yield stress (tension)
      real*4 sut    ! ultimate tensile stress
      real*4 eyc    ! strain at yielding in compression
      real*4 delsteel  ! difference in strains between suc and sbc
      real*4 yini   ! fiber initial modulus
      real*4 yinel  ! fiber inelastic modulus
      real*4 yint   ! fiber intermediate modulus
      real*4 ysh    ! fiber strain-hardening modulus
      real*4 smrf   ! reloading factor
      real*4 s1     !
      real*4 yab    ! shooting modulus
      real*4 de     ! strain increment
      real*4 ee     ! difference in strains at branch #6
      real*4 star   ! target stress
      real*4 piti   ! Dummy
      real*4 smod   ! current moduli
      save yab
c -------------------------------------------------------------------

c --declaration of variables
      syc=smcom(1,1)
      sbc=smcom(1,2)
      suc=smcom(1,3)
      syt=smten(1,1)
      sut=smten(1,2)
      epsu=smcom(2,3)
      eyc=smcom(2,1)
      yini=syc/eyc
```

```
      delsteel=smcom(2,2)-eyc
      yinel=(sbc-syc)/delsteel
      yint=(suc-sbc)/(epsu-smcom(2,2))
      ysh=(sut-syt)/(smten(2,2)-smten(2,1))
C ----------------------------------------START EVENT FACTOR CALCULATION

C ---------------------------------START NEGATIVE DISPLACEMENT INCREMENT
      fact=1
      de=deps
      if(de.lt.0.) then
C ---------------------------------------------------START ELASTIC FIBER
C -- Branch 0-                                               Figure 2.5
        if(brnum.eq.0) then
           ds=yini*de
           if(-syc-sig.ge.ds) then
              fact=(-syc-sig)/ds
              sig=-syc
              etot=etot+fact*de
              de=(1-fact)*de
              brnum=1
              smod=yinel
           else
              sig=sig+ds
              etot=etot+fact*de
              de=0
              smod=yini
           end if

c --------------------------------------------------END ELASTIC FIBER

C ---------------------------------------------START INELASTIC FIBER
        else if(brnum.eq.1) then
c --Branch 1-                                                Figure 2.6
            if(lastt.eq.4) then
               if(emax-(syc+syt)/yini-delsteel-etot.ge.de) then
                  fact=(emax-(syc+syt)/yini-delsteel-etot)/de
                  sig=sst-syc-syt-delsteel*yinel
                  etot=etot+fact*de
                  de=(1-fact)*de
                  brnum=2
                  smod=yint
               else
                  sig=sig+de*yinel
                  etot=etot+fact*de
                  emin=etot
                  de=0
                  smod=yinel
               end if
            else
              if(sbc.ne.syc) then
                 ds=yinel*de
                 if(-sbc-sig.ge.ds) then
                    fact=(-sbc-sig)/(ds)
                    sig=-sbc
                    etot=etot+fact*de
                    de=(1-fact)*de
                    brnum=2
                    smod=yint
                 else
                    sig=sig+ds
                    etot=etot+fact*de
                    emin=etot
                    de=0
                    smod=yinel
                 end if
              else
                 if(emax-(syc+syt)/yini-delsteel-etot.ge.de) then
                    fact=(emax-(syc+syt)/yini-delsteel-etot)/de
                    sig=-sbc
                    etot=etot+fact*de
                    de=(1-fact)*de
                    brnum=2
                    smod=yint
                 else
                    sig=sig+de*yini
                    etot=etot+fact*de
                    emin=etot
                    de=0
                    smod=yinel
                 end if
              end if
```

```
            end if
            piti=316
c---------------------------------

        else if(brnum.eq.2) then
c --Branch 2-                                               Figure 2.8
            ds=yint*de
            if(-suc-sig.le.ds) then
                fact=(-suc-sig)/(ds)
                sig=-suc
                etot=etot+fact*de
                de=(1-fact)*de
                brnum=8
                smod=0
            else
                sig=sig+ds
                etot=etot+fact*de
                emin=etot
                de=0
                smod=yint
            end if
c---------------------------------

        else if(brnum.eq.3) then
c --Branch 3-                                               Figure 2.9
            ds=yini*de
            if(lastc.ne.5 .and. lastc.ne.6) then
                if(ssc-sig.ge.ds) then
                    fact=(ssc-sig)/(ds)
                    sig=ssc
                    etot=etot+fact*de
                    de=(1-fact)*de
                    if(ssc.eq.-suc) then
                       brnum=8
                       smod=0
                    else if(lastc.eq.1) then
                       brnum=1
                       smod=yinel
                    else if(lastc.eq.2) then
                       brnum=2
                       smod=yint
                    end if
                else
                    sig=sig+ds
                    etot=etot+fact*de
                    de=0
                    smod=yini
                end if
            else
                if(lastc.eq.5) then
                    star=(sig+yini*(ezsig-etot))/(1-yini/yab)
                else
                    star=ssc
                end if
                if (sig.ge.0) then
                    if(star-sig.ge.ds) then
                        fact=(star-sig)/ds
                        sig=star
                        etot=etot+fact*de
                        de=(1-fact)*de
                        if(lastc.eq.5) then
                            brnum=5
                            smod=yab
                        else
                            brnum=6
                            smod=0
                        end if
                    else
                        sig=sig+ds
                        etot=etot+fact*de
                        de=0
                        smod=yini
                    end if
                else
                    if(star-sig.ge.ds) then
                        fact=(star-sig)/ds
                        sig=star
                        etot=etot+fact*de
                        de=(1-fact)*de
                        if(lastc.eq.5) then
                            brnum=5
```

```
                    smod=yab
                 else
                    brnum=6
                    smod=0
                 end if
              else
                 sig=sig+ds
                 etot=etot+fact*de
                 de=0
                 smod=yini
              end if
           end if
        end if

c----------------------------------

      else if(brnum.eq.-1) then
c --Branch -1 -                                    Figure 2.11
c        etot=etot+fact*de
        sst=sig
        emax=etot
        brnum=4
        smod=yini
        lastt=-1
c----------------------------------

      else if(brnum.eq.4) then
c --Branch 4-                                      Figure 2.12
        ds=yini*de
        if(lastc.eq.1 .and. (lastt.eq.-1 .or. lastt.eq.7)) then
           if(sst-syt-syc-sig.ge.ds) then
              fact=(sst-syt-syc-sig)/ds
              sig=sst-syt-syc
              etot=etot+fact*de
              de=(1-fact)*de
              brnum=1
              smod=yinel
              lastt=4
           else
              sig=sig+ds
              etot=etot+fact*de
              de=0
              smod=yini
           end if
        else if((lastc.eq.2 .and.(lastt.eq.-1 .or. lastt.eq.7)) .or.
     1         (lastc.eq.8 .and. (lastt.eq.-1 .or. lastt.eq.7))) then
           ezsig=etot-sig/yini

           if(ezsig-etot.ge.de) then
              fact=(ezsig-etot)/de
              etot=ezsig
              de=(1-fact)*de
              sig=0
              brnum=5
              smod=yab
           else
              sig=sig+ds
              etot=etot+fact*de
              de=0
              smod=yini
           end if
        else if((lastt.eq.-1 .or. lastt.eq.7).and. piti.ne.316) then
           if(sst-syt-syc-sig.ge.ds) then
              fact=(sst-syt-syc-sig)/ds
              sig=sst-syt-syc
              etot=etot+fact*de
              de=(1-fact)*de
              brnum=1
              lastt=4
              smod=yinel
           else
              sig=sig+ds
              etot=etot+fact*de
              de=0
              smod=yini
           end if
        end if
c----------------------------------
      else if(brnum.eq.5) then
c --Branch 5-                                      Figure 2.14
        s1=ssc/(emin-ezsig)
```

```
            yab=smrf*s1+(1-smrf)*yini
            ds=yab*de
            if(ssc-sig.ge.ds) then
               fact=(ssc-sig)/ds
               sig=ssc
               etot=etot+fact*de
               de=(1-fact)*de
               brnum=6
               smod=0
            else
               sig=sig+ds
               etot=etot+fact*de
               de=0
               smod=yab
               lastc=5
            end if
c----------------------------------

        else if(brnum.eq.6) then
c --Branch 6-                                          Figure 2.15
            s1=ssc/(emin-ezsig)
            yab=smrf*s1+(1-smrf)*yini
            ee=emin-ezsig-ssc/yab
            if(emin-etot.ge.de) then
               fact=(emin-etot)/de
               etot=etot+fact*de
               de=(1-fact)*de
                  if(ssc.ne.suc) then
                   sig=ssc
                   brnum=2
                   smod=yint
               else
                   sig=suc
                   brnum=8
                   smod=0
               end if
            else if(ee.eq.0) then
               fact=0
               etot=etot+fact*de
               de=(1-fact)*de
               if(ssc.ne.suc) then
                   sig=ssc
                   brnum=2
                   smod=yint
               else
                   sig=suc
                   brnum=8
                   smod=0
               end if
            else
               etot=etot+fact*de
               de=0
               smod=0
               lastc=6
            end if

c----------------------------------
        else if(brnum.eq.7) then
c --Branch 7-                                          Figure 2.17
            emax=etot
            sst=sig
            brnum=4
            lastt=7
            smod=yini
c----------------------------------

        else if(brnum.eq.8) then
c --Branch 8-                                          Figure 2.17
c --                                                      no event
c            sig=-suc
            etot=etot+fact*de
            lastc=8
c            emin=etot
            de=0
            smod=0
C ------------------------------------------------END INELASTIC FIBER
        end if
C -----------------------------------END NEGATIVE DISPLACEMENT INCREMENT
      end if
C ---------------------------------START POSITIVE DISPLACEMENT INCREMENT
      if (de.gt.0) then
```

```
C ----------------------------------------------------START ELASTIC FIBER
       if(brnum.eq.0) then
C --Branch 0-                                                 Figure 2.18
          ds=yini*de
          if(syt-sig.le.ds) then
             fact=(syt-sig)/ds
             sig=syt
             etot=etot+fact*de
             de=(1-fact)*de
             brnum=-1
             smod=ysh
          else
             sig=sig+ds
             etot=etot+fact*de
             smod=yini
             de=0
          end if
C ----------------------------------------------------END ELASTIC FIBER

C -----------------------------------------------START INELASTIC FIBER
       else if(brnum.eq.1) then
c --Branch 1-                                                 Figure 2.20
           etot=etot+fact*de
           emin=etot
           ssc=sig
           brnum=3
           smod=yini
           lastc=1
c---------------------------------

       else if(brnum.eq.2) then
c --Branch 2-                                                 Figure 2.20
           etot=etot+fact*de
           emin=etot
           ssc=sig
           brnum=3
           smod=yini
           lastc=2
c---------------------------------

       else if(brnum.eq.3) then
c --Branch 3-                                                 Figure 2.21
           ds=yini*de
           if(lastc.eq.1) then
               if(ssc+syt+syc-sig.le.ds) then
                  fact=(ssc+syt+syc-sig)/(ds)
                  sig=ssc+syt+syc
                  etot=etot+fact*de
                  de=(1-fact)*de
                  brnum=-1
                  smod=ysh
               else
                  sig=sig+ds
                  etot=etot+fact*de
                  de=0
                  smod=yini
               end if
           else if((lastc.eq.2 .or. lastc.eq.8) .or. (lastc.eq.5
     1                 .or. lastc.eq.6)) then
               if(-sbc+syt+syc-sig.le.ds) then
                  fact=(-sbc+syt+syc-sig)/ds
                  sig=-sbc+syt+syc
                  etot=etot+fact*de
                  de=(1-fact)*de
                  brnum=-1
                  smod=ysh
               else
                  sig=sig+ds
                  etot=etot+fact*de
                  de=0
                  smod=yini
               end if
           end if
c---------------------------------

       else if(brnum.eq.-1) then
c --Branch -1 -                                               Figure 2.19
           ds=ysh*de
           if(sut-sig.le.ds) then
               fact=(sut-sig)/ds
               sig=sut
```

```
              etot=etot+fact*de
              de=(1-fact)*de
              brnum=7
              smod=0
          else
              sig=sig+ds
              etot=etot+fact*de
              emax=etot
              smod=ysh
              de=0
          end if
          lastt=-1
c--------------------------------

      else if(brnum.eq.4) then
c --Branch 4-                                          Figure 2.22
          ds=yini*de
          if(lastt.eq.-1) then
             if(sst-sig.le.ds) then
                fact=(sst-sig)/ds
                sig=sst
                etot=etot+fact*de
                de=(1-fact)*de
                brnum=-1
                smod=ysh
            else
                sig=sig+ds
                etot=etot+fact*de
                de=0
                smod=yini
             end if
          else if(lastt.eq.7) then
             if(sut-sig.le.ds) then
                fact=(sut-sig)/ds
                sig=sut
                etot=etot+fact*de
                de=(1-fact)*de
                brnum=7
                smod=0
             else
                sig=sig+ds
                etot=etot+fact*de
                de=0
                smod=yini
             end if
          end if
c--------------------------------

      else if(brnum.eq.5) then
c --Branch 5-                                          Figure 2.24
          etot=etot+fact*de
          brnum=3
          smod=yini
          lastc=5
c--------------------------------

      else if(brnum.eq.6) then
c --Branch 6-                                          Figure 2.24
          etot=etot+fact*de
          brnum=3
          smod=yini
          lastc=6
c--------------------------------

      else if(brnum.eq.7) then
c --Branch 7-                                          Figure 2.24
c --                                                      no event
          etot=etot+fact*de
          de=0
          smod=0
c--------------------------------

      else if(brnum.eq.8) then
c --Branch 8-                                          Figure 2.20
          emin=etot
          ssc=sig
          brnum=3
          smod=yini
          lastc=8

C ----------------------------------------------END INELASTIC FIBER
```

```
            end if
C -----------------------------------END POSITIVE DISPLACEMENT INCREMENT
         end if
c -----------------------------------------------------------------------
         RETURN
         END
c ***********************************************************************
```

REFERENCES

American Institute of Steel Construction (1999), Load and Resistance Factor Design Specification for Structural Steel Buildings, Second Edition, Chicago, IL.

El-Sheikh, M., Sause, R., Pessiki, S.P., Lu, L.-W., and Kurama, Y. (1997), "Seismic Analysis, Behavior, and Design of Unbonded Post-Tensioned Precast Concrete Frames," Earthquake Engineering Research Report No. EQ-97-02, Department of Civil and Environmental Engineering, Lehigh University, Bethlehem, Pennsylvania.

Kurama, Y. (1997), PhD. Dissertation, "Seismic analysis, behavior, and design of unbonded post-tensioned precast concrete walls," Lehigh University.

Kurama, Y., Pessiki, S.P., Sause, R., Lu, L.-W, and El-Sheikh, M. (1996), "Analytical Modeling and Lateral Load Behavior of Unbonded Post-Tensioned Precast Concrete Walls," Earthquake Engineering Research Report No. EQ-96-02, Department of Civil and Environmental Engineering, Lehigh University, Bethlehem, PA, 1996.

Prakash, V., Powell, G., and Campbell, S. (1993), "DRAIN-2DX Base Program Description and User Guide - Version 1.10," Report No. UCB/SEMM-93/17 and 93/18, Structural Engineering Mechanics and Materials, Department of Civil Engineering, University of California, Berkeley, CA.

Ricles, J., Sause, R., Garlock, M., and Zhao, C. (2001), "Post-tensioned Seismic-Resistant Connections for Steel Frames," Journal of Structural Engineering, Vol. 127, No. 2, February 2001, pp. 113-121.

Shen, Q. and Kurama, Y. (2002), "Nonlinear Behavior of Unbonded Post-Tensioned Hybrid Coupled Wall Subassemblages," Journal of Structural Engineering, ASCE, Vol. 128, No. 10 (in print).

Varma, A.H., Ricles, J.M., and Sause, R. (2001), "Seismic Behavior, Analysis, and Design of High Strength Square Concrete Filled Steel Tube (CFT) Columns," ATLSS Report No. 2001-02, ATLSS Engineering Research Center, Dept. of Civil and Env. Eng., Lehigh University, Bethlehem, PA.