

Structural Engineering Report No. ST-93-5.2

**CYCLIC NONLINEAR ANALYSIS OF
CONCRETE-FILLED STEEL TUBE
BEAM-COLUMNS AND COMPOSITE FRAMES**

Brett C. Gourley and Jerome F. Hajjar

November 1994

**Department of Civil Engineering
Institute of Technology
University of Minnesota
Minneapolis, Minnesota 55455**

Abstract

Concrete-filled steel tubes (CFTs) are being used in an increasing number of structural engineering applications. One particular application which exploits the many advantages a CFT offers is a moment-resisting frame consisting of steel I-beams framing rigidly into CFT beam-columns. This research examines the load-deflection relationship of CFT beam-columns in detail, with the final objective of modeling the cyclic nonlinear behavior of CFT beam-columns in frame structures. An analytical model is presented to simulate both the monotonic and cyclic behavior of single member CFTs and composite frame structures composed of both CFT and steel members.

The first part of this research examines the cross-section strength of a CFT member. Uniaxial stress-strain curves are developed to model the multiaxial stress behavior of the CFT due to the steel tube confining the concrete. These stress-strain curves are incorporated into a fiber element analysis system, which is used to generate accurate CFT cross-section strength surfaces. The cross-section study culminates with the development of an empirical polynomial expression for the three-dimensional (P - M_y - M_z) cross-section strength surface of a CFT. This expression is verified against the fiber model results and is generalized for CFTs having a wide range of material strengths and cross-section dimensions.

The expression for the three-dimensional cross-section strength of a CFT forms the basis for the second part of the research--the development of a compact and efficient macro analytical model to accurately simulate the second order inelastic behavior of a CFT beam-column. This model is incorporated into a finite element analysis computer program, developed for this work, to analyze single members and composite frame structures. The material nonlinear behavior of CFTs is modeled using a concentrated plasticity two-surface bounding surface approach implemented in three-dimensional force-space. The loading and bounding surfaces in the model are asymmetric for CFTs

due to the infilled concrete. A formulation is presented for the kinematic and isotropic hardening of these asymmetric surfaces to model the complex cyclic load-deformation behavior of the CFT beam-columns.

The CFT inelastic beam element formulation includes a number of experimentally-calibrated parameters which model behavioral characteristics such as stiffness degradation, strength degradation, and the decrease in the size of the elastic zone as a CFT is subjected to cyclic loading. A number of monotonic and cyclic experimental tests with varying material and geometric properties are examined to calibrate these parameters of the model. Several additional monotonic and cyclic tests are then presented as verification of the accuracy and broad scope of applicability of the nonlinear CFT beam-column model, including a final example consisting of a cyclically-loaded three-dimensional subassembly of steel I-beams framing into a CFT beam-column.

Acknowledgments

The authors would like to thank Professor Gregory G. Deierlein of Cornell University for providing source code that was modified for incorporation into a portion of this research. Additionally, the authors would like to thank Professors Chiaki Matsui of Kyushu University, Fukuoka, Japan and Shosuke Morino of Mie University, Tsu, Japan for their generous response to requests for further information on their concrete-filled steel tube research. The author thank Paul Schiller, Dan Flemming, Mike Kannel, Jane Lundberg, and Chris Earls for their assistance with this research. This research was funded by the University of Minnesota. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the University of Minnesota.

Table of Contents

Abstract	i
Acknowledgments	iii
Table of Contents	iv
List of Figures	vii
List of Tables	x
1 Introduction	1
1.1 Concrete-Filled Steel Tubes in Frame Structures.....	2
1.2 Research Objectives.....	6
1.3 Scope of Research.....	8
1.4 Organization of Thesis.....	10
2 Analysis of CFT Cross-Section Strength	12
2.1 CFT Cross-Section Behavior.....	14
2.1.1 Axially-Loaded Cross-Sections.....	14
2.1.2 Combined Axial Load and Bending.....	16
2.1.3 Other Cross-Section Effects.....	18
2.2 Fiber Element Method for Computing CFT Cross-Section Strength.....	21
2.2.1 Background on Methods for Cross-Section Strength Computation.....	21
2.2.2 Fiber Element Analysis.....	21
2.2.3 Discretization of the Cross-Section.....	22
2.2.4 Moment-Curvature-Thrust Analysis.....	24
2.2.5 Analysis Parameters.....	27
2.3 Steel and Concrete Constitutive Models.....	28
2.3.1 Concrete Constitutive Model.....	29
2.3.2 Steel Constitutive Model.....	33
2.4 Criteria for Terminating the Moment-Curvature-Thrust Computation.....	35
2.4.1 Discussion of the Termination Criteria.....	36
2.4.2 Values Used for the Termination Criteria.....	38
2.5 Comparison to Experimental Results.....	39
2.6 Equation for the Three-Dimensional Cross-Section Strength Surface.....	51
2.6.1 Selection of CFT Cross-Sections.....	51
2.6.2 Development of the CFT Cross-Section Strength Equation Form.....	53
2.6.3 Modeling the Asymmetry of the Cross-Section Strength Surface.....	57
2.6.4 Development of the Coefficient Equations.....	59
2.6.5 Rectangular CFTs with Aspect Ratios Greater Than 1:1.....	78

2.6.6	CFTs with Higher Strength Steel	84
2.6.7	Nominal Strength of CFTs	85
3	Introduction to the CFT Nonlinear Model	89
3.1	Behavior of Rectangular CFT Beam-Columns	89
3.1.1	Monotonic Behavior of Rectangular CFT Beam-Columns	90
3.1.2	Cyclic Behavior of Rectangular CFT Beam-Columns	93
3.2	CFT Element Formulation.....	94
3.2.1	Background on CFT Behavior Models.....	94
3.2.2	Overview of the Finite Element Macro Model.....	96
3.3	Elastic CFT Stiffness Properties.....	98
3.3.1	Axial Rigidity	99
3.3.2	Flexural Rigidity.....	99
3.3.3	Torsional Rigidity.....	101
3.4	Geometric Nonlinear Formulation	103
3.5	Material Nonlinear Formulation.....	105
3.5.1	Concentrated Plasticity Models	105
3.5.2	Introduction to the CFT Plasticity Model.....	109
3.5.3	Modeling of Inelastic CFT Behavior.....	110
4	CFT Force-Space Plasticity Formulation	113
4.1	Definitions and Assumptions of the Bounding Surface Model.....	114
4.1.1	Loading and Bounding Surfaces.....	114
4.1.2	Basic Assumptions	118
4.2	Derivation of the Plastic Reduction Matrix.....	120
4.3	Calculation of Plastic Stiffness.....	124
4.3.1	Plasticity Coefficients.....	125
4.3.2	Distance Between the Surfaces.....	127
4.4	Isotropic Hardening	128
4.4.1	Calculation of Plastic Work	129
4.4.2	Isotropic Hardening of the Loading and Bounding Surfaces	132
4.5	Variation of κ_2 and E_c	136
4.6	Kinematic Hardening.....	137
5	Verification and Calibration	144
5.1	Verification of the Steel Plasticity Analysis.....	145
5.1.1	Steel Cantilever Beam	145
5.1.2	Steel Beam-Column	148
5.2	CFT Plasticity Analysis Calibration	150
5.2.1	Calibration Parameters and Tests	150
5.2.2	Calibration Procedure and Results	153
5.3	CFT Plasticity Analysis Verification.....	167
5.4	Verification of the Nonlinear Beam-Column Model.....	181

6 Conclusions	185
6.1 Conclusions	186
6.2 Suggestions for Future Research	188
Appendix	191
A CFT Experimental Tests	191
B Key Features of the Computer Implementation	195
C CFTmacro Source Code	201
D List of Symbols	264
List of References	269

List of Figures

1.1	Medium Rise CFT Theme Structure (after Yamanouchi, et al [1993]).....	3
2.1	Cross-Section Strength Curves for Typical CFT Cross-Sections	17
2.2	CFT Geometry and Typical Meshing Scheme	23
2.3	Fiber Element Analysis Definitions (after Sanz-Picon [1992])	25
2.4	Uniaxial Stress-Strain Curve for Concrete in Rectangular CFTs	29
2.5	Uniaxial Stress-Strain Curve for Steel in Rectangular CFTs.....	34
2.6	Moment-Curvature-Thrust Diagram (Tom44a); $P/P_o = 0.13$	41
2.7	Moment-Curvature-Thrust Diagram (Tom44a); $P/P_o = 0.27$	42
2.8	Moment-Curvature-Thrust Diagram (Tom44a); $P/P_o = 0.58$	42
2.9	Moment-Curvature-Thrust Diagram (Tom44b); $P/P_o = 0.0$	43
2.10	Moment-Curvature-Thrust Diagram (Tom44b); $P/P_o = 0.26$	43
2.11	Moment-Curvature-Thrust Diagram (Tom44b); $P/P_o = 0.57$	44
2.12	Moment-Curvature-Thrust Diagram (Tom33); $P/P_o = 0.0$	44
2.13	Moment-Curvature-Thrust Diagram (Tom33); $P/P_o = 0.47$	45
2.14	Moment-Curvature-Thrust Diagram (Tom24); $P/P_o = 0.0$	45
2.15	Moment-Curvature-Thrust Diagram (Tom24); $P/P_o = 0.29$	46
2.16	Moment-Curvature-Thrust Diagram (Tom24); $P/P_o = 0.57$	46
2.17	Two-Dimensional Cross-Section Strength Diagram (Tom44a).....	47
2.18	Two-Dimensional Cross-Section Strength Diagram (Tom44b)	47
2.19	Two-Dimensional Cross-Section Strength Diagram (Tom33)	48
2.20	Two-Dimensional Cross-Section Strength Diagram (Tom24)	48
2.21	Two-Dimensional Cross-Section Strength Diagram (Furl1)	49
2.22	Two-Dimensional Cross-Section Strength Diagram (Furl2)	50
2.23	Two-Dimensional Cross-Section Strength Diagram (Furl3)	50
2.24	Typical CFT Two-Dimensional Cross-Section Strength Diagram	54
2.25	Two-Dimensional Cross-Section Strength Diagrams ($D/t = 24$)	72
2.26	Two-Dimensional Cross-Section Strength Diagrams ($D/t = 48$)	72
2.27	Two-Dimensional Cross-Section Strength Diagrams ($D/t = 72$)	73
2.28	Two-Dimensional Cross-Section Strength Diagrams ($D/t = 96$)	73
2.29	Two-Dimensional Cross-Section Strength Diagrams ($f'_c = 3.5$ ksi)	74
2.30	Two-Dimensional Cross-Section Strength Diagrams ($f'_c = 6.5$ ksi)	74
2.31	Two-Dimensional Cross-Section Strength Diagrams ($f'_c = 10$ ksi)	75
2.32	Two-Dimensional Cross-Section Strength Diagrams ($f'_c = 15$ ksi)	75
2.33	CFT Cross-Section Strength Diagrams-- M_y/M_{y_o} vs. M_z/M_{z_o} ; Section 24A ($D/t = 24$, $f'_c = 3.5$ ksi).....	76
2.34	CFT Cross-Section Strength Diagrams-- M_y/M_{y_o} vs. M_z/M_{z_o} ; Section 48B ($D/t = 48$, $f'_c = 6.5$ ksi).....	76
2.35	CFT Cross-Section Strength Diagrams-- M_y/M_{y_o} vs. M_z/M_{z_o} ; Section 72C ($D/t = 72$, $f'_c = 10$ ksi).....	77

2.36	CFT Cross-Section Strength Diagrams-- M_y/M_{y0} vs. M_z/M_{z0} ; Section 96D ($D/t = 96$, $f'_c = 15$ ksi).....	77
2.37	Two-Dimensional Cross-Section Strength Diagram--Major Axis Moment; (Major Axis $D/t = 48$, Minor Axis $D/t = 24$, $f'_c = 3.5$ ksi)	80
2.38	Two-Dimensional Cross-Section Strength Diagram--Minor Axis Moment; (Major Axis $D/t = 48$, Minor Axis $D/t = 24$, $f'_c = 3.5$ ksi)	80
2.39	CFT Cross-Section Strength Diagrams-- M_y/M_{y0} vs. M_z/M_{z0} ; (Major Axis D/t $= 48$, Minor Axis $D/t = 24$, $f'_c = 3.5$ ksi)	81
2.40	Two-Dimensional Cross-Section Strength Diagram--Major Axis Moment; (Major Axis $D/t = 72$, Minor Axis $D/t = 36$, $f'_c = 15$ ksi)	81
2.41	Two-Dimensional Cross-Section Strength Diagram--Minor Axis Moment; (Major Axis $D/t = 72$, Minor Axis $D/t = 36$, $f'_c = 15$ ksi)	82
2.42	CFT Cross-Section Strength Diagrams-- M_y/M_{y0} vs. M_z/M_{z0} ; (Major Axis D/t $= 72$, Minor Axis $D/t = 36$, $f'_c = 15$ ksi)	82
2.43	Two-Dimensional Cross-Section Strength Diagram--Major Axis Moment; (Major Axis $D/t = 96$, Minor Axis $D/t = 48$, $f'_c = 10$ ksi)	83
2.44	Two-Dimensional Cross-Section Strength Diagram--Minor Axis Moment; (Major Axis $D/t = 96$, Minor Axis $D/t = 48$, $f'_c = 10$ ksi)	83
2.45	CFT Cross-Section Strength Diagrams-- M_y/M_{y0} vs. M_z/M_{z0} ; (Major Axis D/t $= 96$, Minor Axis $D/t = 48$, $f'_c = 10$ ksi)	84
3.1	Typical CFT Load-Deflection Curve	92
3.2	CFT Beam Element.....	97
3.3	Bounding Surface Model	106
3.4	Kinematic Hardening	108
3.5	Isotropic Hardening.....	108
3.6	Cyclic CFT Behavior (after Sakino and Tomii [1981]).....	110
4.1	CFT Bounding Surface Model.....	114
4.2	Effect of κ_2 on the Rate of Element Plastification	126
4.3	Distance Between Loading and Bounding Surface.....	128
4.4	Isotropic Hardening of Loading and Bounding Surfaces	134
4.5	Kinematic Hardening by the Tseng Method	139
5.1	Cantilever Beam.....	145
5.2	Steel Cantilever: Axial Load vs. Displacement	146
5.3	Steel Cantilever: Major Axis Moment vs. Rotation	147
5.4	Steel Cantilever: Minor Axis Moment vs. Rotation	147
5.5	Beam-Column	148
5.6	Steel Beam-Column: Major Axis Bending	149
5.7	Steel Beam-Column: Minor Axis Bending.....	149
5.8	Eccentrically-Loaded Beam-Column.....	153
5.9	Axially-Loaded Beam in Bending.....	153
5.10	Cyclically-Loaded Shear Specimen	153

5.11	Initial Loading Surface Size and Shape	155
5.12	Eccentrically-Loaded Beam-Column (Br3)	161
5.13	Eccentrically-Loaded Beam-Column (Ced2).....	161
5.14	Eccentrically-Loaded Beam-Column (Ced9).....	162
5.15	Eccentrically-Loaded Beam-Column (SK91_4)	162
5.16	Axially-Loaded Beam in Bending (Tom24_3)	163
5.17	Axially-Loaded Beam in Bending (Tom24_6)	163
5.18	Axially-Loaded Beam in Bending (Tom44_3)	164
5.19	Axially-Loaded Beam in Bending (Tom44_6)	164
5.20	Cyclically-Loaded Shear Specimen (Sak24_2).....	165
5.21	Cyclically-Loaded Shear Specimen (Sak34_3).....	165
5.22	Cyclically-Loaded Shear Specimen (Sak46_5).....	166
5.23	Eccentrically-Loaded Beam-Column (Br1)	170
5.24	Eccentrically-Loaded Beam-Column (Br4)	171
5.25	Eccentrically-Loaded Beam-Column (Br5)	171
5.26	Eccentrically-Loaded Beam-Column (Br6)	172
5.27	Eccentrically-Loaded Beam-Column (Br7)	172
5.28	Eccentrically-Loaded Beam-Column (Br8)	173
5.29	Eccentrically-Loaded Beam-Column (Ced1).....	173
5.30	Eccentrically-Loaded Beam-Column (Ced6).....	174
5.31	Eccentrically-Loaded Beam-Column (Ced7).....	174
5.32	Eccentrically-Loaded Beam-Column (Ced10).....	175
5.33	Eccentrically-Loaded Beam-Column (Ced13).....	175
5.34	Eccentrically-Loaded Beam-Column (SK89_2)	176
5.35	Eccentrically-Loaded Beam-Column (SK89_5)	176
5.36	Eccentrically-Loaded Beam-Column (SK91_9)	177
5.37	Eccentrically-Loaded Beam-Column (SK91_10)	177
5.38	Axially-Loaded Beam in Bending (Tom24_2)	178
5.39	Axially-Loaded Beam in Bending (Tom24_4)	178
5.40	Axially-Loaded Beam in Bending (Tom24_5)	179
5.41	Axially-Loaded Beam in Bending (Tom44_2)	179
5.42	Axially-Loaded Beam in Bending (Tom44_4)	180
5.43	Axially-Loaded Beam in Bending (Tom44_5)	180
5.44	Three-Dimensional Subassembly (after Morino et al. [1993])	181
5.45	3D Subassembly: Analytical Model.....	183
5.46	3D Subassembly: Shear-Rotation Hysteresis Curve	184
B.1	Unit Vectors Denoting Global Orientation of Steel and CFT Elements.....	196
B.2	Sample Load Histories	198

List of Tables

2.1	Rectangular CFT Cross-Section Tests	40
2.2	CFT Cross Sections Used for Determining the Cross-Section Strength Surface Equation	52
2.3	Errors in Calculation of Cross-Section Strength Surface Centroids	58
2.4	Optimum CFT Cross-Section Strength Surface Equation Coefficients	60
2.5	Errors in Coefficient c_1	62
2.6	Errors in Coefficient c_2	63
2.7	Errors in Coefficient c_3	64
2.8	Errors in Coefficient c_4	65
2.9	CFT Cross-Section Strength Surface Equation	67
2.10	Cumulative Errors in CFT Cross-Section Strength Surface Equation (Surface Equation vs. Fiber Element Analysis)	68
2.11	Cross-Section Strength Surface Equation Errors for Tension Region	69
2.12	CFT Cross-Section Strength Surface Equation Errors: Additional CFT Cross-Sections	71
2.13	CFT Cross-Section Strength Surface Equation Errors: Rectangular Cross- Sections	79
2.14	CFT Cross-Section Strength Surface Equation Errors: CFT Cross-Sections with Higher Strength Steel	85
2.15	Nominal Moment Calculation Errors	88
3.1	Equations for the Elastic Rigidity of CFT Beam-Columns	103
5.1	List of Calibration Parameters	150
5.2	CFT Calibration Tests	152
5.3	Final Calibrated Parameters	159
5.4	CFT Verification Tests	168
A.1	Rectangular CFT Cross-Section Tests	192
A.2	Monotonic Rectangular Beam-Column Tests	193
A.3	Cyclic Rectangular Beam-Column Tests	194
B.1	Sample Input File for CFTmacro	199
C.1	CFTmacro Source Code	202
C.2	CFTmacro Header Files	254

Chapter 1

Introduction

Composite members have long been recognized as efficient means of resisting loads in structures. These structural elements combine the beneficial qualities of steel and concrete to form a member with qualities superior to the individual components themselves. The steel provides high stiffness and tensile strength; the concrete provides compressive strength and economy. A composite beam-column may take one of two basic forms: 1) a steel-reinforced concrete member (SRC), which consists of a steel W-section encased in reinforced or unreinforced concrete, or 2) a concrete-filled steel tube (CFT), which may be either a circular or a rectangular structural steel tube filled with reinforced or unreinforced concrete.

Steel-reinforced concrete members have been used with increasing regularity in the past decade. Only recently, however, have CFT members become more prevalent in structural framing systems. For building structures, CFTs have been used primarily as columns, most notably forming the primary columns in braced lateral load-resisting and gravity load-resisting systems in high-rise structures [Bode, 1976; Griffis, 1992]. Two unique examples are the 62-story Two Union Square building in Seattle [Godfrey, 1987] and Casselden Place, a high-rise structure in Melbourne [Bridge and Webb, 1993; Webb,

1993]. The Two Union Square building utilizes four 10 foot diameter CFT super-columns with 19 ksi concrete as the primary lateral load-resisting element and additional CFT perimeter columns to resist gravity loads. Casselden Place incorporates CFTs as the gravity load-resisting system into which composite beams frame. The use of CFTs in this structure minimized the required construction labor and maximized the speed of construction. Concrete-filled steel tubes have also been used in seismic applications, in particular for bridge piers [Bode, 1976; Kitada, 1992; Priestley et al., 1994]. Recent trends indicate an increase in the use of CFT members, not only in applications similar to those discussed above, but as beam-columns in new and innovative applications that take advantage of the many benefits a CFT offers [Goel and Yamanouchi, 1993]. The research reported herein presents a computational model for simulating the behavior of CFT beam-columns used in frames subjected to monotonic or cyclic loading.

1.1 Concrete-Filled Steel Tubes in Frame Structures

Concrete-filled steel tubes provide manifold benefits and advantages by exploiting the beneficial properties of steel and concrete and the interaction between the two materials. CFTs offer high strength and stiffness, excellent seismic properties, economy, and efficiency [Tomii et al., 1973; Bode, 1976; Kitada, 1992; Webb, 1993]. The advantages of using CFT members become especially apparent when CFTs are used in braced and unbraced frames. When used as beam-columns in these types of structures, CFTs demonstrate superior economy and strength over traditional steel or reinforced concrete members [Matsui, 1986; Morino et al., 1993].

The usefulness, economy, and efficiency of CFT systems has been recognized by engineers across the world, especially in Japan. The Japanese Planning Group for the 1992 US-Japan Workshop on Hybrid and Composite Structures [Yamanouchi et al., 1993] has identified several CFT structural systems for future large-scale testing. One theme structure presented by this group consists of a two-way unbraced frame structure composed of steel beams and CFT beam-columns (Fig. 1.1). This unbraced composite frame represents one type of structure whose monotonic and cyclic behavior may be assessed using the analytical formulation to be presented in this work.

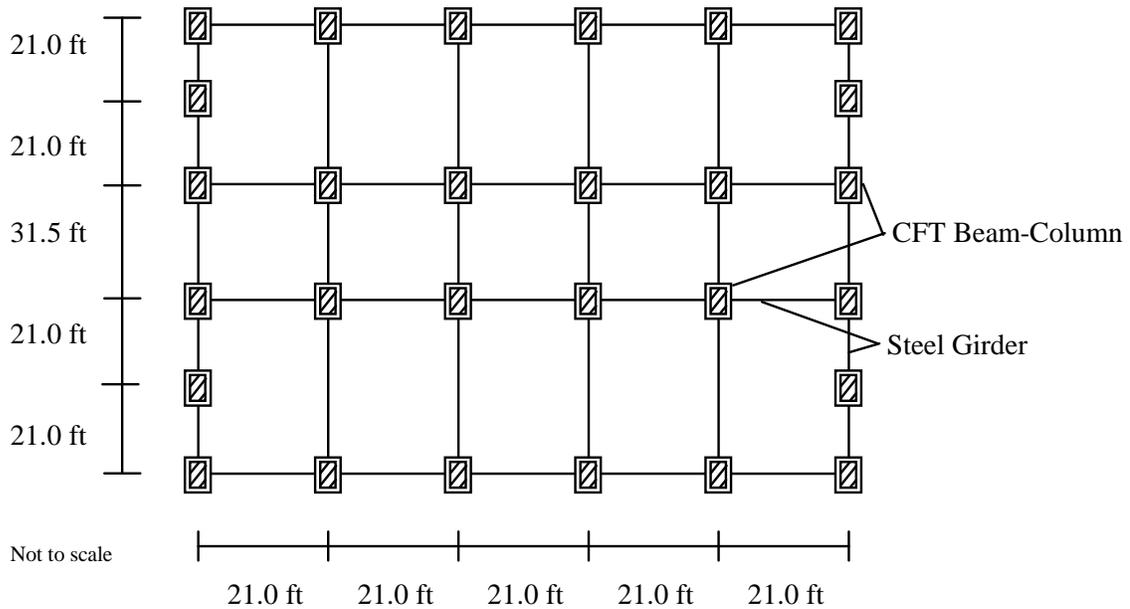


Figure 1.1 Medium Rise CFT Theme Structure (after Yamanouchi et al. [1993])

CFT members perform efficiently under a variety of loading conditions, providing excellent compressive strength, as well as superior flexural and axial stiffness. The orientation of the steel and the concrete in a CFT takes advantage of the properties of both materials. The steel tube forming the exterior of the member, has a much larger

modulus of elasticity and ultimate strength than the concrete and provides a large moment of inertia, leading to high stiffness and flexural capacity [Bridge, 1976]. The concrete, on the other hand, forms an ideal core to withstand the compressive loading typical in beam-column applications.

When a CFT member is subjected to compressive loading, further benefits arise due to the interaction of the steel and concrete. Many authors have recognized that CFTs under compressive loads provide strength in excess of the sum of the individual material strengths of the steel and concrete [Gardner, 1968; Tomii et al., 1973; Ichinohe et al., 1991; Tomii, 1991]. As a CFT undergoes compressive loading, the concrete begins to rapidly expand at a strain of approximately 0.002, inducing an outward pressure on the steel tube [Gardner and Jacobson, 1967; Knowles and Park, 1969]. The steel tube, in turn, exerts a confining pressure on the concrete, resulting in added concrete strength and ductility. Spalling of the concrete, often a problem in reinforced concrete members, is inhibited by the steel tube. Concrete, normally a relatively brittle material, becomes ductile when combined with the steel, resulting in a member with favorable seismic properties [Matsui, 1986; Tomii, 1991].

While the steel improves the properties of the concrete, the concrete contributes to the performance of the steel tube. The presence of the concrete core significantly delays local buckling of the tube by forcing all buckling modes outward [Tomii et al., 1973; Kitada and Nakai, 1991]. The delayed local buckling and increased concrete ductility contribute to the toughness of CFTs--the ability of CFTs to sustain a high proportion of their capacity even after local buckling of the steel and crushing of the concrete has occurred [Bridge, 1976].

In addition to their efficient load resistance on a single member level, CFTs provide a number of economic benefits and additional advantages specific to frame structures. Frame structures composed of CFT beam-columns and wide flange beams,

with proper connection design, provide high energy absorption characteristics and, in turn, very good seismic load resistance [Matsui, 1986]. The symmetry of circular or square CFT cross-sections produces a beam-column with equal resistance to bi-directional moments, a distinct advantage in three-dimensional moment-resisting frames (e.g., Fig. 1.1). In moderate- to high-rise construction, a building with CFT beam-columns can ascend much more quickly than a comparable reinforced concrete structure since the steelwork can precede pouring of the concrete by several floors [Webb, 1993]. The steel tubes also serve as the formwork for the wet concrete during construction, which decreases both material and labor expenses [Bode, 1976; Prion and Boehme, 1989]. The cost of a CFT member on a strength per dollar basis is much less than steel and roughly equivalent to reinforced concrete for low to medium strength concrete. With the use of high-strength concrete, though, CFTs are stronger per square foot than conventional reinforced concrete columns [Webb, 1993]. Finally, in high-strength applications, a smaller column size may be used, increasing the amount of usable floor space in office buildings.

The disadvantages of using CFTs in frame structures should be considered as well as the advantages. The exposed bare steel tube requires expensive fire-protection. The disadvantage of the low fire-resistance of the steel is somewhat mitigated by the concrete core, which provides a larger thermal capacity than air inside a hollow tube, thus decreasing the amount of necessary fire-proofing material [Tomii et al., 1973; Lie and Stringer, 1994]. A CFT, however, still remains more susceptible to fire damage than a reinforced concrete or steel-reinforced concrete member and, at present, must be fire-proofed accordingly.

Another potential drawback of CFTs in frame structures, particularly unbraced frames, is the difficulty and expense of providing rigid beam-to-column connections [Prion and Boehme, 1989]. Tube connections have been studied by a number of

researchers and industry experts, however, and the results are now starting to be synthesized into comprehensive design specifications [Matsui, 1986; Prion and McLellan, 1992; Azizinamini and Prakash, 1993; BMTC, 1994].

Current design specifications for CFTs for both connections and single members provide only limited guidance for the engineer, especially with the use of high strength materials. Lundberg [1993] has compared the AISC LRFD [1994] calculation for beam-column capacity to experimental results and found that the code is inconsistent and may even be unconservative in some cases. The shortcomings of the current design specifications stem in large part from the limited base of experimental and analytical research to quantify CFT behavior.

Despite the potential drawbacks of CFTs, their advantages over traditional structural members promise to establish an expanding role for this versatile structural element in modern construction. With continued analytical and experimental research, many of the current disadvantages of CFTs may be overcome, leading to a more thorough understanding of CFT beam-column behavior and more comprehensive design specifications.

1.2 Research Objectives

The trend toward the use of CFT structural members in frames has presented the need for more comprehensive experimental and analytical research to better understand the complicated behavior of CFTs [Goel and Yamanouchi, 1993]. The analytical requirements for studying CFT frame behavior consist of developing an efficient method to study the overall frame load-deflection behavior in both static and seismic loading

situations. The purpose of this research is to develop an analytical procedure which examines CFT member and frame behavior at the macroscopic level, i.e., each structural member is usually modeled by one to three discrete beam-type finite elements. In particular, the nonlinear finite element model developed in this work will be used to study three-dimensional unbraced moment-resisting frames composed of steel I-girders framing rigidly into rectangular CFT beam-columns and subjected to either monotonic or cyclic loading.

The development of an analytical procedure capable of modeling the nonlinear load-deflection behavior of CFT frames requires a number of tasks. These tasks may be summarized by examining the three primary objectives of this research:

1) Formulate a polynomial expression representing the cross-section strength of a rectangular CFT. This expression should be applicable to a wide range of cross-section sizes and material strengths. Modeling CFT cross-section strength--the capacity of a zero-length CFT section under combined axial and bending loads--requires the development of accurate stress-strain relationships which account for the interaction between the concrete and the steel. Also, conditions must be identified, such as local buckling or concrete crushing, which cause the section to lose its capacity to sustain additional load. The representation of cross-section strength forms the basis for the plasticity model contained in the next objective.

2) Develop a macro finite element beam-column model to reproduce accurately the nonlinear load-deflection behavior of CFT specimens loaded monotonically and cyclically. The beam-column model consists of a concentrated plasticity formulation in which plasticity is confined to the element ends. The plasticity state at either end of the element is modeled using a two-surface, bounding surface model in three-dimensional force space $(P-M_y-M_z)$ ¹. An inner loading surface represents the locus of force points at

¹All symbols are defined in Appendix D.

which inelastic behavior initiates and an outer bounding surface represents the locus of force points that cause full plastification of an element end. The two surfaces translate kinematically and contract or expand isotropically to model load-deflection characteristics such as strength and stiffness degradation. A further objective of the concentrated plasticity model formulation entails calibrating the size of the surfaces, the rate of surface contraction or expansion, and the rate of plastic hardening, such that the set of fixed calibration parameters are applicable to a wide range of CFT cross-section sizes and material strengths.

3) Implement the concentrated plasticity model for CFT beam-columns along with existing formulations for steel beam-columns into a finite element program to analyze the behavior of complete composite moment-resisting frame structures.

1.3 Scope of Research

To achieve the main goals of this research, a general purpose frame analysis computer program was developed based on the use of a standard three-dimensional beam finite element, having a total of twelve degree-of-freedom per element, for both steel and CFT elements. The program performs a fully nonlinear analysis, accounting for both geometric and material nonlinearities. Element stiffness is formulated using a superposition of elastic, geometric, and plastic reduction stiffness matrices. Either single members or entire frame structures in three-dimensional space may be analyzed statically, with loads applied either monotonically or cyclically. In addition, the program has the capability to perform a transient dynamic analysis of single members and frames, although this research concentrates on cyclic static behavior.

As mentioned in Section 1.2, the CFT formulation pertains strictly to rectangular CFT members. The analysis is restricted to member behavior only; the behavior of CFT connections is not included in this work. In the analyses of frame structures, fully restrained connections are assumed. The scope of this research is limited to CFTs which are completely filled with concrete and make no use of reinforcing bars or shear connectors to improve the concrete/steel bond. Additionally, it is assumed that perfect bond between the steel and the concrete is maintained.

Local buckling of the steel tube is not modeled directly in this macro model formulation, although indirect account of local buckling is made in the computation of the cross-section strength of a CFT, and in the isotropic hardening model. Because CFTs rarely exhibit torsional or lateral-torsional buckling, these failure modes are not modeled. Also, since shear and torsional forces are expected to be small, material nonlinearity due to these forces is neglected.

The general CFT beam-column formulation presented in this work is a comprehensive and flexible macro model that is appropriate for modeling a wide range of rectangular CFT cross-sections and material strengths. However, the applicability of the CFT model is limited somewhat by the availability of experimental test results. The formulation for the cross-section strength applies to CFT sections with width/thickness ratios up to approximately 100, concrete compression strengths up to 15 ksi, and yield strengths typical of standard AISC rectangular tubes [AISC, 1994]. The formulation and results of the beam-column model, namely the calibrated plasticity parameters, are somewhat more limited in scope. Experimental monotonic load-deflection curves for CFT members with width/thickness ratios up to approximately 50 and concrete strengths up to 15 ksi were used in the calibration of the model. For the cyclic analysis, sections with a similar range of width/thickness values and concrete strengths of 3 to 5 ksi were analyzed and calibrated. Strictly speaking, the results of the analytical formulation are

meant to apply only for sections within this range. Extension of these preliminary results is pending further experimental research to ensure the accuracy of the analysis parameters.

1.4 Organization of Thesis

The analysis of the cross-section strength of CFTs constitutes the first part of the thesis. A discussion of CFT cross-section behavior introduces Chapter 2. This is followed by a discussion of the numerical procedure used to accurately predict CFT cross-section behavior. The chapter concludes with a discussion of the procedure used to determine a polynomial equation representing the three-dimensional cross-section strength of a general CFT. This cross-section strength equation is incorporated into the concentrated plasticity model for the material nonlinear behavior of CFTs, which is introduced in Chapter 3 and developed in detail in Chapter 4.

Chapter 3 examines the nonlinear CFT beam element formulation. The behavior of CFT beam-columns subjected to monotonic or cyclic loading is discussed first, followed by a general overview of the analytical model. Following this introductory section, each component of the stiffness formulation is discussed--the CFT elastic stiffness, the geometric stiffness, and finally, the plastic reduction stiffness, the final step of the concentrated plasticity formulation. The discussion of the plasticity formulation describes in general terms the method for modeling the observed CFT beam-column behavior, and introduces the plasticity model that is developed in detail in Chapter 4.

The concentrated plasticity model discussion of Chapter 4 is divided into three general sections. The first section discusses the incorporation of the CFT cross-section

strength equation developed in Chapter 2 into the bounding surface model. Next, the plastic reduction stiffness matrix for a single CFT element is developed. The last section of Chapter 4 details the mathematical formulation for the translation and the contraction and expansion of the loading and bounding surfaces.

Chapter 5 presents the calibration of the numerical model to experimental test results and presents a verification of the analytical CFT beam-column model. Example problems are first illustrated to verify the concentrated plasticity routine for steel elements. Following this verification, the model is calibrated to monotonically- and cyclically-loaded CFT members. Several additional CFT experiments are then used to illustrate the accuracy and general applicability of the calibrated model. The last section of the chapter illustrates a final verification problem consisting of a three-dimensional cruciform subassembly composed of steel I-beams framing into a CFT beam-column.

The final chapter summarizes the research on CFT beam-columns and the corresponding nonlinear model of their behavior. Conclusions are presented and possible future research topics are suggested.

Chapter 2

Analysis of CFT Cross-Section Strength

The concentrated plasticity model for the material nonlinear behavior of concrete-filled tube beam-columns requires an explicit equation describing the size and shape of the loading and bounding surfaces. To this end, this chapter details the development of a comprehensive empirical equation for determining the three-dimensional cross-section strength of CFT beam-columns.

The fiber element method, as developed by Sanz-Picon [1992] and El-Tawil et al. [1993], is used in this work to accurately compute the cross-section strength of CFTs. CFT cross-section strength is defined in this research as the combined axial load and bending moment capacity of a zero-length beam-column. The ultimate bending strength of the section for a given applied axial load is determined by executing a moment-curvature-thrust computation using the fiber element approach. This approach is detailed in Section 2.2. The peak moment from the moment-curvature-thrust analysis defines a point in three-dimensional force space (P - M_y - M_z) representing the capacity of the cross-section under a given set of loads. Thus, a complete three-dimensional cross-section

strength surface may be generated by a series of moment-curvature-thrust analyses over a range of axial loads and load eccentricities (i.e., varying degrees of biaxial bending).

The development of an equation for the three-dimensional cross-section strength of CFTs consists of four steps. First, the components of the fiber analysis specific to CFTs are determined: the stress-strain relationships of the steel and concrete, and the criteria for terminating the moment-curvature computation. Experimental tests of short CFT sections are then compared to the results from the fiber element analysis, and the stress-strain relationships and the analysis termination criteria are refined to produce accurate results. Once the results from the fiber element method agree with the experimental results, the third step entails generating three-dimensional cross-section strength surfaces for a series of CFTs with a wide range of dimensions and material strengths. Finally, using the P - M_y - M_z data from the series of fiber element analyses, an empirical polynomial equation is generated to represent the cross-section strength surface of a general CFT surface.

Chapter 2 is structured as follows. Section 2.1 discusses the behavior of CFT cross-sections, highlighting the behavioral aspects that will be modeled by the cross-section strength surface equation. Section 2.2 describes the fiber element analysis procedure used to generate moment-curvature-thrust diagrams. Section 2.3 contains a description of the steel and concrete constitutive relationships and Section 2.4 the conditions for terminating the moment-curvature-thrust computation. Section 2.5 illustrates the accuracy of the fiber element analysis results by comparing the results to experimental CFT test results. In the final portion of this chapter, Section 2.6, the procedure for developing an equation to empirically represent a general three-dimensional cross-section strength surface for rectangular CFTs is detailed.

2.1 CFT Cross-Section Behavior

This section introduces the general behavior of a CFT cross-section and serves a dual purpose. First, it provides a general discussion of CFT behavior which is continued in Chapter 3 for longer members. Second, this section lays the foundation for the remainder of the chapter by discussing the behavioral aspects that will be incorporated into the stress-strain formulations and termination criteria for the moment-curvature thrust computation.

2.1.1 Axially-Loaded Cross-Sections

The cross-section behavior of CFTs may be best understood by first examining a member subjected to a concentric axial load. In typical CFT applications, the axial load will be compressive. The case of pure tensile loading of a CFT is of less interest because the steel and the concrete do not interact. Instead, they will act independently of one another, and the capacity of the member may be accurately approximated by assuming that only the strength of the steel tube contributes to the load resistance [Furlong, 1967]. CFTs in compression, on the other hand, demonstrate unique behavior which is dependent upon a number of factors.

A concentric axial load applied evenly across a short CFT column will cause longitudinal deformation of the steel and the concrete. At low values of axial load, the steel and the concrete deform longitudinally at the same rate, but expand laterally at different rates. At these low levels of strain, Poisson's ratio for the steel exceeds Poisson's ratio for the concrete (0.28 for the steel versus 0.15 to 0.25 for the concrete), resulting in a greater lateral expansion of the steel, and little or no interaction between the two materials [Gardner and Jacobson, 1967; Tsuji et al., 1991]. During this stage of loading, the steel and the concrete sustain load independently of one another. At an axial

strain of approximately 0.001, microcracking in the concrete begins to occur and the lateral expansion rate of the concrete increases and begins to approach the lateral expansion rate of the steel. Between a strain of 0.001 and 0.002, the concrete expansion reinitiates full interactive contact between the two materials, inducing biaxial stresses in the steel and triaxial stresses in the concrete [Knowles and Park, 1970; Bode, 1976; Cai, 1987; Tsuji et al., 1991; Zhang et al., 1991].

The interaction between the concrete and the steel results in an overall increase in the load-carrying capacity of the member. The compressive strength and the ductility of the concrete core is enhanced by the confining pressure of the steel tube. On the other hand, the outward pressure of the concrete on the steel tube decreases the longitudinal capacity of the steel. If the steel tube has not yet yielded, this biaxial state of stress effectively decreases the amount of additional axial load the steel can sustain before yielding occurs. If, on the other hand, the steel tube has begun to yield when the biaxial stresses initiate, the steel will be unable to sustain the longitudinal yield stress. In either case, the tube must shed some of its axial load to the concrete [Gardner and Jacobson, 1967]. While the concrete's expansion has a deleterious effect on the longitudinal steel capacity, the increase in axial strength of the concrete actually outweighs the corresponding decrease in steel strength, resulting in an overall increase in the capacity of the CFT section [Council on Tall Buildings and Urban Habitat, 1979].

The behavior of CFTs after the onset of concrete confinement differs for circular and rectangular tubes. A circular steel tube can maintain lateral pressure on the expanding concrete by developing a circumferential tensile stress. The circumferential or hoop tension developed in the circular tube provides a larger resistance to the expanding concrete and a higher degree of confinement than the flat sides of a rectangular tube, which provide little perpendicular pressure to restrain the expanding concrete [Furlong, 1967; Knowles and Park, 1970]. Consequently, the concrete in rectangular tubes

undergoes little increase in strength due to confinement [Knowles and Park, 1970; Bridge, 1976]. The concrete in a rectangular tube will, however, experience an increase in ductility due to confinement [Tomii et al., 1973], although not to the degree of a circular tube. For both cross-section shapes, an increase in the thickness of the tube results in a corresponding increase in the degree of confinement provided by the steel because of the increased lateral stiffness of the tube.

2.1.2 Combined Axial Load and Bending

A cross-section subjected to a bending moment in addition to axial load exhibits behavior which depends primarily on the D/t ratio (the ratio of the depth or width of the tube to the thickness of the tube), and the strengths of the steel and the concrete. Additionally, local buckling of the steel tube may impact the capacity of the cross-section. Cross-section behavior may be described by an interaction curve which illustrates a section's moment capacity for a given axial load. Figure 2.1 illustrates typical CFT interaction curves for different values of concrete strength, f'_c (this interaction curve is shown in normalized force space; P_o and M_o are, respectively, the ultimate axial load in the presence of no bending and the ultimate bending moment in the presence of no axial load). The figure illustrates that CFT cross-sections initially sustain increasing amounts of moment as the axial load increases from zero. The degree of this characteristic increase in moment, which is manifested in the convex bulge of the interaction curve, depends primarily upon the D/t ratio and the concrete and steel strengths. As axial load is increased further beyond the bulging portion of the curve, CFTs begin a rapid decrease in moment capacity as the axial load approaches its maximum value, P_o .

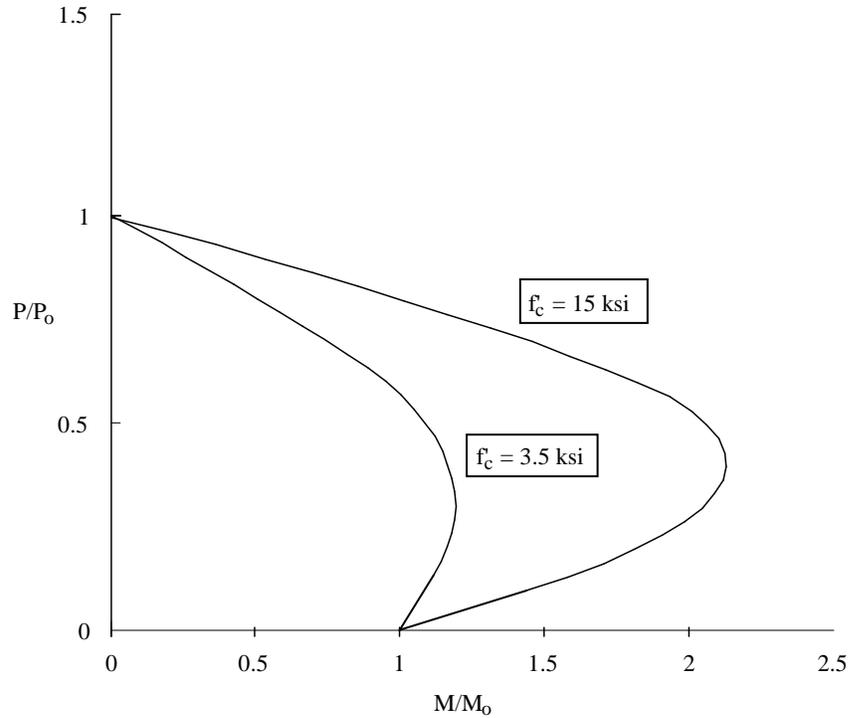


Figure 2.1 Cross-Section Strength Curves for Typical CFT Cross-Sections

The moment capacity of a CFT derives primarily from the resistance of the steel tube and the resistance of the compression concrete. The amount of concrete in compression depends on the location of the neutral axis of the cross-section. For CFTs with a higher concrete strength or larger ratio of concrete to steel area, the neutral axis will lie closer to the top fiber of the section. When increasing amounts of axial load are applied to a given cross-section, the neutral axis moves toward the centroid of the cross-section, increasing the contribution of the compression concrete. Therefore, a section with a larger f'_c or a larger D/t ratio will experience a larger increase in moment capacity (relative to its value of M_o) as the axial load increases from zero, and will thus exhibit a larger bulge in the interaction diagram in the mid-range region of axial load (compare $f'_c = 3.5$ and $f'_c = 15.0$ in Fig. 2.1). The presence of axial load will also enhance the confining effect on the concrete. This illustrates the beneficial behavior of CFT beam-

columns subjected to moderate values of axial load and bending, and demonstrates the advantage of combining high strength concrete with steel tubes that may be quite thin.

While CFT behavior is most advantageous for moderate axial loads, CFTs under high axial load exhibit rapid moment capacity deterioration and brittle failures. For high values of P/P_o , the tensile resistance of the steel tube is underutilized and may, for CFTs with low D/t and low f'_c , still be elastic upon crushing of the compression concrete. In addition, if the steel on the compression side has buckled upon concrete crushing (which is more likely as the D/t ratio increases), an undesirable brittle failure ensues. Because of these characteristics of CFTs subjected to large axial load, most researchers limit their studies of beam-columns to axial loads giving a ratio of P/P_o less than or equal to approximately 0.5 [Tomii and Sakino, 1979a, 1979b; Matsui and Tsuda, 1987].

The capacity of a CFT cross-section must also be examined with respect to the local buckling of the steel tube. This deleterious effect is somewhat mitigated by the presence of the concrete core in a CFT. Matsui [1986] established experimentally that the D/t ratio at which a CFT buckles locally may be increased to 1.5 times that of hollow tubes. Correspondingly, a decrease in the D/t ratio will increase the amount of strain the section can undergo before it buckles locally. In addition, Tomii and Sakino [1979a, 1979b] and Ichinohe et al. [1991] have shown that CFTs with low D/t ratios can maintain their strength long after local buckling has occurred.

2.1.3 Other Cross-Section Effects

Residual Stresses

In addition to stresses due to the applied loads, residual stresses may also be present in the steel tubes. The level of residual stresses in steel tubes is highly dependent

upon the manufacturing process and the shape of the cross section [Sherman, 1992]. The tubes considered in this work are cold-formed, welded seam tubes. These types of tubes are manufactured in sheets, bent into shape, and then welded at the seam. The residual stresses produced by this process are largest in the through-thickness direction [Sherman, 1992] and are assumed in this work to have a negligible effect in the longitudinal and circumferential direction. In addition, to avoid the problem of residual stresses altogether, some researchers have used annealed steel tubes, (i.e., tubes which are heated to a high temperature and then cooled slowly and uniformly to relieve any residual stresses) [Tomii and Sakino, 1979a, 1979b, 1979c; Sakino and Tomii, 1981]. Unless otherwise noted, all of the experimental results cited in this research are tubes which have not been annealed.

Bond

Composite interaction in a CFT member depends in part on the amount of bond between the concrete and the steel. Viridi and Dowling [1980] have identified two types of bond between the concrete and steel in a CFT--microlocking and macrolocking. Microlocking, the primary type of bond, refers to the bonding of the concrete with the surface irregularities (or roughness) of the inside of the tube. Macrolocking refers to the mechanical and frictional interaction between the concrete and steel due to nonuniformities in the tube, e.g., out-of-straightness or out-of-roundness.

The ultimate bond strength between the materials is attained upon local crushing of the concrete at the interface. Viridi and Dowling [1980] studied the effect of a number of parameters and established a characteristic bond strength for CFTs of 150-160 psi. They concluded that surface preparation and the amount of compaction are the only significant parameters that will increase the amount of bond. Parameters such as concrete

strength, length of the concrete/steel interface, the tube thickness, and the tube diameter had only negligible effects on the amount of bond.

Nevertheless, it has yet to be determined comprehensively how much strength, if any, is lost in a CFT member that is part of a frame once the bond between material breaks down [Morino et al., 1993]. It is also difficult to model slip at the interface of the tube and the concrete core in a macro model. Thus, it is assumed in this research that perfect bond is maintained between the steel and the concrete (i.e., that no slip occurs). Most analytical studies of CFTs to date have incorporated this assumption [Neogi et al., 1969; Tomii et al., 1973; Bridge, 1976; Tomii and Sakino, 1979b; Prion and Boehme, 1989; Shakir-Khalil and Zeghiche, 1989; Masuo et al., 1991; Kawaguchi et al., 1991].

Creep and Shrinkage

The effect of creep and shrinkage, an essential consideration in reinforced concrete construction, has a much smaller influence on the behavior of CFTs [Nakai et al., 1991]. The steel tube serves as an enclosed environment in which conditions remain ideally humid, minimizing these effects. Long-term creep and shrinkage tests by Nakai et al. [1991] compared plain concrete specimens to CFTs. Their results indicate that the amount of shrinkage due to drying is negligible compared to the plain concrete. Creep does produce an increase in the longitudinal strains over time in CFTs, but the ratio of the final strain after creep to the initial elastic strain due to the axial load in their CFT specimens was about half of the value obtained for plain concrete. Creep and shrinkage are neglected in this work.

2.2 Fiber Element Method for Computing CFT Cross-Section Strength

2.2.1 Background on Methods for Cross-Section Strength Computation

A number of different methods have been used to analyze the cross-section strength of CFTs. Tomii and Sakino [1979b] computed CFT cross-section strength using a stress block method. In this approach, the stresses in the concrete and steel are represented by uniform rectangular blocks, and the resultant forces over the cross-section are assumed to act at the centroids of the stress blocks. Chen and Atsuta [1976, 1977] derived parametric equations to describe the moment-curvature-thrust relationship of CFT sections. These equations, derived using empirical expressions for the stress-strain relationships of the steel and concrete, relate moment to curvature based on the amount of applied axial load. Numerical integration techniques have also been used by several authors to determine the moment-curvature-thrust relationship of CFT sections. This particular method analyzes a given cross-section by determining the stress and strain at discrete points and then obtaining resultant internal forces by integrating the stresses. The method used in this work, a fiber element analysis, is one such approach. Variations of this method have been used by Bode [1976], Shakir-Khalil and Zeghiche [1989], and Kawaguchi et al. [1993] for uniaxial bending of rectangular CFTs and by Sanz-Picon [1992] and El-Tawil et al. [1993] for biaxial bending of SRC sections.

2.2.2 Fiber Element Analysis

The fiber element method affords a high degree of accuracy as a numerical tool to analyze the moment-curvature-thrust behavior of CFT beam-column cross-sections. Values of stress and strain are monitored at a number of different points on the

discretized cross-section throughout the analysis. The resultant forces on a cross-section (axial force and bending moment) are obtained by numerically integrating the stresses over all of the elements of the cross-section. The accuracy of the solution improves as the cross-section is discretized into a finer and finer mesh of elements.

The moment-curvature-thrust relationship for a given cross-section is generated by performing a series of iterative procedures (detailed in Section 2.2.4). By computing a number of moment-curvature-thrust analyses at specific values of applied axial load ratio, $p = P/P_o$, and angle of eccentricity (i.e., the orientation of the applied moment with respect to the principal axes of the cross-section), a two- or three-dimensional cross-section strength surface may be generated. For a given eccentricity angle of the applied moment, a two-dimensional cross-section strength curve (M-P diagram) is generated by obtaining the maximum moment from separate moment-curvature-thrust analyses performed at increments of axial load ranging from zero to P_o . A piecewise three-dimensional cross-section strength surface (P- M_y - M_z diagram) may then be generated by calculating a series of two-dimensional cross-section strength surfaces with the moment applied at increments of eccentricity angle ranging from 0° (major axis bending) to 90° (minor axis bending).

2.2.3 Discretization of the Cross-Section

The fiber element analysis first requires a specification of the CFT cross-section dimensions and the number of mesh elements. The geometry of a rectangular CFT is defined by three parameters: the width (b), depth (D), and thickness of the tube (t). Figure 2.2 illustrates these dimensions and the meshing scheme incorporated to discretize the rectangular CFT cross-sections analyzed in this research. Each cross-section was divided into five regions, facilitating a nearly uniform element size and an element aspect

ratio as close to 1:1 as possible. This scheme also allows different degrees of mesh fineness to be used for the steel and concrete.

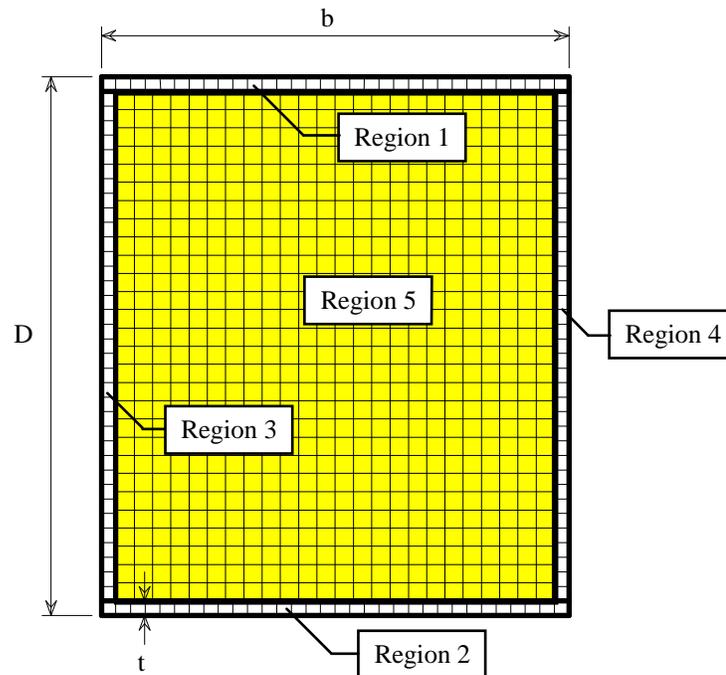


Figure 2.2 CFT Geometry and Typical Meshing Scheme

The density of the mesh for the cross-sections studied in this research was carefully monitored to determine the optimum density with respect to accuracy and analysis time. The mesh densities used were sufficiently refined so that increasing the mesh density did not result in any significant changes in the results. The density of the mesh shown in Fig. 2.2 is somewhat coarser than the density used in the cross-sections of this study. In general, the steel tube was meshed in a somewhat finer grid than the concrete. One layer through the thickness of the steel tube provided sufficiently accurate results for most cross-sections except for thicker steel tubes having a depth to thickness ratio of 24. For these cross-sections, two layers through the thickness were used.

Maintaining a 1:1 aspect ratio resulted in approximately 40 to 60 elements per steel region. The concrete interior (region 5) was also meshed using a 1:1 aspect ratio. For square CFTs, for example, this mesh density results in a grid of concrete elements ranging in number from approximately 36×36 elements to 50×50 elements.

2.2.4 Moment-Curvature-Thrust Analysis

The moment-curvature-thrust analysis used in this research to analyze the cross-section behavior of CFTs was developed by Sanz-Picon [1992] and El-Tawil et al. [1993] for the analysis of steel, steel-reinforced concrete, and reinforced concrete cross-sections. Their fiber element procedure was implemented in an interactive, graphical computer program, COSBIAN [Sanz-Picon, 1992; El-Tawil et al., 1993], which was extended in this work to analyze concrete-filled steel tube cross-sections. The analytical procedure is described in detail in their work; this section only briefly highlights the assumptions and the salient points of the fiber analysis. This section also provides a basis for the descriptions of the CFT stress-strain relationships (Section 2.3) and the criteria for terminating the moment-curvature-thrust computation of CFTs (Section 2.4).

The fiber element procedure may be described by examining one increment of a moment-curvature-thrust analysis. For such an analysis, the moment must be computed for a given value of curvature, ϕ . The procedure for calculating the cross-section moment contains a series of iterative steps. It begins with the initialization of the neutral axis orientation and the top fiber strain, ϵ_{top} , which, combined with the given value of curvature, define the strain distribution over the cross-section. The orientation of the neutral axis (the angle θ between the neutral axis and the centroidal axis of the section) is initialized to the eccentricity of the axial load, α (see Fig. 2.3). The top fiber strain is initialized to its specified increment. From the curvature and the top fiber strain values,

the location of the neutral axis may be calculated. The strain at each element is then computed assuming the strain distribution is linear. This assumption implies that perfect bond exists between the concrete and steel at the material interface (see Section 2.1.3). The strain at any fiber is then computed by multiplying the curvature by the distance from the centroid of the element to the neutral axis, measured with respect to a line perpendicular to the neutral axis (Fig. 2.3).

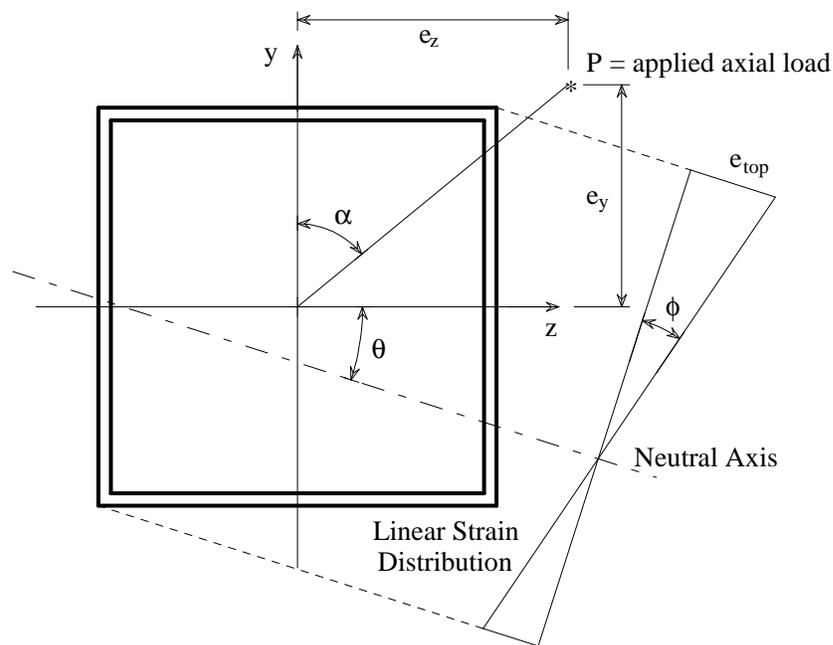


Figure 2.3 Fiber Element Analysis Definitions (after Sanz-Picon [1992])

Once the strain at each fiber element has been determined, stresses are computed from the strains using the constitutive relationships described in Section 2.3. The force on each element is the product of the element stress and area. The resultant internal axial force on the cross-section is computed by numerically integrating the stresses over the cross-section, which is accomplished by summing the force in each fiber element:

$$F = \sum_{i=1}^k \sigma_i \cdot A_i \quad (2.1)$$

where k is the number of fiber elements in the cross-section, A_i is the area of each fiber element, and σ_i is the stress at the centroid of each element. This value of force, F , is then compared to the applied axial load, P . If the values are not within a prescribed tolerance, the top fiber strain is adjusted and the process is repeated until convergence.

For biaxial loading, the correct orientation of the neutral axis must also be established. Once the axial force is equilibrated, moments in the major (z) and minor (y) axis directions are calculated by summing the moment produced by the force in each element (the product of the element's force and its distance to the neutral axis):

$$M_z = P \cdot e_y = \sum_{i=1}^k \sigma_i \cdot A_i \cdot y_i \quad (2.2)$$

$$M_y = P \cdot e_z = \sum_{i=1}^k \sigma_i \cdot A_i \cdot z_i \quad (2.3)$$

The ratio of the calculated moments M_y/M_z is then compared to the tangent of the angle of load eccentricity, α , which is defined as:

$$\tan \alpha = \frac{e_z}{e_y} \quad (2.4)$$

where e_z and e_y are the eccentricities of the applied load as measured from the y and z axis, respectively. If the ratio of the calculated moments and the ratio of the load eccentricities are not within a specified tolerance, the orientation of the neutral axis is adjusted until the two ratios converge. When the neutral axis orientation changes, the strain distribution changes. New strain values are calculated based on the new neutral

axis orientation and the procedure for axial force convergence is repeated. The moment ratio, M_y/M_z , is again compared to Eq. (2.4) and the neutral axis is adjusted if the two ratios are not within a tolerance. These two iterative processes are repeated until both the axial forces converge and the moments converge. Once this has occurred, the resultant moment is calculated by:

$$M = \sqrt{M_z^2 + M_y^2} \quad (2.5)$$

This value defines one point of the moment-curvature-thrust relationship. For a complete moment-curvature-thrust analysis, the curvature is then increased by a specified increment and the iterative process is repeated. The moment-curvature-thrust computation terminates when one of the failure criteria described in Section 2.4 is met.

2.2.5 Analysis Parameters

The fiber element analysis discussed in the previous section requires specified increments and convergence tolerances for the iterative procedures contained within it. Following are the values for these parameters that were used in this research:

Curvature increment ($\Delta\phi$)	= 0.00001
Strain increment ($\Delta\epsilon$)	= 0.0001
Increment in neutral axis orientation ($\Delta\theta$)	= 2.0°
Axial force convergence tolerance	= 0.10 %
M_y/M_z convergence tolerance	= 1.0 %

These values are all based on the recommendations presented by Sanz-Picon [1992].

2.3 Steel and Concrete Constitutive Models

The calculation of the concrete and steel element stresses from the element strains forms a key step in the fiber element approach since the resultant forces on the cross-section for a given curvature are computed directly from the stresses. The accuracy of the moment-curvature-thrust solution, therefore, depends upon a correct formulation of the concrete and steel constitutive relationships.

In this research, empirical uniaxial nonlinear constitutive models were formulated to represent the multiaxial stress-strain behavior of the concrete core and the steel tube. As discussed in Section 2.1, the multiaxial stress-strain behavior of the steel and concrete in a CFT results from the interaction between the concrete and the steel. This interaction often results in additional strength and ductility in the concrete and a decrease in the longitudinal strength of the steel tube (effects which are more pronounced in circular tubes). For the rectangular tubes studied in this research, the effect of confinement is assumed to produce only an increase in the ductility of the concrete and no increase in strength [Tomii and Sakino, 1979b]. Similarly, it is assumed that the concrete/steel interaction in rectangular CFTs produces no decrease in the yield strength of the steel tube.

Because only longitudinal strains are monitored in the fiber element analysis, multi-axial behavior must be accounted for indirectly using uniaxial stress-strain curves. Relatively few authors have addressed the effect of interactive stresses between the steel and the concrete in modeling the behavior of CFTs. Therefore, much of the following discussion and parts of the proposed stress-strain expressions originate from experimental and analytical work performed on confined rectangular reinforced concrete sections.

2.3.1 Concrete Constitutive Model

The stress-strain behavior of the concrete in a CFT is dependent upon its unconfined uniaxial behavior and the relative ratio of concrete to steel (commonly expressed as the depth or width to thickness ratio, or D/t ratio). The D/t ratio affects the degree of concrete confinement provided by the steel, and therefore, the amount of ductility exhibited by the concrete.

The uniaxial stress-strain relationship for concrete in a rectangular CFT consists of four distinct regions: a tensile region, an ascending curve in the compression region, a plateau region, and a descending curve in the compression region [Tomii and Sakino, 1979b]. Figure 2.4 illustrates the basic form used in this work of several typical stress-strain curves for various combinations of concrete strength and D/t ratio. The tensile

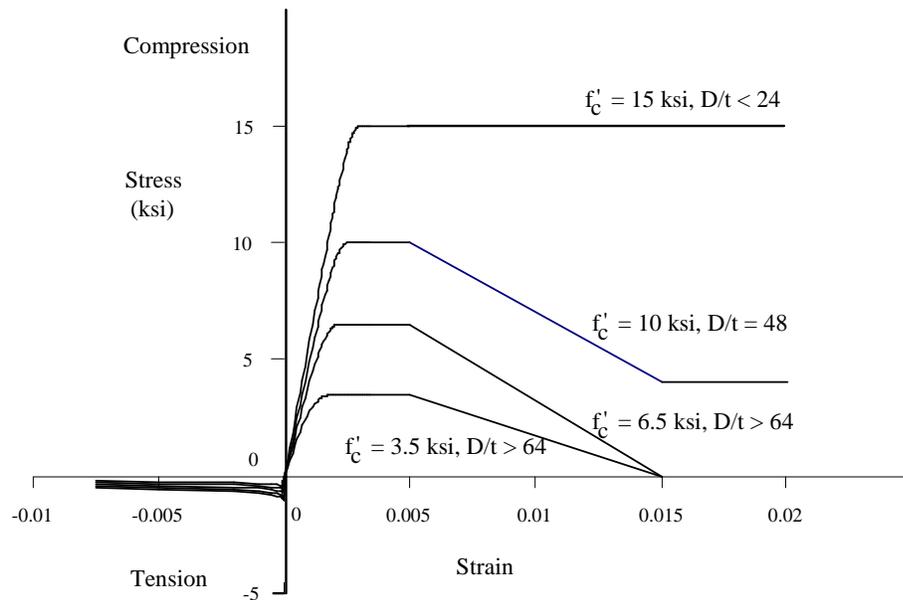


Figure 2.4 Uniaxial Stress-Strain Curve for Concrete in Rectangular CFTs

region consists of a linear curve up to the rupture strength of the concrete, and then a curve that asymptotically approaches zero stress. The first portion of the compression region is an ascending curve from zero to the maximum concrete strength, f'_c . Upon

reaching the maximum strength (typically at a strain of 0.002 to 0.003), the stress remains constant until a strain of 0.005. Beyond this plateau region, the concrete curve descends at a slope based upon the D/t ratio of the CFT. All of the regions depend on the concrete strength, but only the descending portion of the compression curve is a function of the D/t ratio.

The ascending branch formulation used in this work for rectangular CFTs was originally proposed by Popovics [1973] for unconfined concrete and is shown here as modified by Collins et al. [1993]:

$$f_c = f'_c \cdot \frac{\epsilon_c}{\epsilon_o} \cdot \frac{n}{n-1 + (\epsilon_c/\epsilon_o)^n} \quad (2.6)$$

where:

$$n = 0.8 + \frac{f'_c}{2500} \quad (f'_c \text{ in psi units}), \text{ and} \quad (2.7)$$

$$\epsilon_o = \frac{f'_c}{E_c} \frac{n}{n-1}. \quad (2.8)$$

The strain at f'_c , ϵ_o , ranges from 0.0019 for a concrete strength of 3.5 ksi, to 0.0030 for a concrete strength of 15 ksi. The modulus of elasticity of concrete in this region is given by [Collins et al., 1993]:

$$E_c = 40000\sqrt{f'_c} + 1000000 \quad (\text{psi units}) \quad (2.9)$$

Equations (2.6 - 2.9) are appropriate for a wide range of concrete strengths. Collins et al. [1993] confirmed the accuracy of these equations for concrete strengths up to 15 ksi.

Although the expression developed by Popovics was originally proposed for unconfined concrete, it has been used to model the behavior of confined concrete in reinforced concrete sections [Cusson and Paultre, 1993]. Cusson and Paultre showed that the expression produced very accurate results for reinforced concrete sections confined by rectangular ties.

The portion of the rectangular concrete stress-strain expression beyond the strain at the maximum concrete strength is based on the curve proposed by Tomii and Sakino [1979b] for CFTs. Beginning at ϵ_o (the strain at f'_c), the stress remains constant until a strain of 0.005 is reached. This plateau region represents the added ductility provided by the confining action of the steel tube.

The descending branch of the concrete curve extends from a strain of 0.005 to 0.015 at a slope which is a function of the D/t ratio. The larger the D/t ratio, the steeper the slope. This reflects the loss of concrete ductility with a decrease in the amount of confinement provided by the steel. This linear segment descends from a stress of f'_c to f_{cl} , where f_{cl} is given by the following formula:

$$f_{cl} = \left(1.6 - 0.025 \cdot \frac{D}{t} \right) \cdot f'_c. \quad (2.10)$$

Equation (2.10) was developed by Tomii and Sakino [1979b] using experimental data from rectangular CFTs with D/t ratios between 24 and 44. To encompass a comprehensive range of D/t ratios, the Tomii and Sakino model for strains beyond 0.005 was extended as follows. For a D/t ratio of 24, Eq. (2.10) equals f'_c . It is assumed in this work that, regardless of the D/t ratio, the concrete does not exceed its cylinder strength (f'_c). Therefore, sections with a D/t ratio less than 24 maintain a constant concrete stress of f'_c as the strain increases from 0.005 to 0.015 or beyond. Sections with D/t ratios

ranging between 24 and 64 have a non-zero value of f_{ct} given by Eq. (2.10). At a D/t ratio of 64, f_{ct} equals zero. All sections with D/t ratios larger than this follow the same curve as the section with a D/t ratio of 64, dropping from f'_c at a strain of 0.005 to zero stress at a strain of 0.015. For strain values beyond 0.015, the stress in all sections is assumed to remain at a constant value (f_{ct}) for any subsequent value of strain [Tomii and Sakino, 1979b]. Refer again to Fig. 2.4 for clarification.

For rectangular tubes (as differentiated from square tubes), the D/t of the section is different in the major and minor axis directions of the CFT². The D/t ratio used in the Eq. (2.10) for rectangular CFTs is the average of the major axis D/t ratio and the minor axis D/t ratio.

The formulation for the concrete tensile response used in this work for rectangular CFTs is adopted from the formula proposed by Vecchio and Collins [1986], which was also used by Sanz-Picon [1992] for the analysis of steel-reinforced concrete cross-sections. The concrete is assumed to follow a linear elastic curve given by the equation:

$$f_{ct} = \frac{f'_c}{0.001} \epsilon_{ct} \quad (2.11)$$

until it reaches the rupture strength of the concrete. The rupture strength is given by:

$$f_r = 7.5\sqrt{f'_c} \quad (2.12)$$

Once the concrete reaches this stress, its strength decreases according to the following equation:

²The dimension D in the D/t ratio for rectangular (as differentiated from square) tubes refers to the dimension in the plane parallel to the plane of bending. The nomenclature for a 24×12×1/2 tube, for example, is as follows. The major, or strong, axis D/t ratio is 48 (the depth of the section, 24, divided by 1/2) and the minor, or weak, axis D/t ratio is 24 (the width, 12, divided by 1/2).

$$f_{ct} = \frac{f_r}{1 + \sqrt{200\epsilon_{ct}}}. \quad (2.13)$$

The subscript ct in Eqs. (2.11) and (2.13) denotes concrete tensile stress and strain. All values of strength in Eqs. (2.11 - 2.13) are in psi.

2.3.2 Steel Constitutive Model

In the stress-strain formulation for CFTs contained in this work, confinement of the concrete in a rectangular CFT is assumed to increase only the ductility of the concrete (Section 2.3.1). Because the concrete strength is not enhanced, the stress-strain expression for steel in a rectangular CFT correspondingly assumes that any biaxial stresses in a rectangular tube are relatively small for most ranges of loading. Therefore, the steel stress-strain expression is represented by a curve similar to that used for tube steel. The tension and compression regions of the curve are modeled by the same curve, with one exception. To account for the interaction due to the small degree of confinement for rectangular CFTs in compression, strain hardening is not modeled in the compression region [Tomii and Sakino, 1979b]. Any interaction that may occur between the steel and concrete in the tension region is neglected because the concrete in tension offers little resistance. Therefore, steel in the tension region acts independently of the concrete, and is modeled as if it were a hollow tube, including strain hardening.

The steel stress-strain curve is illustrated in Fig. 2.5. In the compression region, the steel exhibits linear elastic behavior up to the yield point of the material (f_y) and then

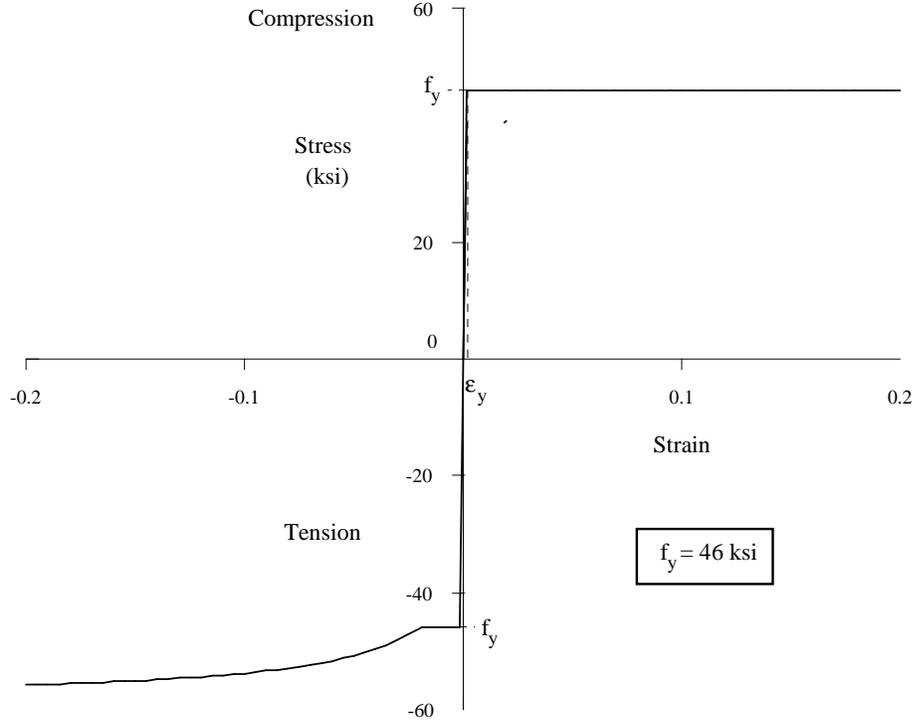


Figure 2.5 Uniaxial Stress-Strain Curve for Steel in Rectangular CFTs

follows a perfectly-plastic plateau, remaining at the yield stress, f_y , for subsequent values of strain. In the tension region, the stress-strain behavior mirrors the linear elastic, perfectly-plastic behavior of the compression region until the onset of strain hardening. In this research the value of strain at which strain hardening begins, ϵ_{sh} , was set to 0.0186 [Sanz-Picon, 1992]. The stress at strain values beyond ϵ_{sh} increases according to the following equation [Sanz-Picon, 1992]:

$$f_s = \frac{E_{sh}(\epsilon_s - \epsilon_{sh})}{\sqrt[N]{1 + \left[\frac{E_{sh}(\epsilon_s - \epsilon_{sh})}{f_u - f_y} \right]^N}} + f_y \quad (\text{psi}) \quad (2.14)$$

The steel stress in this equation asymptotically converges to f_u . This so-called *power equation*, first developed by Richard and Abbott [1975], uses the parameter N to define the shape of the curve. For the analyses performed here, N is set to a value of 2 [Sanz-Picon, 1992]. As described in Section 2.1.3, residual stresses are assumed to be zero in the cross-section strength analyses in this work.

2.4 Criteria for Terminating the Moment-Curvature-Thrust Computation

The generation of a moment-curvature-thrust diagram requires a set of criteria for determining when the computation should terminate, thus identifying the conditions at which the section can no longer carry additional load. Rectangular CFT cross-sections in which the concrete remains essentially unconfined and plays a large role in the strength of the member, such as CFTs with a large ratio of concrete to steel (i.e., a high D/t ratio) and subjected to a high axial load, the stress-strain expressions produce a moment-curvature-thrust diagram with a distinct peak moment. The descending branch of the concrete stress-strain curve for these sections will cause a similar descending slope in the moment-curvature-thrust diagram. Termination of the moment-curvature-thrust computation in this case occurs when the moment decreases to a certain percentage of the maximum moment attained. However, in the case of rectangular tubes with low D/t ratios and small values of axial load ratio, the steel maintains its characteristic ductility and the concrete remains ductile as well due to confinement. In such a case, the stress-strain curves of the steel and concrete suggest that the section can withstand constant or increasing moment to very large strains. This large capacity, however, will not be reached because of other failure mechanisms such as local buckling or concrete

crushing. Therefore, alternate means of terminating the moment-curvature-thrust diagram incorporating these additional failure criteria must be employed.

2.4.1 Discussion of the Termination Criteria

Four separate criteria were established to determine the termination point of the moment-curvature-thrust diagrams for CFTs. The criteria include the following: 1) the moment-curvature-thrust diagram drops below 95% of its peak moment value; 2) a specified percentage of steel yields and a specified percentage of the concrete in compression crushes; 3) a specified percentage of the steel in compression buckles and a specified percentage of the concrete in compression crushes; and 4) any portion of the steel reaches a tensile strain equal to a specified rupture strain of the steel. The use of several criteria permits an accurate prediction of different failure mechanisms during a three-dimensional cross-section strength analysis in which strengths are determined for a wide variety of loading, ranging from pure bending to pure axial load.

The first criterion applies primarily to CFT sections with medium to high D/t ratios subjected to a combination of axial load and bending. Once the peak moment is reached, the moment will decrease with any further increase in curvature due to the descending branch of the concrete stress-strain curve. The moment-curvature-thrust computation is terminated when the decreasing moment drops to 95% of the maximum moment.

The second criterion terminates the analysis when a certain percentage of the steel yields (typically close to 100%) and a certain percentage of the concrete in compression crushes. Crushing of the concrete combined with steel yielding (or with steel buckling, as in criterion three) is the typical method of failure for most CFT cross-sections, especially for sections subjected to high axial loads [Gardner and Jacobson, 1967; Chen and Chen,

1973; Shakir-Khalil and Zeghiche, 1989]. The strain at which crushing of the concrete occurs has been established as 0.005 in this work. This value was identified by Tomii and Sakino [1979b], who indicated in their studies that the concrete stress-strain curve consistently begins to descend at this strain.

Failure of a CFT may also occur by concrete crushing combined with a specified amount of local buckling or bulging of the steel [Kitada et al., 1987; Shakir-Khalil and Mouli, 1990]. After local buckling occurs, the concrete resists some of the compressive load shed by the steel [Matsui, 1986]. As in criterion two, the strain at which the compression concrete crushes is assumed to be 0.005. The same percentage of crushed concrete specified for criterion two is used here as well. The strain at which local buckling of the steel tube is assumed to occur is computed based on the equation specified in the AISC LRFD Specification [1994]. This equation limits the D/t ratio of steel tubes to prevent local buckling before the steel yields [SSRC, 1979; AISC LRFD, 1994]. For a rectangular CFT, this limit is:

$$\frac{D}{t} \leq \sqrt{\frac{3 \cdot E_s}{f_y}} \quad (2.15)$$

Solving for the yield strain, $\epsilon_y = f_y / E_s$, and setting this strain equal to the local buckling strain, ϵ_{lb} , results in the following expression:

$$\epsilon_{lb} = \frac{3}{(D/t)^2} \quad (2.16)$$

Experiments have shown that CFT sections often provide additional capacity after the initiation of local buckling [Tomii and Sakino, 1979ab]. Also, the presence of concrete in the CFT delays the onset of local buckling [Matsui, 1986; Matsui and Tsuda, 1987; Tsuji

et al., 1991], rendering Eq. (2.16) conservative, since it was developed for hollow tubes. Therefore, rather than terminating the computation immediately at the onset of local buckling, a specified percentage of compression steel must buckle and a certain percentage of the concrete must crush before the computation is terminated.

The fourth and final criterion limits the tensile strain in any given steel element to a value of 0.2. At this strain the steel will rupture, causing a catastrophic failure of the section. This criterion may apply to CFTs having very thin tubes (high D/t ratio) and subjected to little or no axial load (i.e., pure bending conditions [Prion and Boehme, 1989]).

2.4.2 Values Used for the Termination Criteria

The values of the criteria for terminating the moment-curvature-thrust computation remained constant throughout this research. The values of these parameters are as follows:

Percent of yielded steel = 98%	(criterion 2)
Percent of crushed compression concrete = 50%	(criteria 2 & 3)
Percent of locally buckled compression steel = 25%	(criterion 3)
Maximum steel fiber strain (ϵ_{\max}) = 0.2	(criterion 4)

The majority of the sections in this research were governed by either the first criterion (moment dropping to 95% of the maximum) or the third criterion (local buckling of the compression steel and crushing of the compression concrete). The former criterion typically governed for sections with high D/t ratios combined with high axial load ratios. The third criterion governed for most D/t ratios at lower axial load ratios. Generally, the

percentage of crushed concrete dictated the stopping point rather than the percentage of local buckling, because the crushing strain is larger than the buckling strain and, as specified, 50% of the compression concrete must crush compared to 25% of the compression steel buckling.

Experimental tests indicate that large curvatures may be reached beyond the point of initial local buckling [Tomii and Sakino, 1979a]. Also, the local buckling of the steel tube in a CFT will be delayed due to the concrete forcing all buckling modes outward [Furlong, 1967; Tsuji et al., 1991]. However, the value of 25% was used (rather than a higher value) since at present [1994] there is little experimental data justifying a higher value.

Unlike reinforced concrete sections, the initiation of concrete crushing will not cause an immediate loss of member capacity because the concrete in a CFT is confined. Therefore, considerably more concrete may crush before failure. The value of 50% was selected based on comparison of the results to experimental data.

The percent of steel yielding is the value recommended by Sanz-Picon [1992]. The value of the rupture strain used in criterion 4 is a typical value specified in the steel literature [Salmon and Johnson, 1990].

2.5 Comparison to Experimental Results

The accuracy of the constitutive relationships and the criteria for terminating the computation of the moment-curvature-thrust relationship was verified by using the fiber element method to analyze cross-sections tested experimentally by other researchers. The constitutive relationships and termination criteria were refined and adjusted as necessary to produce the best correlation to the experimental data. Once the available experimental

results are accurately reproduced for a select number of sections, a wide range of sizes and strengths of CFT sections may be modeled accurately.

Table A.1 in Appendix A lists rectangular CFT cross-section tests presented in sufficient detail to allow calibration of the analytical results; Table 2.1 illustrates the salient data for each of these tests. Most of the calibration for rectangular CFTs was performed using the tests by Tomii and Sakino [1979a, 1979b]. These authors provide the most complete results, documenting both moment-curvature-thrust and two-dimensional cross-section strength relationships. Tests by Furlong [1967] were also used. As of 1994, there are almost no other well documented experimental results of short rectangular CFTs subjected to combined axial force and flexure.

Table 2.1 Rectangular CFT Cross-Section Tests

Test	Figure Notation	Tube Dimensions (in.)	L/D	D/t	f'_c (ksi)	f_y (ksi)	Other Data
Furlong, 1967	Furl1	$5.0 \times 5.0 \times 0.19$	7.2	26.5	6.5	70.3	--
Furlong, 1967	Furl2	$4.0 \times 4.0 \times 0.084$	9.0	47.6	3.4	48.0	--
Furlong, 1967	Furl3	$4.0 \times 4.0 \times 0.125$	9.0	32.0	4.2	48.0	--
Tomii and Sakino, 1979a	Tom44a	$3.9 \times 3.9 \times 0.089$	3.0	44.0	6.6	28.0	annealed tube
Tomii and Sakino, 1979a	Tom44b	$3.9 \times 3.9 \times 0.089$	3.0	44.0	3.8	49.2	annealed tube
Tomii and Sakino, 1979a	Tom33	$3.9 \times 3.9 \times 0.118$	3.0	33.0	3.6	42.0	annealed tube
Tomii and Sakino, 1979a	Tom24	$3.9 \times 3.9 \times 0.162$	3.0	24.0	3.2	42.0	annealed tube

Figures 2.6 to 2.16 compare the moment-curvature-thrust results of the fiber element analysis to the experimental results of Tomii and Sakino [1979a, 1979b]. Each figure is referenced by the notation in column 2 of Table 2.1. Tomii and Sakino performed four series of tests, with D/t ranging from 24 to 44 and f'_c ranging from 3.2 to 6.6 ksi. The steel tubes in these tests were annealed to remove residual stresses. The fiber analysis results are quite accurate for moment-curvature-thrust curves with a low- to mid-range P/P_o and slightly less accurate for higher P/P_o (Figs. 2.8, 2.11, and 2.13). The latter tests show a maximum error of approximately 15.0 % and a maximum error in the peak moment of 3.0 %. Figures 2.17 to 2.20 illustrate the close correlation of the fiber element analysis results with the results of Tomii and Sakino's experiments; all errors are less than 10%. Both the fiber analysis and the experimental results are normalized by the ultimate axial and bending loads calculated in the fiber analysis.

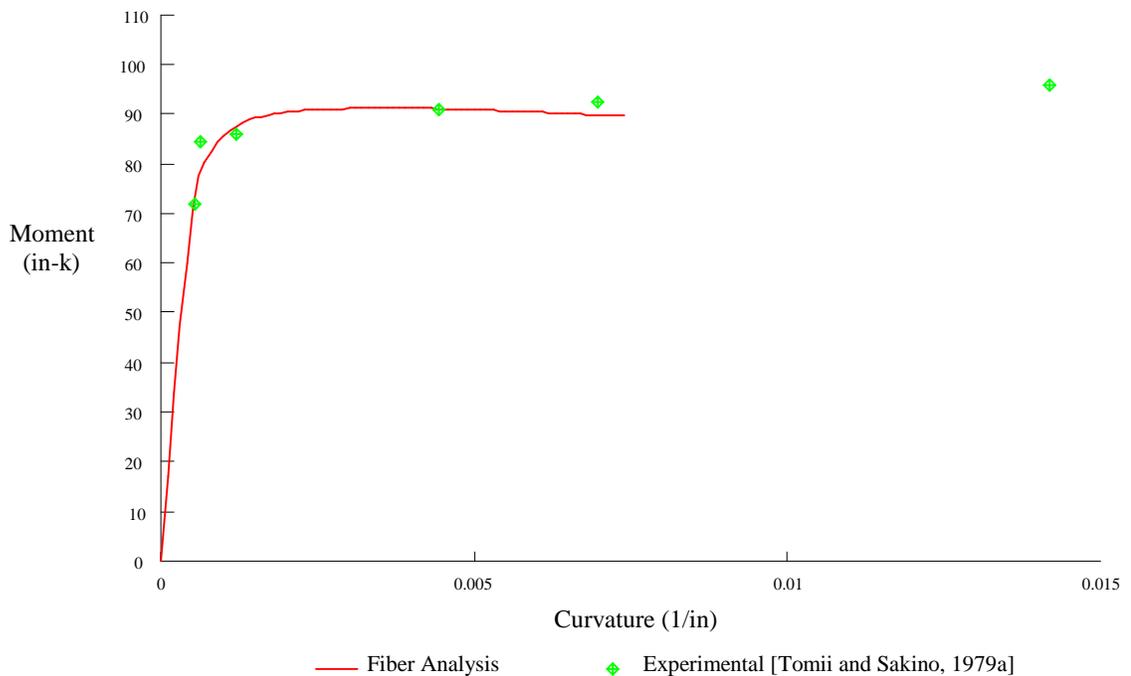


Figure 2.6 Moment-Curvature-Thrust Diagram (Tom44a); $P/P_o = 0.13$

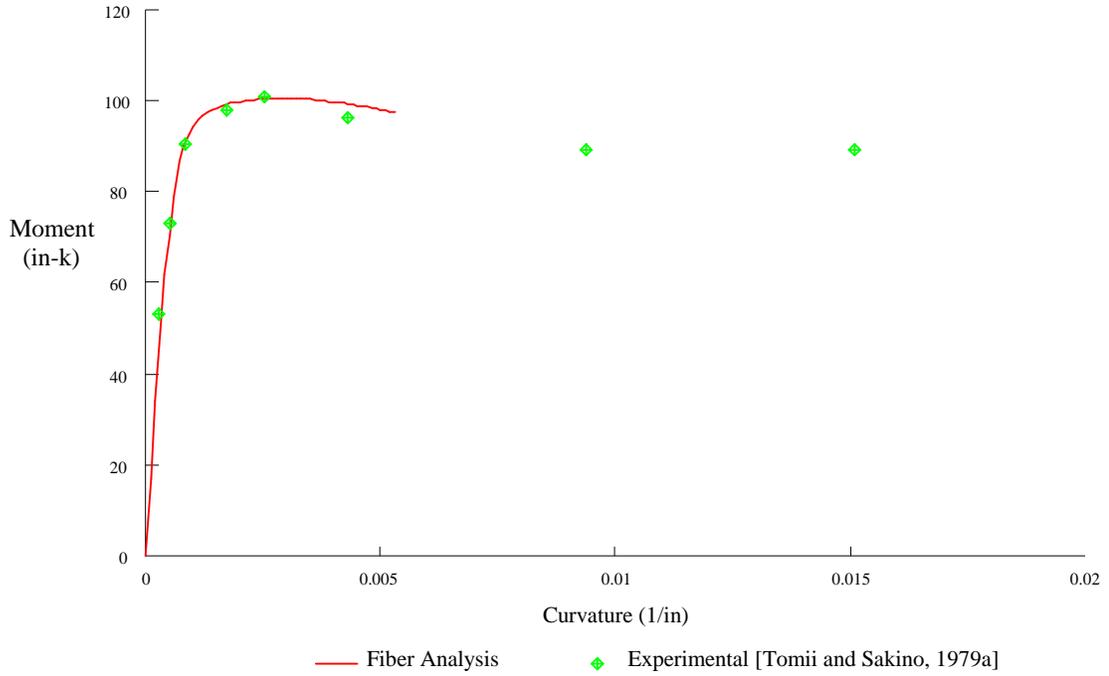


Figure 2.7 Moment-Curvature-Thrust Diagram (Tom44a); $P/P_0 = 0.27$

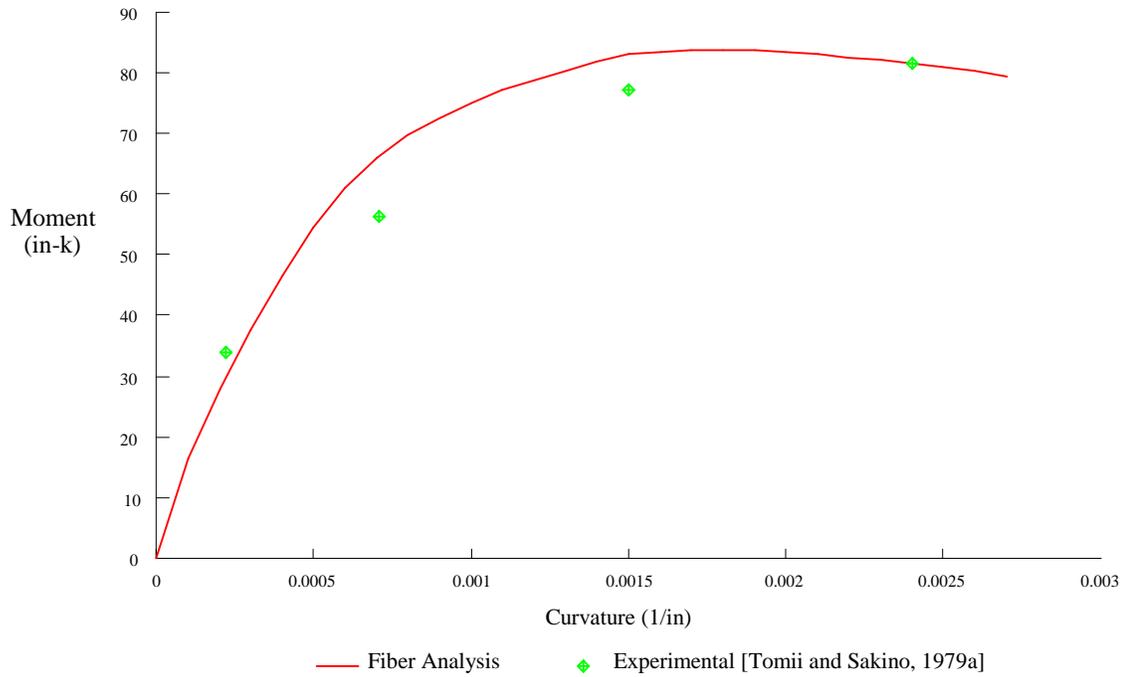


Figure 2.8 Moment-Curvature-Thrust Diagram (Tom44a); $P/P_0 = 0.58$

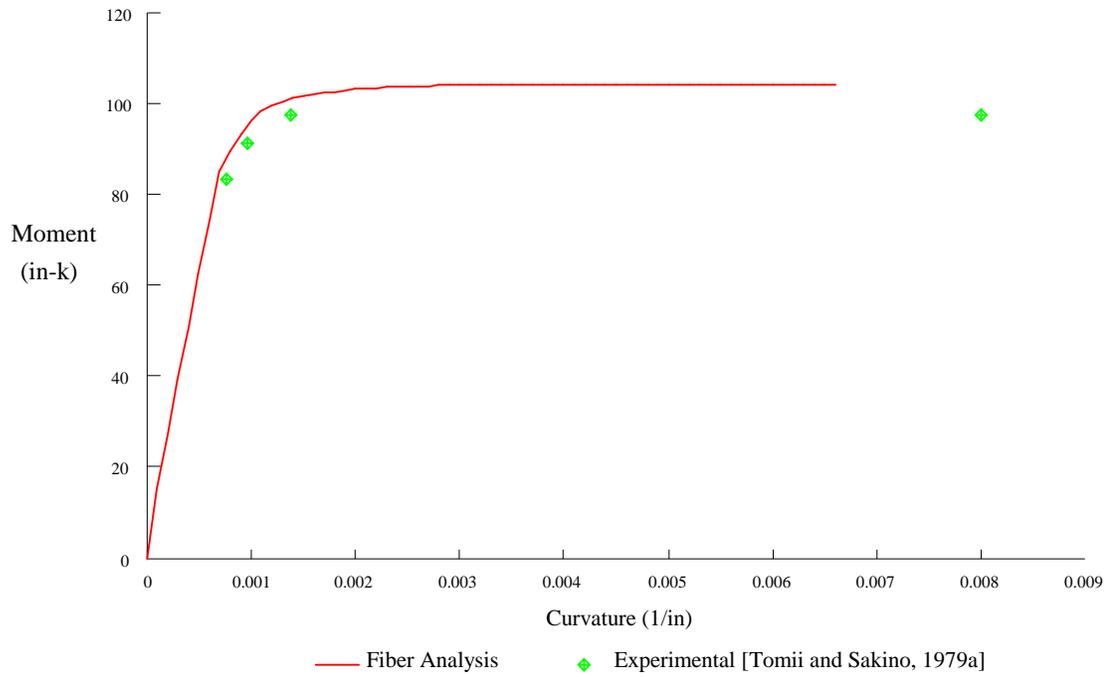


Figure 2.9 Moment-Curvature-Thrust Diagram (Tom44b); $P/P_0 = 0.0$

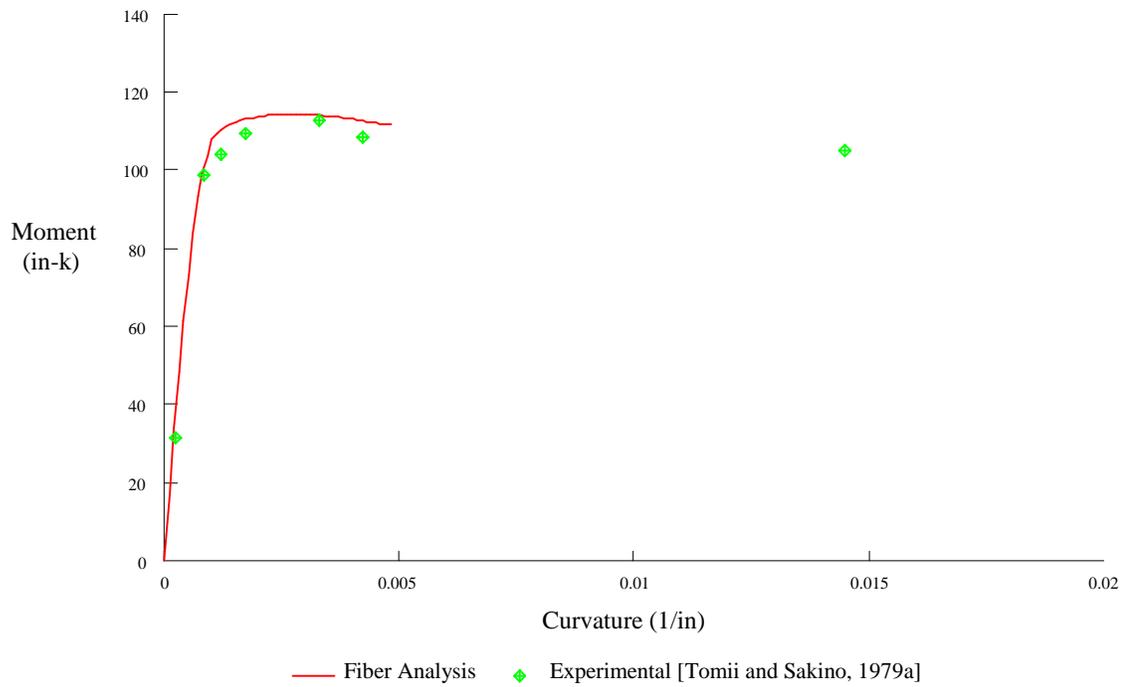


Figure 2.10 Moment-Curvature-Thrust Diagram (Tom44b); $P/P_0 = 0.26$

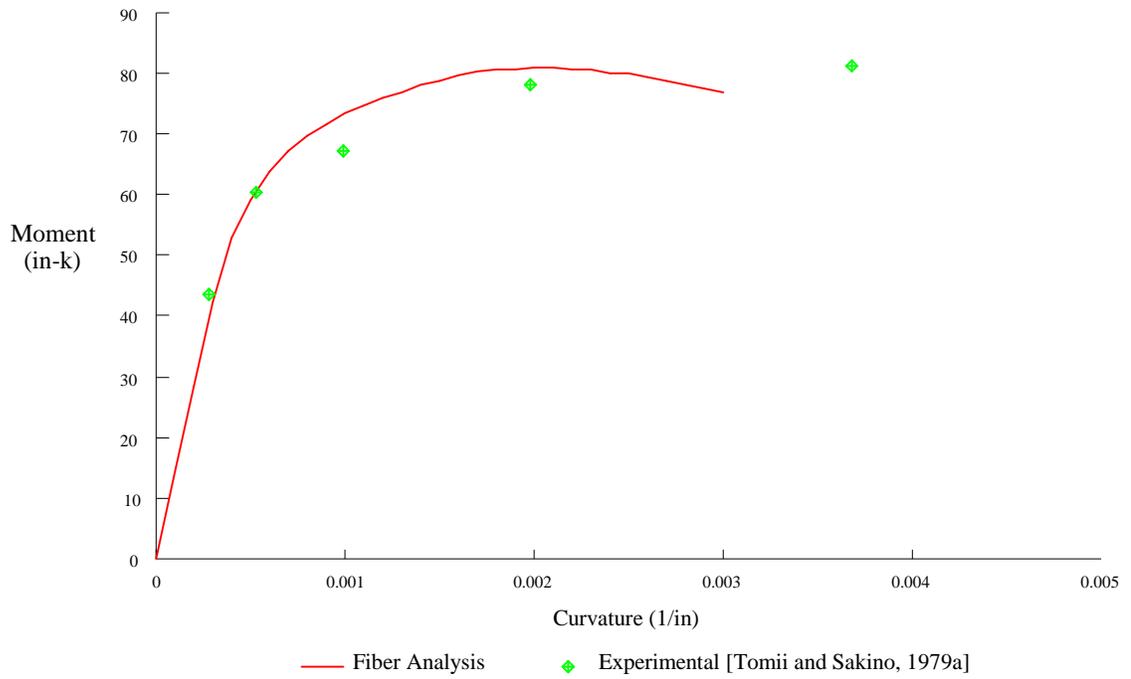


Figure 2.11 Moment-Curvature-Thrust Diagram (Tom44b); $P/P_0 = 0.57$

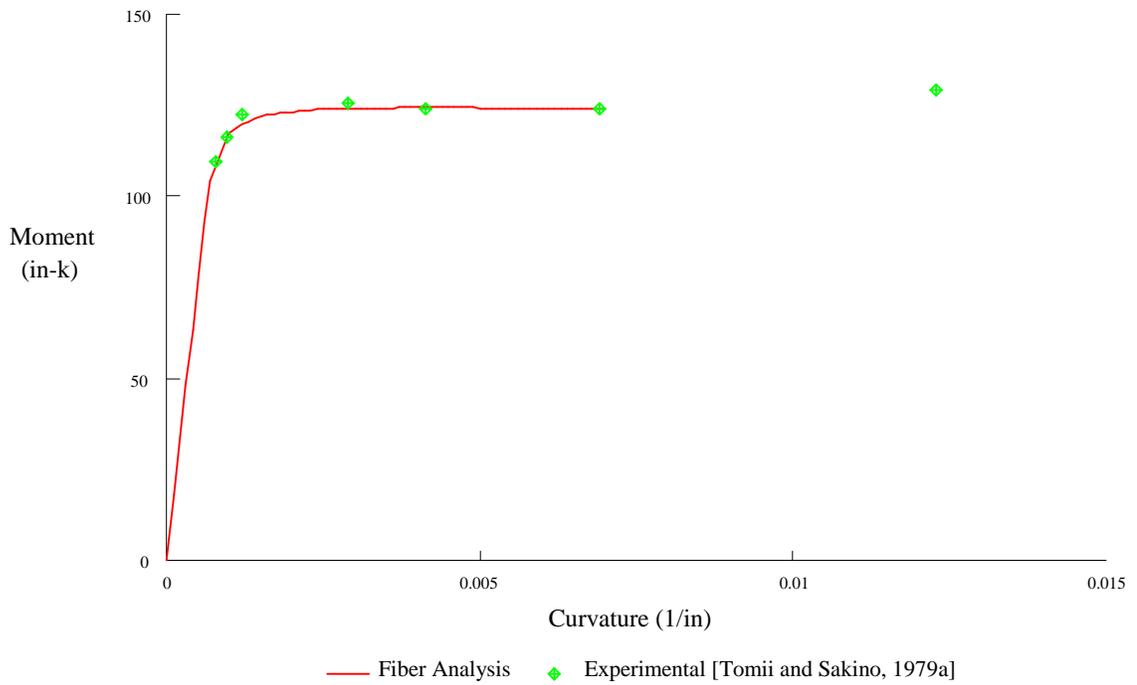


Figure 2.12 Moment-Curvature-Thrust Diagram (Tom33); $P/P_0 = 0.0$

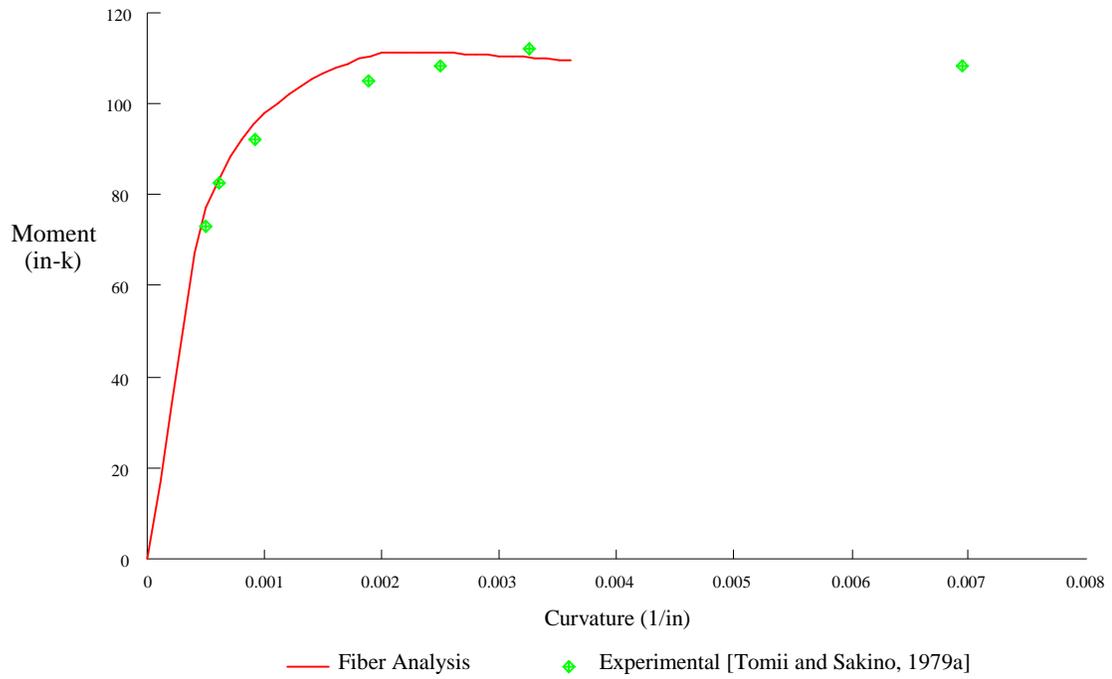


Figure 2.13 Moment-Curvature-Thrust Diagram (Tom33); $P/P_0 = 0.47$

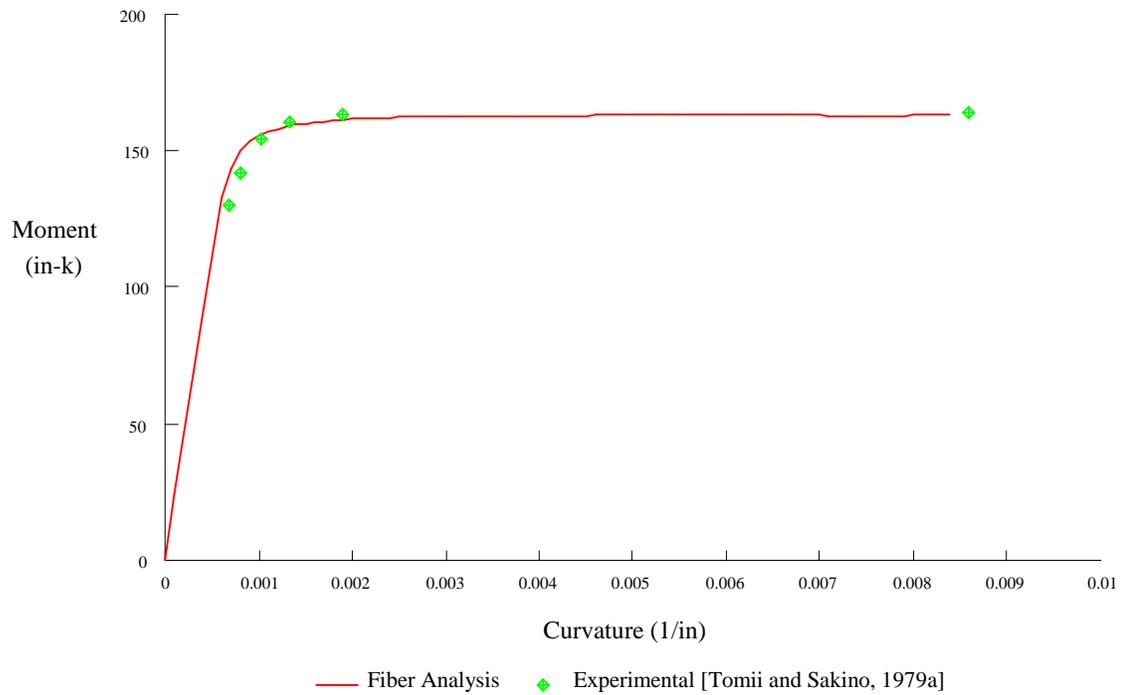


Figure 2.14 Moment-Curvature-Thrust Diagram (Tom24); $P/P_0 = 0.0$

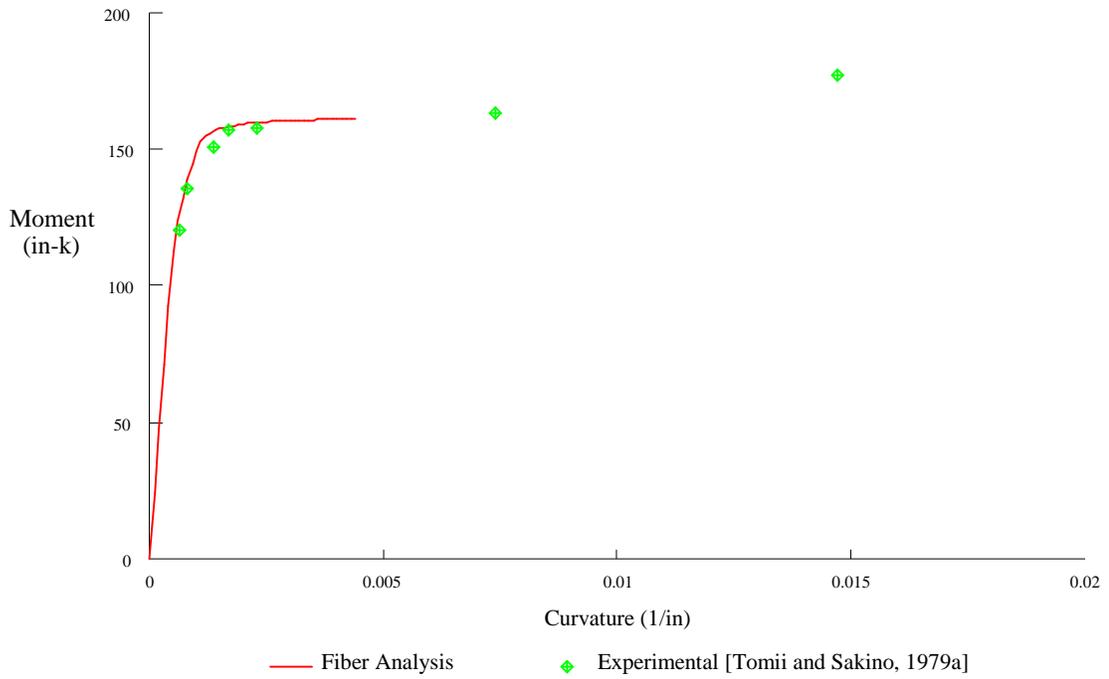


Figure 2.15 Moment-Curvature-Thrust Diagram (Tom24); $P/P_o = 0.29$

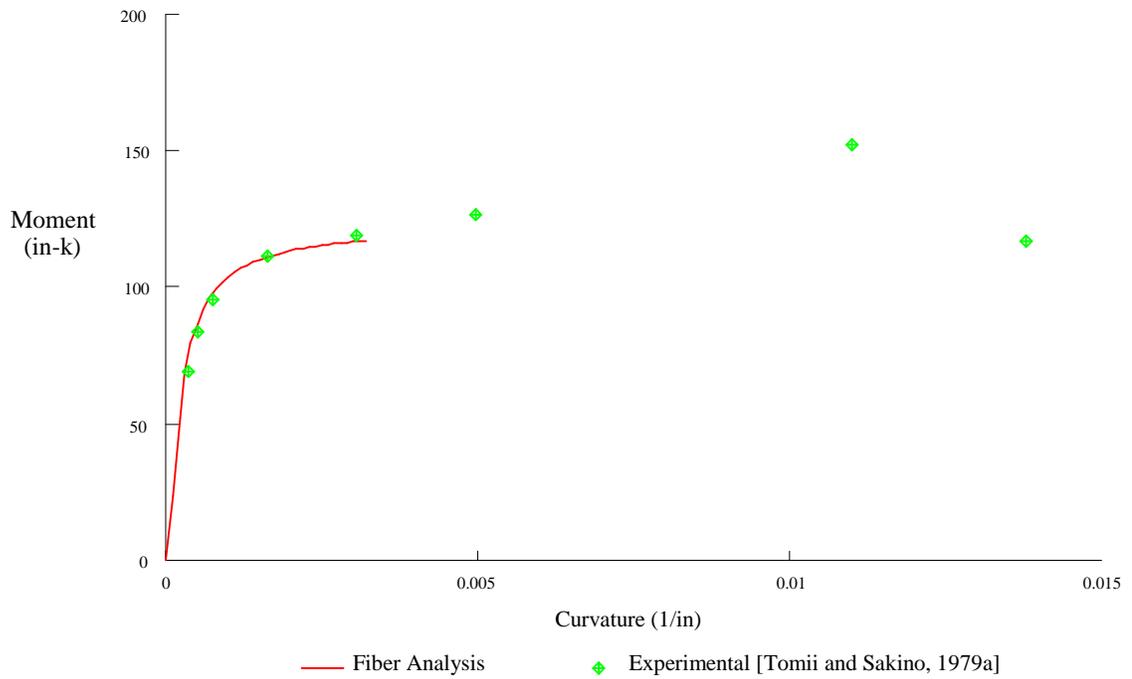


Figure 2.16 Moment-Curvature-Thrust Diagram (Tom24); $P/P_o = 0.57$

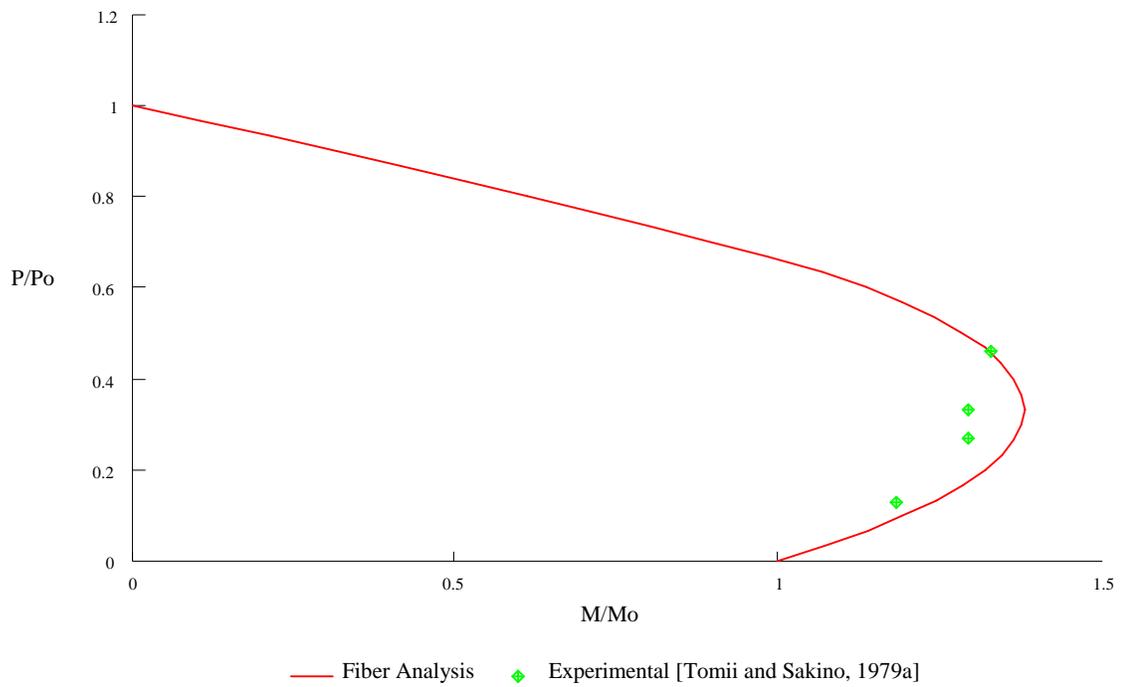


Figure 2.17 Two-Dimensional Cross-Section Strength Diagram (Tom44a)

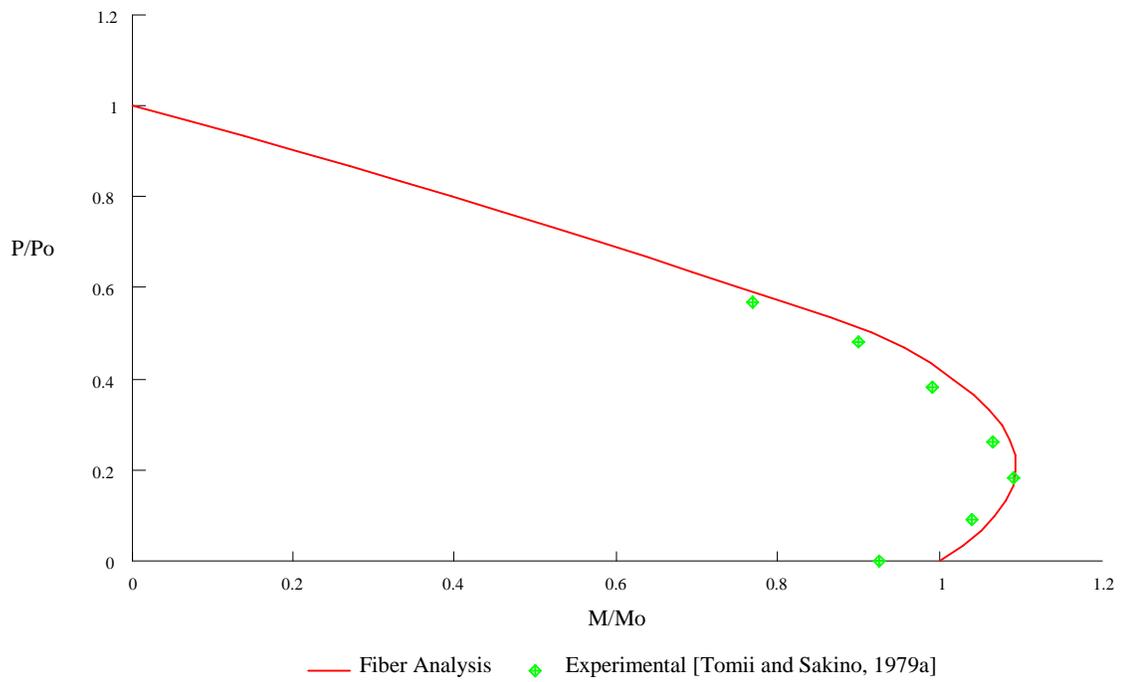


Figure 2.18 Two-Dimensional Cross-Section Strength Diagram (Tom44b)

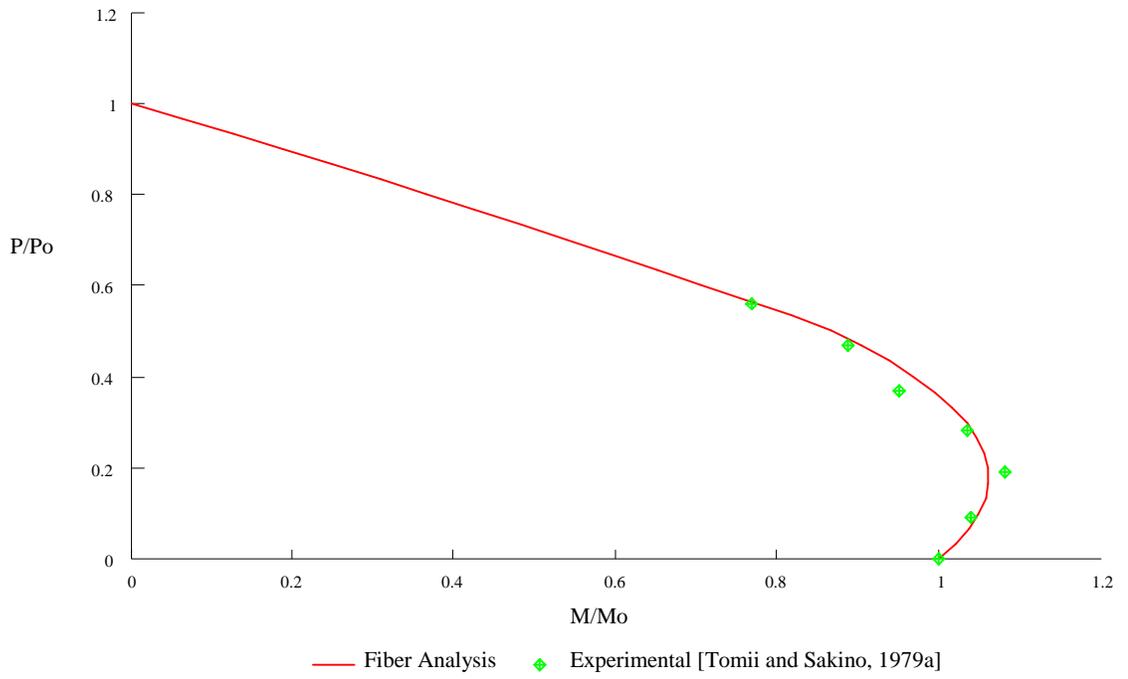


Figure 2.19 Two-Dimensional Cross-Section Strength Diagram (Tom33)

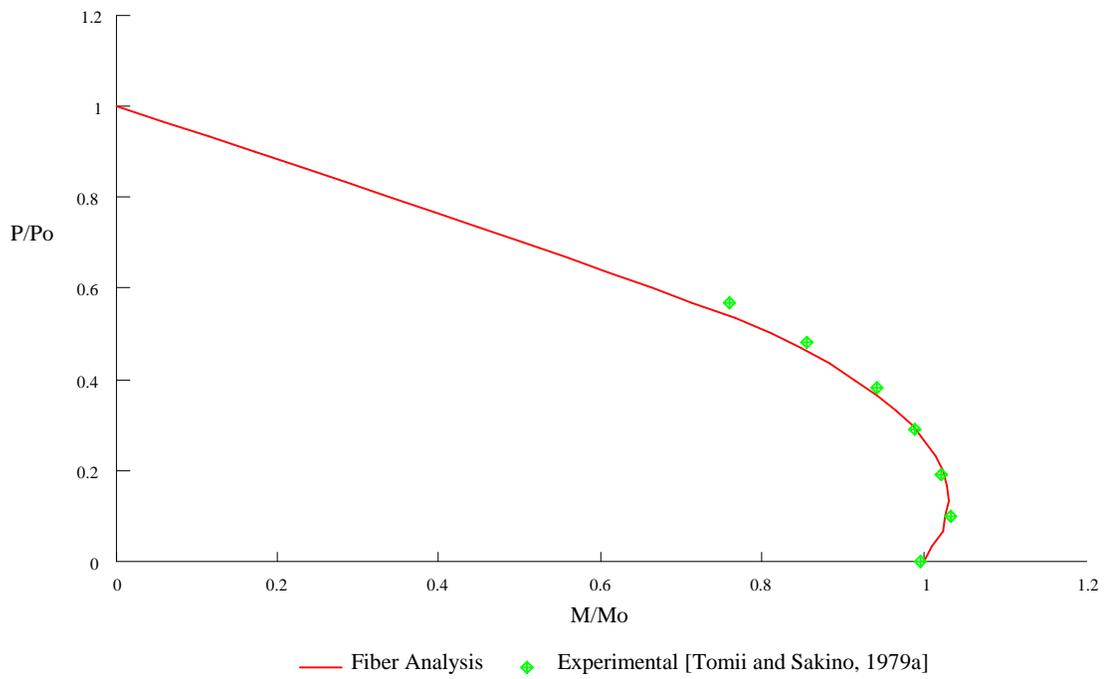


Figure 2.20 Two-Dimensional Cross-Section Strength Diagram (Tom24)

Figures 2.21 to 2.23 illustrate the fiber analysis results versus the two-dimensional cross-section strength data presented by Furlong [1967]; each figure is referenced by the notation listed in column 2 of Table 2.1. These experimental results show a considerable degree of scatter, in contrast to the consistent results presented by Tomii and Sakino. Furlong used a slow rate of loading and stated that additional strength could be achieved with a more rapid loading rate. This may explain the over-prediction of strength shown in Figs. 2.21 and 2.22. On the other hand, the under-prediction in strength shown in Fig. 2.23 may be due to the inaccuracy of the steel properties used in the analysis. The yield stress of the tubes tabulated by Furlong was specified by the supplier, and was not determined prior to the tests.

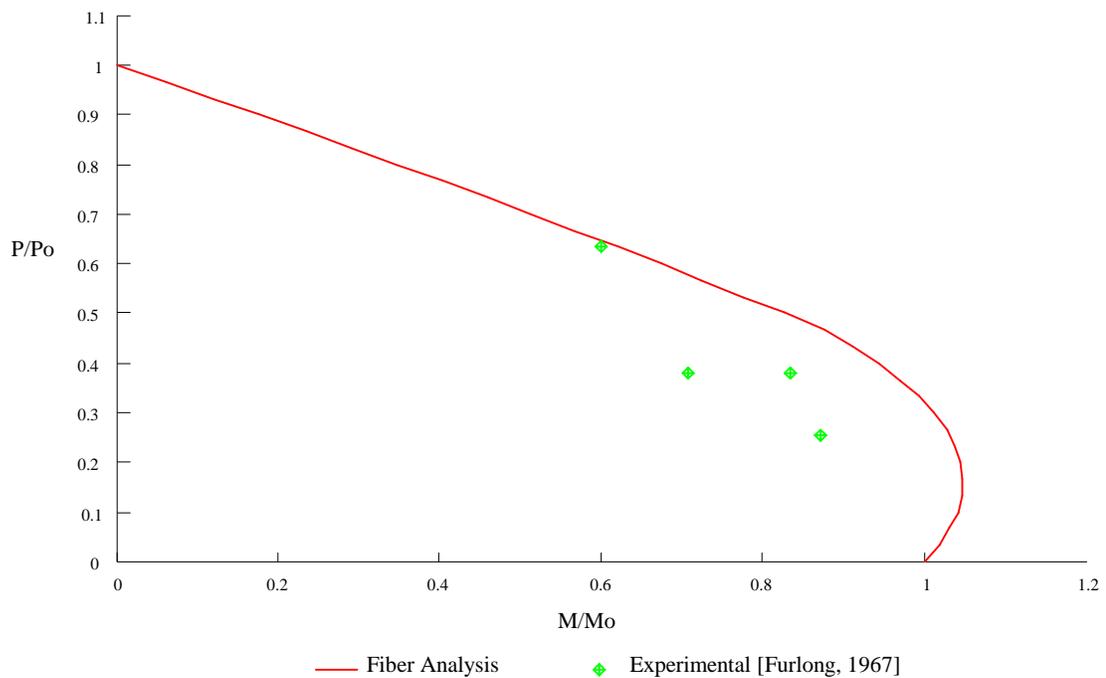


Figure 2.21 Two-Dimensional Cross-Section Strength Diagram (Furl1)

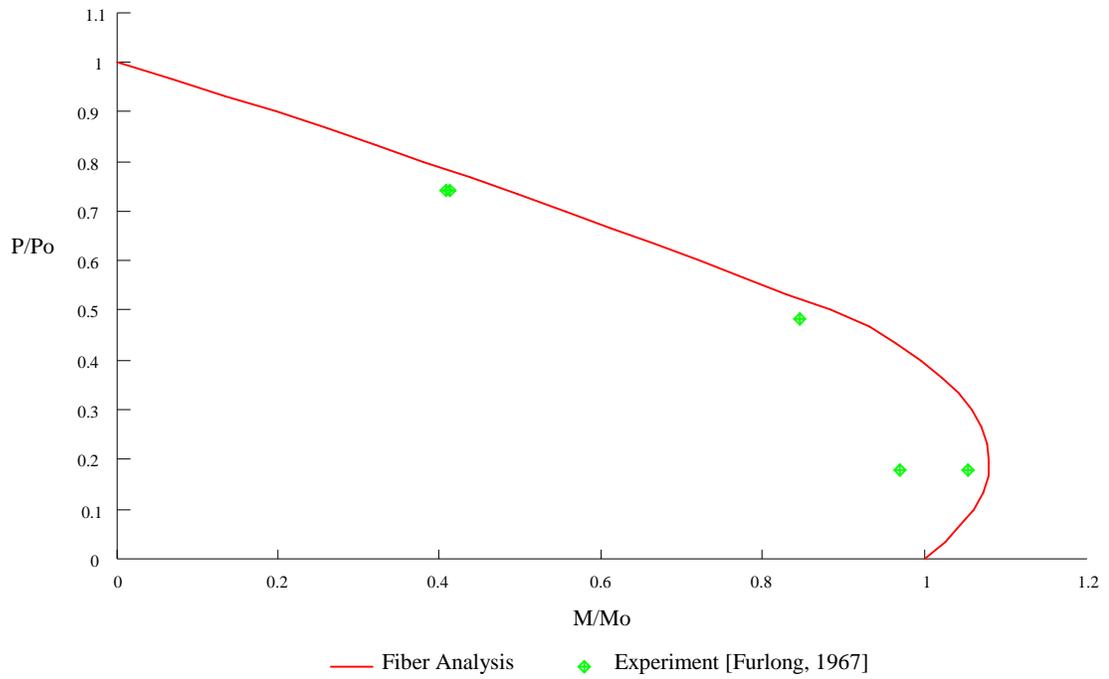


Figure 2.22 Two-Dimensional Cross-Section Strength Diagram (Furl2)

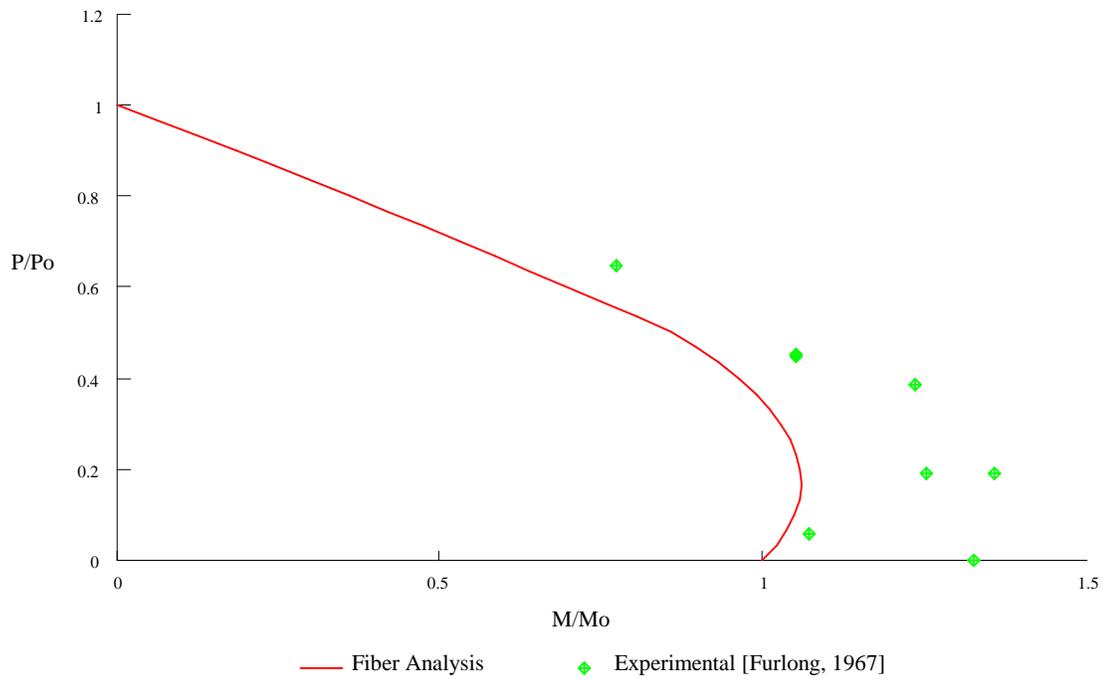


Figure 2.23 Two-Dimensional Cross-Section Strength Diagram (Furl3)

2.6 Equation for the Three-Dimensional Cross-Section Strength Surface

To develop an empirical expression for the cross-section strength of a general CFT, the calibrated fiber element analysis procedure was used to analyze a wide range of cross-sections. A series of CFT sections was selected and analyzed, producing sets of force point (P - M_y - M_z) data to which an equation could be fit. The resulting expression for the three-dimensional cross-section strength surface is meant to apply to any CFT within the range of selected sections. This expression, in turn, forms an integral part of the CFT beam-element concentrated plasticity model.

2.6.1 Selection of CFT Cross-Sections

The first step in the development of an expression for the CFT cross-section strength surface entailed selecting a wide range of cross-sections representative of CFTs used in both present and future construction practice. The two properties which most directly affect the behavior of the CFT cross-section are the ratio of the concrete area to the steel area and the ratio of the concrete strength to the steel strength. Cross-sections may thus be identified by two dimensionless ratios: the ratio of the tube width to tube thickness (D/t ratio), which accounts for the ratio of the areas, and the ratio of the concrete compression strength to the steel yield strength (f'_c/f_y). Four series of square tubes were selected with D/t ratios ranging from 24 to 96. Within each series, four sections with different f'_c/f_y ratios were chosen. The concrete strength ranged from 3.5 ksi to 15 ksi; the steel yield strength of the tubes was 46 ksi. Therefore, f'_c/f_y ranged from 0.0761 to 0.326. The sections chosen represent a range of A_s/A_t , all above a value of 0.04, the minimum required by the AISC LRFD Specification [1993] to constitute a composite column. These cross-sections are summarized in Table 2.2.

Table 2.2 CFT Cross Sections Used for Determining the Cross-Section Strength Surface Equation

Steel tubes ($f_y = 46$ ksi, $f_u = 58$ ksi):

- *Series 24:* $12 \times 12 \times \frac{1}{2}$ ($D/t = 24$, $A_s/A_t = 0.160$)
- *Series 48:* $18 \times 18 \times \frac{3}{8}$ ($D/t = 48$, $A_s/A_t = 0.0816$)
- *Series 72:* $27 \times 27 \times \frac{3}{8}$ ($D/t = 72$, $A_s/A_t = 0.0548$)
- *Series 96:* $36 \times 36 \times \frac{3}{8}$ ($D/t = 96$, $A_s/A_t = 0.0412$)

Concrete properties:

- *Type A:* $f'_c = 3.5$ ksi ($f'_c/f_y = 0.0761$)
- *Type B:* $f'_c = 6.5$ ksi ($f'_c/f_y = 0.141$)
- *Type C:* $f'_c = 10$ ksi ($f'_c/f_y = 0.217$)
- *Type D:* $f'_c = 15$ ksi ($f'_c/f_y = 0.326$)

The tube sizes were chosen based on the listing of available shapes published by AISC [1994]. The two smallest cross-sections in Table 2.2, $12 \times 12 \times \frac{1}{2}$ ($D/t = 24$) and $18 \times 18 \times \frac{3}{8}$ ($D/t = 48$), are standard manufactured shapes. The next largest section, $27 \times 27 \times \frac{3}{8}$ ($D/t = 72$), although not listed by AISC, was chosen to provide a uniform increment in D/t values. This cross-section is, however, still representative of a typical structural tube since standard tubes with dimensions of $26 \times 26 \times \frac{3}{8}$ and $28 \times 28 \times \frac{3}{8}$ are produced. The largest section in Table 2.2 ($D/t = 96$) was also chosen to provide a uniform spacing of D/t ratios, and it represents the larger sections that may be used in future construction [Goel and Yamanouchi, 1993]. The tube with the largest D/t ratio in the AISC listing of structural tubes is a $30 \times 30 \times \frac{3}{8}$ ($D/t = 80$). The tube thickness of this section was maintained and the width and depth were increased by six inches to obtain a D/t ratio of 96. All of the tubes in Table 2.2 are cold-formed square tubes with a nominal yield strength of 46 ksi and an ultimate strength of 58 ksi. Additionally, the following stress/strain parameters were used for the steel (see Section 2.3.2):

$$\text{Modulus of elasticity } (E_s) = 29,000,000 \text{ psi}$$

$$\text{Strain at which strain hardening begins } (\epsilon_{sh}) = 0.0186$$

$$\text{Strain hardening modulus } (E_{sh}) = 300,000 \text{ psi}$$

Additional rectangular CFT sections with an aspect ratio (the ratio of the long side of the tube to the short side) ranging from 1 to 2, and CFTs with tube strengths up to 70 ksi were checked following the development of the cross-section strength equation using the 16 sections of Table 2.2. These additional results are summarized in Sections 2.6.5 and 2.6.6.

For each combination of D/t and f'_c/f_y shown in Table 2.2, a cross-section strength surface consisting of 100 P- M_y - M_z points was generated using the fiber element analysis. Each three-dimensional surface consisted of a series of 10 two-dimensional cross-section strength curves generated at load eccentricity increments of 10 degrees, ranging from 0° (major axis bending) to 90° (minor axis bending). For each two-dimensional cross-section strength curve, 10 points in the M-P plane were computed.

2.6.2 Development of the CFT Cross-Section Strength Equation Form

The equation describing the three-dimensional cross-section strength surface of rectangular CFTs is a polynomial expression with terms consisting of products of the normalized force points -- p , m_y , and m_z . The most accurate form for the equation was determined by performing a least squares fit to the set of cross-section analyses described in Section 2.6.1 (for the CFTs of Table 2.2). The equation consists of six terms: three terms representing the normalized loads (p , m_y , and m_z) and three cross-product terms ($m_y \cdot p$, $m_z \cdot p$, and $m_z \cdot m_y$). The normalized loads p , m_y , and m_z are given by the following equations:

$$p = \frac{P - \phi \cdot P_o}{P_o} \quad m_y = \frac{M_y}{M_{y_o}} \quad m_z = \frac{M_z}{M_{z_o}} \quad (2.17)$$

P_o , M_{y_o} , and M_{z_o} are the respective axial compression, ultimate minor axis bending moment, and major axis bending moment capacities for the cross-section (see Section 2.6.7 for a description of their calculation), and ϕ is the location in normalized force space of the centroid of the three-dimensional cross-section strength surface. The value of ϕ , which varies from 0 to 1, is required because of the asymmetry of the three-dimensional cross-section strength surface of a CFT about the moment axes (a typical CFT cross-section strength diagram is shown in Fig. 2.24). As illustrated in Fig. 2.24, an actual CFT cross-section strength surface is, however, approximately symmetric about a moment axis located at the axial load ratio, ϕ , producing the maximum moment in the section. Thus, by using the parameter ϕ , the axial load is normalized with respect to this shifted moment axis (the dotted line in the figure). Section 2.6.3 discusses the calculation of the value of the centroid of the cross-section strength surface, ϕ .

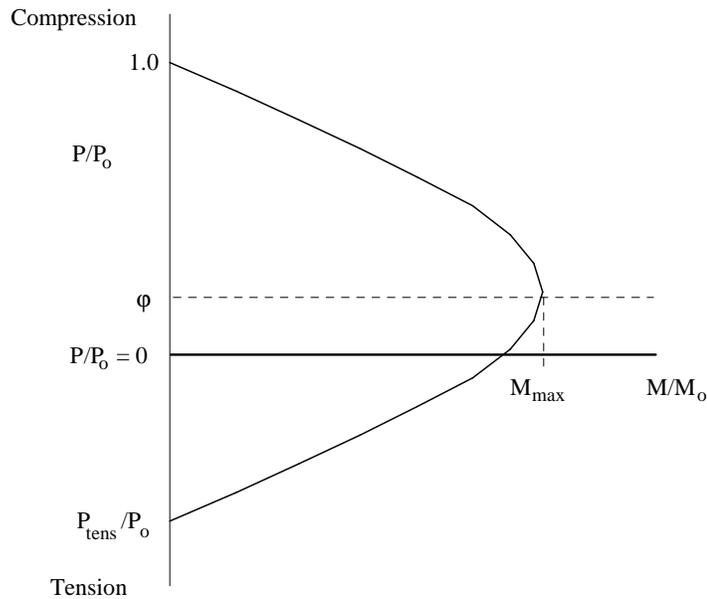


Figure 2.24 Typical CFT Two-Dimensional Cross-Section Strength Diagram

The equation for the cross-section strength surface takes the following general form (see Orbison et al. [1982] and Zhao [1993] for a description of steel wide-flange cross-section strength surfaces of a similar form, and Duan and Chen [1990] for a description of similar hollow tube cross-section strength surfaces):

$$c_1 \cdot m_y^{n_1} + c_2 \cdot m_z^{n_2} + c_3 \cdot p^{n_3} + c_4 \cdot (m_y^{n_4} \cdot p^{n_5}) + c_5 \cdot (m_z^{n_6} \cdot p^{n_7}) + c_6 \cdot (m_y^{n_8} \cdot m_z^{n_9}) = 1.0 \quad (2.18)$$

where c_1, c_2, \dots, c_6 are coefficients. The exponents of the terms in Eq. (2.18) (n_1, n_2, \dots, n_9) must be even integers to insure that the behavior about the coordinate axes is symmetric (see Section 2.6.3). Because the 16 cross-sections used in the least squares analysis were square, the properties in the y and z directions are identical. Therefore, the m_y and m_z terms and the $m_y \cdot p$ and $m_z \cdot p$ terms are the same. This decreases the number of coefficients in the surface equation to four, and the number of exponents to five:

$$c_1 \cdot (m_y^{n_1} + m_z^{n_1}) + c_2 \cdot p^{n_2} + c_3 \cdot (m_y^{n_3} \cdot p^{n_4} + m_z^{n_3} \cdot p^{n_4}) + c_4 \cdot (m_y^{n_5} \cdot m_z^{n_5}) = 1.0 \quad (2.19)$$

The four coefficients of Eq. (2.19) are each functions of the two cross-section parameters, the D/t ratio and the f'_c/f_y ratio, rather than remaining constant for all CFT sections. Figure 2.1 in Section 2.1.2 compares the two-dimensional cross-section strength curves from the fiber analyses of section types 96A and 96D (refer to Table 2.1 for this nomenclature). The surfaces are similar in shape, but the section with the larger f'_c/f_y ratio, 96D, produces a substantially larger surface. A similar increase in the size of the cross-section strength surface occurs as the D/t ratio increases from 24 to 96. This comparison illustrates the necessity of using coefficients in Eq. (2.19) that are functions

of the section and material properties. Both of the surfaces shown in Fig. 2.1 could not be accurately described using fixed coefficient values. Note that cross-section strength surfaces which have been developed for similar steel wide-flange sections [Orbison et al., 1982; Zhao, 1993] use the same coefficient values regardless of section and material properties, since the cross-section strengths of a wide range of steel I-beams are nearly identical in normalized force space [Chen and Atsuta, 1976, 1977]. Section 2.6.4 discusses the development of the equations used to describe the coefficients.

The general form of Eq. (2.19) was first optimized for section type 48B by explicitly using different combinations of exponent values. For each selected set of exponents, the coefficients c_1 , c_2 , c_3 , and c_4 were determined by a least squares procedure using the normalized axial load and moment data from the fiber analysis of this section. The accuracy of the resulting expression with the optimized coefficients was then checked against the original fiber analysis data. Several different combinations of exponents were tried using this procedure. The best set of exponents was determined based upon a combination of achieving the smallest average error of all combinations of m_z , m_y , and p generated by the fiber analysis, and achieving the smallest standard deviation from the average.

Forms of the equation in which certain terms were neglected were also analyzed, but it became clearly evident that every term of Eq. (2.19) was necessary. The five or six most accurate forms of the equation for section 48B were then checked using sections with different combinations of D/t and f'_c/f_y . From the least squares analyses of these additional sections, the optimum form of the equation was selected:

$$c_1 \cdot (m_y^2 + m_z^2) + c_2 \cdot p^2 + c_3 \cdot (m_y^2 \cdot p^2 + m_z^2 \cdot p^2) + c_4 \cdot m_y^2 \cdot m_z^2 = 1.0 \quad (2.20)$$

2.6.3 Modeling the Asymmetry of the Cross-Section Strength Surface

As shown in Fig. 2.24, the typical CFT cross-section strength surface is unsymmetric about the moment axes. This is due to the concrete, which provides much more strength in compression than in tension. Therefore, the largest moment capacity occurs when the cross-section is subjected to a moderate compressive axial load.

As discussed in Section 2.6.2, the normalized cross-section strength surface of all CFTs was found by inspection to be nearly symmetric about an axis represented by the axial load ratio, ϕ , producing the maximum moment capacity. The centroid of the surface, ϕ , is a function of the relative ratio of concrete to steel and the concrete strength, f'_c . The most accurate formulation for determining the surface centroid was obtained by using the normalized average of the tensile and compressive strengths of the concrete. This calculated value is expressed as:

$$\phi_{calc} = \frac{(A_c \cdot f'_c + A_c \cdot f_r)/2}{P_o} \quad (2.21)$$

where P_o is the ultimate axial load capacity of the section in the presence of no bending (see Section 2.6.7) and the rupture strength of the concrete, f_r , is given by Eq. (2.12). All values in Eq. (2.21) are taken as positive except for f_r , which is negative, and ϕ_{calc} is the calculated position of the centroid on the compressive axis of P/P_o .

Table 2.3 compares the calculated value of the surface centroid, ϕ_{calc} , with the centroid value obtained from the fiber analysis, ϕ_{fa} (i.e., the axial load ratio at the point of maximum moment). The values of ϕ_{calc} show excellent correlation with ϕ_{fa} --the maximum error between these values for the 16 cross-sections of Table 2.2 is 7.0 % and the majority of the values are within an error of 2.0 %. These results indicate that using ϕ

from Eq. (2.21) in Eq. (2.17), as is done in this work, serves as an accurate method of representing the asymmetry of the cross-section strength surface.

Table 2.3 Errors in Calculation of Cross-Section Strength Surface Centroids

Section Type	Φ_{calc} Eq. (2.21)	Φ_{fa} (fiber analysis)	% error
24A	0.125	0.133	-6.02 %
24B	0.193	0.200	-3.50 %
24C	0.247	0.250	-1.20 %
24D	0.297	0.300	-1.00 %
48A	0.201	0.200	0.50 %
48B	0.278	0.283	-1.76 %
48C	0.328	0.333	-1.50 %
48D	0.369	0.367	0.54 %
72A	0.248	0.250	-0.80 %
72B	0.322	0.317	1.58 %
72C	0.365	0.350	4.29 %
72D	0.399	0.400	-0.25 %
96A	0.279	0.300	-7.00 %
96B	0.348	0.350	-0.57 %
96C	0.386	0.383	0.78 %
96D	0.415	0.400	3.75 %

2.6.4 Development of the Coefficient Equations

For each of the 16 cross-sections of the data set in Table 2.2, the coefficients of Eq. (2.20) were optimized using a least squares procedure, producing 16 values of each of the four coefficients, c_1 , c_2 , c_3 , and c_4 . These values are listed in Table 2.4. This table reemphasizes the need for functional coefficients, as stressed in Section 2.6.2.

Coefficients c_1 and c_4 , for example, range from 0.2214 to 0.9379 and from 0.03667 to 0.3962, respectively. Consequently, using fixed coefficients, or even using coefficients that are linear functions of D/t and f'_c/f_y , would introduce large errors into Eq. (2.20).

For each coefficient, c_i , a least squares analysis was performed using the 16 respective coefficient values in Table 2.4 to obtain expressions in terms of the D/t ratio and the f'_c/f_y ratio of the cross-section. Two equation forms were examined for the optimization of the coefficients, a full quadratic equation:

$$c_i = a_1 + a_2 \cdot \left(\frac{D}{t}\right) + a_3 \cdot \left(\frac{f'_c}{f_y}\right) + a_4 \cdot \left(\frac{D}{t}\right) \cdot \left(\frac{f'_c}{f_y}\right) + a_5 \cdot \left(\frac{D}{t}\right)^2 + a_6 \cdot \left(\frac{f'_c}{f_y}\right)^2 \quad (2.22)$$

and a full cubic equation:

$$c_i = b_1 + b_2 \cdot \left(\frac{D}{t}\right) + b_3 \cdot \left(\frac{f'_c}{f_y}\right) + b_4 \cdot \left(\frac{D}{t}\right) \cdot \left(\frac{f'_c}{f_y}\right) + b_5 \cdot \left(\frac{D}{t}\right)^2 + b_6 \cdot \left(\frac{f'_c}{f_y}\right)^2 + b_7 \cdot \left(\frac{D}{t}\right) \cdot \left(\frac{f'_c}{f_y}\right)^2 + b_8 \cdot \left(\frac{D}{t}\right)^2 \cdot \left(\frac{f'_c}{f_y}\right) + b_9 \cdot \left(\frac{D}{t}\right)^3 + b_{10} \cdot \left(\frac{f'_c}{f_y}\right)^3 \quad (2.23)$$

where a_1, \dots, a_6 and b_1, \dots, b_{10} are constant coefficients. The optimization of coefficient c_1 using Eqs. (2.22) and (2.23) produces the following quadratic and cubic equations:

Table 2.4 Optimum CFT Cross-Section Strength Surface Equation Coefficients

D/t	f'_c (ksi)	c_1	c_2	c_3	c_4
24	3.5	0.9379	1.598	1.961	0.3962
	6.5	0.8671	1.890	1.875	0.3538
	10	0.8130	2.187	1.771	0.3281
	15	0.6994	2.542	1.492	0.2453
48	3.5	0.8322	1.917	2.324	0.3919
	6.5	0.6962	2.381	2.143	0.3014
	10	0.5684	2.812	1.804	0.2091
	15	0.4521	3.287	1.420	0.1354
72	3.5	0.7297	2.160	2.146	0.3182
	6.5	0.5589	2.686	1.866	0.2014
	10	0.4255	3.123	1.544	0.1250
	15	0.3025	3.573	1.149	0.06702
96	3.5	0.6568	2.350	1.966	0.2580
	6.5	0.4648	2.893	1.650	0.1397
	10	0.3301	3.320	1.307	0.07567
	15	0.2214	3.749	0.9194	0.03667

$$c_1 = 1.235 - 0.009683 \cdot \left(\frac{D}{t}\right) + 0.00005267 \cdot \left(\frac{D}{t}\right)^2 - 0.8031 \cdot \left(\frac{f'_c}{f_y}\right) + 0.01335 \cdot \left(\frac{f'_c}{f_y}\right)^2 + 0.01127 \cdot \left(\frac{D}{t}\right) \cdot \left(\frac{f'_c}{f_y}\right) \quad (2.24)$$

$$c_1 = 1.077 - 0.002646 \cdot \left(\frac{D}{t}\right) + 0.00002304 \cdot \left(\frac{D}{t}\right)^2 - 1.128 \cdot 10^{-7} \cdot \left(\frac{D}{t}\right)^3 + 0.3745 \cdot \left(\frac{f'_c}{f_y}\right) - 1.299 \cdot \left(\frac{f'_c}{f_y}\right)^2 - 0.04193 \cdot \left(\frac{f'_c}{f_y}\right)^3 - 0.06913 \cdot \left(\frac{D}{t}\right) \cdot \left(\frac{f'_c}{f_y}\right) + 0.0002339 \cdot \left(\frac{D}{t}\right)^2 \cdot \left(\frac{f'_c}{f_y}\right) + 0.07542 \cdot \left(\frac{D}{t}\right) \cdot \left(\frac{f'_c}{f_y}\right)^2 \quad (2.25)$$

Table 2.5 shows the errors in the coefficient c_1 obtained using Eqs. (2.24) and (2.25) for each of the CFT cross-sections. The errors are computed based upon a comparison to the optimum values obtained from the least squares analysis of each individual cross-sections. These optimum coefficient values are listed in Table 2.4 and repeated in column 2 of Table 2.5. Equations similar to Eqs. (2.24) and (2.25) were obtained for coefficients c_2 , c_3 , and c_4 . The corresponding errors for these coefficients are tabulated in Tables 2.6, 2.7, and 2.8, respectively.

The coefficients obtained from the quadratic form of the coefficient equation, Eq. (2.22), produced errors in Eq. (2.20) of well in excess of 10% for force points of several of the cross-sections, especially in the ranges of high D/t and high f'_c/f_y . This was due to the large errors in the quadratic equation for coefficients c_1 and c_4 (shown in Tables 2.5 and 2.8, respectively). Therefore, the cubic coefficient equations, which produced much smaller errors in Eq. (2.20), were adopted in this work. For continuity, the cubic form was used for all four coefficient equations, although coefficients c_1 and c_4 particularly required it. The final form of the CFT three-dimensional cross-section

Table 2.5 Errors in Coefficient c_1

Section Type	Optimum Value of c_1 (Table 2.4)	Quadratic Equation Value of c_1	% Error in Quadratic Equation	Cubic Equation Value of c_1	% Error in Cubic Equation
24A	0.9379	0.9513	-1.43%	0.9406	-0.29%
24B	0.8671	0.8819	-1.70%	0.8731	-0.69%
24C	0.813	0.8006	1.52%	0.7993	1.68%
24D	0.6994	0.6844	2.14%	0.7032	-0.54%
48A	0.8322	0.7894	5.14%	0.8210	1.34%
48B	0.6962	0.7024	-0.89%	0.6975	-0.19%
48C	0.5684	0.6006	-5.66%	0.5776	-1.62%
48D	0.4521	0.4549	-0.61%	0.4519	0.05%
72A	0.7297	0.6881	5.69%	0.7297	-0.00%
72B	0.5589	0.5836	-4.41%	0.5678	-1.59%
72C	0.4255	0.4612	-8.39%	0.4222	0.77%
72D	0.3025	0.2860	5.45%	0.2962	2.09%
96A	0.6568	0.6476	1.41%	0.6574	-0.09%
96B	0.4648	0.5254	-13.04%	0.4745	-2.09%
96C	0.3301	0.3825	-15.88%	0.3238	1.91%
96D	0.2214	0.1778	19.68%	0.2268	-2.45%

Table 2.6 Errors in Coefficient c_2

Section Type	Optimum Value of c_2 (Table 2.4)	Quadratic Equation Value of c_2	% Error in Quadratic Equation	Cubic Equation Value of c_2	% Error in Cubic Equation
24A	1.598	1.510	5.52%	1.581	1.09%
24B	1.890	1.898	-0.44%	1.901	-0.56%
24C	2.187	2.256	-3.15%	2.188	-0.04%
24D	2.542	2.585	-1.70%	2.548	-0.24%
48A	1.917	1.961	-2.30%	1.943	-1.37%
48B	2.381	2.384	-0.13%	2.392	-0.45%
48C	2.812	2.782	1.06%	2.797	0.53%
48D	3.287	3.170	3.58%	3.265	0.66%
72A	2.160	2.236	-3.52%	2.146	0.63%
72B	2.686	2.694	-0.29%	2.667	0.71%
72C	3.123	3.132	-0.29%	3.124	-0.03%
72D	3.573	3.578	-0.13%	3.606	-0.91%
96A	2.350	2.335	0.65%	2.355	-0.21%
96B	2.893	2.827	2.28%	2.891	0.07%
96C	3.320	3.306	0.43%	3.334	-0.41%
96D	3.749	3.809	-1.61%	3.734	0.41%

Table 2.7 Errors in Coefficient c_3

Section Type	Optimum Value of c_3 (Table 2.4)	Quadratic Equation Value of c_3	% Error in Quadratic Equation	Cubic Equation Value of c_3	% Error in Cubic Equation
24A	1.961	2.066	-5.36%	1.964	-0.13%
24B	1.875	1.909	-1.82%	1.898	-1.23%
24C	1.771	1.729	2.37%	1.751	1.14%
24D	1.492	1.478	0.95%	1.488	0.24%
48A	2.324	2.211	4.85%	2.310	0.62%
48B	2.143	2.008	6.32%	2.109	1.61%
48C	1.804	1.773	1.74%	1.833	-1.61%
48D	1.420	1.443	-1.59%	1.445	-1.77%
72A	2.146	2.175	-1.34%	2.168	-1.03%
72B	1.866	1.924	-3.11%	1.885	-1.00%
72C	1.544	1.634	-5.84%	1.544	0.02%
72D	1.149	1.226	-6.66%	1.120	2.57%
96A	1.966	1.956	0.49%	1.964	0.09%
96B	1.650	1.659	-0.53%	1.652	-0.11%
96C	1.307	1.314	-0.54%	1.308	-0.05%
96D	0.919	0.827	10.08%	0.937	-1.92%

Table 2.8 Errors in Coefficient c_4

Section Type	Optimum Value of c_4 (Table 2.4)	Quadratic Equation Value of c_4	% Error in Quadratic Equation	Cubic Equation Value of c_4	% Error in Cubic Equation
24A	0.3962	0.4356	-9.95%	0.4019	-1.44%
24B	0.3538	0.3579	-1.17%	0.3554	-0.46%
24C	0.3281	0.2923	10.91%	0.3187	2.87%
24D	0.2453	0.2458	-0.19%	0.2475	-0.89%
48A	0.3919	0.3642	7.06%	0.3827	2.35%
48B	0.3014	0.2819	6.48%	0.2937	2.56%
48C	0.2091	0.2107	-0.76%	0.2241	-7.17%
48D	0.1354	0.1563	-15.41%	0.1377	-1.69%
72A	0.3182	0.3033	4.67%	0.3212	-0.95%
72B	0.2014	0.2163	-7.39%	0.2080	-3.28%
72C	0.1250	0.1396	-11.68%	0.1270	-1.61%
72D	0.0670	0.0773	-15.29%	0.0562	16.21%
96A	0.2580	0.2530	1.95%	0.2588	-0.32%
96B	0.1397	0.1612	-15.38%	0.1397	-0.00%
96C	0.0757	0.0790	-4.41%	0.0688	9.14%
96D	0.0367	0.0088	76.07%	0.0442	-20.51%

strength surface equation and the four corresponding cubic coefficient equations are shown in Table 2.9.

Using the equations of Table 2.9, the errors in each of the 16 cross-sections of Table 2.2 were examined and documented. For a given cross-section, the 100 fiber analysis data points were each substituted into Eq. (2.20) and the error was again determined by checking the deviation of the solution to Eq. (2.20) from a value of 1.0. The cross-section strength errors (e.g., average error and standard deviation of the error for the 100 points, as well as the maximum positive and negative errors) are compiled in Table 2.10. The cumulative errors are very good for all of the sections. The average error is never greater than 2.0 % and the largest error between the results of the surface equation and the fiber analysis for any p - m_y - m_z point of the 16 cross-sections is only 10.14 %.

Table 2.11 examines the accuracy of the cross-section strength equation in the tensile region. Only the steel tube is assumed to contribute to the tensile capacity of a CFT, which may be expressed as $P_{\text{tens}} = A_s \cdot f_y$. This value is normalized by P_o and compared to the value predicted by Eq. (2.20) for the case of pure axial tension, P_{eq}/P_o . The errors between the two values are quite accurate except for the sections with a high f'_c . These errors are, however, acceptable since the tensile region was modeled primarily to maintain a symmetric and continuous equation and it is presumed in this work that the CFT beam-column for which the equation was developed will never be subjected to a purely axial tensile load.

Table 2.9 CFT Cross-Section Strength Surface Equation

CFT Cross-Section Strength Surface Equation:

$$c_1 \cdot (m_y^2 + m_z^2) + c_2 \cdot p^2 + c_3 \cdot (m_y^2 \cdot p^2 + m_z^2 \cdot p^2) + c_4 \cdot m_y^2 \cdot m_z^2 = 1.0$$

$$p = \frac{P - \phi \cdot P_o}{P_o} \quad m_y = \frac{M_y}{M_{y0}} \quad m_z = \frac{M_z}{M_{z0}}$$

$$\phi = \frac{P}{P_o} @ M_{\max} = \frac{(A_c \cdot f'_c + A_s \cdot f_r) / 2}{P_o}$$

Coefficient Equations:

$$c_1 = 1.08 - 0.00265 \cdot x + 0.0000230 \cdot x^2 - 1.13 \cdot 10^{-7} \cdot x^3 + 0.374 \cdot y - 1.30 \cdot y^2 - 0.0419 \cdot y^3 - 0.0691 \cdot x \cdot y + 0.000234 \cdot x^2 \cdot y + 0.0754 \cdot x \cdot y^2$$

$$c_2 = 0.628 + 0.0259 \cdot x - 0.000367 \cdot x^2 + 1.99 \cdot 10^{-6} \cdot x^3 + 4.50 \cdot y - 14.9 \cdot y^2 + 22.4 \cdot y^3 + 0.164 \cdot x \cdot y - 0.000756 \cdot x^2 \cdot y - 0.126 \cdot x \cdot y^2$$

$$c_3 = 0.420 + 0.0892 \cdot x - 0.00122 \cdot x^2 + 5.13 \cdot 10^{-6} \cdot x^3 + 4.90 \cdot y - 16.5 \cdot y^2 + 16.2 \cdot y^3 - 0.165 \cdot x \cdot y + 0.000713 \cdot x^2 \cdot y + 0.120 \cdot x \cdot y^2$$

$$c_4 = 0.346 + 0.00912 \cdot x - 0.000127 \cdot x^2 + 4.98 \cdot 10^{-7} \cdot x^3 - 0.391 \cdot y + 4.55 \cdot y^2 - 10.3 \cdot y^3 - 0.0592 \cdot x \cdot y + 0.000245 \cdot x^2 \cdot y + 0.0659 \cdot x \cdot y^2$$

$$x = \frac{D}{t} \quad y = \frac{f'_c}{f_y}$$

Table 2.10 Cumulative Errors in CFT Cross-Section Strength Surface Equation (Surface Equation vs. Fiber Element Analysis)

D/t	f'_c (ksi)	Standard Deviation	Average Error	Maximum Positive Error	Maximum Negative Error
24	3.5	3.11 %	-0.20 %	5.67 %	-8.57 %
	6.5	2.80 %	0.59 %	6.00 %	-6.76 %
	10	2.75 %	-1.38 %	5.37 %	-7.55 %
	15	2.77 %	0.16 %	5.84 %	-7.58%
48	3.5	2.59 %	0.88 %	5.18 %	-6.35 %
	6.5	2.55 %	-0.25 %	4.63 %	-6.48 %
	10	3.33 %	0.99 %	7.85 %	-6.93 %
	15	4.57 %	-0.39 %	9.28 %	-9.87 %
72	3.5	2.33 %	-0.22 %	4.92 %	-5.90 %
	6.5	2.29 %	0.87 %	4.47 %	-5.83 %
	10	2.59 %	-0.60 %	4.70 %	-6.74 %
	15	4.28 %	-1.83 %	5.55 %	-9.05 %
96	3.5	2.19 %	-0.06 %	4.88 %	-5.36 %
	6.5	1.96 %	1.04 %	4.44 %	-4.74 %
	10	2.19 %	-1.46 %	3.70 %	-5.97 %
	15	4.01 %	1.89 %	10.14 %	-7.00 %

Table 2.11 Cross-Section Strength Surface Equation Errors for Tension Region

D/t	f'_c (ksi)	P_{tens}/P_o	P_{eq}/P_o	% error
24	3.5	0.714	0.666	7.21 %
	6.5	0.574	0.534	7.49 %
	10	0.466	0.429	8.62 %
	15	0.368	0.330	11.52 %
48	3.5	0.539	0.521	3.45 %
	6.5	0.386	0.370	4.32 %
	10	0.290	0.268	8.21 %
	15	0.214	0.183	16.94 %
72	3.5	0.432	0.432	0.00 %
	6.5	0.291	0.288	1.04 %
	10	0.210	0.200	5.00 %
	15	0.151	0.130	16.15 %
96	3.5	0.361	0.373	-3.22 %
	6.5	0.233	0.240	-2.92 %
	10	0.165	0.163	1.23%
	15	0.117	0.101	15.84 %

Two-dimensional graphical representations of the cross-section strength surface equation are shown in Figs. 2.25 to 2.36. These figures compare the equation results to the p - m_x - m_y data points obtained from the fiber analysis of the cross-sections in Table 2.2. Figures 2.25 to 2.28 show P/P_o vs. M/M_o for each series as f'_c varies from 3.5 to 15 ksi. Figures 2.29 to 2.32 illustrate the same results for constant values of f'_c and D/t ratios ranging from 24 to 96. Figures 2.33 to 2.36 show selected plots of M_y/M_{y_o} vs. M_z/M_{z_o} for sections 24A, 48B, 72C, and 96D. In these last figures, each graph shows contours of M_y/M_{y_o} vs. M_z/M_{z_o} for different constant values of axial load ratio, P/P_o , varying from 0 to 1.

The accuracy of the three-dimensional cross-section strength surface equation was checked by examining sections with combinations of D/t and f'_c/f_y ratios that are different from the cross-sections in Table 2.2. These sections are all tubes that are available from manufacturers [AISC LRFD, 1994]. The errors between the surface equation and the fiber analysis for these cross-sections are tabulated in Table 2.12. Sections having intermediate values of f'_c and D/t were chosen as well as sections with the same D/t as sections from the Table 2.2 but with a larger or smaller cross-sectional area. All of these sections produced acceptable results.

Table 2.12 CFT Cross-Section Strength Surface Equation Errors:
Additional CFT Cross-Sections

Section (D/t ratio)	f'_c (ksi)	Standard Deviation	Average	Maximum Positive Error	Maximum Negative Error
$6 \times 6 \times \frac{1}{4}$ (24)	8.0	2.71 %	0.14 %	5.93 %	-6.74 %
$16 \times 16 \times \frac{1}{2}$ (32)	12.5	3.79 %	-0.39 %	7.45 %	-9.67 %
$14 \times 14 \times \frac{3}{8}$ (37.3)	5.0	2.83 %	-1.03 %	5.57 %	-7.59 %
$24 \times 24 \times \frac{1}{2}$ (48)	15.0	4.50 %	-0.24 %	9.28 %	-9.87 %
$28 \times 28 \times \frac{1}{2}$ (56)	12.5	3.67 %	0.22 %	7.49 %	-7.70 %
$24 \times 24 \times \frac{3}{8}$ (64)	5.0	2.27 %	0.50 %	4.74 %	-5.61 %
$30 \times 30 \times \frac{3}{8}$ (80)	8.0	2.11 %	0.34 %	4.24 %	-5.95 %

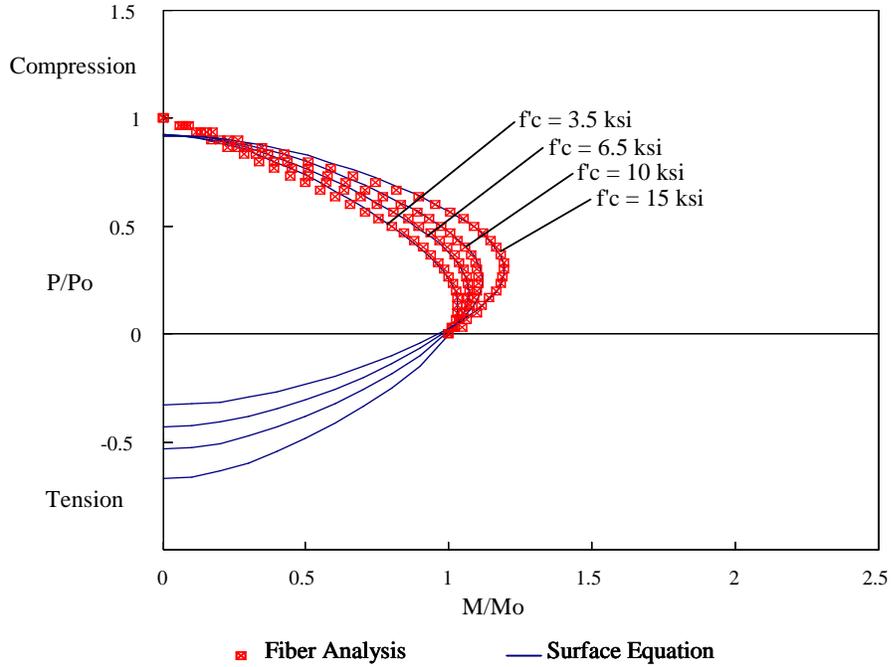


Figure 2.25 Two-Dimensional Cross-Section Strength Diagrams ($D/t = 24$)

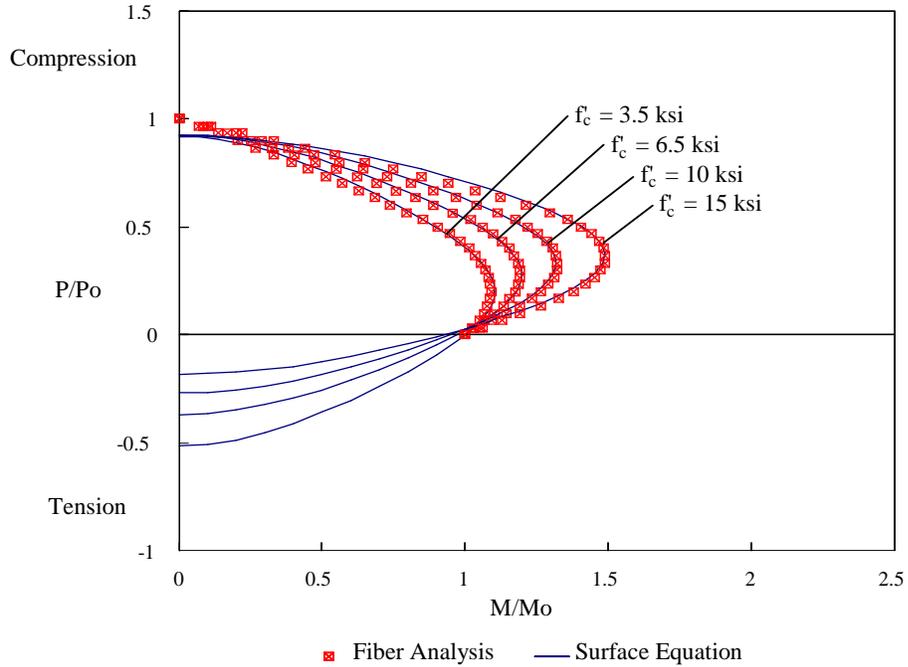


Figure 2.26 Two-Dimensional Cross-Section Strength Diagrams ($D/t = 48$)

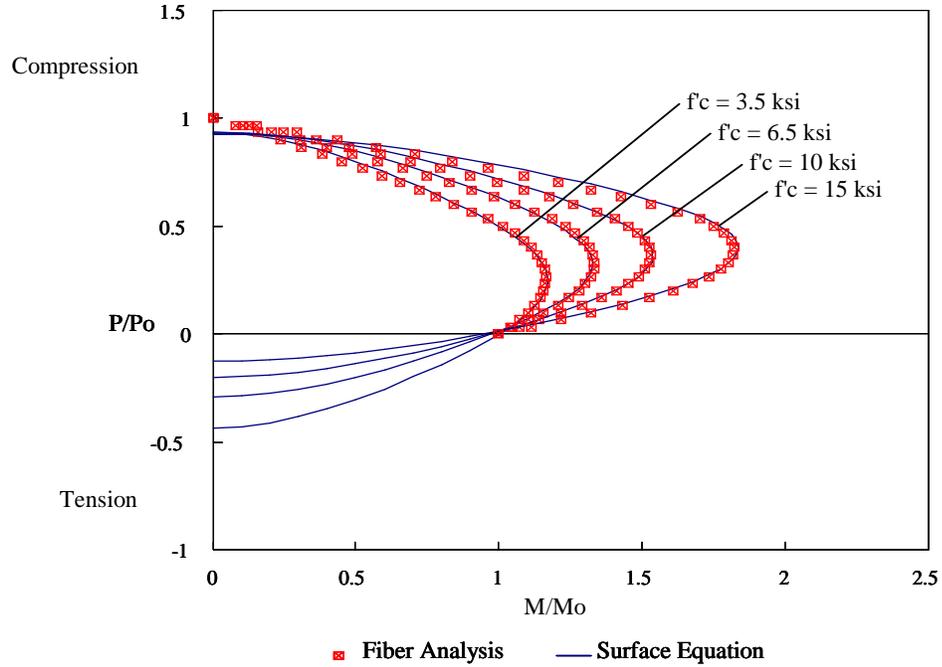


Figure 2.27 Two-Dimensional Cross-Section Strength Diagrams ($D/t = 72$)

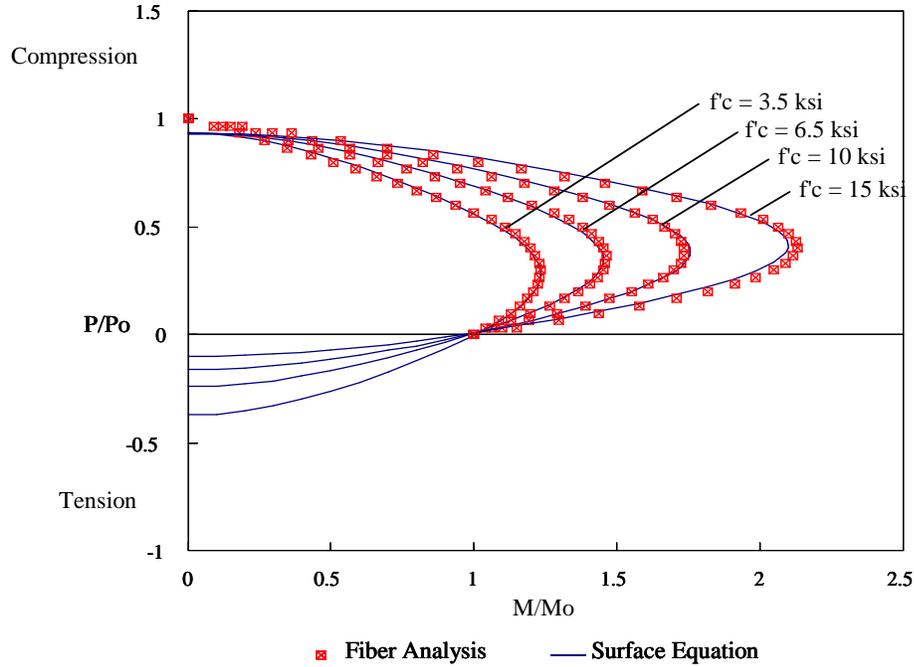


Figure 2.28 Two-Dimensional Cross-Section Strength Diagrams ($D/t = 96$)

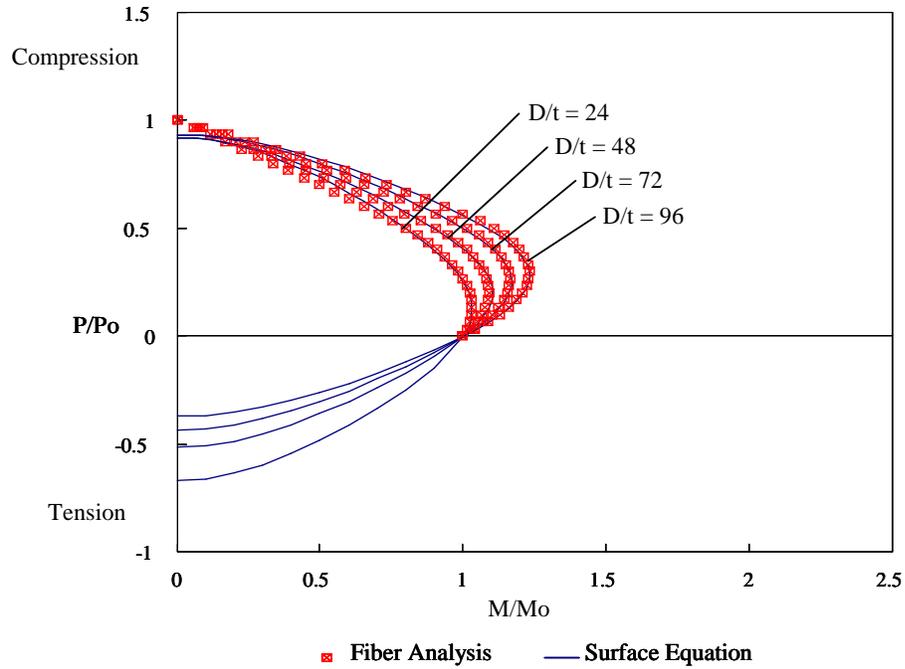


Figure 2.29 Two-Dimensional Cross-Section Strength Diagrams ($f'_c = 3.5$ ksi)

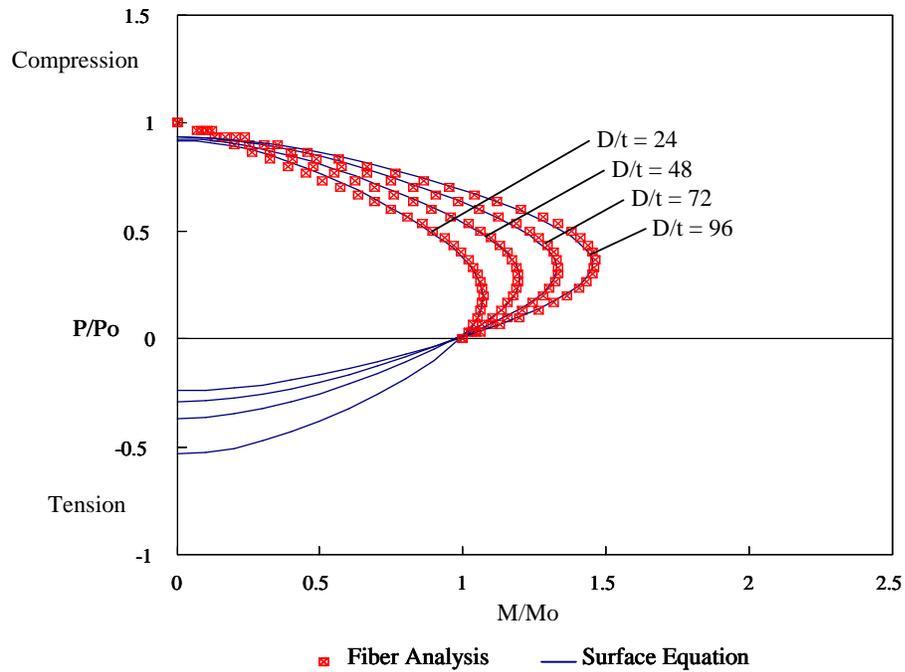


Figure 2.30 Two-Dimensional Cross-Section Strength Diagrams ($f'_c = 6.5$ ksi)

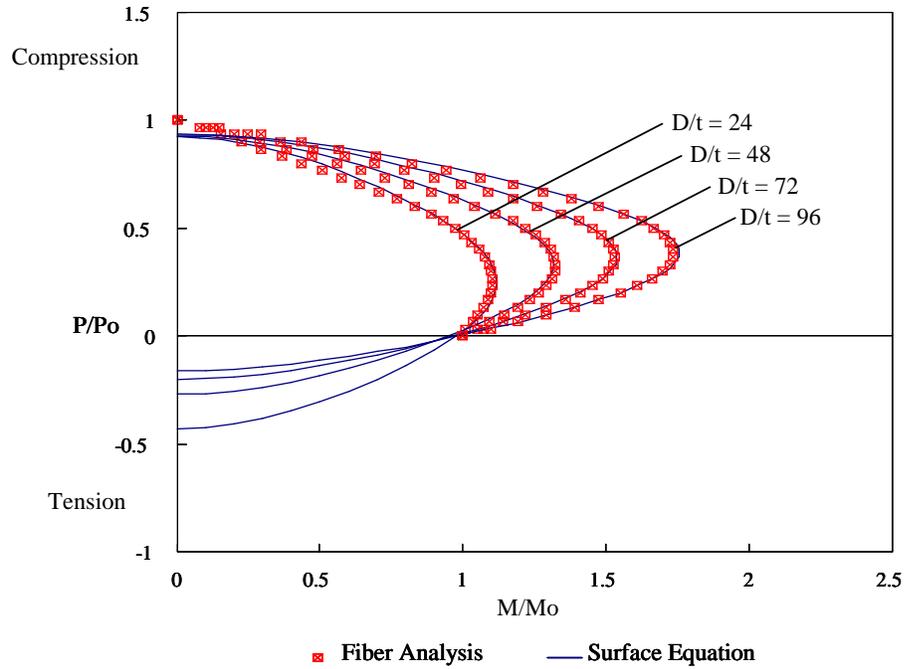


Figure 2.31 Two-Dimensional Cross-Section Strength Diagrams ($f'_c = 10$ ksi)

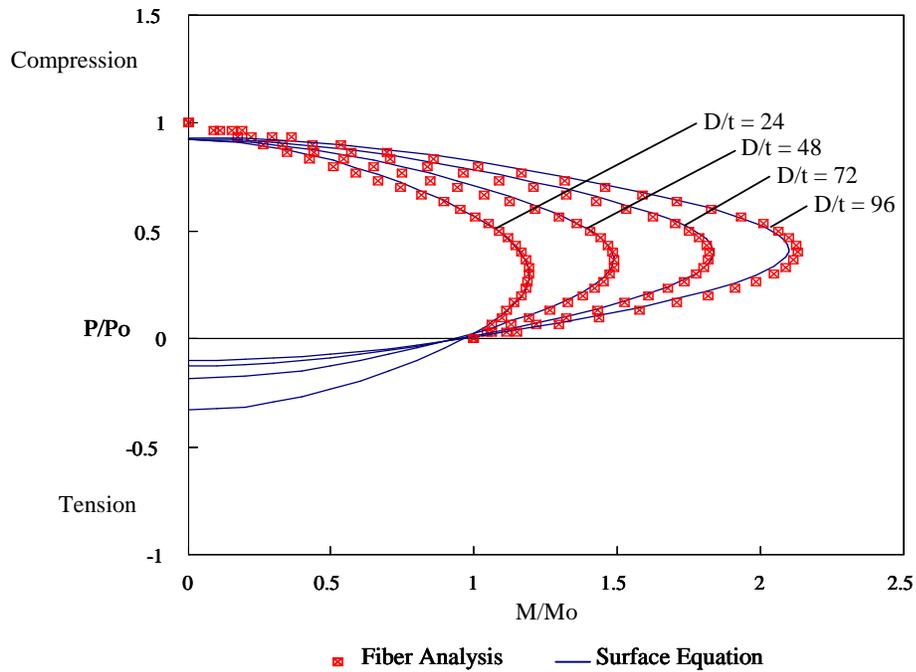


Figure 2.32 Two-Dimensional Cross-Section Strength Diagrams ($f'_c = 15$ ksi)

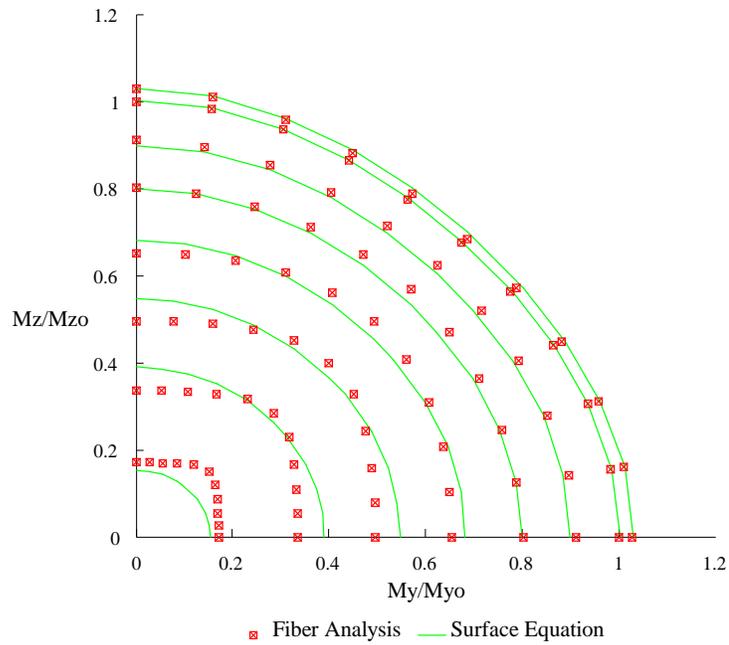


Figure 2.33 CFT Cross-Section Strength Diagrams-- M_y/M_{y0} vs. M_z/M_{z0}
Section 24A ($D/t = 24$, $f'_c = 3.5$ ksi)

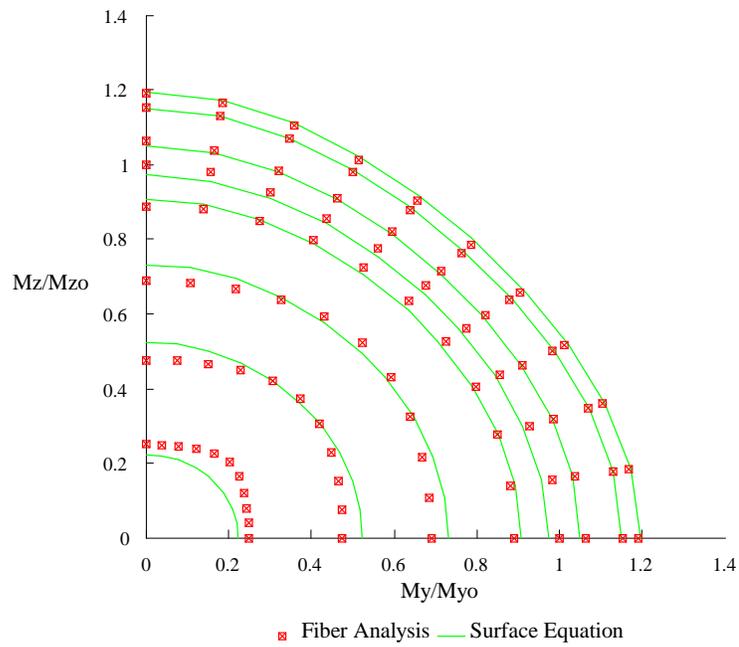


Figure 2.34 CFT Cross-Section Strength Diagrams-- M_y/M_{y0} vs. M_z/M_{z0}
Section 48B ($D/t = 48$, $f'_c = 6.5$ ksi)

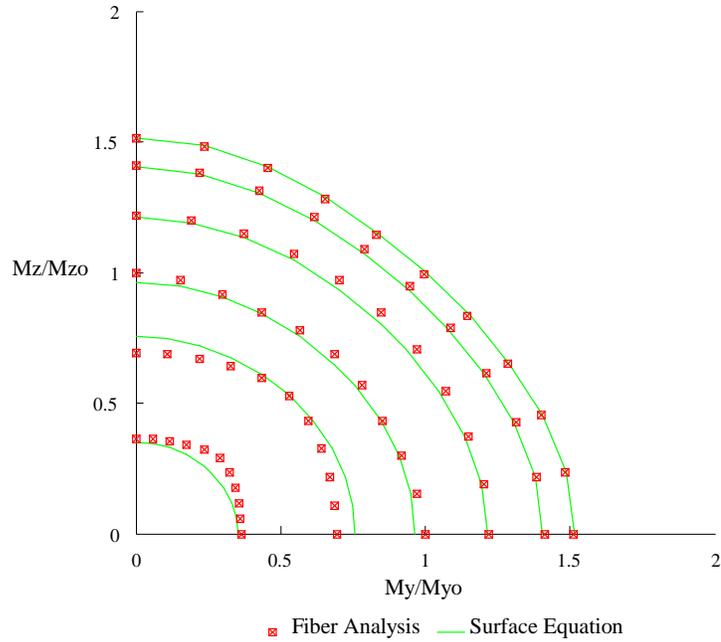


Figure 2.35 CFT Cross-Section Strength Diagrams-- M_y/M_{y0} vs. M_z/M_{z0}
Section 72C ($D/t = 72$, $f'_c = 10$ ksi)

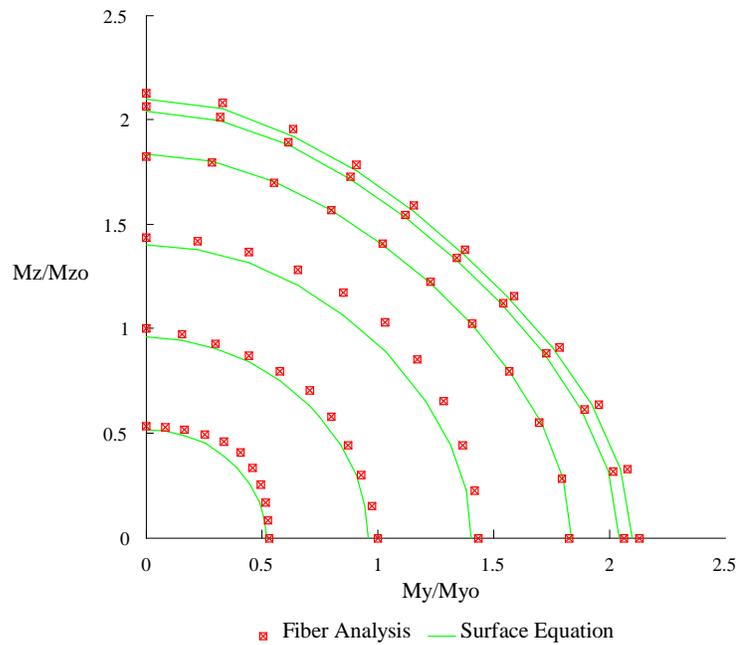


Figure 2.36 CFT Cross-Section Strength Diagrams-- M_y/M_{y0} vs. M_z/M_{z0}
Section 96D ($D/t = 96$, $f'_c = 15$ ksi)

2.6.5 Rectangular CFTs with Aspect Ratios Greater Than 1:1

The cross-section strength surface equation developed above for square CFT cross-sections should be accurate for rectangular sections as well. The equations of Table 2.9 may be used for rectangular cross-sections by modifying the calculation of the coefficient values to account for the different D/t ratio in the major axis direction and the minor axis direction. Each coefficient value for a rectangular cross-section is computed by averaging two coefficient values: the coefficient value obtained using the major axis D/t ratio in the coefficient equations of Table 2.9 and the value obtained using the minor axis D/t ratio. This approach produces slightly smaller errors than averaging the major axis and minor axis D/t ratios and using this average D/t value in the coefficient equations of Table 2.9.

The errors using average coefficients for seven tested sections with aspect ratios between 1:1 and 2:1 are shown in Table 2.13. The errors are greater for the larger aspect ratios. Although some of the maximum errors exceed 10 %, the average error plus or minus one standard deviation is, except for one case, always less than 10%. Figures 2.37 to 2.45 illustrate a comparison of the surface equation to the fiber analysis results for selected sections. Figures 2.37, 2.38, 2.40, 2.41, 2.43, and 2.44 show P/P_o vs. M_z/M_{zo} (major axis) and P/P_o vs. M_y/M_{yo} (minor axis) plots, and Figs. 2.39, 2.42, and 2.45 show plots of M_y/M_{yo} vs. M_z/M_{zo} .

Table 2.13 CFT Cross-Section Strength Surface Equation Errors:
Rectangular Cross-Sections

Section (D/t ratios: major axis, minor axis)	f'_c (ksi)	Standard Deviation	Average	Maximum Positive Error	Maximum Negative Error
$12 \times 6 \times \frac{1}{4}$ (48, 24)	3.5	4.80 %	-3.83 %	7.17 %	-14.14 %
$12 \times 6 \times \frac{1}{4}$ (48, 24)	10	5.02 %	-5.17 %	5.74 %	-16.16 %
$18 \times 9 \times \frac{1}{4}$ (72, 36)	6.5	4.48 %	-4.26 %	5.73 %	-13.49 %
$18 \times 9 \times \frac{1}{4}$ (72, 36)	15	5.10 %	-2.06 %	7.90 %	-14.40 %
$20 \times 12 \times \frac{5}{16}$ (64, 38)	15	5.14 %	-1.69 %	7.59 %	-14.38 %
$30 \times 24 \times \frac{3}{8}$ (80, 64)	6.5	2.43 %	0.21 %	4.20 %	-7.07 %
$36 \times 18 \times \frac{3}{8}$ (96, 48)	10	4.38 %	-3.47 %	4.54 %	-12.88 %

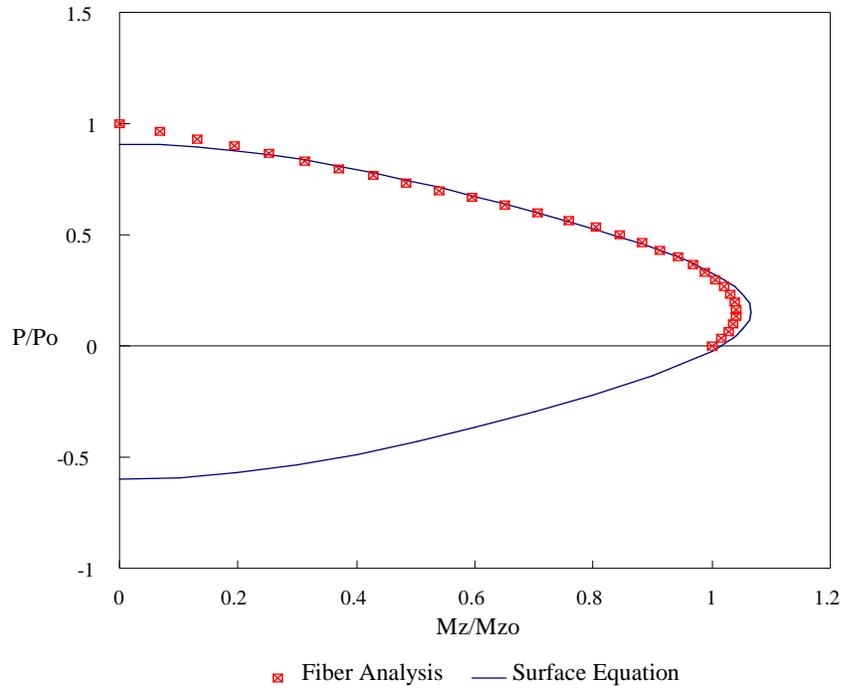


Figure 2.37 Two-Dimensional Cross-Section Strength Diagram--Major Axis Moment
(Major Axis $D/t = 48$, Minor Axis $D/t = 24$, $f'_c = 3.5$ ksi)

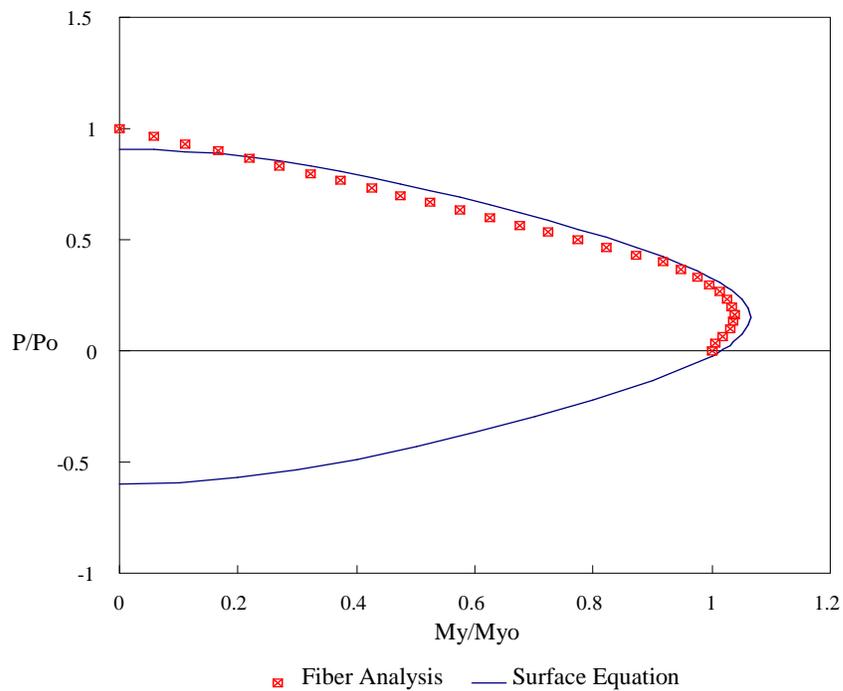


Figure 2.38 Two-Dimensional Cross-Section Strength Diagram--Minor Axis Moment
(Major Axis $D/t = 48$, Minor Axis $D/t = 24$, $f'_c = 3.5$ ksi)

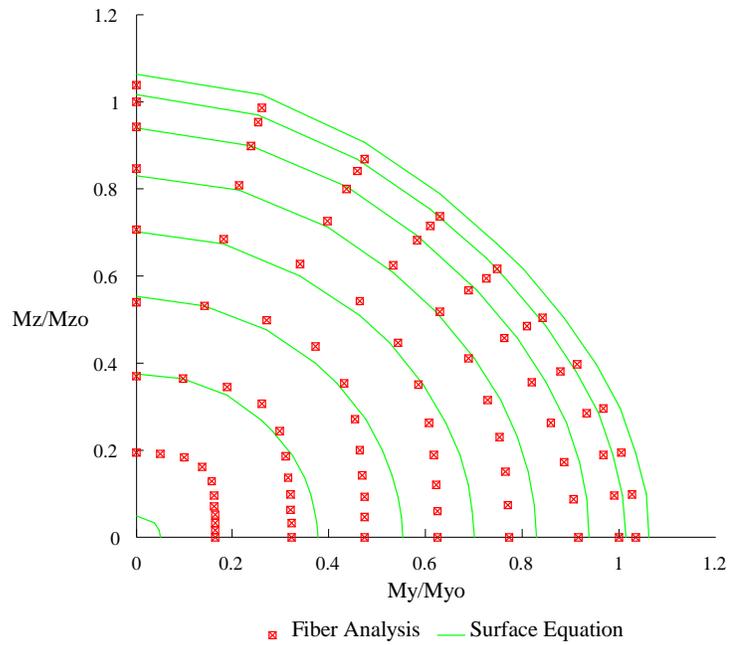


Figure 2.39 CFT Cross-Section Strength Diagrams-- M_y/M_{y0} vs. M_z/M_{z0}
(Major Axis $D/t = 48$, Minor Axis $D/t = 24$, $f'_c = 3.5$ ksi)

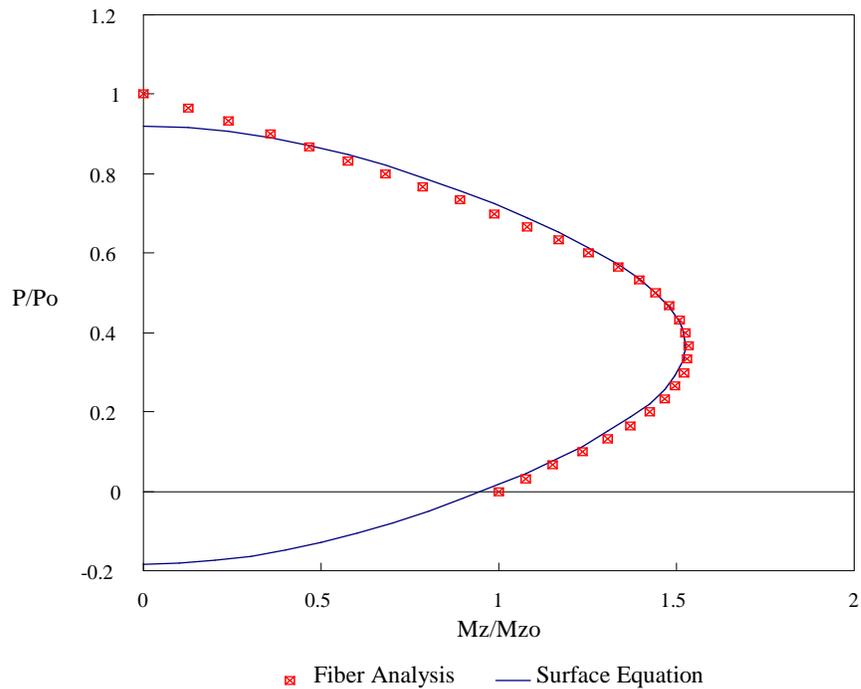


Figure 2.40 Two-Dimensional Cross-Section Strength Diagram--Major Axis Moment
(Major Axis $D/t = 72$, Minor Axis $D/t = 36$, $f'_c = 15$ ksi)

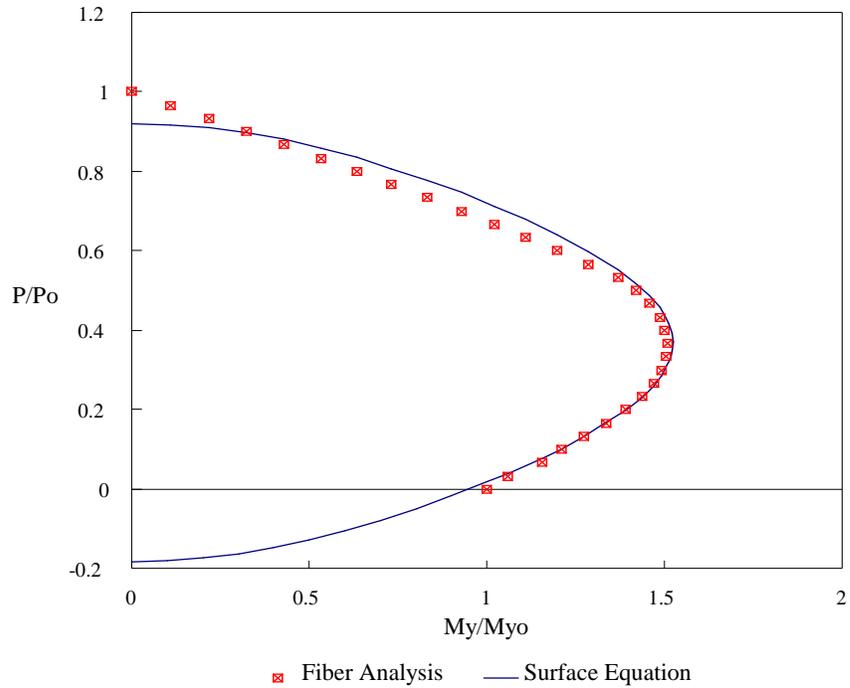


Figure 2.41 Two-Dimensional Cross-Section Strength Diagram--Minor Axis Moment
(Major Axis $D/t = 72$, Minor Axis $D/t = 36$, $f'_c = 15$ ksi)

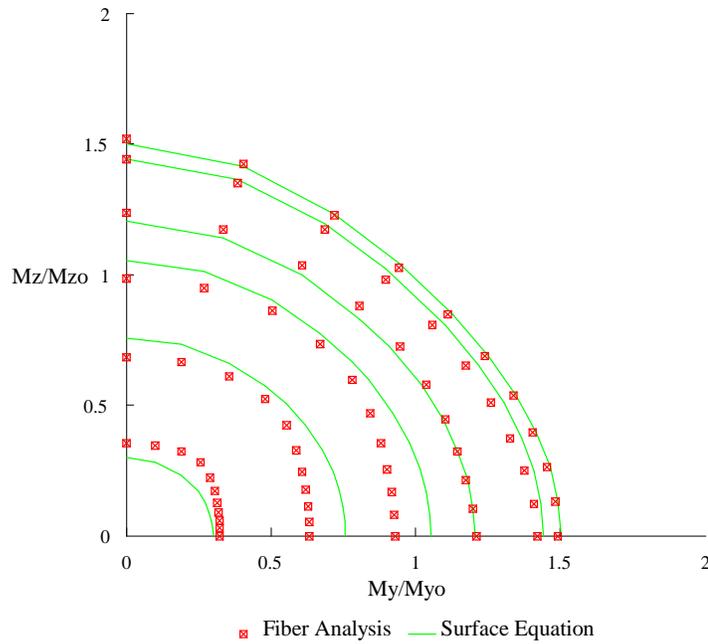


Figure 2.42 CFT Cross-Section Strength Diagrams-- M_y/M_{y0} vs. M_z/M_{z0}
(Major Axis $D/t = 72$, Minor Axis $D/t = 36$, $f'_c = 15$ ksi)

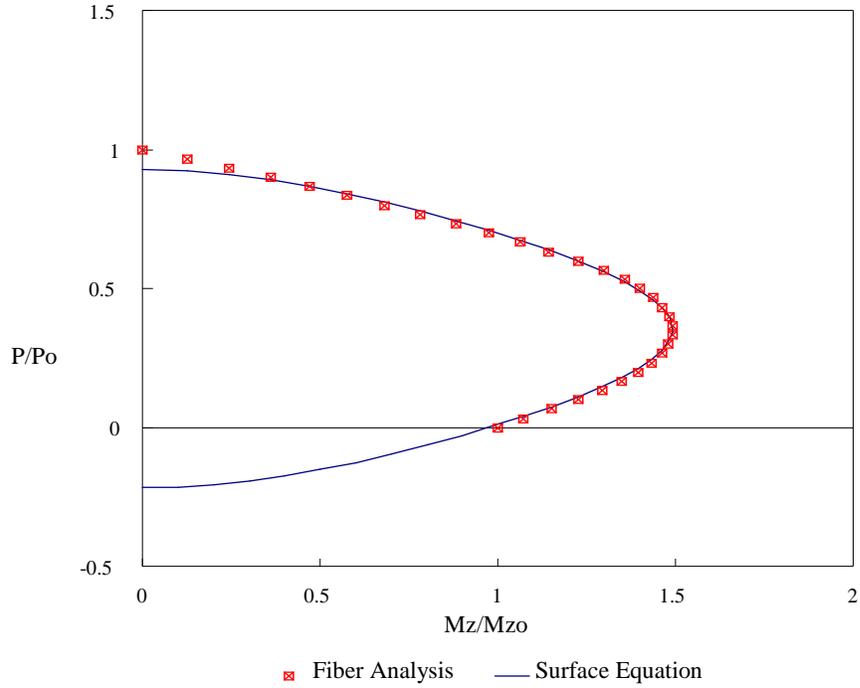


Figure 2.43 Two-Dimensional Cross-Section Strength Diagram--Major Axis Moment
 (Major Axis $D/t = 96$, Minor Axis $D/t = 48$, $f'_c = 10$ ksi)

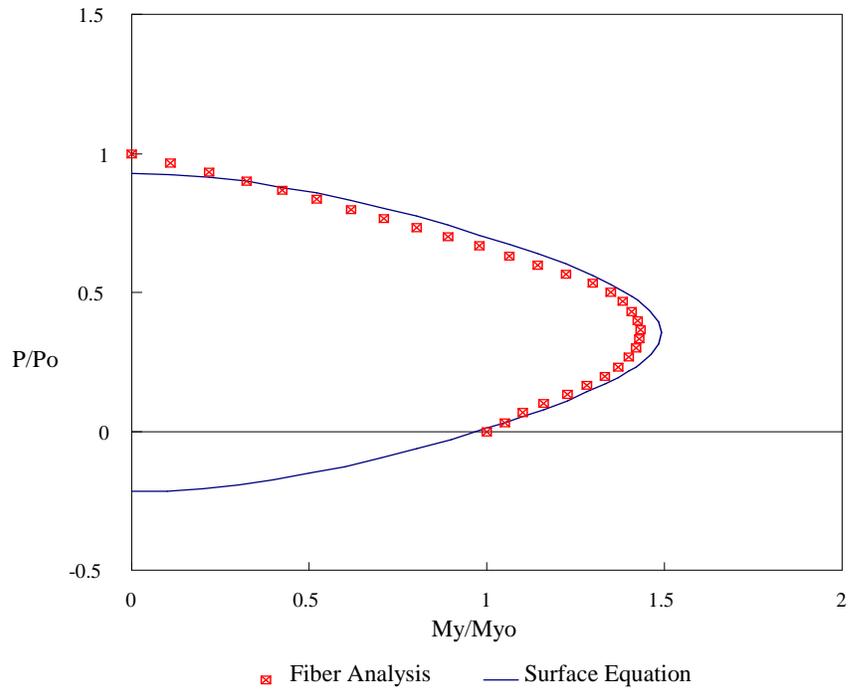


Figure 2.44 Two-Dimensional Cross-Section Strength Diagram--Minor Axis Moment
 (Major Axis $D/t = 96$, Minor Axis $D/t = 48$, $f'_c = 10$ ksi)

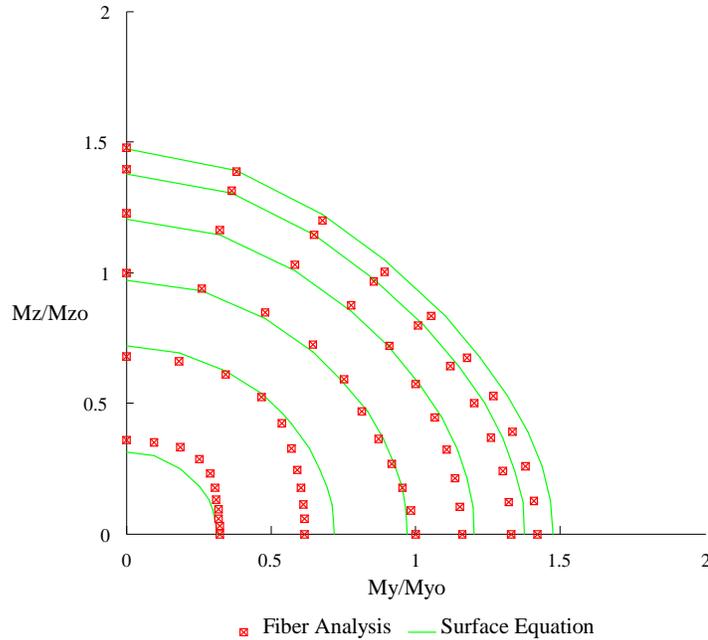


Figure 2.45 CFT Cross-Section Strength Diagrams-- M_y/M_{y0} vs. M_z/M_{z0}
(Major Axis $D/t = 96$, Minor Axis $D/t = 48$, $f'_c = 10$ ksi)

2.6.6 CFTs with Higher Strength Steel

Several sections were tested to check the validity of the f'_c/f_y ratio used as a variable parameter in the formulation of the cross-section strength surface equation. Specifically, the equation should accurately predict the cross-section strength surface for a constant value of the f'_c/f_y ratio with an increasing value of f_y . Table 2.14 shows the results of five tested sections with steel yield strengths of up to 70 ksi. The errors were comparable to those obtained with the original formulation using 46 ksi steel.

Table 2.14 CFT Cross-Section Strength Surface Equation Errors:
CFT Cross-Sections with Higher Strength Steel

Section (D/t ratio)	f_y (ksi)	f'_c (ksi)	Standard Deviation	Average	Maximum Positive Error	Maximum Negative Error
$12 \times 12 \times \frac{1}{2}$ (24)	50	10.87	2.72 %	-0.84 %	5.13 %	-7.84 %
$12 \times 12 \times \frac{1}{2}$ (24)	70	15.22	4.13 %	3.61 %	7.72 %	-7.30 %
$18 \times 18 \times \frac{3}{8}$ (48)	60	8.48	3.43 %	-0.96 %	5.54 %	-9.05 %
$27 \times 27 \times \frac{3}{8}$ (72)	60	13.02	4.50 %	0.70 %	8.87 %	-8.08 %
$36 \times 36 \times \frac{3}{8}$ (96)	70	5.33	3.37 %	-0.73 %	4.52 %	-8.17 %

2.6.7 Nominal Strength of CFTs

The final requirement necessary to complete the equation to represent the three-dimensional cross-section strength of a CFT member is a method for calculating the nominal axial and bending moment capacity of the section. In the above procedure, these values were calculated for each analysis by the fiber element procedure. For implementation into the concentrated plasticity model to follow, however, these values must be computed explicitly by a simple yet accurate set of equations.

The nominal axial load capacity may be calculated directly as it is done in the fiber analysis program. Thus, the axial load capacity is simply the sum of the strengths of the two constituent materials [Tomii and Sakino, 1979a, 1979b]:

$$P_o = A_s \cdot f_y + A_c \cdot f'_c \quad (2.26)$$

The calculation of the nominal moment capacity of the section (i.e., the maximum moment in the presence of no axial load) requires a more complicated formulation. In the fiber element analysis, the nominal moment is calculated iteratively by the moment-curvature analysis procedure (Section 2.2) for the case when the axial load ratio is zero. In the simplified formulation that follows, the moment resistance of the concrete is calculated by assuming stress blocks equal to 0.85 times the compressive area and 0.50 times the tensile area, a procedure similar in some respects to that used in the analysis of reinforced concrete beams. The addition of the tensile moment capacity is included for the following reason. For larger CFT sections with a high D/t ratio and high concrete strength, the neutral axis of the section will be very close to the top fiber of the section and a large portion of the concrete will be in tension. Although the concrete strength in tension is only approximately one-tenth of the compression strength, the large percentage of concrete in tension will nevertheless have a significant effect on the moment resistance of the CFT. Neglecting this effect produced significantly less accurate results. The nominal moment resistance of the steel is computed assuming the entire cross-section has reached the yield stress.

Because the concrete in tension and the concrete in compression have different strengths, the neutral axis of the section will not be at the centroid and must be calculated prior to the moment calculation. For a rectangular CFT cross-section with width b and depth D , the neutral axis, x_n (measured from the top fiber of the cross-section), is computed by the following equation:

$$x_n = \frac{0.85 \cdot f'_c \cdot (b \cdot t - 2 \cdot t^2) + 0.5 \cdot f_r \cdot (D - t) \cdot (b - 2 \cdot t) + f_y \cdot (2 \cdot D \cdot t)}{0.85 \cdot f'_c \cdot (b - 2 \cdot t) + 0.5 \cdot f_r \cdot (b - 2 \cdot t) + f_y \cdot (4 \cdot t)} \quad (2.27)$$

The maximum tensile strength of the concrete, f_r , is given by Eq. (2.12). Given the neutral axis, the nominal moment for a general rectangular CFT cross-section may be calculated by:

$$M_o = 0.85 \cdot f'_c \cdot \left[0.5 \cdot (b - 2 \cdot t) \cdot (x_n - t)^2 \right] + 0.5 \cdot f_r \cdot \left[0.5 \cdot (b - 2 \cdot t) \cdot (D - x_n - t)^2 \right] + f_y \cdot \left[(2 \cdot t) \cdot \left(\frac{D^2}{2} + x_n^2 + t^2 - D \cdot t - D \cdot x_n \right) + (b \cdot t) \cdot (D - t) \right] \quad (2.28)$$

The results of the moment computations were compared to the results from the fiber element analysis for the 16 cross-sections of Table 2.2, and to theoretical results presented by Chen and Chen [1973]. The results of Eqs. (2.27) and (2.28) are illustrated in Table 2.15. The majority of the computed nominal moments are within 5.0 % of the fiber element and the theoretical results.

Table 2.15 Nominal Moment Calculation Errors

Cross-Section	f_y (ksi)	f'_c (ksi)	f_{ct} (ksi)	M_o (Calc.)	M_o (Theory)*	% error
Chen, '73 3×3×0.129	47	5.9	0.58	84.6	81.5	3.85%
24A 12×12× $\frac{1}{2}$	46	3.5	0.44	5988	5018	-0.59%
24B 12×12× $\frac{1}{2}$	46	6.5	0.61	5215	5302	-1.64%
24C 12×12× $\frac{1}{2}$	46	10	0.75	5405	5669	-4.66%
24D 12×12× $\frac{1}{2}$	46	15	0.92	5598	5944	-5.82%
48A 18×18× $\frac{3}{8}$	46	3.5	0.44	9414	9466	-0.54%
48B 18×18× $\frac{3}{8}$	46	6.5	0.61	9997	10 275	-2.70%
48C 18×18× $\frac{3}{8}$	46	10	0.75	10 440	10 910	-4.28%
48D 18×18× $\frac{3}{8}$	46	15	0.92	10 880	11 710	-7.07%
72A 27×27× $\frac{3}{8}$	46	3.5	0.44	22 630	22 716	-0.38%
72B 27×27× $\frac{3}{8}$	46	6.5	0.61	24 230	24 949	-2.90%
72C 27×27× $\frac{3}{8}$	46	10	0.75	25 440	26 762	-4.96%
72D 27×27× $\frac{3}{8}$	46	15	0.92	26 640	28 316	-5.90%
96A 36×36× $\frac{3}{8}$	46	3.5	0.44	42 290	42 814	-1.22%
96B 36×36× $\frac{3}{8}$	46	6.5	0.61	45 560	47 303	-3.68%
96C 36×36× $\frac{3}{8}$	46	10	0.75	48 070	50 564	-4.94%
96D 36×36× $\frac{3}{8}$	46	15	0.92	50 630	53 418	-5.22%

*Theoretical nominal moments refer to either moments provided by other authors (e.g. Chen, '73) or the moments computed using the fiber element analysis (e.g. Section 24A).

Chapter 3

Introduction to the CFT Nonlinear Model

This chapter presents an introduction to the nonlinear analytical formulation to model the load-deflection behavior of frame structures composed of CFT beam-columns and steel I-beams and subjected to either monotonic or cyclic loading. The first section of this chapter discusses the behavior of rectangular CFT beam-columns, providing an introduction to the characteristics modeled by the analytical procedure, which is discussed in subsequent sections. The analytical procedure is examined with respect to each component of the nonlinear beam-column model--the elastic stiffness, the geometric stiffness, and the plastic reduction stiffness. The section on the plastic reduction stiffness introduces the concentrated plasticity model, the topic of Chapter 4.

3.1 Behavior of Rectangular CFT Beam-Columns

The primary loads on a CFT beam-column in a frame structure will be a combination of axial compression and uniaxial or biaxial bending. The behavior of CFT

beam-columns under a given set of loads will depend upon the member length, the strength of the concrete and steel, and the ratio of the concrete area to the steel area. Additionally, the behavior of CFT beam-columns is largely influenced by the pattern of applied loading--monotonic or cyclic.

Chapter 2 examined the behavior of short beam-columns, members governed exclusively by cross-section behavior. These sections are characterized by an ultimate strength that is independent of the effect of member instability. For longer beam-columns, stability becomes an important factor in the behavior of the member. Depending upon the state of stresses at failure, beam-columns may be further classified as long (or slender)--beam-columns that reach their capacity when the column fails due primarily to elastic instability [Shakir-Khalil and Zeghiche, 1989], or intermediate--beam-columns that undergo partial plastification of the cross-section at some point along the length of the beam-column and fail due to a combination of geometric and material nonlinearity [Bridge, 1976; Cederwall et al., 1990].

In most CFT applications, especially frame structures, the CFT member will be of a length sufficient to induce lateral instability at medium to high axial loads. The following sections investigate the monotonic and cyclic strength and stiffness of intermediate and long beam-columns typical of frame structures. The discussion highlights the behavioral characteristics of beam-columns that will be modeled by the subsequent analytical formulation.

3.1.1 Monotonic Behavior of Rectangular CFT Beam-Columns

The behavior of CFT beam-columns is, in many respects, similar to CFT cross-section behavior (see Chapter 2). The main exception is the aspect of length, which imparts two primary effects on CFT beam-column behavior. As the length increases,

geometric nonlinearities such as the $P-\delta$ effect (member curvature) and the $P-\Delta$ effect (member chord rotation) [Galambos, 1988] cause the forces in the member to be amplified. The length will also affect the degree to which the concrete is confined by the steel tube, directly affecting the material properties of the CFT. For long beam-columns, overall member buckling will occur at longitudinal strains below the level at which volumetric expansion of the concrete begins, preventing any increase in concrete strength and ductility [Tomii et al., 1973]. Long CFT beam-columns are therefore undesirable in practice because the material capacity of the CFT is largely underutilized due to early failure by instability [Shakir-Khalil and Mouli, 1990].

Failure of an intermediate length CFT beam-column generally results from a combination of steel yielding and concrete crushing, which leads to flexural buckling [Bridge, 1976; Shakir-Khalil and Zeghiche, 1989]. CFT columns of intermediate length undergo a transfer of the load resistance from the concrete to the steel as the moment on the section is increased. Bridge [1976] calculated the percent strength contribution of the concrete core for a CFT with a D/t ratio of 20.0. The concrete core provides only 7.5% of the capacity in a member subjected to pure bending versus 30% of the capacity of a member subjected to pure axial load. Although this CFT contains a relatively large proportion of steel, the results underscore the increased usefulness of CFTs for members subjected to moderate axial loads.

Figure 3.1 illustrates the axial load versus mid-height deflection curve for a simply-supported CFT beam-column subjected to an axial load applied at an eccentricity, e . The stiffness of the CFT, even under small loads, is nonlinear due to early tensile cracking of the concrete, (A). The beam-column remains fairly stiff, though, until the steel tube begins to yield. (B). As yielding progresses to both the compression and tension flanges of the beam-column, (C), the beam-column stiffness degrades severely

and the concrete core begins to crush, eventually leading to an instability failure, (D) [Bridge, 1976; Shakir-Khalil and Mouli, 1990; Cederwall et al., 1990].

The interaction between the concrete and the steel and the inherently ambiguous stiffness properties of concrete complicate the calculation of stiffness properties for CFTs. The bending stiffness, $E \cdot I$, and the axial stiffness, $E \cdot A$, are well known for steel, but these properties are difficult to predict for CFTs because of the inhomogeneity of concrete. Section 3.3 details the formulations that were used in this work for these elastic stiffness parameters.

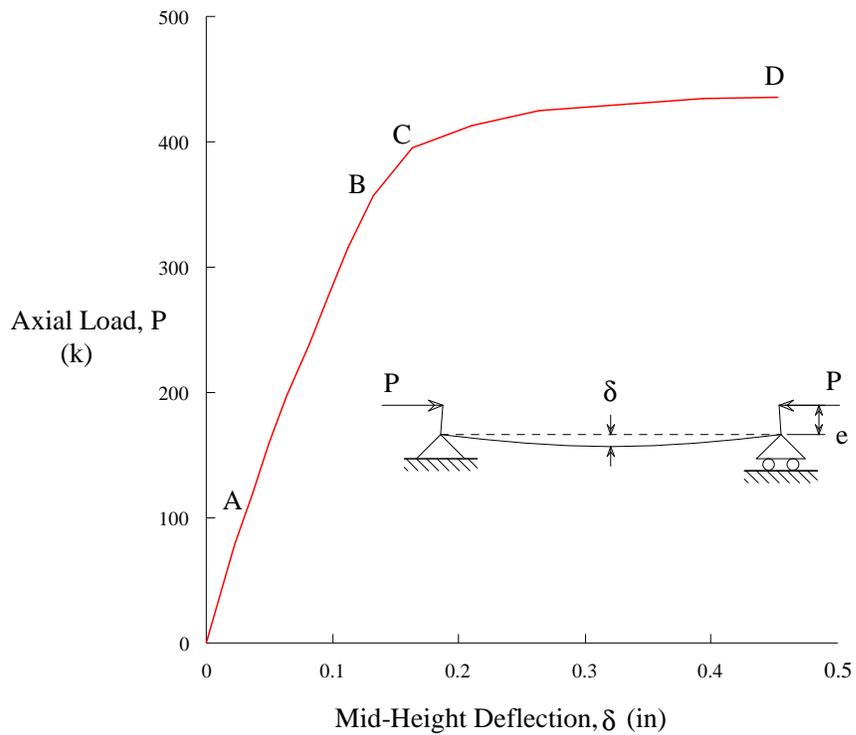


Figure 3.1 Typical CFT Load-Deflection Curve

3.1.2 Cyclic Behavior of Rectangular CFT Beam-Columns

Many of the benefits and advantages of CFTs are realized when these members are subjected to cyclic loading. The addition of concrete to a hollow tube significantly improves the cyclic behavior of the member [Matsui and Tsuda, 1987]. The presence of the concrete leads to an increase in the capacity of the section and greater ductility because the local buckling of the steel tube is delayed [Kawaguchi et al., 1991]. This ductility of CFTs is manifested in very full hysteresis loops, indicating a capacity to dissipate large amounts of energy. Additionally, CFT specimens exhibit some cyclic strain hardening, resulting in an increase in capacity before degradation due to local buckling and concrete crushing occurs [Sakino and Tomii, 1981].

The behavior of a CFT beam-column subjected to cyclic loading is most affected by the D/t ratio and the applied axial load ratio. The significance of the D/t ratio lies in the occurrence of local buckling of the steel tube. While the presence of the concrete will delay local buckling, tubes having high values of D/t or tubes subjected to large plastic displacements may undergo extensive local buckling. Combined with crushing of the concrete, local buckling will cause a degradation in strength and lead to eventual member failure [Sugano et al., 1992]. It has been observed by some researchers, however, that rectangular CFTs tend to behave as circular tubes after several cycles, as the buckling of the steel tube at the point of maximum force transforms the critical regions from rectangular to somewhat circular in shape [Sakino and Tomii, 1981; Kawaguchi et al., 1993]. Circular members have more stable hysteresis loops and a greater ductility than rectangular tubes due to confining effects. Therefore, this transformation of the rectangular tube geometry tends to stabilize the degrading hysteresis loops. As a result, CFTs often exhibit tough behavior, maintaining a high percentage of their initial capacity, even for relatively large cyclic displacements [Sugano et al., 1992; Kawaguchi et al., 1993].

The effect of the axial load ratio on the cyclic behavior of CFTs is much the same as the D/t ratio. For cyclically loaded specimens subjected to moderate to high axial loads, an increase in the axial load leads to a larger and more rapid strength degradation [Sakino and Tomii, 1981], resulting in less energy dissipation.

Experimental testing of cyclically-loaded rectangular beam-columns has been primarily limited to low- to medium-strength materials. The variable parameters in most tests have been member geometry of the specimen and axial load ratio. Sugano et al. [1992], however, have examined CFTs with concrete strengths from 4.5 ksi to 12.8 ksi and indicated that the hysteresis curves for square columns will be fuller as the strength of the concrete decreases. These results were based on tests performed by Yamaguchi et al. [1989], which were documented in Japanese and unavailable for this research.

3.2 CFT Element Formulation

3.2.1 Background on CFT Behavior Models

Much of the analytical research conducted on CFTs to date has focused on computing the ultimate capacity of members. In contrast, relatively few researchers have examined the comprehensive load-deflection behavior of CFTs. The numerical analysis methods for CFTs fall into three general categories: finite element "macro" model analyses, fiber element analyses, and three-dimensional continuum analyses. The former two methods are formulations in which each member is represented by one or more line elements, each having a specified number of degrees-of-freedom.

Finite element macro model analysis, as referred to in this work, is an analytical method in which the structural member is divided into one or more line elements along its length and deflections and forces are monitored only at the centroids of the cross-sections at the ends of each element. Macro models often incorporate simplifying assumptions such as perfect bond [Neogi et al., 1969; Chen and Chen, 1973; Bridge, 1976; Shakir-Khalil and Zeghiche, 1989; Masuo et al., 1991] and simplified uniaxial stress-strain curves for the steel and concrete [Neogi et al., 1969; Chen and Chen, 1973; Bridge, 1976; Shakir-Khalil and Zeghiche, 1989]. Load-deflection relationships may be formulated by using moment-curvature-thrust relationships to compute inelastic member response [Bridge, 1976]; by assuming a deflected beam-column shape and iteratively computing element forces to equilibrate the applied load for successive increments in displacement [Neogi et al., 1969; Shakir-Khalil and Zeghiche, 1989; Masuo et al., 1991]; or by using a parametric representation of moment and curvature which, in turn, may be used to compute deflections for a given loading [Chen and Chen, 1973].

The second type of analysis using line elements involves a finer discretization of each CFT. Each member is composed of a number of elements along its length, each of which are then, in turn, subdivided into a number of longitudinal fiber elements. This type of analysis allows stress and strain to be monitored at select points along the cross-section and along the member length, resulting in a more accurate model but a computationally more expensive model as well. Kawaguchi et al. [1993] used such a fiber element analysis to model the degrading stress-strain and load-displacement relationships due to local buckling of the steel. Formulation of this type of model for three-dimensional CFT beam-columns is part of ongoing research at the University of Minnesota.

The third general method used to model CFT behavior is three-dimensional continuum, or micro analysis. In this type of analysis, the member is divided into a three-

dimensional mesh of continuum finite elements, which facilitates a detailed examination of bond, initial imperfections, and residual stresses. Additionally, member inelasticity may be explicitly modeled at the multiaxial stress-strain level throughout the member using this method. Ge and Usami [1994] incorporated this type of model into an elasto-plastic finite displacement analysis of rectangular CFT stub columns.

3.2.2 Overview of the Finite Element Macro Model

For computational speed and efficiency, and to facilitate the analysis of entire frame structures, a macro finite element model is adopted in this work. The macro model is based on a standard twelve degree-of-freedom beam finite element (displacements for the twelve degrees-of-freedom are shown in Fig. 3.2) in which transverse displacements are obtained assuming cubic Hermetian shape functions [Weaver and Gere, 1990]. The finite element computer program developed in this research to model the load-deflection behavior of structures composed of a combination of CFT and steel I-beam members is based upon the direct stiffness approach and an incremental updated Lagrangian formulation. Incremental nonlinear finite element analysis has been described by many authors (see, for example, Bathe [1982]) and is only briefly presented here. In this work, for static analyses, the total applied load is divided into a series of increments, each applied in a load step. For transient time history analyses, the loading is divided into a series of time steps. While the program developed in this work can execute both static and dynamic time history analysis, the incremental formulation is presented in terms of static analysis, since there is little experimental evidence at present to verify dynamic analysis of structures containing CFTs. Additional detail on the computer implementation of the analytical procedure is provided in the appendix.



Figure 3.2 CFT Beam Element

The fundamental step in the analytical procedure is the formulation of the total stiffness matrix for each beam element of the structure. The stiffness, $[k^e]$, for a single structural element may be expressed by the following equation:

$$[k^e] = [k_e^e] + [k_g^e] + [k_r^e] \quad (3.1)$$

The three components of the total element stiffness given by Eq. (3.1) are the elastic stiffness, $[k_e^e]$, the geometric stiffness, $[k_g^e]$, and the plastic reduction stiffness, $[k_r^e]$.

Obtaining accurate displacements from the given set of applied loads is entirely dependent upon an accurate stiffness formulation. The following sections of this chapter examine each of the three stiffness components of Eq. (3.1) in turn. The elastic stiffness matrix of a structural element, $[k_e^e]$, has been very well documented for decades.

Nevertheless, effective elastic stiffness values which constitute the CFT stiffness matrix--axial rigidity, $E \cdot A$, bending rigidity, $E \cdot I$, and torsional rigidity, $G \cdot J$ --must be determined to model the composite elastic stiffness of the steel and concrete in a CFT.

The geometric stiffness matrix, $[k_g^e]$, incorporated in this research is based primarily upon the work of Orbison [1982]. The discussion of the geometric stiffness matrix in Section 3.4 will focus first on the background of the present formulation and second, on the behavioral aspects which the matrix models, namely the stiffness loss in a element due to the action of external loads acting through element displacements.

The third component of the stiffness formulation, the plastic reduction stiffness, $[k_r^e]$, forms the core of this research. Section 3.5 introduces the concentrated plasticity model for the formulation of the plastic reduction matrix.

3.3 Elastic CFT Stiffness Properties

The standard 12×12 elastic stiffness matrix for a three-dimensional beam element requires terms representing the element's axial, bending and torsional rigidity. For a steel element, these values are well known. For a CFT element, however, the interaction of the steel and the concrete obviates using a simple superposition of stiffnesses, which may not necessarily be an accurate representation of the CFT element stiffness. This section discusses the axial, bending, and torsional stiffnesses composing the elastic CFT matrix and presents formulas representative of the combined stiffnesses of the steel and the concrete. The formulation for each type of behavior developed assuming the elastic stiffness is decoupled from the nonlinear stiffness terms. The equations for the CFT stiffness terms presented in this section are based upon the results presented by Schiller et al. [1994].

3.3.1 Axial Rigidity

As discussed for CFT cross-sections in Chapter 2, the concrete and the steel sustain load independently of one another for small values of axial strain [Gardner and Jacobson, 1967; Neogi et al., 1969; Tsuji et al., 1991]. Confinement effects generally do not begin until near the point of steel yielding [Knowles and Park, 1970]. Therefore, in the elastic region, the behavior of a CFT (in this case, the axial stiffness of a CFT) may be accurately represented by summing the individual stiffness components of the steel and the concrete:

$$(E \cdot A)_{cft} = E_s \cdot A_s + E_c \cdot A_c \quad (3.2)$$

For CFTs under pure axial load, Eq. (3.2) is generally regarded as the most accurate model of axial rigidity [Neogi et al., 1969; Tomii et al., 1973; Zhong and Miao, 1987].

For cyclic load applications, the axial stiffness of a CFT will degrade due to concrete cracking as evidenced by the test results of Liu and Goel [1988]. In this work, to account for the reduction in elastic rigidity, the elastic modulus of the concrete, E_c , is reduced based on the amount of accumulated plastic work. The next section on flexural rigidity explains the reduction of E_c in more detail.

3.3.2 Flexural Rigidity

Tomii and Sakino [1979b] presented one of the first detailed formulations of a composite flexural stiffness for rectangular CFTs (see also [Bridge, 1976]). They presented a formula in which the flexural rigidity of a CFT is a summation of the rigidity of the steel tube and a reduced concrete rigidity which accounts for concrete cracking in tension:

$$(E \cdot I)_{cft} = E_s \cdot I_s + \beta \cdot E_c \cdot I_c \quad (3.3)$$

The reduction factor, β , is a function of the applied axial load, and is given by:

$$\beta = 0.311 + 0.619 \cdot \frac{P}{P_o} + 0.457 \cdot \left(\frac{P}{P_o} \right)^2 \quad (3.4)$$

For a larger applied axial load ratio, β and the corresponding stiffness given by Eq. (3.3) are larger due to the greater inhibition of concrete cracking over the cross-section. Tomii and Sakino conducted experimental tests for a range of values of D/t , f'_c , f_y , and P/P_o to validate their theoretical formulation. For CFT specimens with axial load ratios, P/P_o , of less than 0.5, Eqs. (3.3) and (3.4) predicted a composite stiffness within approximately 5% of the experimental results.

The matrix model contained in this work, however, requires that the linear elastic stiffness properties be independent of the loading, rendering Eqs. (3.3) and (3.4) inapplicable as expressed. Since β equals 0.3 (rounded to one significant digit) for the case of no axial load, an expression for flexural rigidity that is decoupled from axial force may be expressed as:

$$(E \cdot I)_{cft} = E_s \cdot I_s + 0.3 \cdot E_c \cdot I_c \quad (3.5)$$

The results of using Eq. (3.5) in the finite element model in this work produced an underestimation of the elastic stiffness exhibited by most experimental CFT beam-column tests. The primary reason for this, as discussed above, is the decreased concrete cracking due to the presence of moderate axial force in the CFT beam-columns in this study. In the Tomii and Sakino model, the presence of axial force results in a larger value

of β than was assumed in Eq. (3.5) for pure flexure. Therefore, an alternate stiffness formulation was employed in the elastic range of CFT behavior. This formulation consists of using the full stiffness of both the concrete and steel and is given by the following equation [Gardner and Jacobson, 1967; Neogi et al., 1969; Tomii et al., 1973; Liu and Goel, 1988]:

$$(E \cdot I)_{cft} = E_s \cdot I_s + E_c \cdot I_c \quad (3.6)$$

Equation (3.6) neglects the detrimental effect of concrete cracking on the overall CFT stiffness. In this work, extensive concrete cracking is assumed to become prevalent and significantly influence the CFT stiffness only after the onset of steel yielding. To account for concrete cracking beyond this point, the elastic modulus of the concrete, E_c , decreases toward a final value equal to 30% of its original value, resulting in a final CFT flexural stiffness defined by Eq. (3.5). The rate of the decrease in E_c is a calibrated parameter which is a function of the amount of accumulated plastic work for a given element. This calibration is explained in Chapter 5.

A comparison of the initial elastic flexural stiffness predicted by Eq. (3.6) and the results of experimental tests may be found in Schiller et al. [1994]. The verification of the nonlinear model for CFTs presented in Chapter 5 also confirms the accuracy of Eq. (3.6) and better illustrates the effect of decreasing the concrete elastic modulus on overall CFT load-deflection behavior.

3.3.3 Torsional Rigidity

Rectangular CFTs provide excellent torsional resistance, exhibiting both high strength and large ductility [Bridge, 1976; Kitada and Nakai, 1991]. Although this

observation has been well established, there has been relatively little experimental research conducted to quantify the behavior of rectangular CFTs in torsion. Kitada and Nakai [1991] present the most comprehensive results. These authors illustrate that, due to the composite interaction of the steel and the concrete, the ultimate torsional moment of a rectangular CFT is about 1.2 times the summation of the ultimate torsional moments of the steel tube and the concrete core. With regard to torsional stiffness, their tests showed that the torsional rigidity of a rectangular CFT subjected to relatively low values of torsion (as in this work) may be accurately estimated by assuming only the steel tube contributes torsional resistance. This result is given by the formula:

$$(G \cdot J)_{cft} = G_s \cdot J_s \quad (3.7)$$

The rigidity of the CFT remains relatively high for moderate values of torsion due to the benefit of the infilled concrete. The rigidity begins to decline once shear cracking of the concrete core initiates. At the point of maximum torsional resistance of the section, the CFT has undergone some nonlinear behavior and the rate of twist is close to the value that would be predicted using a secant stiffness equal to the original elastic rigidity of the steel alone [Kitada and Nakai, 1991]. Therefore, since nonlinear torsional stiffness is not modeled in this work, Eq. (3.7) provides an accurate estimate of a secant stiffness.

The equations that were incorporated in this work for the axial, flexural, and torsional rigidity of a CFT beam-column are shown in Table 3.1. Two equations are shown for flexural rigidity--the initial rigidity and the rigidity after the elastic modulus of the concrete has been reduced to its final value.

Table 3.1 Equations for the Elastic Rigidity of CFT Beam-Columns

Elastic Rigidity	Equation(s)	Equation Number
Axial	$(E \cdot A)_{cft} = E_s \cdot A_s + E_c \cdot A_c$	(3.2)
Flexural	$(E \cdot I)_{cft} = E_s \cdot I_s + E_c \cdot I_c$ (initial)	(3.6)
	$(E \cdot I)_{cft} = E_s \cdot I_s + 0.3 \cdot E_c \cdot I_c$ (final)	(3.5)
Torsional	$(G \cdot J)_{cft} = G_s \cdot J_s$	(3.7)

3.4 Geometric Nonlinear Formulation

The second component of element stiffness, the geometric stiffness, models the reduction in element stiffness due to the amplification of internal forces resulting from external applied loads acting through large displacements. Geometric nonlinearities may include member chord rotation (P- Δ effect) and member curvature (P- δ effect) [Galambos, 1988]. The geometric stiffness matrix used in this work is taken from Orbison [1982] and is based on the work of Bathe and Bolourchi [1979], Argyris et al. [1979], and others (readers are referred to Orbison for the terms of the matrix). The stiffness matrix is formulated based on the principle of minimum potential energy, assuming a cubic polynomial variation of the transverse displacements along the element length (i.e., cubic Hermetian shape functions). In addition, small strains and large displacements and rotations are assumed.

The geometric formulation is based on an updated Lagrangian formulation, in which equilibrium is formed on the deformed configuration of the structure [Bathe,

1982]. The nodal coordinates of the structure are thus updated at the end of each load step based on the incremental displacements. In the formulation of $[k_g^e]$, the elements are assumed to be straight at the beginning of each load step and prismatic. Both CFT and steel elements are modeled using the same form of the geometric stiffness matrix.

The accuracy of the geometric formulation is a function of the number of load steps in the analysis and the number of elements per member. It is possible, nonetheless, to obtain excellent accuracy with a reasonable number of load steps and elements. The analysis used in this work is a simple incremental approach, in lieu of equilibrating the forces for each load step (i.e., using a Newton-Raphson or similar solution scheme) which would require fewer load steps to obtain the same accuracy. For the tests contained in this work, the number of load steps in the analysis was doubled until the accuracy of the solution did not noticeably change. The number of load steps, although important for modeling geometric nonlinearities, depended to a greater extent on the requirements of the material nonlinear analysis.

Based on the geometric nonlinear formulation, each structural member in a frame may be accurately modeled using one to three elements [White and Hajjar, 1991]. For members subjected to an axial load ratio, P/P_e of less than 0.4, where $P_e = \frac{\pi^2 \cdot E \cdot I}{L^2}$, one element sufficiently accounts for both $P-\Delta$ effects and $P-\delta$ effects. The largest error in any term of the geometric stiffness matrix in this case is less than one percent. For axial load ratios beyond this value, however, up to 3 elements may be required to accurately model stiffness loss due to member curvature, or $P-\delta$, effects. Additionally, more than one element is necessary if initial imperfections in the beam-column are to be modeled explicitly. The members in this work generally consisted of 1 to 4 beam elements, depending upon the structure and loading. Four elements, rather than three, were used to model beam-columns for which the displacement at mid-span was desired.

3.5 Material Nonlinear Formulation

The two line element formulations outlined in Section 3.2 have different approaches to modeling plasticity in a structural element [White and Chen, 1993]. Macro finite element analyses typically employ plastic hinge or concentrated plasticity formulations, for which it is assumed that plasticity is restricted to zero-length hinges at the element ends. Material nonlinear fiber formulations, often referred to as plastic zone or distributed plasticity analyses, account for the spread of plasticity along an element's length as well as through the cross-section. Both types have advantages and disadvantages. The distributed plasticity analysis requires more computational effort but provides a more detailed and often more accurate analysis. The concentrated plasticity model, on the other hand, is computationally efficient and easily incorporated into matrix based analysis programs. This type of analysis is especially appealing for unbraced frame structures, in which the beam-columns are typically in double curvature and the maximum moments are at the element ends. Additionally, the concentrated plasticity analysis need not be confined to an elastic-perfectly plastic analysis, but may model strain hardening, concrete crushing, and the gradual plastification of an element end. This method becomes less accurate, however, for members such as columns subjected to high axial loads or beams with uniform bending moments, where the plasticity may be distributed along much of the member's length.

3.5.1 Concentrated Plasticity Models

The concentrated plasticity approach incorporated in this work is the bounding surface model, first developed by Dafalias and Popov [1975] to model the stress-strain behavior of metals subjected to cyclic loading. This formulation is summarized briefly here; the reader is referred to papers by Dafalias [Dafalias and Popov, 1975, 1976;

Dafalias, 1986, 1992] and others [Krieg, 1975; Tseng and Lee, 1983; Yang et al., 1985; McDowell, 1985, 1987; Yoder and McDowell, 1989; Sfakianakis and Fardis, 1991] for in depth discussions of the stress space bounding surface model.

Figure 3.3 illustrates a schematic representation in principal stress-space of the model proposed by Dafalias and Popov. The model consists of an inner loading surface representing the locus of points in two-dimensional stress space at which the initiation of yielding begins. The outer bounding surface represents the stress state at which a limiting or bounding stiffness is attained. The two surfaces generally are of the same shape, but have different radii, R_{LS} and R_{BS} . Three-dimensional stresses may also be represented by this model.

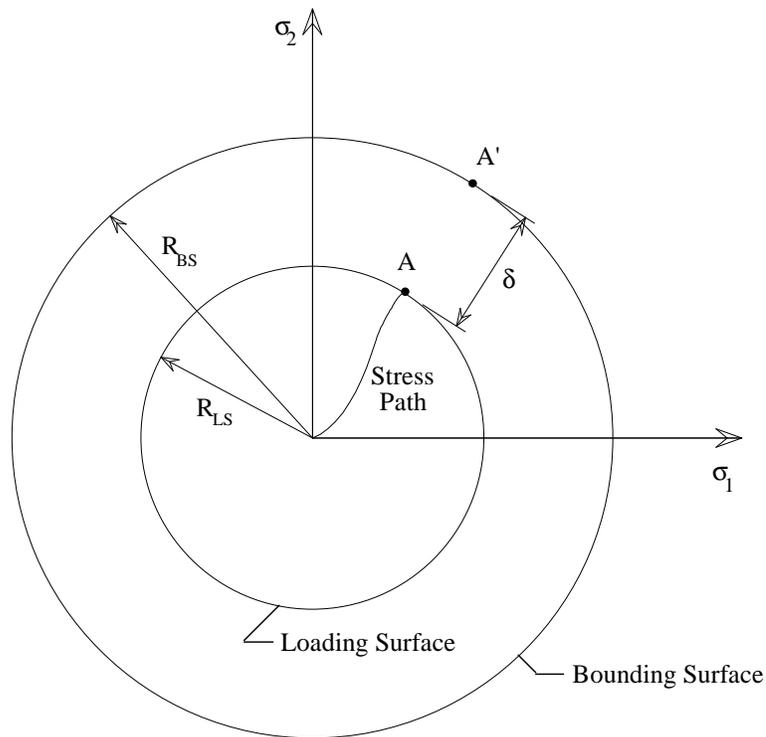


Figure 3.3 Bounding Surface Model

The material remains elastic for stress points within the loading surface. When the stress point contacts the loading surface--point *A* in Fig. 3.3--the material undergoes inelastic behavior. The plastic stress response is governed by a number of hardening rules which determine subsequent inelastic behavior [Armen, 1978]. As the material is stressed inelastically, the surfaces may translate (kinematic hardening), or contract or expand (isotropic hardening), to model parameters such as cyclic softening, cyclic hardening, and cyclic creep [Dafalias and Popov, 1975]. The degree of plasticity in the material is a function of the relative distance between the two surfaces, denoted by δ in Fig. 3.3 (i.e., the distance between *A* and *A'*, a point on the bounding surface whose location depends on the particular kinematic hardening approach that is selected [McDowell, 1985]).

Figure 3.4 schematically illustrates kinematic hardening of the loading surface. Figure 3.5 shows a similar schematic representation for isotropic hardening of the loading surface. While Fig. 3.5 illustrates an expansion of the loading surface, isotropic hardening may entail a shrinking of the loading surface as well (referred to as isotropic softening). Additionally, the bounding surface may also kinematically and isotropically harden. Once the force point contacts the bounding surface, the stress-strain curve increases at a constant slope--the bounding, or limiting slope.

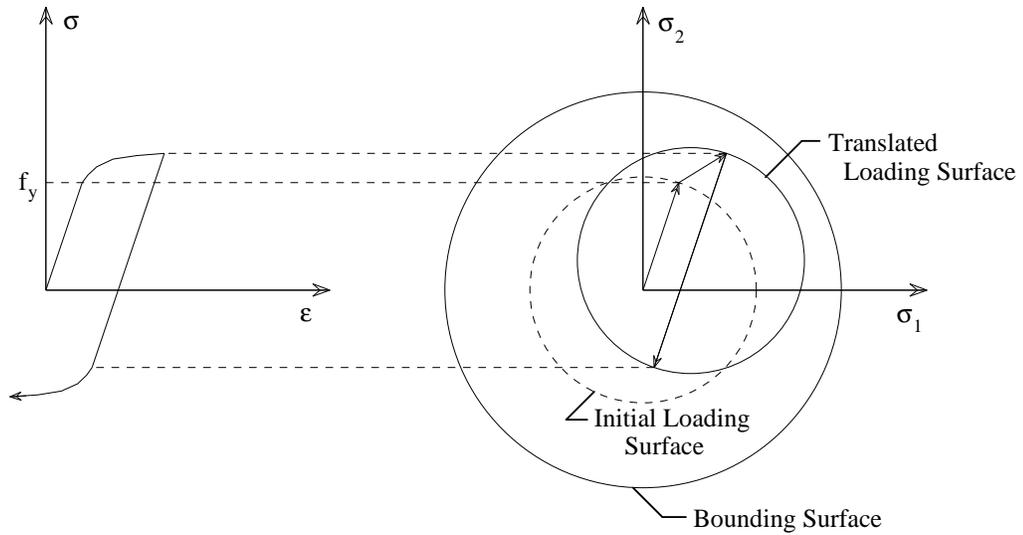


Figure 3.4 Kinematic Hardening

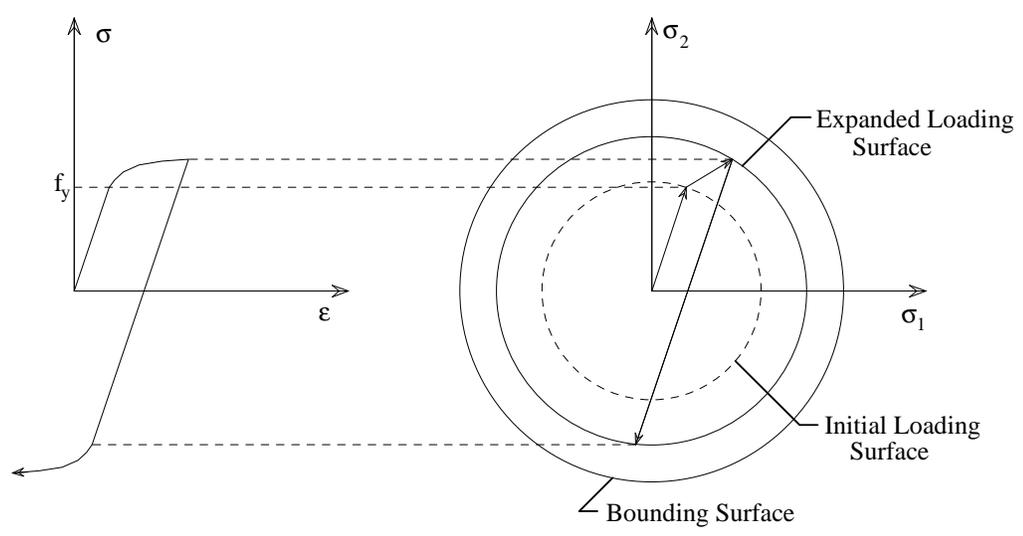


Figure 3.5 Isotropic Hardening

The bounding surface model in stress space has since been incorporated in different forms by several authors [Krieg, 1975; Tseng and Lee, 1983; Yang et al., 1985; McDowell, 1985, 1987; Sfakianakis and Fardis, 1991]. More recently, the stress space model was extended to force space by Hilmy and Abel [1985]. The force space model is a macroscopic model consisting of surfaces representing resultant stresses (or forces)--

typically axial force and major and minor axis bending moment. Orbison [1982] developed an empirical equation defining the shape of the surfaces for selected steel W-sections which Hilmy and Abel implemented in their force space bounding surface model. Zhao [1993] later refined and improved Hilmy's force space formulation and used it to model the plastification of steel elements. The model proposed by Zhao has been incorporated in this work to model the behavior of steel members. Additionally, the work conducted by Hilmy and Zhao forms the basis for much of the CFT plasticity model contained herein.

3.5.2 Introduction to the CFT Plasticity Model

The development of a bounding surface model for CFT elements first requires equations representing the loading surface and bounding surface in three-dimensional force space. Since element plasticity in the macroscopic model is constrained to zero-length hinges at the ends of the elements, the force state at an element end may be examined irrespective of length, i.e., by examining the CFT cross-section strength. The polynomial equation representing the three-dimensional cross-section strength surface of a general CFT member developed in Chapter 2 thus forms the basic mathematical expression of the loading and bounding surfaces. Both surfaces have the same basic shape and their respective sizes may be obtained by specifying appropriate surface radii (see Chapter 4). The inner loading surface represents forces which cause the initiation of yielding at some point on the element end cross-section. The outer surface, as in the Dafalias and Popov model, represents the force state at which a limiting stiffness of the CFT element end is achieved [Hilmy and Abel, 1985; Zhao, 1993] (see Zhao [1993] for further description of the bounding stiffness). The two surfaces harden kinematically and isotropically to model the monotonic and cyclic load-deflection characteristics of CFT

beam-columns. The following section demonstrates how the cyclic and monotonic behavioral characteristics of CFTs are modeled through isotropic and kinematic hardening of the two surfaces.

3.5.3 Modeling of Inelastic CFT Behavior

Figure 3.6 illustrates a typical cyclic hysteresis curve for a CFT beam-column tested by Sakino and Tomii [1981]. The test setup is shown along with the curve. The loading pattern consists of a constant axial load, P , and a cyclic shear, Q , applied over three full cycles at increasing increments of mid-height rotation from 0.5% to 2.5%.

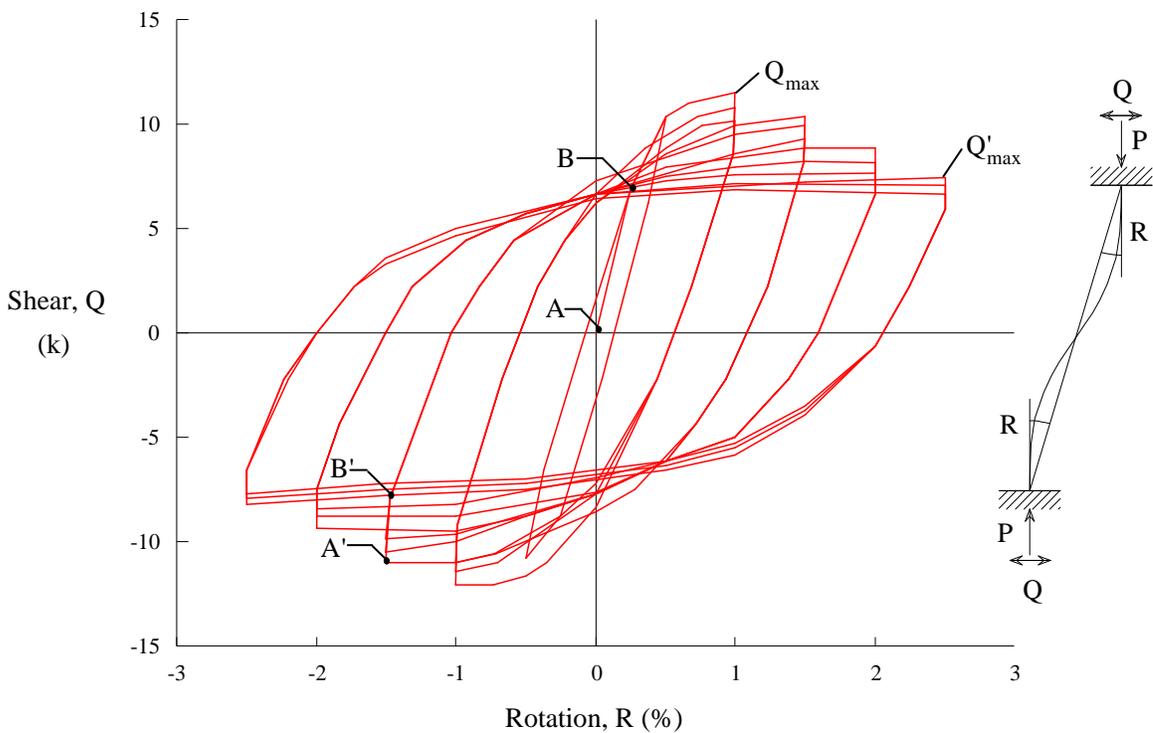


Figure 3.6 Cyclic CFT Behavior (after Sakino and Tomii [1981])

Figure 3.6 illustrates several key characteristics of cyclic CFT behavior that must be modeled by the concentrated plasticity formulation. The first noticeable characteristic

of the curve is the decrease in the size of the elastic zone with successive cycles of plasticity. As a CFT specimen is cycled, the concrete crushes, leading to an early loss of elastic strength. The elastic strength loss propagates further as the steel undergoes cycles of local buckling (see Section 3.1.2). Line A-B in Fig. 3.6 represents the elastic zone for the first cycle and line A'-B' represents the elastic zone after several cycles of loading. It is clearly evident that this region shrinks as the member undergoes repeated cycles of plasticity, but does not vanish completely. By shrinking the size of the loading surface, the decrease in the size of the elastic zone may be modeled. If the loading surface size is decreased with plastic loading, for each successive cycle the force point will have a smaller distance to traverse before plasticity reoccurs, thus creating a smaller elastic zone.

A second behavioral characteristic that may be observed in Fig. 3.6 is the change in maximum strength as the specimen is cycled. The section initially exhibits an increase in capacity due to cyclic strain hardening of the tube, and then the strength begins to degrade (e.g., the strength degradation from Q_{\max} to Q'_{\max} in the figure) due to concrete crushing and local buckling of the steel [Sakino and Tomii, 1981; Sugano et al., 1992]. The nonlinear model accounts for these effects primarily by first increasing and then decreasing the size of the bounding surface, which results in a corresponding change in the load at which the bounding stiffness is reached.

The concept of a bounding stiffness may be illustrated by examining the last three cycles in Fig 3.6. As the specimen approaches 2.5% rotation in the positive load region, the value of shear force levels out, showing only a slight increase with a further increase in rotation. This steady, relatively shallow slope evolves due to the stabilizing effect of the steel tube after significant local buckling (see Section 3.1) [Sakino and Tomii, 1981; Kawaguchi et al., 1993]. This slope may be thought of as the bounding stiffness. In the plasticity model, once the force point contacts the bounding surface, the force may

increase at a relatively small but constant rate based on a calibrated parameter that models the observed slope of the experimental curve.

A third characteristic of the cyclic behavior of CFT specimens, which is also prominent in the cyclic behavior of metals, is the Bauschinger effect. If the specimen is loaded inelastically into the positive quadrant of Fig. 3.6, upon unloading, less force will be required to reinitiate plastic behavior in the negative region than would be required if the specimen were initially loaded into the negative quadrant from its virgin state. Modeling this characteristic is the prime reason for kinematic hardening of the loading surface. The loading surface translates as the specimen is loaded into the positive region. Then upon unloading, the force point contacts the loading surface earlier, i.e., at a smaller magnitude of force, because the surface has translated. This is illustrated in Fig 3.4.

CFT specimens also exhibit a gradual softening behavior from the initiation of plasticity to the point at which they reach the bounding stiffness, as evidenced in each cycle of the curve in Fig. 3.6. Modeling this gradual softening is the chief advantage of the bounding surface model. Once the force point contacts the loading surface at the initiation of plasticity, the loading surface is dragged toward the bounding surface. As the loading surface translates, the distance between the surfaces decreases, causing a corresponding decrease in the element stiffness. Chapter 4 describes how this distance between the surfaces correlates to a mathematical formulation of the element stiffness.

The above characteristics of CFT behavior vary somewhat as the D/t ratio and material strengths of the section change, which necessitates variable calibration parameters, as will be described in Chapter 5. With these behavioral characteristics as a base, however, the plasticity model is formulated in detail in Chapter 4.

Chapter 4

CFT Force-Space Plasticity Formulation

This chapter presents the formulation of the concentrated plasticity model for CFTs. Portions of the formulation are based on the work of Hilmy and Abel [1985] and Zhao [1993] for steel members and are briefly summarized. The CFT plasticity model, however, has several unique characteristics, which are the focus of this chapter. The first section of the chapter describes the formulation of the CFT limit surfaces, which are based on the cross-section studies of Chapter 2. Following this section, the plastic reduction matrix for the CFT finite element is presented. The assumptions implicit in the derivation of this matrix are outlined, followed by a description of the plastic stiffness matrix, the main component of the plastic reduction matrix. The final two sections of the chapter detail the isotropic and kinematic hardening models, elaborating upon the introduction presented in Chapter 3 by describing the mathematical formulation of the hardening rules.

4.1 Definitions and Assumptions of the Bounding Surface Model

4.1.1 Loading and Bounding Surfaces

As described in Section 3.5, the bounding surface model consists of two surfaces, an inner loading surface and an outer bounding surface, which both translate and contract in force space to model the load-deflection behavior of CFT elements. The two surfaces are each convex and continuous [Drucker, 1951], and are of the same shape but have different sizes. Figure 4.1a illustrates the initial unstressed state of the loading surface and bounding surface for a CFT element end in two-dimensional, normalized force space. Figure 4.1b then shows an arbitrary configuration of the bounding surface model. The vector $\{s\}$ denotes the normalized vector of internal forces at the element end. This vector resides on the loading surface during plastic loading (point F in Fig 4.1b). The normalized centroids of the loading surface and bounding surface are shown as the vectors $\{\alpha_{LS}\}$ and $\{\alpha_{BS}\}$, respectively.

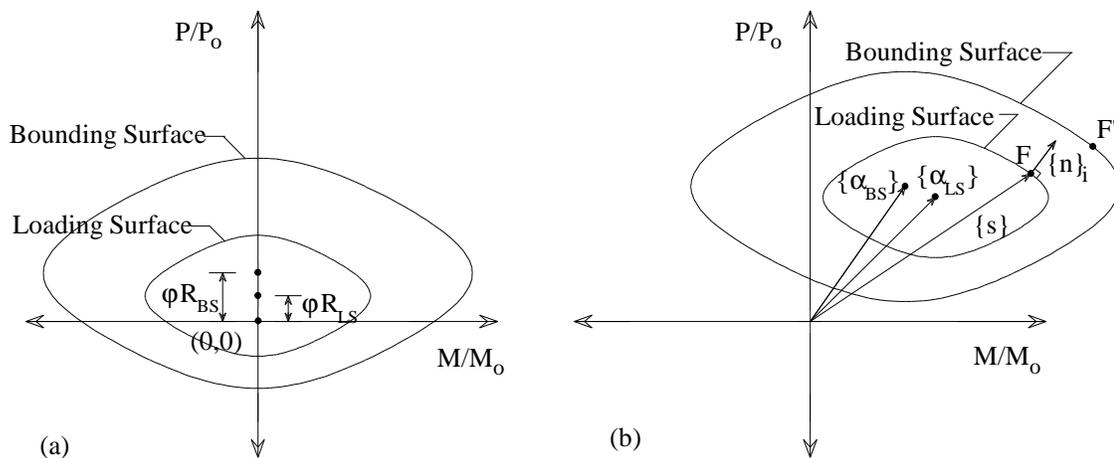


Figure 4.1 CFT Bounding Surface Model

The equation for the three-dimensional cross-section strength of a CFT, as derived in Chapter 2, may now be rewritten as the equation defining the loading or bounding surface. Using the equation of Table 2.9, the loading surface equation in three-dimensional normalized force space may be rewritten as a function of the loading surface "radius", R_{LS} , and the vector from the normalized centroid of the surface to the normalized force point, $\{s\} - \{\alpha_{LS}\}$:

$$\begin{aligned}
 f(\{s\} - \{\alpha_{LS}\}, R_{LS}) = & c_1 \cdot \left[\left(\frac{M_y - a_{LS}^{my}}{R_{LS} \cdot M_{yo}} \right)^2 + \left(\frac{M_z - a_{LS}^{mz}}{R_{LS} \cdot M_{zo}} \right)^2 \right] + \\
 & c_2 \cdot \left(\frac{P - a_{LS}^p - \varphi \cdot R_{LS} \cdot P_o}{R_{LS} \cdot P_o} \right)^2 + c_3 \cdot \left[\left(\frac{M_y - a_{LS}^{my}}{R_{LS} \cdot M_{yo}} \right)^2 + \left(\frac{M_z - a_{LS}^{mz}}{R_{LS} \cdot M_{zo}} \right)^2 \right] \cdot \\
 & \left(\frac{P - a_{LS}^p - \varphi \cdot R_{LS} \cdot P_o}{R_{LS} \cdot P_o} \right)^2 + c_4 \cdot \left(\frac{M_y - a_{LS}^{my}}{R_{LS} \cdot M_{yo}} \right)^2 \cdot \left(\frac{M_z - a_{LS}^{mz}}{R_{LS} \cdot M_{zo}} \right)^2 - 1 = 0 \quad (4.1)
 \end{aligned}$$

Equation (4.1) modifies the cross-section strength surface equation developed in Chapter 2 by multiplying the nominal strengths (P_o , M_{yo} , and M_{zo}) by R_{LS} . Although this value is not a radius per se (i.e., the surface is not circular in shape), it serves as a convenient representation of the size of the surface relative to the full cross-section strength surface developed in Chapter 2. With this modification, the normalized force vector with respect to the loading surface for an element end in three dimensions becomes:

$$\{s\} = \left\{ \frac{P}{R_{LS} \cdot P_o} \quad \frac{M_y}{R_{LS} \cdot M_{yo}} \quad \frac{M_z}{R_{LS} \cdot M_{zo}} \right\}^T \quad (4.2)$$

The vector representing the normalized centroid of the loading surface in three dimensions, $\{\alpha_{LS}\}$, may be written as:

$$\{\alpha_{LS}\} = \left\{ \frac{a_{LS}^p + \varphi \cdot R_{LS} \cdot P_o}{R_{LS} \cdot P_o} \quad \frac{a_{LS}^{my}}{R_{LS} \cdot M_{yo}} \quad \frac{a_{LS}^{mz}}{R_{LS} \cdot M_{zo}} \right\}^T \quad (4.3)$$

For an element that has not undergone any plastic deformation (i.e., the surfaces have not moved), the normalized centroid of the loading surface given by Eq. (4.3) will lie on the axial force axis a distance $\varphi \cdot R_{LS}$ above the moment axis (Fig. 4.1a) to account for the asymmetry of the surface (refer to Chapter 2). Thus, the initial, normalized centroidal coordinates of the loading surface, before any plastic deformation has occurred in the CFT element, are:

$$\{\alpha_{LS}\} = \{\varphi \cdot R_{LS} \quad 0 \quad 0\}^T \quad (4.4)$$

Although Eqs. (4.1) to (4.4) provide a concise representation of the loading surface, the ensuing CFT plasticity formulation considers forces in unnormalized force space (this simplifies many calculations). In unnormalized force space, the force point is written as:

$$\{S\} = \{P \quad M_y \quad M_z\}^T \quad (4.5)$$

and the centroid of the loading surface is represented by the vector $\{A_{LS}\}$, where:

$$\{A_{LS}\} = \{a_{LS}^p + \varphi \cdot R_{LS} \cdot P_o \quad a_{LS}^{my} \quad a_{LS}^{mz}\}^T \quad (4.6)$$

The unnormalized centroid of the loading surface given by Eq. (4.6) is the summation of two vectors: $\{A_{LS}\} = \{a_{LS}\} + \{\Phi_{LS}\}$, where $\{a_{LS}\}$ represents the "backforce"

vector, and $\{\Phi_{LS}\}$ represents the centroidal "offset" vector. These vectors are expressed as:

$$\{a_{LS}\} = \{a_{LS}^p \quad a_{LS}^{my} \quad a_{LS}^{mz}\}^T \quad (4.7a)$$

$$\{\Phi_{LS}\} = \{\varphi \cdot R_{LS} \cdot P_o \quad 0 \quad 0\}^T \quad (4.7b)$$

Previous stress-space and force-space bounding surface models [Dafalias and Popov, 1975; Hilmy and Abel, 1985; Zhao, 1993] identify the location of the loading and bounding surfaces by their respective backforce vector, $\{a_{LS}\}$ and $\{a_{BS}\}$, which both equal the zero vector for a specimen that has not been loaded. For symmetric surfaces, this backforce point coincides with the centroid of the surface, i.e., there is no initial centroid offset from the origin. For asymmetric surfaces, however, the additional offset vector, $\{\Phi\}$, must be added to the backforce to obtain the location of the centroid (Eq. (4.6)).

In the CFT plasticity formulation to follow, the surface locations are effectively identified by the backforce vector. However, to facilitate a more succinct representation of the plasticity equations, the formulation is written in terms of the surface centroids. For clarity, the centroidal vectors should nonetheless be thought of as the summation of the two vectors--the backforce, $\{a\}$, plus an offset, $\{\Phi\}$, to account for the asymmetry.

Similar equations to the loading surface equations given by Eqs. (4.1) through (4.7) may be written to describe the bounding surface, $f(\{s\} - \{\alpha_{BS}\}, R_{BS})$. The equation for the bounding surface is identical to Eq. (4.1) except for the radius and the centroidal vector. The bounding surface is represented by a larger radius, R_{BS} , and a normalized centroid, $\{\alpha_{BS}\}$, given by:

$$\{\alpha_{BS}\} = \left\{ \frac{a_{BS}^p + \varphi \cdot R_{BS} \cdot P_o}{R_{BS} \cdot P_o} \quad \frac{a_{BS}^{my}}{R_{BS} \cdot M_{yo}} \quad \frac{a_{BS}^{mz}}{R_{BS} \cdot M_{zo}} \right\}^T \quad (4.8)$$

The initial value of the normalized bounding surface centroid (Fig. 4.1a) is given by:

$$\{\alpha_{BS}\} = \{\varphi \cdot R_{BS} \quad 0 \quad 0\} \quad (4.9)$$

The unnormalized centroid is expressed as the sum of the bounding surface backforce vector and the centroidal offset of the bounding surface:

$$\{A_{BS}\} = \{a_{BS}\} + \{\Phi_{BS}\} = \{a_{BS}^p + \varphi \cdot R_{BS} \cdot P_o \quad a_{BS}^{my} \quad a_{BS}^{mz}\}^T \quad (4.10)$$

where

$$\{a_{BS}\} = \{a_{BS}^p \quad a_{BS}^{my} \quad a_{BS}^{mz}\}^T \quad (4.11a)$$

$$\{\Phi_{BS}\} = \{\varphi \cdot R_{BS} \cdot P_o \quad 0 \quad 0\}^T. \quad (4.11b)$$

The values for the surface radii, R_{LS} and R_{BS} , are parameters that must be calibrated. The discussion in Chapter 5 addresses the calibration of the surface radii and specifies the actual values used in this work.

4.1.2 Basic Assumptions

When the force point contacts the loading surface, the element end undergoes plastic deformation. Subsequent inelastic behavior is governed by several rules and assumptions. First, the force point must remain on the loading surface whenever plastic

loading occurs [Armen, 1978]. This rule is referred to as the consistency condition and must be adhered to when the surfaces harden isotropically and kinematically (movement of the surfaces is discussed further in Sections 4.4 and 4.5) [Hilmy and Abel, 1985; Zhao, 1993].

Two additional conditions were postulated by Drucker [1951] for stress-space plasticity models of work-hardening materials. They were adopted to force space plasticity by Hilmy and Abel [1985] and are retained in this work. The first condition states that, for work-hardening materials, the plastic potential surface is convex [Drucker, 1951]. This condition is satisfied by Eq. (4.1). While concrete exhibits strain-softening behavior [Bazant, 1982], the load-deflection behavior of CFTs rarely exhibits softening behavior except when subjected to extreme cyclic loading [Kawaguchi et al., 1993]. The force-space plasticity model is thus treated as "work-hardening" in this research. Figure 3.6 indicates the continual hardening behavior exhibited by a typical CFT beam-column.

A second condition, the normality condition, states that plastic deformations are normal to the plastic potential surface [Hilmy and Abel, 1985; Zhao, 1993], i.e., in the direction of the gradient of the surface. In this work, associated flow is assumed, such that the initial yield (loading) surface is taken to be the same as the plastic potential surface. The gradient vector equals the partial derivative of the loading surface, f , with respect to the current force point, and is denoted by $\{n\}_i$, where the subscript i denotes the hinge at the i -end of the element. In three-dimensional force space, the gradient for an element end is given by:

$$\{n\}_i = \frac{\partial f}{\{\partial S\}} = \left\{ \frac{\partial f}{\partial P} \quad 0 \quad 0 \quad 0 \quad \frac{\partial f}{\partial M_y} \quad \frac{\partial f}{\partial M_z} \right\}^T \quad (4.12)$$

where the zero terms correspond to the shear in the strong and weak axis directions and the torsion. These terms remain zero since the plasticity model considers only axial force and bending. The gradient vector to the loading surface is illustrated in two-dimensional force space in Fig. 4.1.

For convenience in later calculations, the gradient vector of the entire element may be represented by a single matrix by combining the vectors at both ends, i and j , of the element:

$$[N] = \begin{bmatrix} \{n\}_i & \{0\} \\ \{0\} & \{n\}_j \end{bmatrix} \quad (4.13)$$

The gradient terms in Eq. (4.13) for an elastic element end are zero and therefore do not affect the calculation of the plastic reduction matrix.

4.2 Derivation of the Plastic Reduction Matrix

The stiffness formulation presented in Chapter 3 introduced the plastic reduction matrix, $[k_r^e]$, for a general element. This section briefly discusses the derivation of this matrix. This formulation is derived from incremental plasticity theory and is based on the work of Zhao [1993]. The reader is referred there for a more extensive derivation.

In this approach, the relationship between incremental forces and incremental displacements may be expressed as:

$$\{dS\} = ([k_r^e] + [k_r^e]) \cdot \{dq\} \quad (4.14)$$

where $[k_T^e]$ represents the elastic tangent stiffness matrix and is equal to the sum of the elastic and geometric stiffness matrices, $[k_e^e] + [k_g^e]$. The incremental displacement and force vectors for end i of an element are given by (see Fig. 3.2):

$$\begin{aligned} \{dq\}_i &= \{du_{ix} \quad du_{iy} \quad du_{iz} \quad d\theta_{ix} \quad d\theta_{iy} \quad d\theta_{iz}\}^T \\ \{dS\}_i &= \{dF_{ix} \quad dF_{iy} \quad dF_{iz} \quad dM_{ix} \quad dM_{iy} \quad dM_{iz}\}^T \end{aligned} \quad (4.15)$$

Similar expressions apply to end j of the element. The complete incremental displacement and force vectors for the element are:

$$\begin{aligned} \{dq\} &= \{\{dq\}_i \quad \{dq\}_j\}^T \\ \{dS\} &= \{\{dS\}_i \quad \{dS\}_j\}^T \end{aligned} \quad (4.16)$$

Several assumptions are incorporated into the derivation of the plastic reduction stiffness matrix. First, it is assumed that the incremental displacements can be decomposed into elastic and plastic displacements [Hilmy and Abel, 1985; Zhao, 1993]:

$$\{dq\} = \{dq_e\} + \{dq_p\} \quad (4.17)$$

Then, by the normality condition [Drucker, 1951], incremental plastic displacements are normal to the loading surface, i.e., in the direction of the gradient to the loading surface.

$$\{dq_p\} = [N] \cdot \{d\lambda\} \quad (4.18)$$

where $\{d\lambda\} = \{d\lambda_i \quad d\lambda_j\}^T$ and $d\lambda_i$ represents the magnitude of incremental plastic displacement at element end i . In isotropic stress-space plasticity formulations, this parameter may be calibrated directly. In multi-dimensional force space, however, the behavior is orthotropic, and an additional constraint is required [Zhao, 1993]. This orthotropic behavior is expressed in the relation between the plastic stiffness matrix and the incremental plastic displacements as follows:

$$\{dS^*\} = [k_p^e] \cdot \{dq_p\} \quad (4.19)$$

where $[k_p^e]$ is referred to as the plastic stiffness matrix. In this work, it is diagonal but not proportional to the identity matrix [Zhao, 1993]. Thus, the incremental force vector, $\{dS^*\}$, is not parallel to $\{dq_p\}$, as it would be if the force space behavior were isotropic. Zhao then related this vector to the actual incremental force vector, $\{dS\}$, by the following constraint [Zhao, 1993]:

$$[N]^T \cdot \{dS^*\} = [N]^T \cdot \{dS\} \quad (4.20)$$

Through the use of this constraint, in the force space formulation the element plastic stiffness, $[k_p^e]$, is calibrated rather than $\{d\lambda\}$ (see Section 4.3).

The relationships from Eqs. (4.17) to (4.20) may now be used to derive the plastic reduction matrix. Substituting Eqs. (4.18) and (4.19) into Eq. (4.20) results in:

$$[N]^T \cdot \{dS\} = [N]^T \cdot [k_p^e] \cdot \{dq_p\} = [N]^T \cdot [k_p^e] \cdot [N] \cdot \{d\lambda\} \quad (4.21)$$

Solving Eq. (4.21) for the magnitude of the incremental plastic displacement produces:

$$\{d\lambda\} = \frac{[N]^T \cdot \{dS\}}{[N]^T \cdot [k_p^e] \cdot [N]} \quad (4.22)$$

The elastic force-displacement relationship is given by:

$$\{dS\} = [k_T^e] \cdot \{dq_e\} \quad (4.23)$$

By substituting Eq. (4.17) into Eq. (4.23), the total incremental force vector may also be expressed as:

$$\{dS\} = [k_T^e] \cdot \{dq\} - [k_T^e] \cdot \{dq_p\} \quad (4.24)$$

or, substituting Eq. (4.18) into Eq. (4.24):

$$\{dS\} = [k_T^e] \cdot \{dq\} - [k_T^e] \cdot [N] \cdot \{d\lambda\} \quad (4.25)$$

Using Eqs. (4.22) and (4.25) and rearranging terms, the magnitudes of the incremental plastic displacements may be written in terms of the total incremental displacements:

$$\{d\lambda\} = \frac{[N]^T \cdot [k_T^e]}{[N]^T \cdot ([k_T^e] + [k_p^e]) \cdot [N]} \cdot \{dq\} \quad (4.26)$$

Combining Eqs. (4.25) and (4.26), the incremental force vector may be expressed as:

$$\{dS\} = [k_T^e] \cdot \{dq\} + \left\{ -[k_T^e] \cdot [N] \cdot ([N]^T \cdot ([k_T^e] + [k_p^e]) \cdot [N])^{-1} \cdot [N]^T \cdot [k_T^e] \right\} \cdot \{dq\} \quad (4.27)$$

The bracketed expression represents the plastic reduction matrix as derived by Hilmy and Abel [1985] and Zhao [1993]:

$$[k_r^e] = -[k_T^e] \cdot [N] \cdot ([N]^T \cdot ([k_T^e] + [k_p^e]) \cdot [N])^{-1} \cdot [N]^T \cdot [k_T^e] \quad (4.28)$$

Since the plastic reduction matrix is negative, this stiffness is subtracted from the tangent stiffness. Therefore, increasing its magnitude results in a decrease in the total stiffness. For the case when the plastic stiffness matrix is infinite (i.e., no plastic hinges), the plastic reduction matrix is zero and the element stiffness is equal to the elastic tangent stiffness, $[k_T^e]$.

4.3 Calculation of Plastic Stiffness

The plastic stiffness matrix, $[k_p^e]$, is a diagonal matrix composed of twelve independent terms, six at each element end. Plastic displacements due to shear and torsion are neglected in this model. Therefore, the shear and torsion terms will be infinite, resulting in zero plastic displacements for these components. The diagonal matrix for a single element takes the form:

$$[k_p^e] = \text{diag} \left((k_{pi}^e)_p, \infty, \infty, \infty, (k_{pi}^e)_{my}, (k_{pi}^e)_{mz}, (k_{pj}^e)_p, \infty, \infty, \infty, (k_{pj}^e)_{my}, (k_{pj}^e)_{mz} \right) \quad (4.29)$$

The terms of the plastic stiffness matrix, $[k_p^e]$, are determined by the bounding surface model, and are thus computed based upon the elastic stiffness, the distance

between the loading surface and the bounding surfaces for the current load step, δ , the initial distance between the surfaces at the first contact of the force point with the loading surface, δ_{in} , and calibrated plasticity coefficients, κ_1 and κ_2 . The general bounding surface formula for a single plastic stiffness term is represented by [Dafalias and Popov, 1975; Hilmy and Abel, 1985]:

$$k_p = k_e \cdot \left(\kappa_1 + \kappa_2 \cdot \frac{\delta}{\delta_{in} - \delta} \right) \quad (4.30)$$

Each plastic stiffness term will take the form of Eq. (4.30) but may contain different values of elastic stiffness and calibration coefficients, thus resulting in orthotropic plasticity [Hilmy and Abel, 1985]. The elastic stiffnesses in the three force directions are calculated as follows:

$$(k_e)_p = \frac{(E \cdot A)_{cft}}{L} \quad (k_e)_{my} = \frac{(E \cdot I_y)_{cft}}{L} \quad (k_e)_{mz} = \frac{(E \cdot I_z)_{cft}}{L} \quad (4.31)$$

The remaining variables of Eq. (4.30)--the calibrated plasticity coefficients and the distances between the surfaces--are detailed in the following two sections.

4.3.1 Plasticity Coefficients

The plasticity coefficients, κ_1 and κ_2 , are calibrated separately in each of the three force space directions included in the plasticity model. The details of the calibration of these parameters are presented in Chapter 5. The parameter κ_1 represents the bounding slope of the element stiffness (see Section 3.5). When the force point reaches the

bounding surface, the distance between the surfaces, δ , is zero and the plastic stiffness reaches its bounding or limiting stiffness, given by:

$$k_p^b = \kappa_1 \cdot k_e \quad (4.32)$$

The parameter κ_2 represents the rate of plastification, i.e., the rate at which the element stiffness decreases from the elastic state to the bounding limit. Figure 4.2 illustrates the effect of increasing the value of κ_2 . A large value of κ_2 results in a relatively slow rate of degradation of stiffness until the force is close to the bounding surface, at which point the stiffness decreases rapidly. A small value of κ_2 , on the other hand, results in a more gradual plastification.

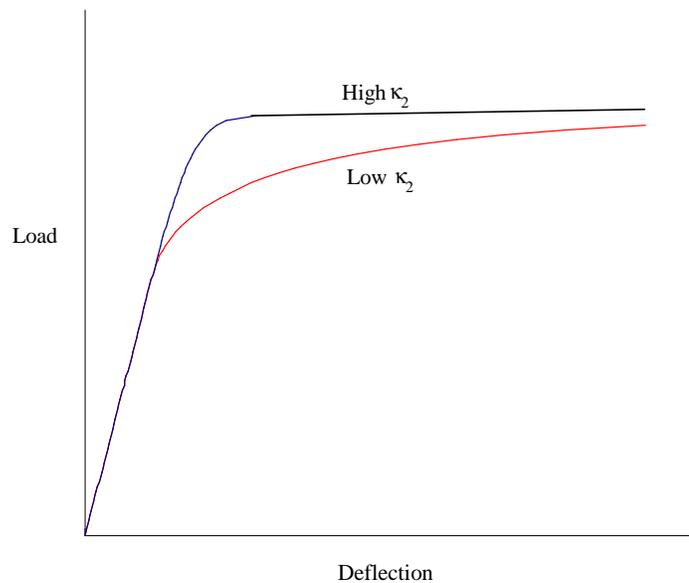


Figure 4.2 Effect of κ_2 on the Rate of Element Plastification

4.3.2 Distance Between the Surfaces

In this work, the distance between the two surfaces, δ , is computed along a vector from the beginning of step force point, denoted by the vector $\{S\}$ in Fig. 4.3, which resides on the loading surface, to the conjugate force point on the bounding surface, denoted by $\{S'\}$. This vector, referred to as the Mroz vector after Mroz [1967], is given by:

$$\{v\} = \{S'\} - \{S\} \quad (4.33)$$

where the conjugate force point vector, $\{S'\}$, is given by:

$$\{S'\} = \frac{R_{BS}}{R_{LS}} \cdot (\{S\} - \{A_{LS}\}) + \{A_{BS}\} \quad (4.34)$$

and where, as discussed in Section 4.1.1, $\{A_{LS}\} = \{a_{LS}\} + \{\Phi_{LS}\}$ (see Fig. 4.3), and $\{A_{BS}\} = \{a_{BS}\} + \{\Phi_{BS}\}$. (See Dafalias and Popov [1986] for a discussion of alternate methods of measuring the distance between surfaces in stress-space formulations for use in Eq. (4.30)).

The distance between the two surfaces may then be determined by taking the norm of the Mroz vector:

$$\delta = \sqrt{\{v\}^T \cdot \{v\}} \quad (4.35)$$

The value of the distance between the two surfaces for the load step at which the loading surface is first contacted is denoted as δ_{in} . This initial distance is updated each time the force point moves from an elastic state (i.e., inside the loading surface) to a plastic state

(i.e., on the loading surface). Therefore, if a hinge unloads, δ_{in} is recalculated upon reinitiation of contact between the force point and the loading surface (see Hilmy [1984] and Chaboche [1986] for a discussion on proper updating of δ_{in} , e.g., for cases of unloading/reloading).

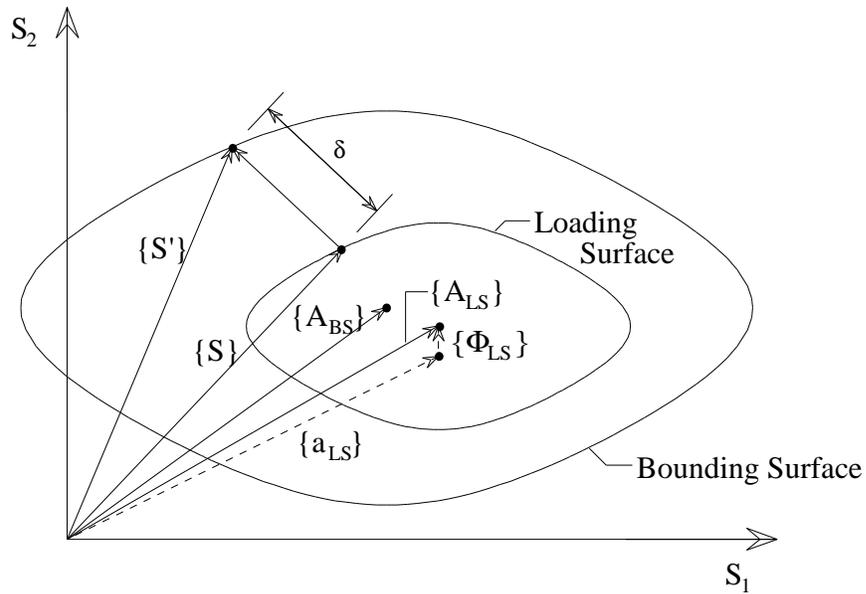


Figure 4.3 Distance Between Loading and Bounding Surface

4.4 Isotropic Hardening

The isotropic hardening formulation presented in this work to change the size of the loading surface and bounding surface is based upon the accumulated amount of plastic work for a given element end. As plastic work accumulates, the surfaces contract

or expand at exponential rates based upon the amount of accumulated plastic work and based upon rate parameters that are calibrated to experimental results.

4.4.1 Calculation of Plastic Work

The incremental plastic work is computed for each inelastic element end and is expressed as the dot product of the incremental force vector and the incremental plastic displacement vector for the element end:

$$dW_{p_i} = \{dS\}_i^T \cdot \{dq_p\}_i \quad (4.36)$$

where i denotes the i -end of the element. The incremental force vector is known for each load step and the incremental plastic displacement vector may be calculated as per the discussion in Section 4.2. Expressing Eq. (4.18) in terms of a single element end, the incremental plastic displacements are given by:

$$\{dq_p\}_i = \{n\}_i \cdot d\lambda_i \quad (4.37)$$

where the magnitude of incremental plastic displacement at an element end, $d\lambda_i$, may be written as:

$$d\lambda_i = \frac{\{n\}_i^T \cdot \{dS\}_i}{\{n\}_i \cdot [k_p^e]_i \cdot \{n\}_i} \quad (4.38)$$

where $[k_p^e] = \text{diag}\left\{\left(k_{pi}^e\right)_p \quad \infty \quad \infty \quad \infty \quad \left(k_{pi}^e\right)_{my} \quad \left(k_{pi}^e\right)_{mz}\right\}$. The incremental plastic work may be calculated using Eqs. (4.36) to (4.38). The accumulated plastic work is updated for each plastic element end by integrating the incremental plastic work:

$$W_{p_i} = \int_0^{t_{final}} dW_{p_i} = \sum_0^{t_{final}} dW_{p_i} \quad (4.39)$$

The plastic work is continually updated from the initial onset of plastification and is not reset to zero upon unloading. Therefore, when an element end reloads, the entire previous plastic load history of that end is considered in the formulation.

Sections having the same material properties and D/t ratio but different dimensions (e.g., a $6 \times 6 \times \frac{1}{4}$ section and a $12 \times 12 \times \frac{1}{2}$ section) will produce different values of plastic work for the same amount of displacement. Therefore, since the same calibration parameters will be used for both of these sections (see Section 5.2), the accumulated work of Eq. (4.39) must be normalized by some value. In this work, the accumulated plastic work for a given element end is normalized by the axial and flexural elastic strain energy present in the element at the first initiation of plastic behavior (i.e., at the first contact of the force point with the loading surface). Once this elastic work for an element is established, it remains constant for the remainder of the analysis.

The axial strain energy for a single element is expressed as:

$$U_p = \int_0^L \left(\frac{P^2}{2 \cdot E \cdot A} \right) dx = \frac{P^2 \cdot L}{2 \cdot E \cdot A} \quad (4.40)$$

The axial rigidity, $E \cdot A$, and the axial load, P , are both constant over the element length and therefore the strain energy is simply the integrand multiplied by the element length,

as shown in the latter portion of Eq. (4.40). The strain energy for a single element due to bending (shown here for the strong axis moment, M_z), is given by:

$$U_{mz} = \int_0^L \frac{M_z^2}{2 \cdot E \cdot I} \cdot dx \quad (4.41)$$

The flexural rigidity, $E \cdot I$, is constant along the element length, but the moment is a function of the distance along the element and the moments at the i - and j -ends of the element. With a linear relationship between end moments, the strain energy integral becomes:

$$U_{mz} = \frac{1}{2 \cdot E \cdot I} \cdot \int_0^L \left[M_{zi}^2 + \frac{x^2}{L^2} \cdot (M_{zi} - M_{zj})^2 - 2 \cdot M_{zi} \cdot \frac{x}{L} \cdot (M_{zi} - M_{zj}) \right] \cdot dx \quad (4.42)$$

Integrating Eq. (4.42) produces the expression for major axis bending strain energy:

$$U_{mz} = \frac{1}{2 \cdot E \cdot I} \cdot \left[M_{zi}^2 \cdot L + \frac{L}{3} \cdot (M_{zi} - M_{zj})^2 - M_{zi} \cdot L \cdot (M_{zi} - M_{zj}) \right] \quad (4.43)$$

A similar expression may be written for the bending moment in the minor axis direction. The summation of the three strain energy terms produces the elastic strain energy for the element (note that the strain energy due to shear and torsion is neglected):

$$U_T = U_p + U_{my} + U_{mz} \quad (4.44)$$

The accumulated plastic work for an element end is then normalized by the elastic strain energy to produce a normalized value of plastic work, Ω_p , which now allows the same calibration parameters to be used regardless of material strength and section geometry:

$$\Omega_p = \frac{W_p}{U_T} \quad (4.45)$$

4.4.2 Isotropic Hardening of the Loading and Bounding Surfaces

The size of the loading surface and bounding surface, represented by their respective radii, R_{LS} and R_{BS} , are updated based upon the total amount of normalized plastic work represented by Eq. (4.45). The surface radii are also functions of the initial and final values of the respective surfaces, and the isotropic hardening parameters, ξ_{LS} and ξ_{BS} . The initial and final values of the surfaces and the isotropic hardening parameters are all obtained based on a calibration to experimental results (see Chapter 5).

The equation defining the updated loading surface size may be represented by (see Ricles and Popov [1994] for a related formula for isotropic hardening of steel members):

$$(R_{LS})_{new} = (R_{LS})_{final} - [(R_{LS})_{final} - (R_{LS})_{init}] \cdot e^{-\xi_{LS} \cdot \Omega_p} \quad (4.46)$$

When the normalized plastic work, Ω_p , is zero, the exponent term equals one and the loading surface equals the initial size, $(R_{LS})_{init}$. As Ω_p becomes increasingly larger, the surface size asymptotically approaches the final size, $(R_{LS})_{final}$, at a rate dependent upon ξ_{LS} .

Based upon the new radius, the magnitude of incremental loading surface translation due to isotropic hardening is expressed as the ratio of the incremental change in the loading surface size to the old loading surface size:

$$\eta_{LS} = \frac{dR_{LS}}{(R_{LS})_{old}} = \frac{(R_{LS})_{old} - (R_{LS})_{new}}{(R_{LS})_{old}} \quad (4.47)$$

The loading surface centroid moves in the direction of the vector from the centroid, $\{A_{LS}\}$, to the force point, $\{S\}$, by an amount, η_{LS} :

$$\{dA_{LS}\}_{iso} = \eta_{LS} \cdot (\{S\} - \{A_{LS}\}) \quad (4.48)$$

This formulation contracts the loading surface directly about the force point, not the surface centroid. By doing this, the consistency condition (see Section 4.1) is not violated, i.e., the force point remains on the surface [Zhao, 1993]. Figure 4.4 illustrates the contraction of the two surfaces.

The bounding surface initially hardens isotropically (i.e., increases in size), and then, at a calibrated value of normalized plastic work, $(\Omega_p)_{intm}$, the bounding surface isotropically softens in the same manner as the loading surface (see Section 3.5.3). For values of normalized plastic work less than $(\Omega_p)_{intm}$, the bounding surface size increases by the formula:

$$(\mathbf{R}_{BS})_{new} = (\mathbf{R}_{BS})_{intm} - [(\mathbf{R}_{BS})_{intm} - (\mathbf{R}_{BS})_{init}] \cdot e^{-\xi_{BS} \cdot \Omega_p} \quad (4.49)$$

where $(R_{BS})_{intm}$ denotes the intermediate, or maximum size of the bounding surface. Once the intermediate plastic work value, $(\Omega_p)_{intm}$, is attained, the bounding surface decreases according to the following formula:

$$(R_{BS})_{new} = (R_{BS})_{final} - [(R_{BS})_{final} - (R_{BS})_{intm}] \cdot 1.1^{-\xi_{BS} \cdot |\Omega_p - (\Omega_p)_{intm}|} \quad (4.50)$$

To avoid a discontinuity in the formulation for the bounding surface size, the value of $(R_{BS})_{intm}$ in Eq. (4.50) is updated to the actual bounding surface size at $(\Omega_p)_{intm}$ (the original intermediate size, $(R_{BS})_{intm}$, is only asymptotically approached in Eq. (4.49) and will not be reached). A second characteristic of Eq. (4.50) to note is the base of the exponential, which is set to 1.1 rather than e . The latter value results in an undesirably rapid decrease in the size of the loading surface, whereas the value of 1.1 provides the more moderate decrease necessary to model actual CFT behavior (see Chapter 5).

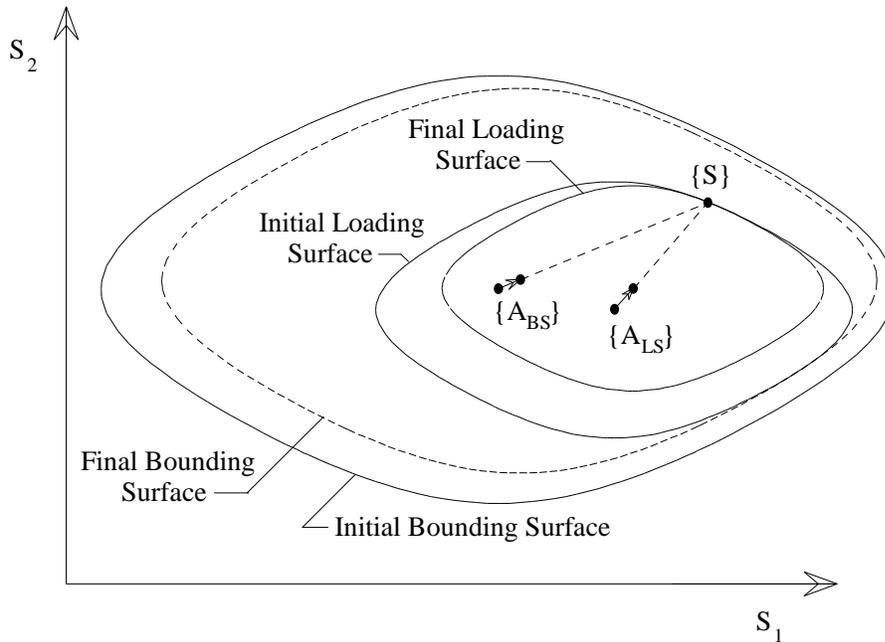


Figure 4.4 Isotropic Hardening of Loading and Bounding Surfaces

At no time during the analysis does the loading surface radius size exceed the size of the bounding surface. If this occurred, the surfaces would overlap, thus violating the bounding principle [Dafalias and Popov, 1975]. To ensure that overlap does not occur, the initial and final sizes of the bounding surface must be larger than the respective sizes of the loading surface, and the loading surface should generally contract at a faster rate (the calibrated parameters given in Chapter 5 adhere to this constraint). If experimental results demand a faster bounding surface contraction, then the user must ensure that the bounding surface size does not decrease below the loading surface size.

The incremental movement of the bounding surface due to isotropic hardening is given by:

$$\eta_{BS} = \frac{dR_{BS}}{(R_{BS})_{old}} = \frac{(R_{BS})_{old} - (R_{BS})_{new}}{(R_{BS})_{old}} \quad (4.51)$$

The bounding surface contracts by moving the centroid of the bounding surface in the direction of the vector pointing from the centroid to the force point, $\{S\}$ (Fig. 4.4):

$$\{dA_{BS}\}_{iso} = \eta_{BS} \cdot (\{S\} - \{A_{BS}\}) \quad (4.52)$$

If the force point (and thus the loading surface) ever contact the bounding surface, the isotropic hardening formulation remains the same as shown in Eqs. (4.46) to (4.52). Additionally, when the bounding surface increases in size (i.e., $(R_{BS})_{new} > (R_{BS})_{old}$), Eq. (4.51) produces a negative value of η_{BS} . In this case, the formulation in Eq. (4.52) moves the centroid away from the force point. When the bounding surface decreases in size, the centroid moves toward the force point as shown in Fig. 4.4.

4.5 Variation of κ_2 and E_c

As will be shown in Chapter 5, the values of the κ_2 parameters must increase as plastic work accumulates in order to simulate experimental results for cyclically loaded CFT specimens. This is especially true for CFT sections with low D/t ratios, as will be discussed in Chapter 5.

Alternately, the concrete elastic modulus, E_c , must decrease during plastic loading to simulate the degrading behavior of CFT beam-columns as the concrete cracks in tension and crushes in compression (Section 3.3.2). Beginning with the initial contact between the force point and the loading surface, E_c decreases to a final value equal to $0.3 \cdot E_c$ as discussed in Section 3.3.2.

The rate at which both parameters change from their initial value to their final value is based upon the same plastic work-based exponential equation used for isotropic hardening (e.g., Eq. (4.40)), but with different exponential rates. The updated value of κ_2 for each step is given by:

$$(\kappa_2)_{\text{new}} = (\kappa_2)_{\text{final}} - [(\kappa_2)_{\text{final}} - (\kappa_2)_{\text{init}}] \cdot e^{-\xi_{\kappa_2} \cdot \Omega_p} \quad (4.53)$$

where ξ_{κ_2} denotes the calibrated rate of change in κ_2 .

For the concrete elastic modulus, E_c , the increase is given by a similar equation.

Using the initial and final values given in Section 3.3.2:

$$(E_c)_{\text{new}} = 0.3 \cdot E_c - [0.3 \cdot E_c - E_c] \cdot e^{-\xi_{E_c} \cdot \Omega_p} \quad (4.54)$$

where ξ_{E_c} denotes the calibrated rate of change in E_c .

4.6 Kinematic Hardening

In the bounding surface formulation presented in this work, the loading surface and bounding surface are isotropically hardened first, and then the surfaces are hardened kinematically. In this way, when the surfaces translate, it is with respect to the surface radii and the centroid location after isotropic hardening, and the consistency condition (which must also be enforced in the kinematic hardening formulation) is not violated. Consequently, the two hardening methods are essentially decoupled (this approach was first proposed by Zhao [1993] after determination that satisfying the consistency condition during simultaneous isotropic and kinematic hardening is quite complex in orthotropic force space).

The formulation for the kinematic hardening of the loading and bounding surfaces requires both a magnitude and a direction of translation. Both the loading surface and the bounding surface are assumed to move in the same direction in this work, but the magnitude of their respective movement differs. Only when the loading surface contacts the bounding surface do the two surfaces move at the same rate.

The direction of kinematic hardening may be specified by a number of different methods [Dafalias and Popov, 1975; Armen, 1978; McDowell, 1985; Hilmy and Abel, 1985]. One of the first methods was proposed by Prager [1956] for stress-space models. In Prager's method, the loading surface translates in the direction of the normal to the surface at the stress point. Ziegler [1959] modified Prager's hardening rule by specifying that the loading surface move in the direction of the vector from the centroid of the surface to the force point. These methods were used primarily with single surface models. In 1967, Mroz [1967], proposed finding the conjugate point on the bounding surface and translating the loading surface in the direction of a vector pointing from the force point to the conjugate bounding surface point. Recall that the distance between the

loading surface and the bounding surface (Section 4.3.2) was calculated along this vector. Additionally, Zhao [1993] used the Mroz kinematic hardening rule in his bounding surface model for steel elements. The Mroz direction of surface movement is retained for the steel elements contained in this work (see Zhao [1993] for further details on this method).

Tseng and Lee [1983] provided an alternative translation direction which has been shown to produce superior results for stress-space plasticity of both metals [McDowell, 1985] and concrete [Fardis et al., 1983]. Based on observations of the experimental results of Phillips [Phillips and Weng, 1975; Phillips and Lee, 1979], Tseng postulated that the loading surfaces generally translate kinematically in the direction of the incremental stress vector. For CFT elements, the Tseng rule of kinematic hardening is adopted to three-dimensional force-space in this work [Tseng and Lee, 1983].

The Tseng method is illustrated schematically in Fig. 4.5. The surfaces kinematically harden in a direction that is obtained as follows. The incremental force vector, $\{dS\}$, is extended to the bounding surface. The intersection of the extended incremental force vector and the bounding surface is represented by point F' in Fig. 4.5. The conjugate point on the loading surface is then located--point F in Fig. 4.5. The vector from point F to point F' denotes the Tseng direction, $\{\nu\}$, the direction in which the surfaces kinematically translate. One feature of this formulation is that this direction of motion is optimal for insuring that the loading surface never overlaps the bounding surface (presuming both are of the same shape).

Algebraically, the formulation for the Tseng kinematic hardening direction is represented in force-space by the following equations for spherical loading surfaces [Tseng and Lee, 1983]. First, the distance between the beginning of step force point and the point F' on the bounding surface is computed using the law of cosines:

$$\rho_{ds} = \left\{ \left[\left(\{S\} - \{A_{BS}\} \right)^T \cdot \frac{\{dS\}}{\|\{dS\}\|} \right]^2 + (R_{BS})^2 - \left\| \left(\{S\} - \{A_{BS}\} \right)^T \right\|^2 \right\}^{\frac{1}{2}} \quad (4.55)$$

$$- \left(\{S\} - \{A_{BS}\} \right)^T \cdot \frac{\{dS\}}{\|\{dS\}\|}$$

where $\|\{ \} \|$ represents the magnitude of the vector $\{ \}$. The unit vector, $\{\tau\}$, denotes the direction from the centroid of the bounding surface to point F', as shown in Fig. 4.5, and is given by:

$$\{\tau\} = \frac{\{S\} + \rho_{ds} \cdot \frac{\{dS\}}{\|\{dS\}\|} - \{A_{BS}\}}{\left\| \{S\} + \rho_{ds} \cdot \frac{\{dS\}}{\|\{dS\}\|} - \{A_{BS}\} \right\|} \quad (4.56)$$

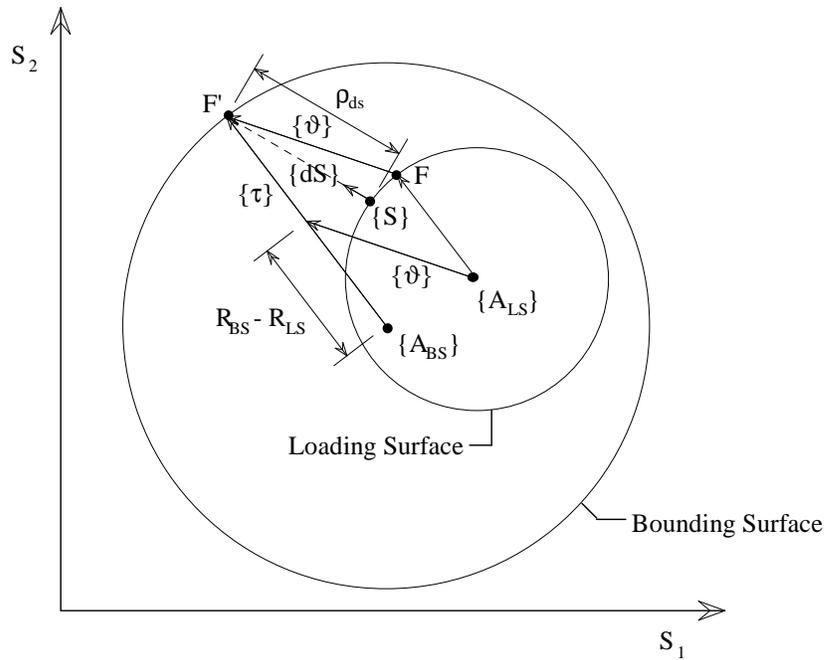


Figure 4.5 Kinematic Hardening by the Tseng Method

Finally, the Tseng direction unit vector, $\{\vartheta\}$, the vector denoting the direction of kinematic hardening, is illustrated in Fig. 4.5 and is given by:

$$\{\vartheta\} = \frac{(R_{BS} - R_{LS}) \cdot \{\tau\} - (\{A_{LS}\} - \{A_{BS}\})}{\|(R_{BS} - R_{LS}) \cdot \{\tau\} - (\{A_{LS}\} - \{A_{BS}\})\|} \quad (4.57)$$

In the above formulation, the radii of the loading and bounding surfaces are used in the equations. Because the CFT surfaces are not spherical like the surfaces in the model for which Tseng originally proposed these equations (isotropic metal plasticity), the use of radii does not give appropriate results in Eqs. (4.55) to (4.57). Therefore, in this work, a different approach is used to obtain the Tseng vector, $\{\vartheta\}$.

In the computer implementation, the radii of the surfaces need not be used and the calculations may be performed in unnormalized force space. The first step in this formulation entails locating the intersection of the incremental force vector extension with the bounding surface. Successive increments are added to the incremental force vector until the total vector length intersects the bounding surface. The increment that crosses the bounding surface is then bisected to locate the point of intersection within a tolerance. Given the intersection point, denoted now in vector notation, $\{F'\}$, the Tseng vector may be calculated as the vector from the conjugate point on the loading surface, $\{F\}$, to the bounding surface intersection point, $\{F'\}$:

$$\{\vartheta\} = \{F'\} - \{F\} \quad (4.58)$$

where the conjugate point on the loading surface, $\{F\}$, is given by:

$$\{F\} = (\{F'\} - \{A_{BS}\}) \cdot \left(\frac{R_{LS}}{R_{BS}} \right) + \{A_{LS}\} \quad (4.59)$$

The ratio of the surface sizes, $\frac{R_{LS}}{R_{BS}}$, is applicable regardless of the shape of the surfaces

(i.e., Eq. (4.59) holds for the non-spherical CFT surfaces as well as spherical surfaces).

The amount of kinematic hardening that the loading surface undergoes is a function of the loading surface gradient, the incremental force vector, and the Tseng direction vector, $\{\vartheta\}$, and is obtained by satisfying the consistency condition [Zhao, 1993]:

$$\{dA_{LS}\}_{kin} = \frac{\{n\}^T \cdot \{dS\}}{\{n\}^T \cdot \{\vartheta\}} \cdot \{\vartheta\} \quad (4.60)$$

The bounding surface moves in the same direction, $\{\vartheta\}$, as the loading surface, but by a smaller amount, which is a function of the amount of kinematic hardening the loading surface undergoes and the terms of the plastic stiffness matrix (see Section 4.3 for their derivation). The incremental bounding surface movement due to kinematic hardening is given by [Zhao, 1993]:

$$\{dA_{BS}\}_{kin} = [\Lambda] \cdot \{dA_{LS}\}_{kin} \quad (4.61)$$

where the magnitude of translation is given by:

$$[\Lambda] = \text{diag} \left(\frac{(k_p^b)_p}{(k_p)_p}, \frac{(k_p^b)_{my}}{(k_p)_{my}}, \frac{(k_p^b)_{mz}}{(k_p)_{mz}} \right) \quad (4.62)$$

The k_p^b terms represent the bounding stiffness terms and are calculated by Eq. (4.32) as presented in Section 4.3.1. The plastic stiffness terms will always be greater than or

equal to these values, producing terms in $[\Lambda]$ that are less than one. Therefore, the bounding surface will move at a slower rate than the loading surface, and, with continued loading in one direction, the loading surface will eventually contact the bounding surface. At the point of contact, k_p will equal k_p^b and the two surfaces translate at the same rate.

Once contact occurs, the surfaces move together in the direction of the normal to the loading surface (this equation results from equating $\{\vartheta\}$ to $\{n\}$ in Eq. (4.60)):

$$\{dA_{LS}\}_{kin} = \{dA_{BS}\}_{kin} = \frac{\{n\}^T \cdot \{dS\}}{\|\{n\}^T\|} \cdot \{n\} \quad (4.63)$$

For plastic displacements beyond the contact point, the plastic stiffness remains equal to the bounding stiffness, $[k_p^b]$.

The isotropic and kinematic hardening may be summarized in a single expression representing the total movement of the surface centroids. For the loading surface:

$$\{dA_{LS}\} = \{dA_{LS}\}_{iso} + \{dA_{LS}\}_{kin} \quad (4.64)$$

Similarly, for the bounding surface:

$$\{dA_{BS}\} = \{dA_{BS}\}_{iso} + \{dA_{BS}\}_{kin} \quad (4.65)$$

To reiterate, the surfaces are referenced by the backforce vectors, $\{a_{LS}\}$ and $\{a_{BS}\}$, corrected by the offset vectors, $\{\Phi_{LS}\}$ and $\{\Phi_{BS}\}$, to account for the asymmetry of the surfaces (see Section 4.1.1). Equations (4.64) and (4.65) also presume that isotropic hardening is performed first, followed by kinematic hardening, which is based upon the updated radii and surface centroids.

Following isotropic and kinematic hardening, force point drift control is performed to fulfill the consistency condition [Hilmy et al., 1987]. Using a bisection algorithm, the force point is returned to the loading surface (within a specified tolerance) in the direction of the normal to the loading surface. Using a sufficiently small load step size, the force point never drifts far from the loading surface. This is, however, an essential attribute of the plasticity formulation analysis, since the assumptions contained within it require the consistency condition to be maintained.

Chapter 5

Verification and Calibration

Verification of the fully-nonlinear frame analysis program developed in this research will be presented in three sections. Further details of this program, CFTmacro, are outlined in Appendix B and C. The steel plasticity formulation will be verified first. The steel formulation, based on the work of Zhao [1993], uses his calibrated parameters, and the examples will be compared to his results. Additionally, these examples verify the geometric nonlinear formulation and general analysis system. The inelastic analysis of CFT beam-columns will then be calibrated and verified. The CFT examples include both monotonic and cyclic studies. A select group of CFT beam-column sections with varying material strength, D/t ratio, and method of applied loading will first be examined. The CFT plasticity model is calibrated to these tests, and the procedure and results are documented. Additional monotonic examples are then presented to verify the model and the calibration parameters. The final study consists of a comparison of the analytical results to experimental data for a cyclically-loaded three-dimensional subassembly consisting of steel I-girders framing from four sides into a CFT beam-column.

5.1 Verification of the Steel Plasticity Analysis

The main application of the program contained in this work is the analysis of one-way or two-way unbraced steel or composite frames. This section examines the accuracy of the program in modeling the material nonlinear behavior of steel beam-column elements required for such frame studies. In addition, the results presented in this section serve to verify the accuracy of the geometric nonlinear formulation (which is the same for the CFT beam-column element). The steel plasticity formulation is based on the work of Zhao [1993]. The two examples presented in this section were presented by Zhao [1993] and use the calibration parameters that he recommended in his work.

5.1.1 Steel Cantilever Beam

The first steel example consists of a W8×31 cantilever subjected to axial load, major axis bending, and minor axis bending, each load applied separately (Fig. 5.1). This example served as a calibration study for Zhao [1993]. Only kinematic hardening was performed; the surfaces were not isotropically hardened. The parameters used in Zhao's work are retained for the analyses performed here and are as follows:

$$R_{LS} = 0.5 \quad R_{BS} = 1.0 \quad \kappa_1 = 0.001 \quad \kappa_2^P = 8.0 \quad \kappa_2^{my} = 5.0 \quad \kappa_2^{mz} = 11.0$$

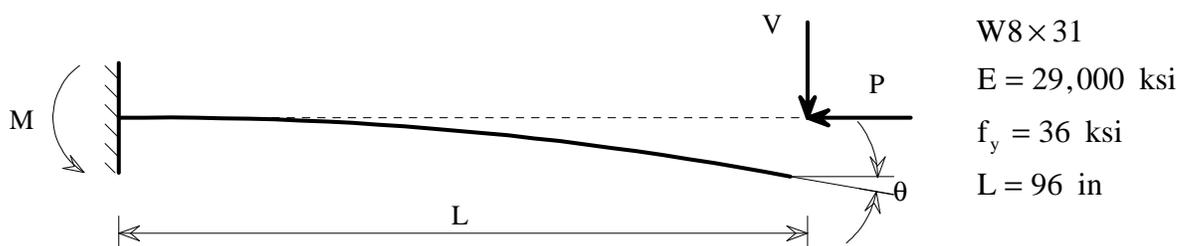


Figure 5.1 Cantilever Beam

The program results shown in the Figs. 5.2 through 5.4 were obtained using four elements along the member length, as per Zhao [1993]. The first load case, the results of which are shown in Figure 5.2, consists of an axial load applied to the end of the member. The beam-column is restrained in the minor axis direction, forcing buckling to occur in the major axis plane of the member. Figures 5.3 and 5.4 illustrate a comparison of the results of CFTmacro to Zhao's proposed model for major and minor axis bending of the beam-column. In these cases, only the transverse load, V , is applied. All results match Zhao's.

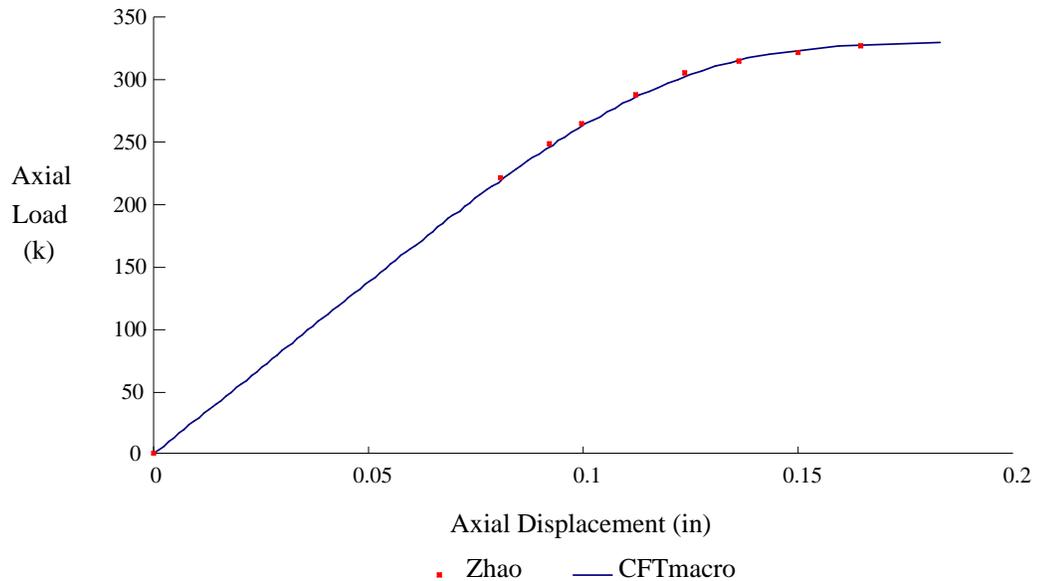


Figure 5.2 Steel Cantilever: Axial Load vs. Displacement

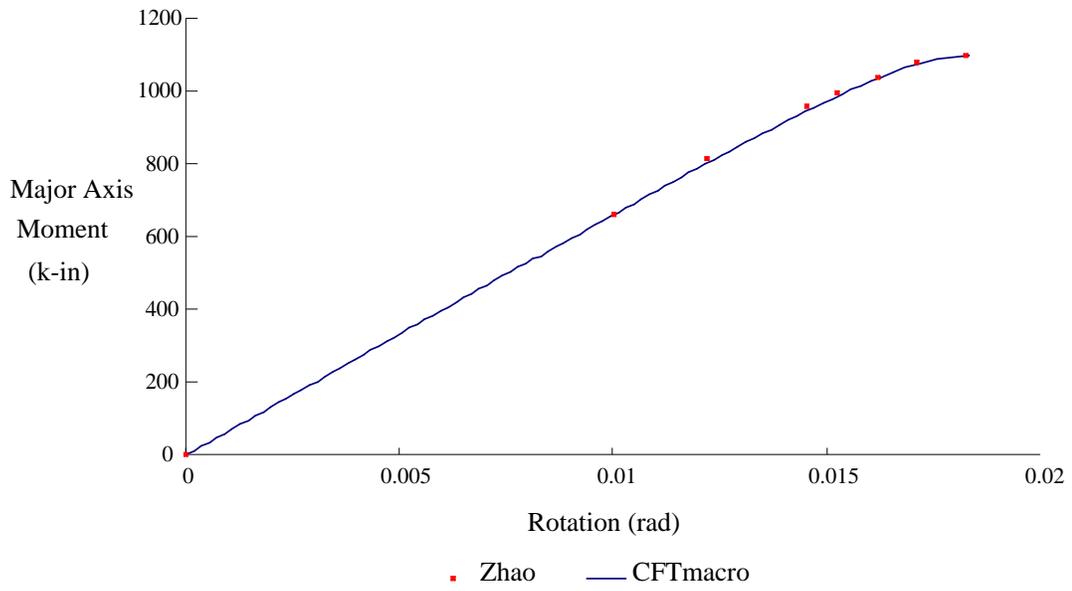


Figure 5.3 Steel Cantilever: Major Axis Moment vs. Rotation

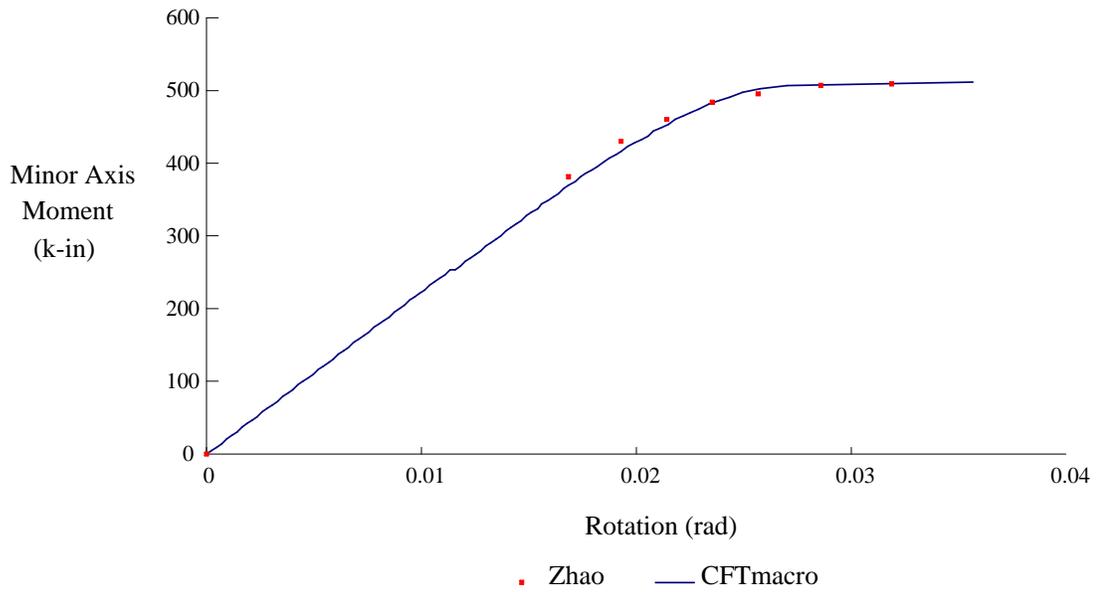


Figure 5.4 Steel Cantilever: Minor Axis Moment vs. Rotation

5.1.2 Steel Beam-Column

The second steel plasticity verification problem examines an axially-loaded beam-column with varying end conditions [Zhao, 1993]. For each end condition (pinned-pinned, pinned-fixed, and fixed-fixed) the beam-column is alternately loaded in major axis bending and minor axis bending. A W8×31 with an initial imperfection of $L/1000$ is used for the beam-column (Fig. 5.5). Isotropic hardening is not included; the beam-column is divided into four elements; and the surface radii and plasticity parameters, κ_1 and κ_2 , used in Example 5.1.1 are used in this example as well [Zhao, 1993].

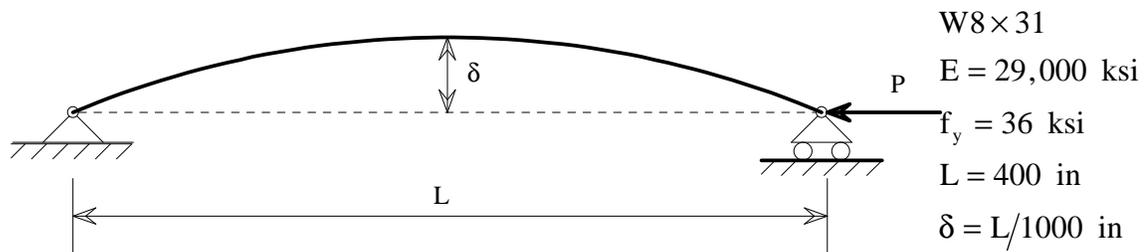


Figure 5.5 Beam-Column

Figures 5.6 and 5.7 illustrate the excellent correlation between the program results and Zhao's results for major axis and minor axis bending, respectively. Zhao, in turn, demonstrated that his proposed model provides sufficient accuracy when compared to a more detailed finite element analysis [Zhao, 1993].

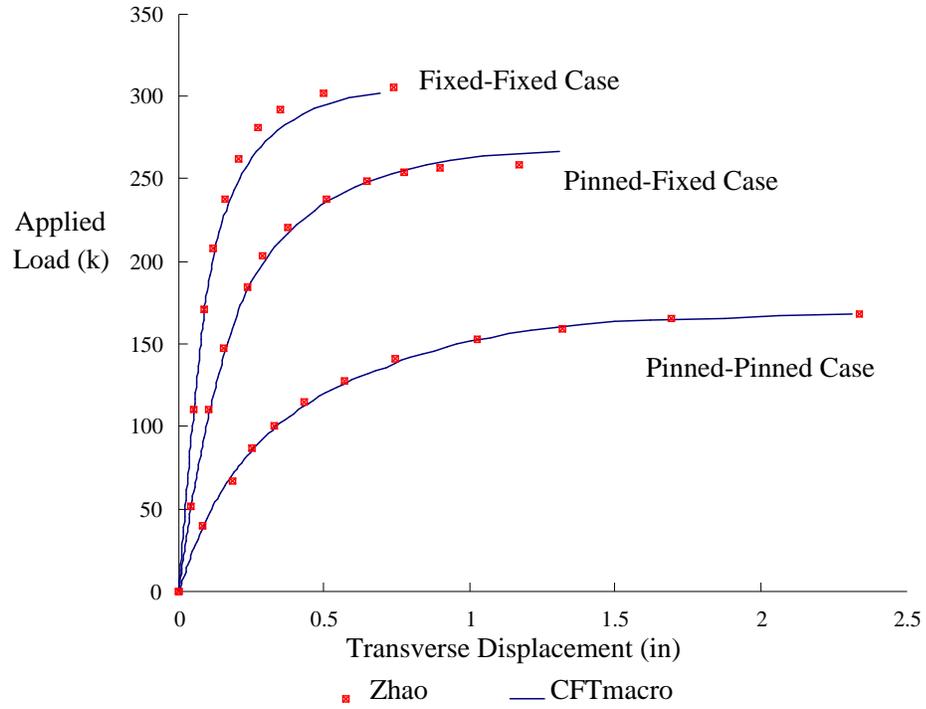


Figure 5.6 Steel Beam-Column: Major Axis Bending

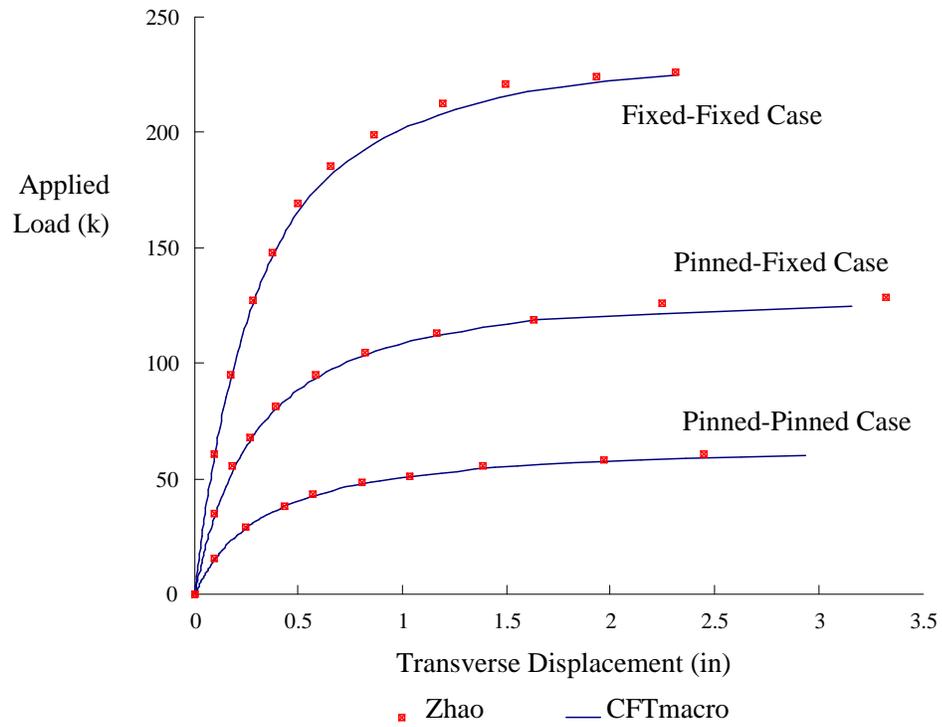


Figure 5.7 Steel Beam-Column: Minor Axis Bending

5.2 CFT Plasticity Analysis Calibration

The calibration of the CFT plasticity model was performed by comparing the analytical results of CFTmacro with several different experimental studies of individual CFT beam-columns. The experimental calibration studies were selected to provide the widest range of test variables (e.g., material strength, D/t ratio, and the method of applied loading). Section 5.2.1 reviews the plasticity parameters that require calibration and itemizes the CFT tests that were used in the procedure. Section 5.2.2 discusses the procedure of selecting the parameter values and presents the final calibration values.

5.2.1 Calibration Parameters and Tests

The calibration parameters that were introduced in Chapter 4 are summarized in Table 5.1. Each parameter is listed along with its corresponding symbol.

Table 5.1 List of Calibration Parameters

Parameter	Symbol
Initial Loading Surface Radius	$(R_{LS})_{init}$
Final Loading Surface Radius	$(R_{LS})_{final}$
Isotropic Hardening Rate of L.S.	ξ_{LS}
Initial Bounding Surface Radius	$(R_{BS})_{init}$
Intermediate Bounding Surface Radius	$(R_{BS})_{intm}$
Final Bounding Surface Radius	$(R_{BS})_{final}$
Isotropic Hardening Rate of B.S.	ξ_{LS}
Normalized Plastic Work at $(R_{BS})_{intm}$	$(\Omega_p)_{intm}$
κ_1 Parameters	$\kappa_1^p, \kappa_1^{my}, \kappa_1^{mz}$
Initial κ_2 Parameters	$(\kappa_2^p)_{init}, (\kappa_2^{my})_{init}, (\kappa_2^{mz})_{init}$
Final κ_2 Parameters	$(\kappa_2^p)_{final}, (\kappa_2^{my})_{final}, (\kappa_2^{mz})_{final}$
Rate of κ_2 Increase	ξ_{κ_2}
Rate of E_c Decrease	ξ_{Ec}

For the calibration to CFT experimental tests, it was desired to obtain a wide range of beam-column test sections. Appendix A tabulates papers containing experimental test results for rectangular CFTs. Tables A.2 and A.3, respectively, list monotonic and cyclic CFT papers and the number of tests for each paper; each table is categorized by concrete strength, f'_c , and D/t ratio. The tests that were used for the calibration of the CFT element model are listed in Table 5.2 along with their salient geometric and material parameters. The group of tests includes 3 cyclic tests and 8 monotonic tests, 4 of which were loaded proportionally (including 3 uniaxial and 1 biaxial test), and 4 of which were loaded non-proportionally (i.e., a constant axial load is applied, followed by a gradual increase in the applied bending load). The 3 cyclic tests were all loaded non-proportionally. The figure reference given in column 2 of Table 5.2 refers to the appropriate figure, Fig. 5.8, 5.9, or 5.10, illustrating the type of test. These figures show the schematic representation of the analytical structural model and the applied loading for the three basic types of experimental tests used in the calibration. Four elements per member were used for the tests represented by Figs. 5.8 and 5.9. For CFT members represented by Fig. 5.10, two elements per member were used in the analytical model. For rectangular beam-columns, the L/D and D/t ratios are tabulated for both the major and minor axis directions. The first number denotes the D/t or L/D value with respect to the dimension in the plane of the applied eccentric load and the second number, in parentheses, denotes the out-of-plane value. Additionally, tubes that were annealed to remove residual stresses are noted in the table.

The sections in Table 5.2 represent the widest range of parameters currently available in the CFT literature [Gourley and Hajjar, 1993]. As mentioned in Chapter 1, the applicability of the calibration parameters presented in this work is therefore necessarily limited in scope to the range given by the sections in Table 5.2.

Table 5.2 CFT Calibration Tests

Test (Notation)	Type of Test* (Figure)	Dimen- sions (in.)	L/D	D/t	f'_c (ksi)	f_y (ksi)	Other Data**
Bridge, 1976 (Br3)	Ecc. Load, Pr Monotonic (Fig. 5.8)	8.0 × 8.0 × 0.395	10.5	20.0	5.0	45.4	e = 1.5 in. $\alpha = 30^\circ$
Cederwall et al., 1990 (Ced2)	Ecc. Load, Pr Monotonic (Fig. 5.8)	4.7 × 4.7 × 0.20	25.0	24.0	6.7	63.5	e = 0.79 in
Cederwall et al., 1990 (Ced9)	Ecc. Load, Pr Monotonic (Fig. 5.8)	4.7 × 4.7 × 0.32	25.0	15.0	14.9	55.0	e = 0.79 in
Shakir-Khalil, 1991 (SK91_4)	Ecc. Load, Pr Monotonic (Fig 5.8)	5.9 × 3.9 × 0.20	25.5 (38.3)	30.0 (20.0)	5.8	50.8	e = 1.77 in
Tomii and Sakino, 1979a (Tom24_3)	Bm-Col, NPr Monotonic (Fig 5.9)	3.9 × 3.9 × 0.17	3.0	24.0	3.2	41.5	$P/P_o =$ 0.29 annealed
Tomii and Sakino, 1979a (Tom24_6)	Bm-Col, NPr Monotonic (Fig 5.9)	3.9 × 3.9 × 0.17	3.0	24.0	3.4	41.5	$P/P_o =$ 0.57 annealed
Tomii and Sakino, 1979a (Tom44_3)	Bm-Col, NPr Monotonic (Fig 5.9)	3.9 × 3.9 × 0.087	3.0	44.0	3.75	49.2	$P/P_o =$ 0.26 annealed
Tomii and Sakino, 1979a (Tom44_6)	Bm-Col, NPr, Monotonic (Fig 5.9)	3.9 × 3.9 × 0.087	3.0	44.0	3.75	42.0	$P/P_o =$ 0.57 annealed
Sakino and Tomii, 1981 (Sak24_2)	Cyclic Shear, NPr (Fig 5.10)	3.9 × 3.9 × 0.164	6.0	24.0	3.5	42.7	$P/P_o =$ 0.20 annealed
Sakino and Tomii, 1981 (Sak34_3)	Cyclic Shear, NPr (Fig. 5.10)	3.9 × 3.9 × 0.116	6.0	34.0	4.0	42.7	$P/P_o =$ 0.30 annealed
Sakino and Tomii, 1981 (Sak46_5)	Cyclic Shear, NPr (Fig 5.10)	3.9 × 3.9 × 0.086	6.0	46.0	4.0	42.7	$P/P_o =$ 0.50 annealed

* *Pr* denotes proportional loading; *NPr* denotes non-proportional loading

** *e* denotes the applied load eccentricity; α denotes angle of applied load relative to centroidal axis of cross-section for biaxially-loaded specimens

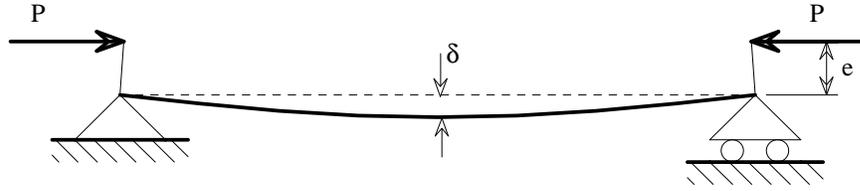


Figure 5.8 Eccentrically-Loaded Beam-Column

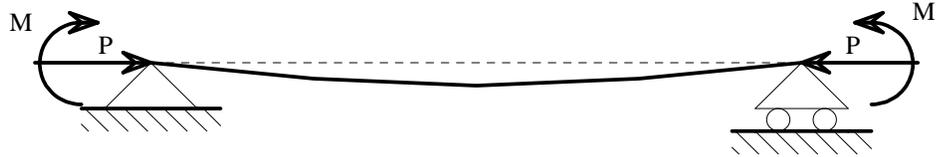


Figure 5.9 Axially-Loaded Beam in Bending

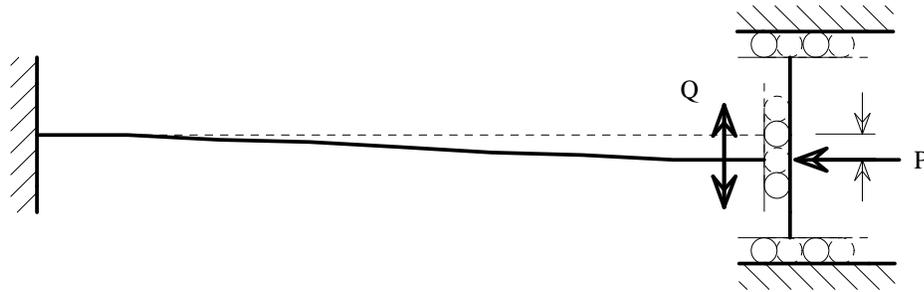


Figure 5.10 Cyclically-Loaded Shear Specimen

5.2.2 Calibration Procedure and Results

Initial Loading Surface Size

The initial size of the loading surface was the first parameter to be calibrated. Recall that the initial loading surface represents the locus of force points at which inelastic behavior begins. To simplify the plasticity formulation and insure that the surfaces do not overlap, the loading surface is a scaled version of the surface representing the CFT section's cross-section strength (see Section 4.1.1). The initial yield surface of an actual CFT, however, takes a different shape, as illustrated in Fig. 5.11 (only the

positive force quadrant in two dimensions is shown). The initial loading surface, LS_1 , represents one possible locus of force points at which steel yielding or concrete crushing initiates. This surface, LS_1 , was obtained using a radius, R_{LS} , equal to the ratio to the yield moment of the steel tube to the nominal moment of the CFT, M_{yld}/M_o . For pure bending, the actual initial yield condition is represented well by this loading surface. For mid-range values of axial load, however, the loading surface overshoots the initial yield surface, predicting higher forces than are actually required to cause plasticity. Additionally, for higher D/t ratios or higher f'_c/f_y ratios (Fig 5.11 illustrates a section with $D/t = 20.0$ and $f'_c/f_y = 0.1$), the bulge in the loading surface is even more conspicuous (see Figs. 2.25 through 2.32 in Section 2.6), while the initial yield surface remains almost linear, since only the steel dictates the initial yield for nearly all cross-sections. Since the objective of the CFT study in this work focuses on beam-columns existing in frame structures, these observations suggest that, for the mid-range axial load values typical in frame applications, the loading surface should be further scaled back from M_{yld}/M_o based upon the D/t ratio and the f'_c/f_y ratios of the section. The loading surface labeled LS_2 in Fig. 5.11 illustrates this result.

Based on the above discussion, the initial loading surface radius was calibrated by optimizing the loading surface size for each of the tests of Table 5.2 and then developing a linear relation between the ratio of the radius and the relative axial capacity of the concrete to the axial capacity of the entire section, represented by $P_{co}/(P_{co} + P_{so})$. This term implicitly accounts for both the D/t ratio and the f'_c/f_y ratio. The calibrated equation for the initial radius of the loading surface is expressed as:

$$(R_{LS})_{init} = \frac{M_{yld}}{M_o} - 0.30 \cdot \frac{P_{co}}{P_{co} + P_{so}} \quad (5.1)$$

where M_o is calculated by Eqs. (2.27) and (2.28) in Section 2.6.7 and $M_{yld} = f_y \cdot S$, where S is the section modulus of the steel tube in the plane of bending. The axial compression capacity of the steel and concrete, respectively, are $P_{so} = A_s \cdot f_y$ and $P_{co} = A_c \cdot f'_c$. For CFT sections with a low D/t and a low f'_c/f_y ratio (e.g., Tom24_3), the steel will dominate the section behavior and a loading surface scaled to M_{yld}/M_o will be accurate. In this case, Eq. (5.1) produces a value very near M_{yld}/M_o . As the D/t ratio and the f'_c/f_y ratio increase, a larger value is subtracted from M_{yld}/M_o , effectively moving the bulge in the surface for mid-range axial force closer to the actual initial yield surface.

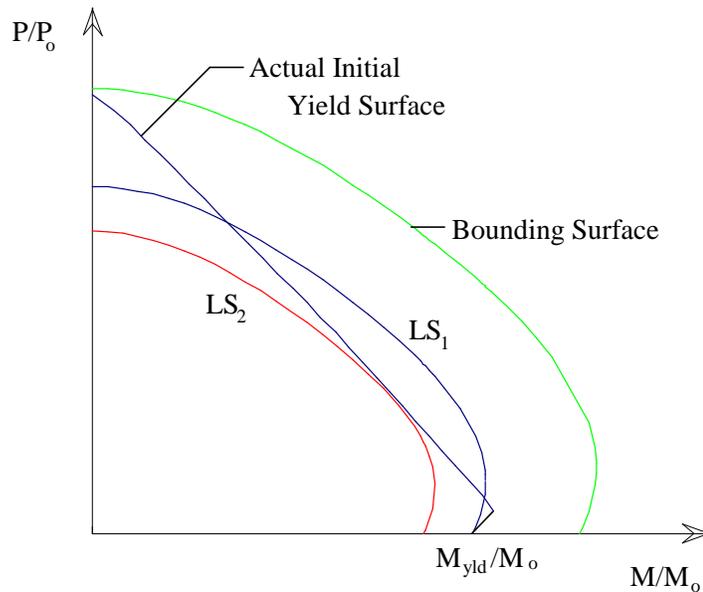


Figure 5.11 Initial Loading Surface Size and Shape

Initial Value of κ_2

The initial values of the κ_2 coefficients were calibrated based on the relative strengths of the concrete and steel, thus accounting implicitly for both the D/t ratio and

the f'_c/f_y ratio. Using a methodology similar to that used to obtain Eq. (5.1), the initial axial κ_2 coefficient value is given as:

$$\left(\kappa_2^p\right)_{init} = \frac{P_{co}}{P_{so}} \quad (5.2)$$

The κ_2 coefficient values for the moment terms are equal to half of the axial value:

$$\left(\kappa_2^{my}\right)_{init} = \left(\kappa_2^{mz}\right)_{init} = \frac{\left(\kappa_2^p\right)_{init}}{2.0} \quad (5.3)$$

Initial Bounding Surface Size

After specifying an initial loading surface radius and κ_2 coefficients, the initial bounding surface size was calibrated using primarily the monotonic test results. The bounding surface was set to an initial value of 1.0 to represent the cross-section strength of a CFT; this value proves quite accurate for the tests of Table 5.2.

Value of κ_1

A value of κ_1 (the bounding slope of the load-deflection curve) equal to 0.0010 produces accurate results. The non-zero value of the bounding slope results from the continued strength gain observed in CFTs due to gradual strain hardening of the steel tube [Sakino and Tomii, 1981]. A constant value of κ_1 accurately portrays the observed bounding slope of CFT specimens. This slope does not vary substantially with continued loading since strain hardening of the steel tube occurs gradually over the cross-section.

Final Loading Surface Size

Having established the initial calibration values, the cyclic tests were used to calibrate the remaining parameters. Cyclic tests indicate that the radius of the loading surface and the corresponding size of the elastic zone decrease to a very small value due to extensive concrete crushing and steel local buckling (however, as discussed in Section 3.5.2, the elastic zone rarely vanishes completely). For the plasticity model a minimum, or final radius, $(R_{LS})_{final}$, of 0.10 was thus established. The isotropic softening rate, ξ_{LS} , was calibrated to be 1.0×10^1 . For the cyclic tests in Table 5.2, this rate shrinks the loading surface to its final size several cycles before the completion of the test.

Final Value of κ_2

In conjunction with a decrease in the loading surface size, the κ_2 coefficients simultaneously change from their initial to final values. A calibrated rate of $\xi_{\kappa_2} = 3.0 \times 10^1$ provides the best results. The final κ_2 values are a function of the initial κ_2 values and may be larger or smaller than this initial value depending upon the section properties. The linear relationship between initial and final values is expressed by the following calibrated equations:

$$\begin{aligned} (\kappa_2^p)_{final} &= 5.0 - 4.5 \cdot (\kappa_2^p)_{init} \\ (\kappa_2^m)_{final} &= (\kappa_2^p)_{final} / 2.0 \end{aligned} \quad (5.4)$$

For sections with a low D/t ratio or a low f'_c/f_y ratio, and therefore a low value of $(\kappa_2)_{init}$ (see Eqs. (5.2) and (5.3)), κ_2 will increase. This simulates experimental CFT load-deflection behavior which, despite the decrease in the size of the elastic zone, exhibits relatively little strength degradation, due to strain hardening and confinement of the concrete [Sakino and Tomii, 1981]. Reexamining Fig. 4.2 in Section 4.3.1 illustrates that

a higher value of κ_2 results in less strength degradation for low to moderate magnitudes of displacement.

If the CFT section has a higher D/t ratio or lower steel strength, less strain hardening and concrete confinement will occur and the section will exhibit a more severe degradation. In this case, κ_2 will decrease with continued plasticity, resulting in a lower strength for low to moderate magnitudes of displacement (refer again to Fig. 4.2).

Intermediate and Final Bounding Surface Sizes

As discussed in Section 4.4.2, the bounding surface first increases to model cyclic strain hardening and then decreases at a calibrated value of plastic work to model concrete crushing and steel local buckling. Therefore, initial, intermediate, and final bounding surface sizes are required, as well as an isotropic hardening rate and a value of normalized plastic work at which the bounding surface begins to decrease. The same rate of isotropic hardening, $\xi_{BS} = 5.0 \times 10^1$, was used for the isotropic hardening and isotropic softening. This value was constant for all tests, as were the intermediate bounding surface size, $(R_{BS})_{intm}$ and the normalized plastic work at which the bounding surface begins to shrink, $(\Omega_p)_{intm}$. These two values were calibrated to be 1.4 and 0.020, respectively. Upon reaching the maximum size, the bounding surface decreases toward a final size, which is given by the following calibrated equation:

$$(R_{BS})_{final} = 1.3 - 0.88 \cdot \frac{P_{co}}{P_{so}} \quad (5.5)$$

The final bounding surface size becomes smaller as the D/t ratio increases or the strength of the steel decreases. This models the decreased effect of strain hardening and concrete confinement and the increased severity of local buckling corresponding to a smaller quantity of steel or a lower steel strength.

Rate of Change of E_c

The final calibrated parameter is the rate of decrease of the elastic concrete modulus from its initial value (Eq. (3.6)) to its final reduced value (Eq. (3.5)). This value, ξ_{Ec} , must rapidly decrease to model the early tensile cracking of the concrete and the degradation of concrete stiffness. A relatively high rate of 3.0×10^3 was thus established. This results in a degradation of the concrete elastic modulus, E_c , even at low magnitudes of plastic displacement.

Table 5.3 summarizes the calibration values and formulas presented above. Note that these equations apply only to sections within the range of experimental studies incorporated in this work (see Section 1.3).

Table 5.3 Final Calibrated Parameters

Parameter	Initial Value	Intermediate Value	Final Value	Rate, ξ
R_{LS}	$\frac{M_{yld}}{M_o} - 0.30 \cdot \frac{P_{co}}{P_{co} + P_{so}}$	--	0.10	$\xi_{LS} = 1.0 \times 10^1$
R_{BS}	1.0	1.4	$1.3 - 0.88 \cdot \frac{P_{co}}{P_{so}}$	$\xi_{BS} = 5.0 \times 10^1$
$(\Omega_p)_{intm}$	0.020	--	0.020	--
$\kappa_1^p, \kappa_1^{my}, \kappa_1^{mz}$	0.0010	--	0.0010	--
κ_2^p	$\frac{P_{co}}{P_{so}}$	--	$5.0 - 4.5 \cdot (\kappa_2^p)_{init}$	$\xi_{\kappa_2} = 3.0 \times 10^1$
$\kappa_2^{my}, \kappa_2^{mz}$	$\frac{(\kappa_2^p)_{init}}{2.0}$	--	$\frac{(\kappa_2^p)_{final}}{2.0}$	$\xi_{\kappa_2} = 3.0 \times 10^1$
E_c	E_c	--	$0.3 \cdot E_c$	$\xi_{Ec} = 3.0 \times 10^3$

Results of Calibration Studies

The following figures illustrate the comparison between the analytical and experimental load-deflection curves for each of the calibration tests listed in Table 5.2. Each test is referenced by the notation given in column 1 of Table 5.2. As a whole, these figures illustrate the applicability of the calibrated parameters of Table 5.3 to a wide range of experimental tests. Figure 5.12 illustrates a biaxially-loaded specimen, which was as accurate as the uniaxial tests. The mid-height deflection shown in Fig. 5.12 denotes the deflection in the plane of the applied loading. Both proportional (Figs. 5.12 to 5.15) and non-proportional tests (Figs 5.16 to 5.19) produced equally accurate results.

The cyclic tests shown in Figs. 5.20 to 5.22 illustrate the accuracy of the rate parameters and the final values of the loading surface, bounding surface, and κ_2 parameters. Better results could have been obtained for these tests had they each been calibrated individually. The objective of this work, however, is to establish calibration parameters of general applicability. The equations listed in Table 5.3 (Eqs. (5.4) and (5.5) specifically) were developed as linear functions of the optimum results from these three tests. This reemphasizes the need to recalibrate if sections with properties beyond this range of tests are analyzed, but also that this CFT formulation is quite accurate within this wide range of CFT sizes and strengths.

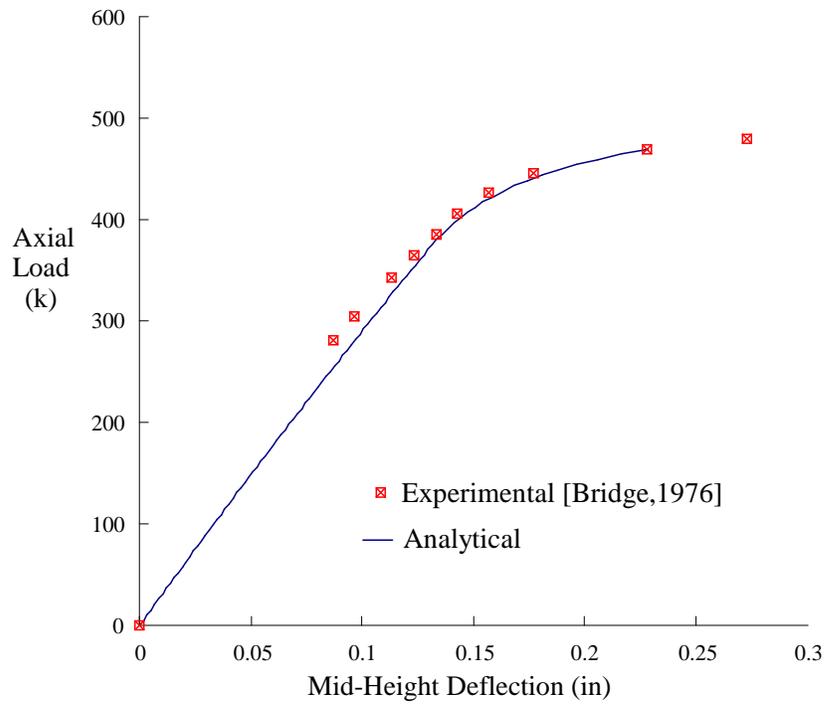


Figure 5.12 Eccentrically-Loaded Beam-Column (Br3)

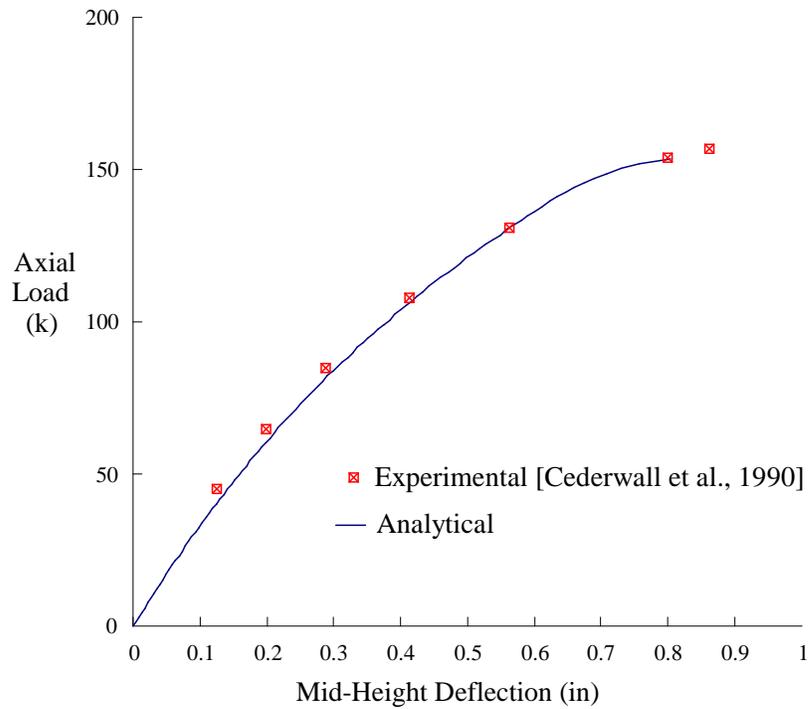


Figure 5.13 Eccentrically-Loaded Beam-Column (Ced2)

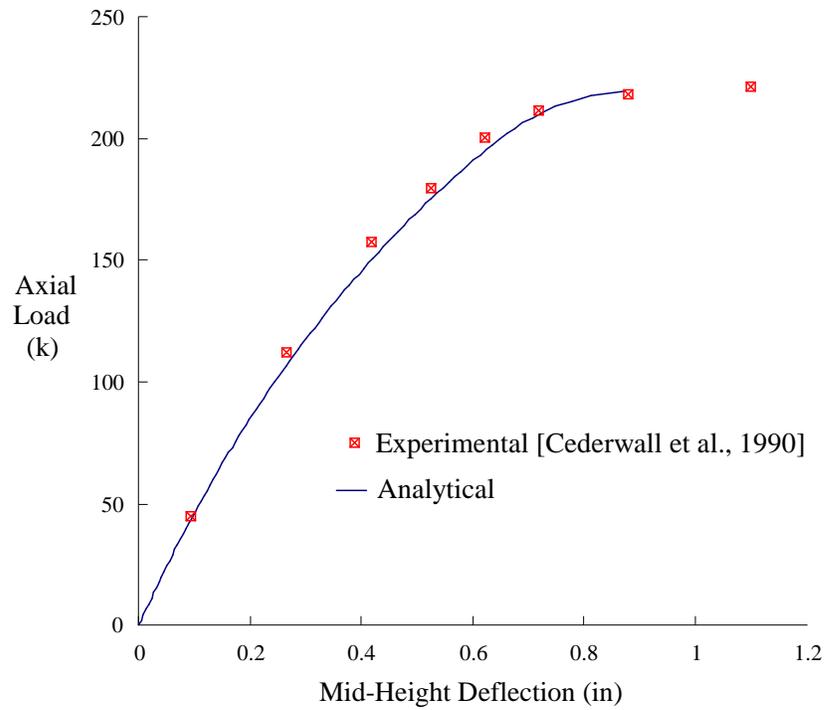


Figure 5.14 Eccentrically-Loaded Beam-Column (Ced9)

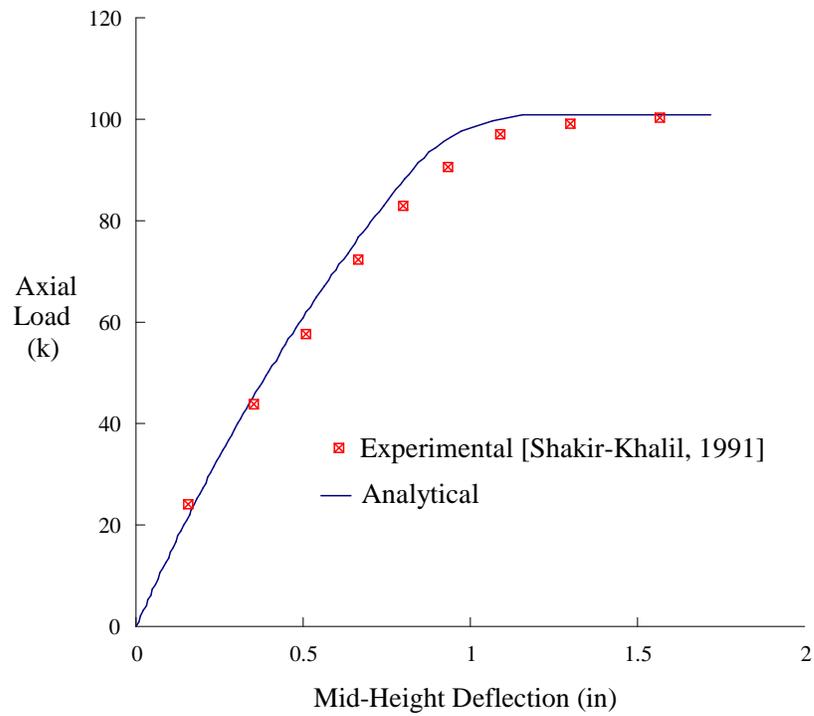


Figure 5.15 Eccentrically-Loaded Beam-Column (SK91_4)

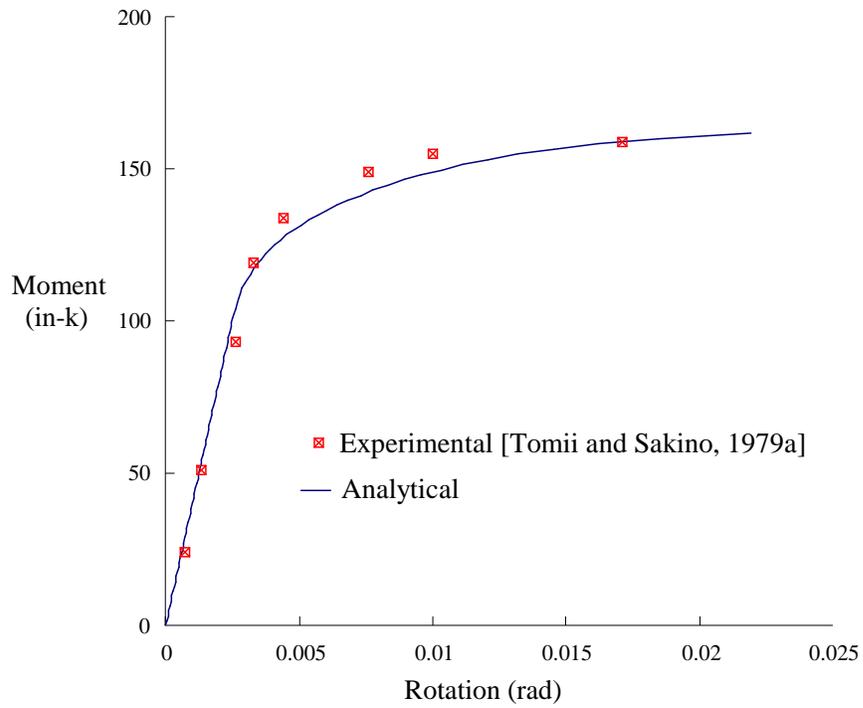


Figure 5.16 Axially-Loaded Beam in Bending (Tom24_3)

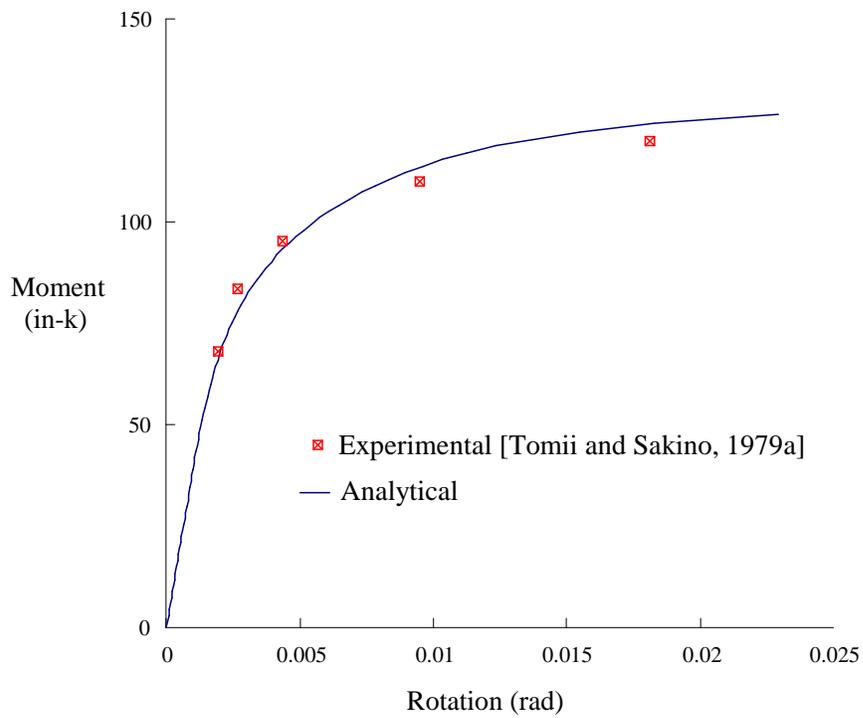


Figure 5.17 Axially-Loaded Beam in Bending (Tom24_6)

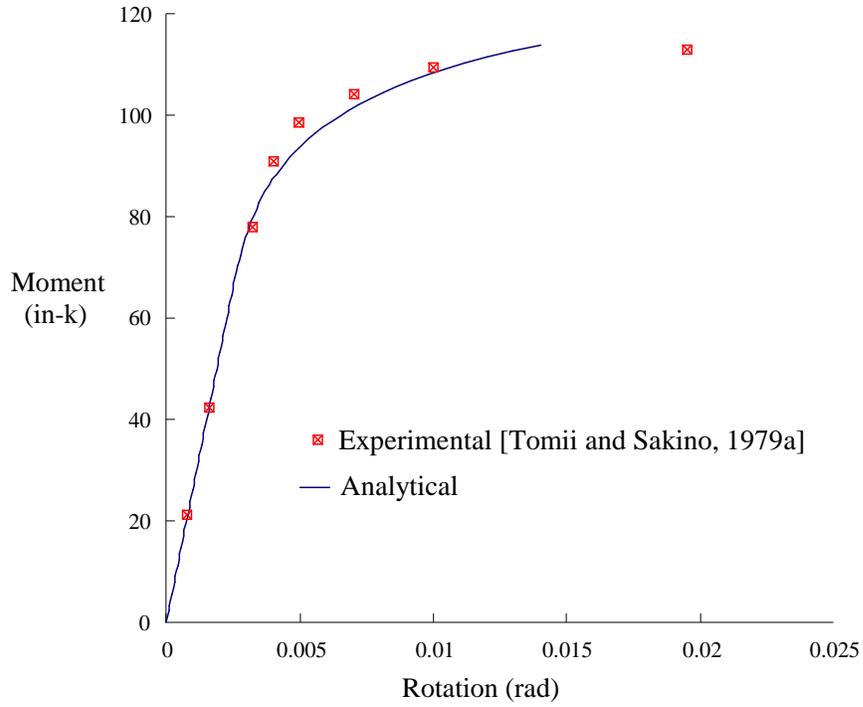


Figure 5.18 Axially-Loaded Beam in Bending (Tom44_3)

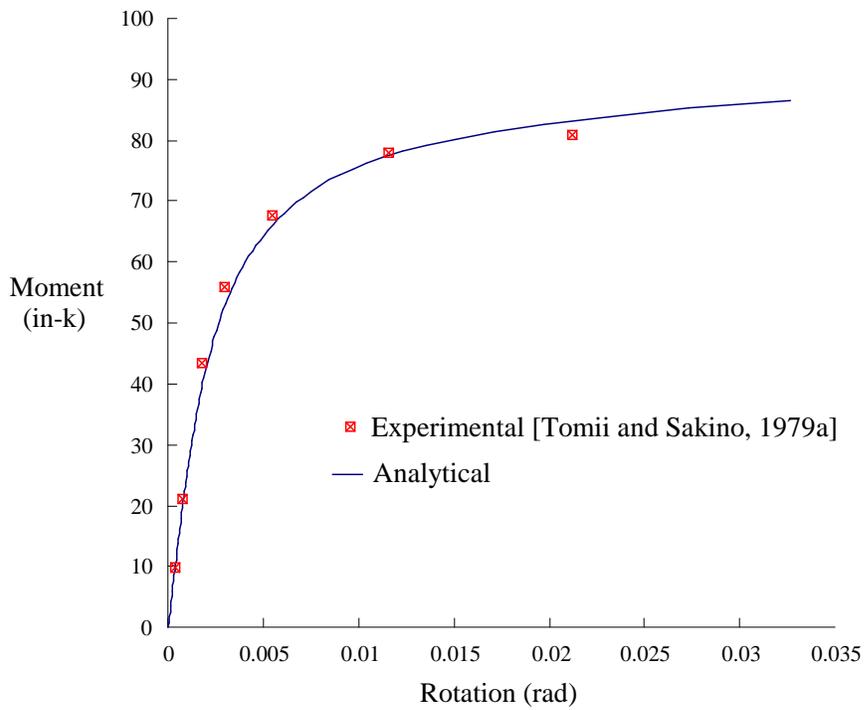


Figure 5.19 Axially-Loaded Beam in Bending (Tom44_6)

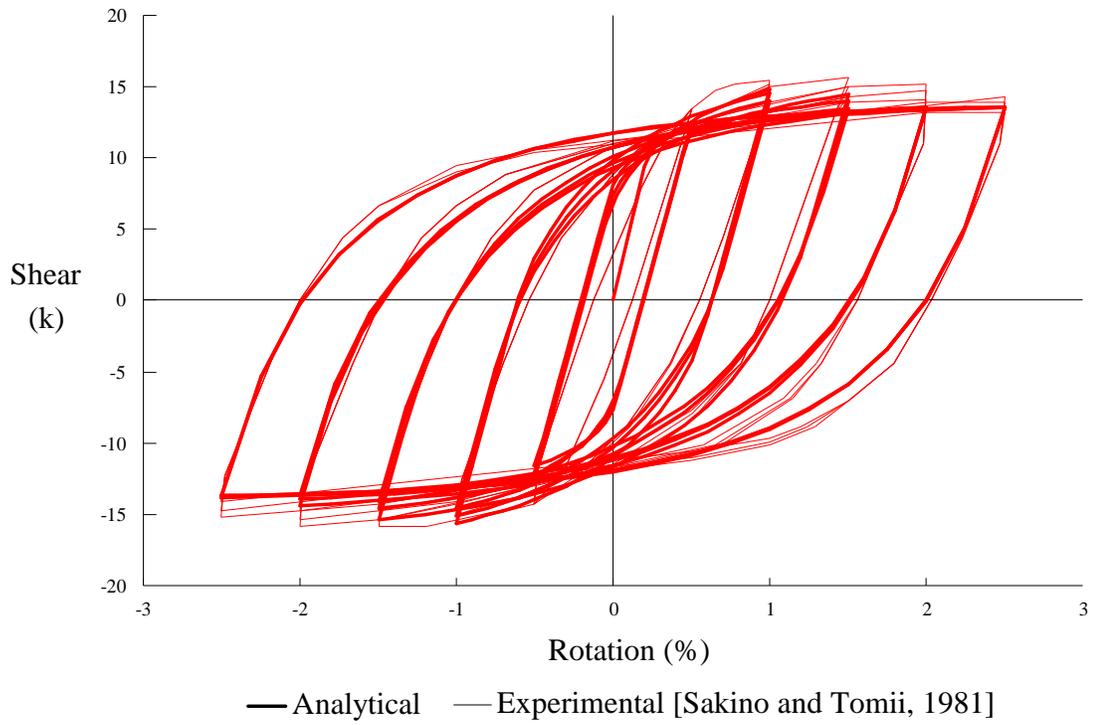


Figure 5.20 Cyclically-Loaded Shear Specimen (Sak24_2)

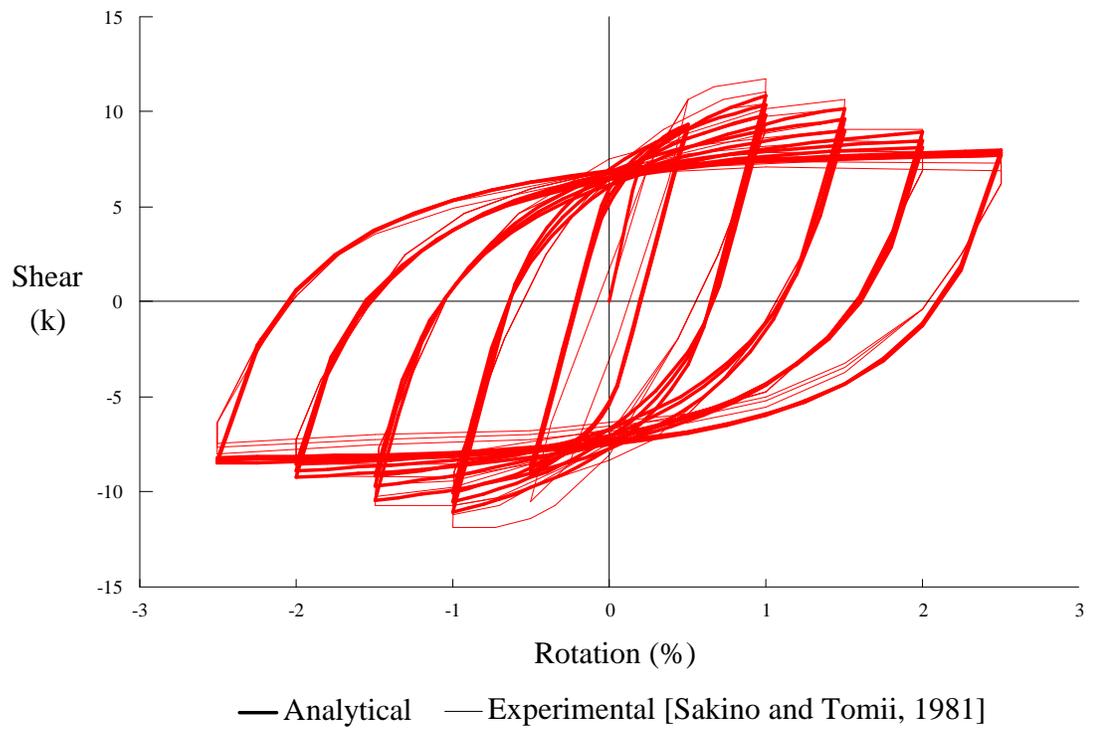


Figure 5.21 Cyclically-Loaded Shear Specimen (Sak34_3)

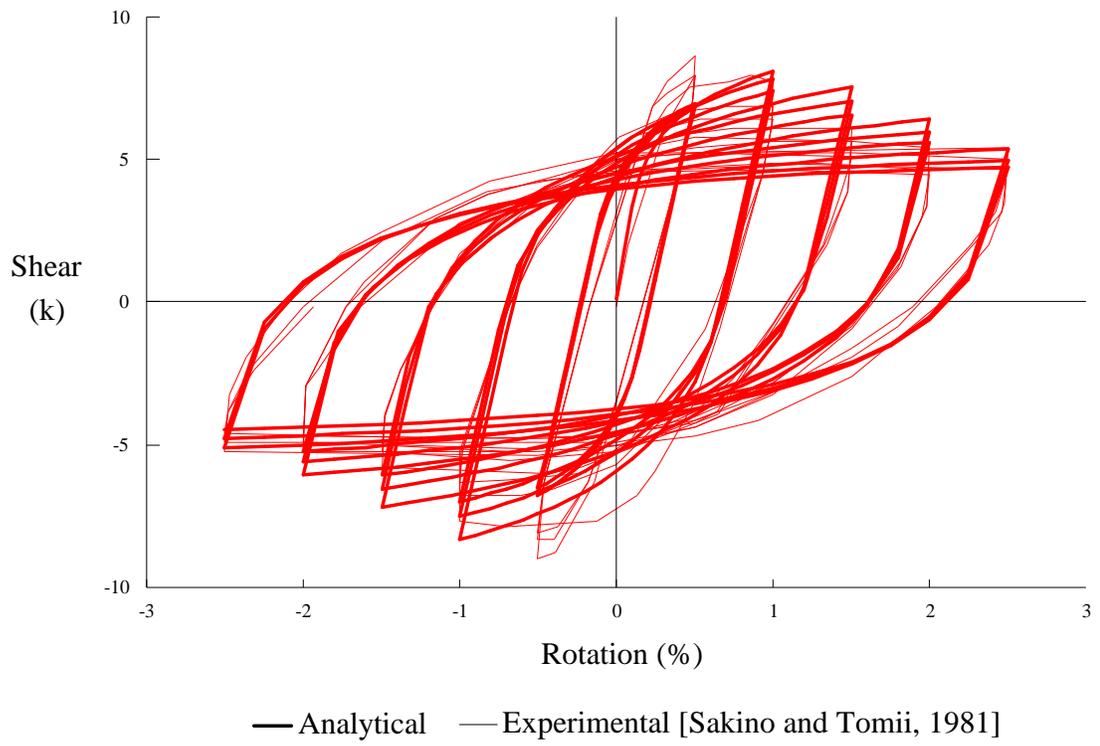


Figure 5.22 Cyclically-Loaded Shear Specimen (Sak46_5)

5.3 CFT Plasticity Analysis Verification

The verification of the CFT element involves comparing the analytical results to the results of a number of additional CFT beam-column tests without deviating from the calibration parameters of Table 5.3. This section contains additional monotonic tests, which together reflect a thorough range of test parameters within the bounds established by the tests of Table 5.2. Table 5.4 presents the CFT verification tests in the same format as Table 5.2. The figure number given in column 2 again refers to the appropriate analytical structural model illustrated in the previous section (i.e., Fig. 5.8, 5.9, or 5.10).

Figures 5.23 through 5.43 illustrate each test in turn, showing the analytical results versus the experimental results. Figures 5.23 through 5.28 contain eccentrically-loaded tests by Bridge [1976]. The maximum percent error in the axial load, P , for these tests was 12.7 % for specimen Br8 (Fig 5.28). The biaxially-loaded specimens--Br4, Br5, and Br6--are modeled well by the analytical formulation, showing a maximum error in the axial load of only 7.3 %.

Figures 5.29 through 5.33 illustrate eccentrically-loaded tests performed by Cederwall et al. [1990]. These tests, along with Ced2 and Ced9 in Section 5.2, illustrate the accuracy of the model for a range of concrete strengths. The maximum error in the axial load was 8.9 % (specimen Ced10--Fig. 5.32).

Figures 5.34 through 5.37 illustrate the analytical results versus the results from tests performed by Shakir-Khalil et al. [1989, 1991]. The maximum error between the analytical and experimental results was 10.5 % (specimen SK91_10--Fig 5.37). These tests provide the only rectangular (as opposed to square) sections in the verification, and they include relatively long members. Specimens SK89_2 and SK91_9 are loaded in the minor axis direction and specimens SK89_5 and SK91_10 in the major axis direction. The analytical results for these tests produced reasonably accurate results.

Table 5.4 CFT Verification Tests

Test (Notation)	Type of Test* (Figure)	Dimen- sions (in)	L/D	D/t	f'_c (ksi)	f_y (ksi)	Other Data**
Bridge, 1976 (Br1)	Ecc. Load, Pr Monotonic (Fig. 5.8)	8.0 × 8.0 × 0.392	10.5	20.0	4.4	42.2	e = 1.5 in.
Bridge, 1976 (Br4)	Ecc. Load, Pr Monotonic (Fig. 5.8)	7.9 × 7.9 × 0.394	10.5	20.0	4.9	45.4	e = 1.5 in. $\alpha = 45^\circ$
Bridge, 1976 (Br5)	Ecc. Load, Pr Monotonic (Fig. 5.8)	7.9 × 7.9 × 0.394	15.0	20.0	5.5	45.4	e = 1.5 in. $\alpha = 30^\circ$
Bridge, 1976 (Br6)	Ecc. Load, Pr Monotonic (Fig. 5.8)	8.0 × 8.0 × 0.385	15.0	20.0	4.7	45.4	e = 2.5 in. $\alpha = 45^\circ$
Bridge, 1976 (Br7)	Ecc. Load, Pr Monotonic (Fig. 5.8)	6.0 × 6.0 × 0.256	20.0	23.5	5.1	36.8	e = 1.5 in.
Bridge, 1976 (Br8)	Ecc. Load, Pr Monotonic (Fig. 5.8)	6.0 × 6.0 × 0.392	20.0	23.5	5.1	36.8	e = 2.5 in.
Cederwall et al, 1990 (Ced1)	Ecc. Load, Pr Monotonic (Fig. 5.8)	4.7 × 4.7 × 0.20	25.0	24.0	6.8	44.1	e = 0.79 in
Cederwall et al, 1990 (Ced6)	Ecc. Load, Pr Monotonic (Fig. 5.8)	4.7 × 4.7 × 0.32	25.0	15.0	6.7	43.5	e = 0.79 in
Cederwall et al, 1990 (Ced7)	Ecc. Load, Pr Monotonic (Fig. 5.8)	4.7 × 4.7 × 0.32	25.0	15.0	6.8	54.5	e = 0.79 in
Cederwall et al, 1990 (Ced10)	Ecc. Load, Pr Monotonic (Fig. 5.8)	4.7 × 4.7 × 0.32	25.0	15.0	5.7	55.0	e = 0.79 in
Cederwall et al, 1990 (Ced13)	Ecc. Load, Pr Monotonic (Fig. 5.8)	4.7 × 4.7 × 0.32	25.0	15.0	11.6	56.6	e = 0.79 in

Table 5.4 (cont'd)

Test (Notation)	Type of Test* (Figure)	Dimen- sions (in)	L/D	D/t	f_c (ksi)	f_y (ksi)	Other Data**
Shakir-Khalil and Zeghiche, '89 (SK89_2)	Ecc. Load, Pr Monotonic (Fig 5.8)	4.7 × 3.2 × 0.18	23.0 (34.5)	24.0 (16.0)	5.2	56.0	e = 0.95 in
Shakir-Khalil and Zeghiche, '89 (SK89_5)	Ecc. Load, Pr Monotonic (Fig 5.8)	4.7 × 3.2 × 0.18	34.5 (23.0)	16.0 (24.0)	5.6	49.8	e = 1.57 in minor axis bending
Shakir-Khalil, 1991 (SK91_9)	Ecc. Load, Pr Monotonic (Fig 5.8)	5.9 × 3.9 × 0.20	31.6 (47.4)	30.0 (20.0)	5.5	50.8	e = 2.95 in
Shakir-Khalil, 1991 (SK91_10)	Ecc. Load, Pr Monotonic (Fig 5.8)	5.9 × 3.9 × 0.20	40.0 (26.6)	20.0 (30.0)	5.9	53.4	e = 1.18 in minor axis bending
Tomii and Sakino, 1979a (Tom24_2)	Bm-Col, NPr Monotonic (Fig 5.9)	3.9 × 3.9 × 0.17	3.0	24.0	3.2	41.5	P/P _o = 0.19 annealed
Tomii and Sakino, 1979a (Tom24_4)	Bm-Col, NPr Monotonic (Fig 5.9)	3.9 × 3.9 × 0.17	3.0	24.0	3.2	41.5	P/P _o = 0.38 annealed
Tomii and Sakino, 1979a (Tom24_5)	Bm-Col, NPr Monotonic (Fig 5.9)	3.9 × 3.9 × 0.17	3.0	24.0	3.4	41.5	P/P _o = 0.48 annealed
Tomii and Sakino, 1979a (Tom44_2)	Bm-Col, NPr Monotonic (Fig 5.9)	3.9 × 3.9 × 0.087	3.0	44.0	3.75	49.2	P/P _o = 0.18 annealed
Tomii and Sakino, 1979a (Tom44_4)	Bm-Col, NPr Monotonic (Fig 5.9)	3.9 × 3.9 × 0.087	3.0	44.0	3.75	42.0	P/P _o = 0.38 annealed
Tomii and Sakino, 1979a (Tom44_5)	Bm-Col, NPr Monotonic (Fig 5.9)	3.9 × 3.9 × 0.087	3.0	44.0	3.75	42.0	P/P _o = 0.48 annealed

* *Pr* denotes proportional loading; *NPr* denotes non-proportional loading

** *e* denotes the applied load eccentricity; α denotes angle of applied load relative to centroidal axis of cross-section for biaxially-loaded specimens

Figures 5.38 to 5.43 show the results of the tests performed by Tomii and Sakino [1979a, 1979b] for a wide range of applied axial loads. The maximum percent error between the experimental and analytical moments for these tests was -8.1 % (specimen Tom24_2--Fig. 5.38). These tests, in which the specimens were loaded nonproportionally, illustrate the accuracy of the formulation for the initial loading surface size. The mid-range axial loads result in very accurate results while the low range tests produce slightly less accurate results. For a low value of axial load, the loading surface, LS_2 , in Fig. 5.12 predicts a lower yield moment, producing premature yielding in the analytical results.

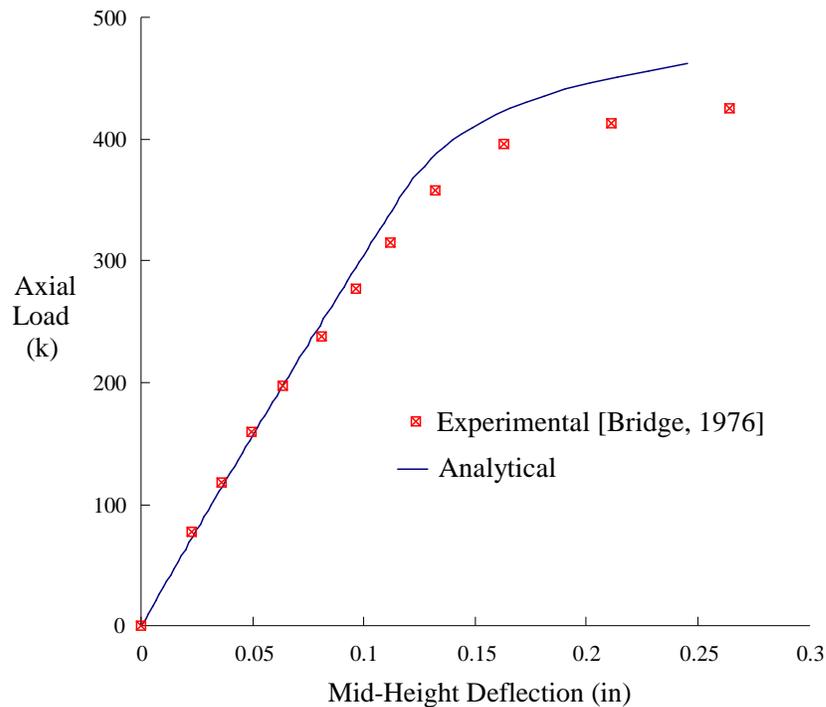


Figure 5.23 Eccentrically-Loaded Beam-Column (Br1)

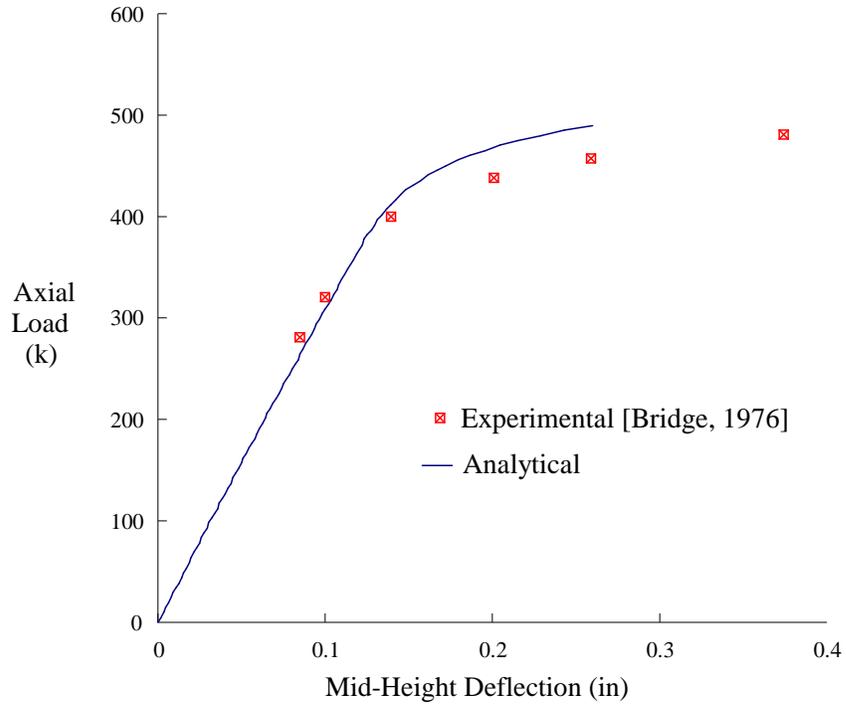


Figure 5.24 Eccentrically-Loaded Beam-Column (Br4)

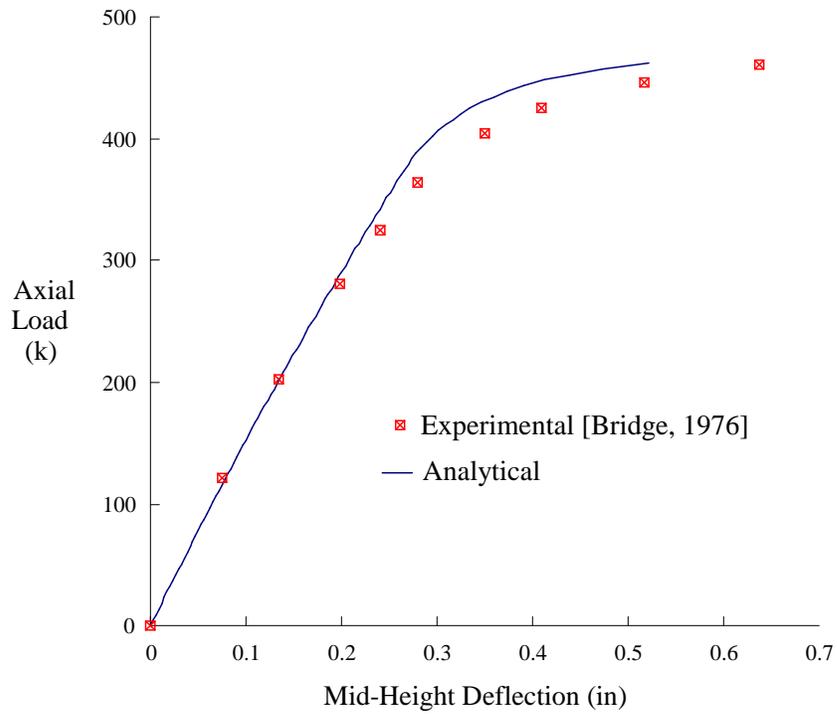


Figure 5.25 Eccentrically-Loaded Beam-Column (Br5)

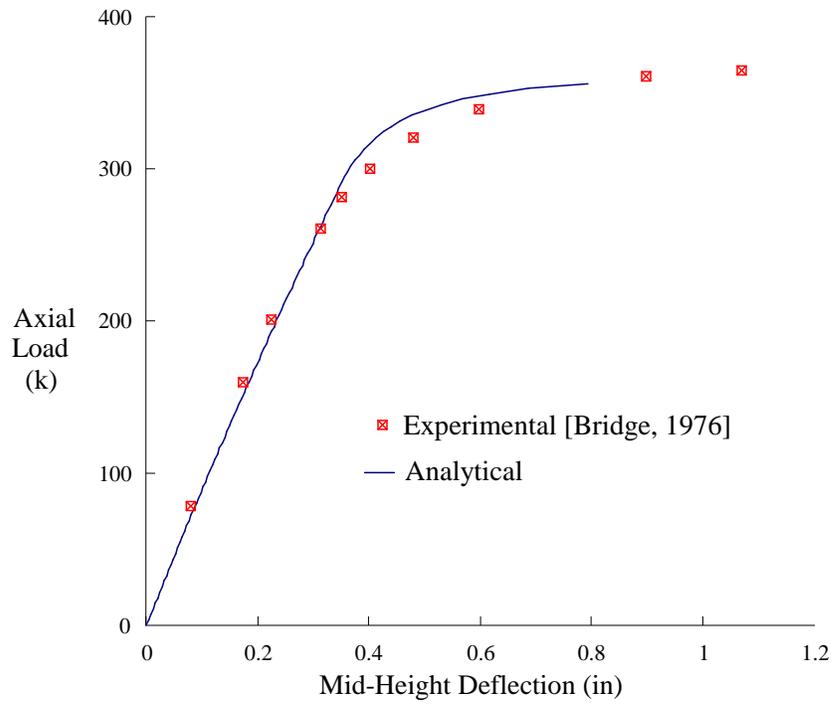


Figure 5.26 Eccentrically-Loaded Beam-Column (Br6)

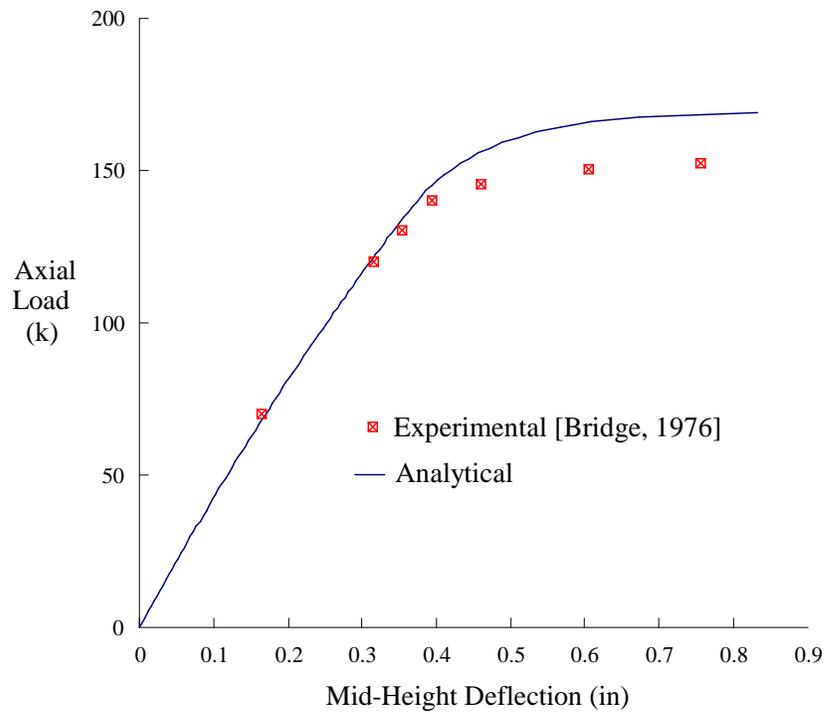


Figure 5.27 Eccentrically-Loaded Beam-Column (Br7)

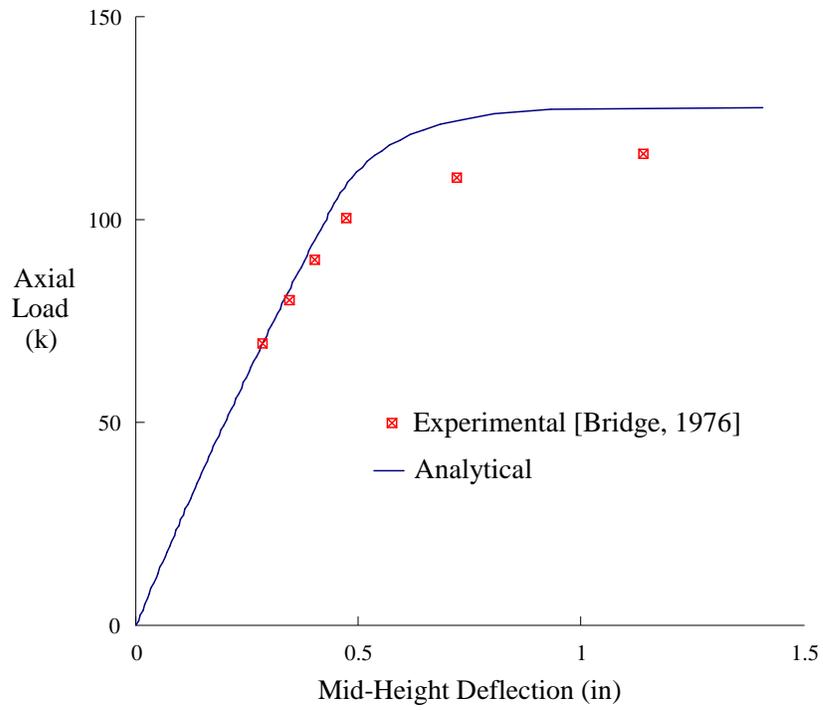


Figure 5.28 Eccentrically-Loaded Beam-Column (Br8)

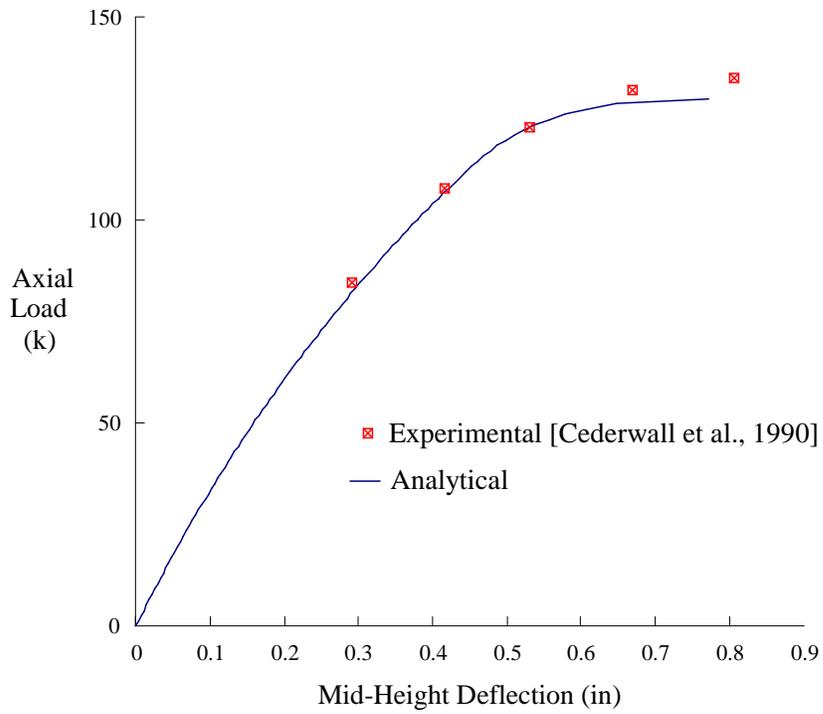


Figure 5.29 Eccentrically-Loaded Beam-Column (Ced1)

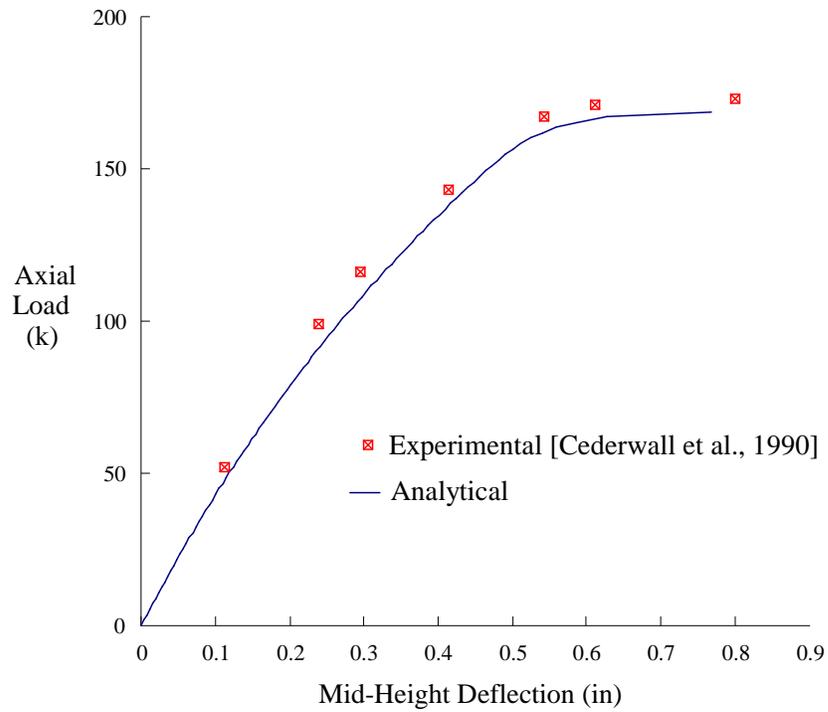


Figure 5.30 Eccentrically-Loaded Beam-Column (Ced6)

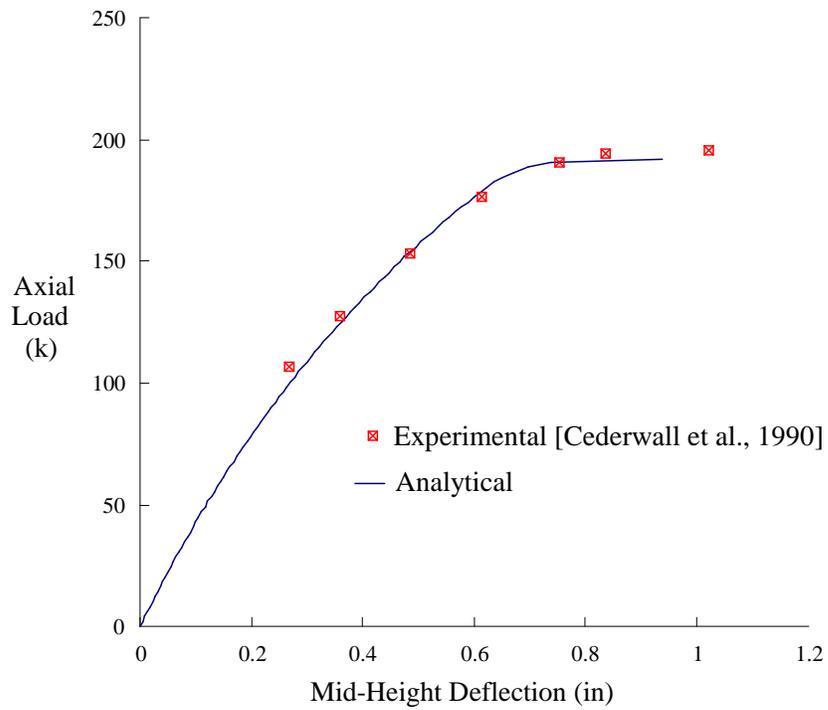


Figure 5.31 Eccentrically-Loaded Beam-Column (Ced7)

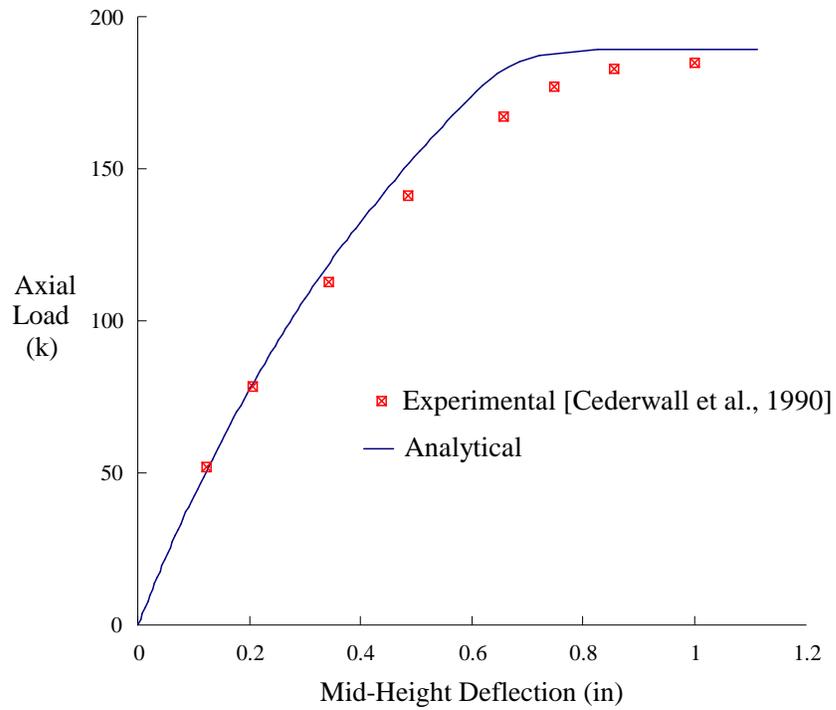


Figure 5.32 Eccentrically-Loaded Beam-Column (Ced10)

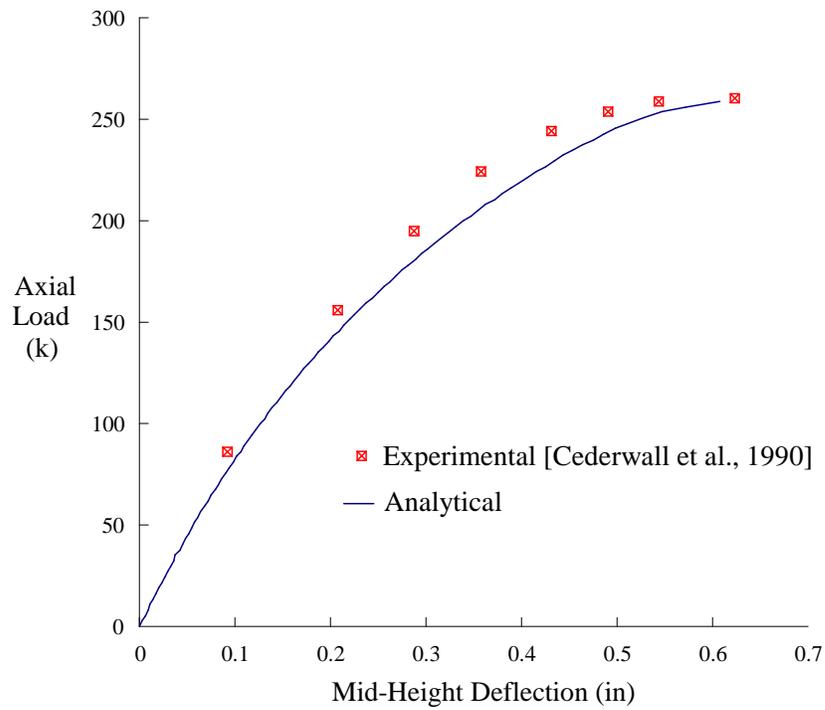


Figure 5.33 Eccentrically-Loaded Beam-Column (Ced13)

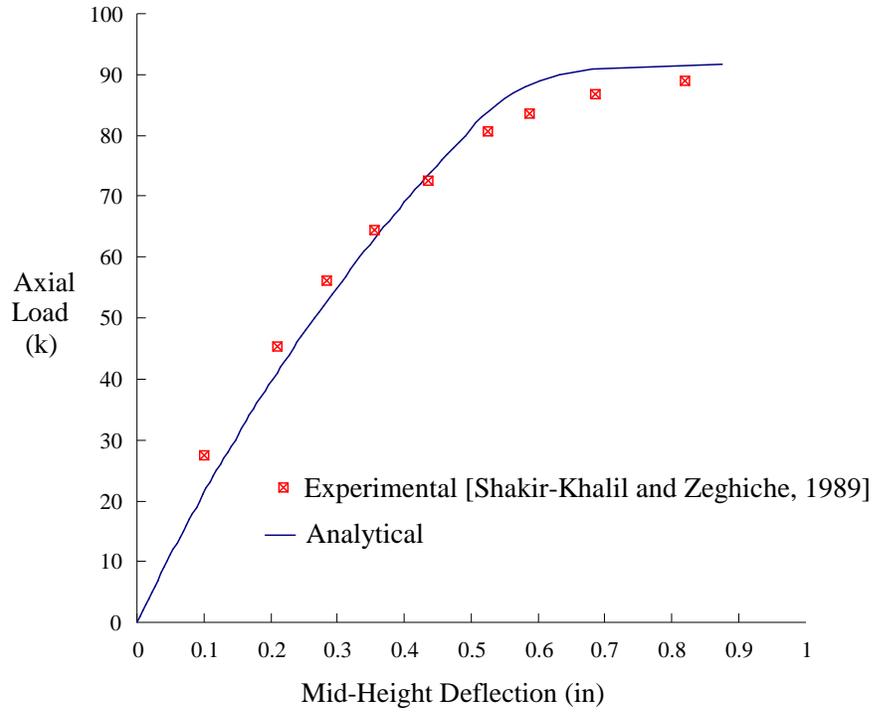


Figure 5.34 Eccentrically-Loaded Beam-Column (SK89_2)

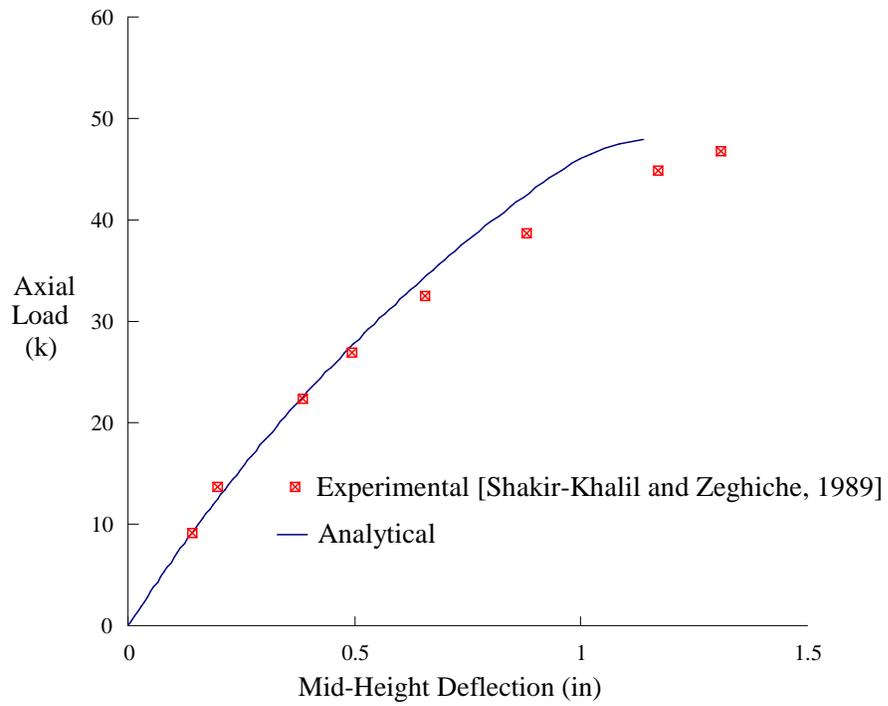


Figure 5.35 Eccentrically-Loaded Beam-Column (SK89_5)

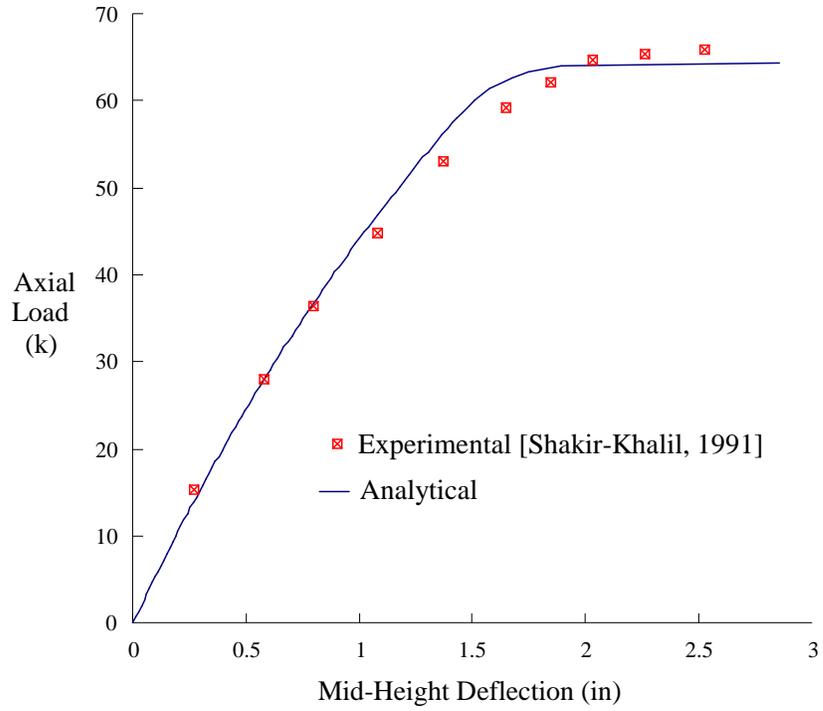


Figure 5.36 Eccentrically-Loaded Beam-Column (SK91_9)

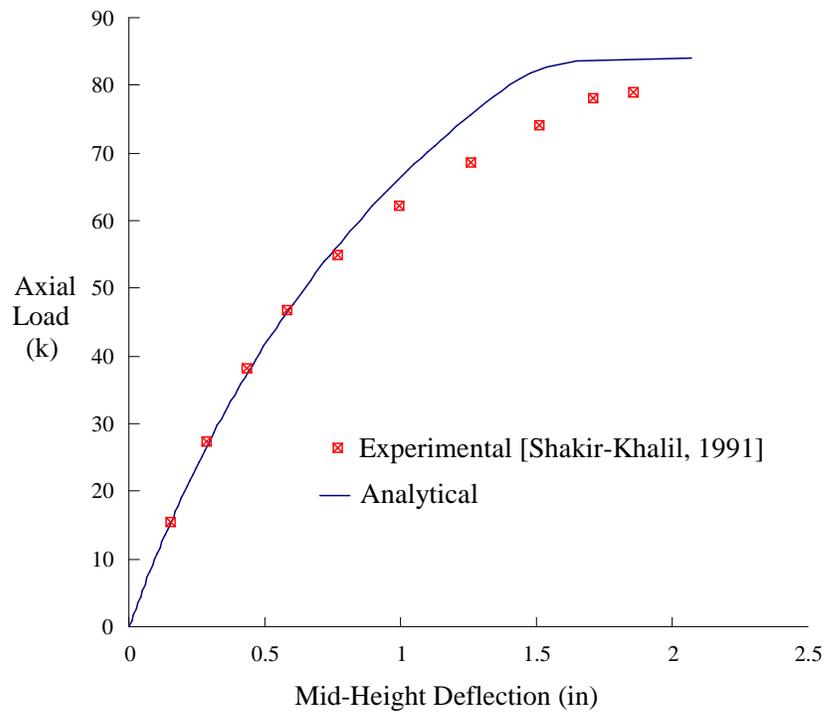


Figure 5.37 Eccentrically-Loaded Beam-Column (SK91_10)

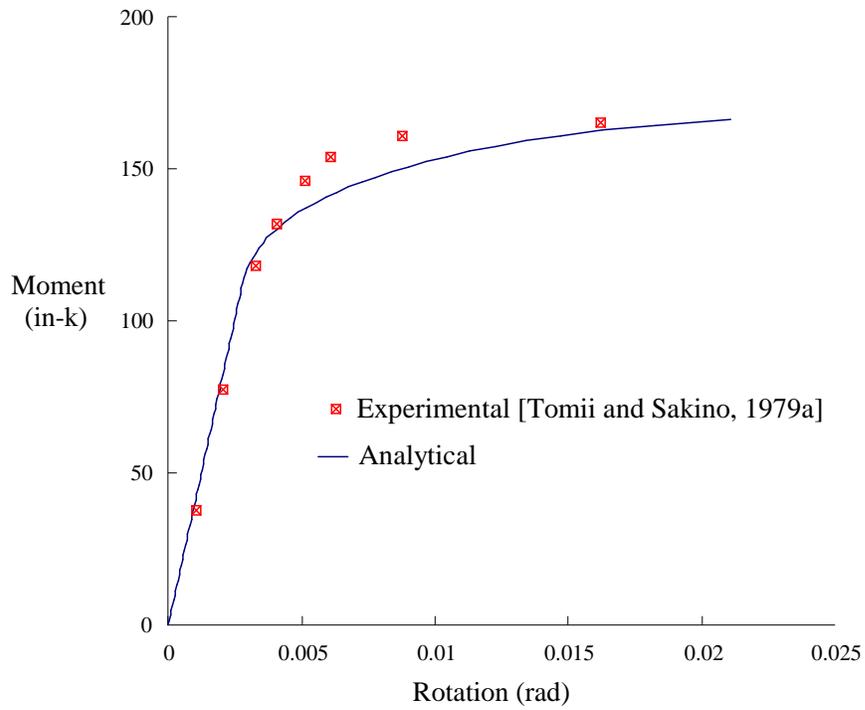


Figure 5.38 Axially-Loaded Beam in Bending (Tom24_2)

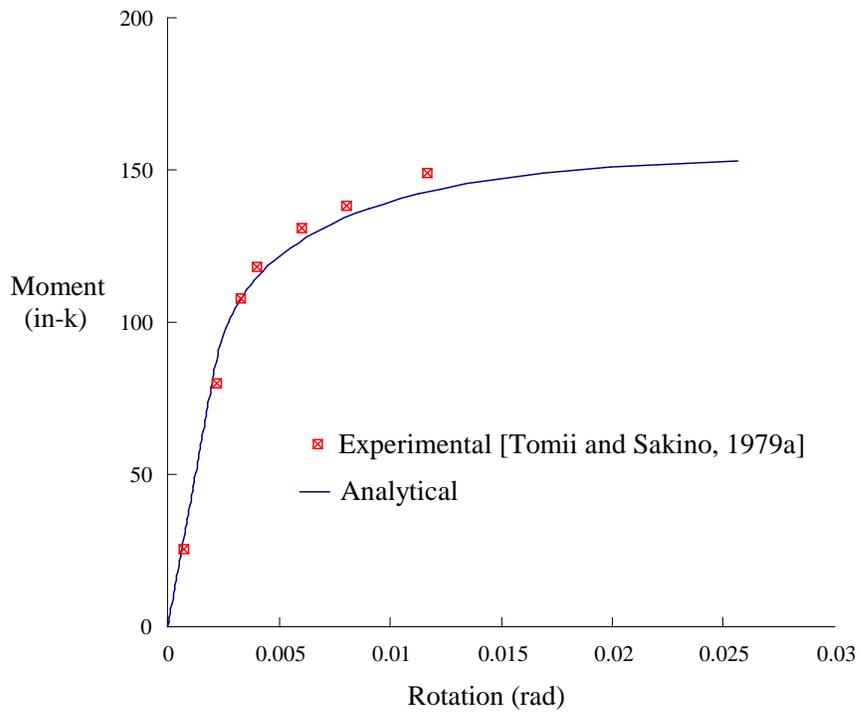


Figure 5.39 Axially-Loaded Beam in Bending (Tom24_4)

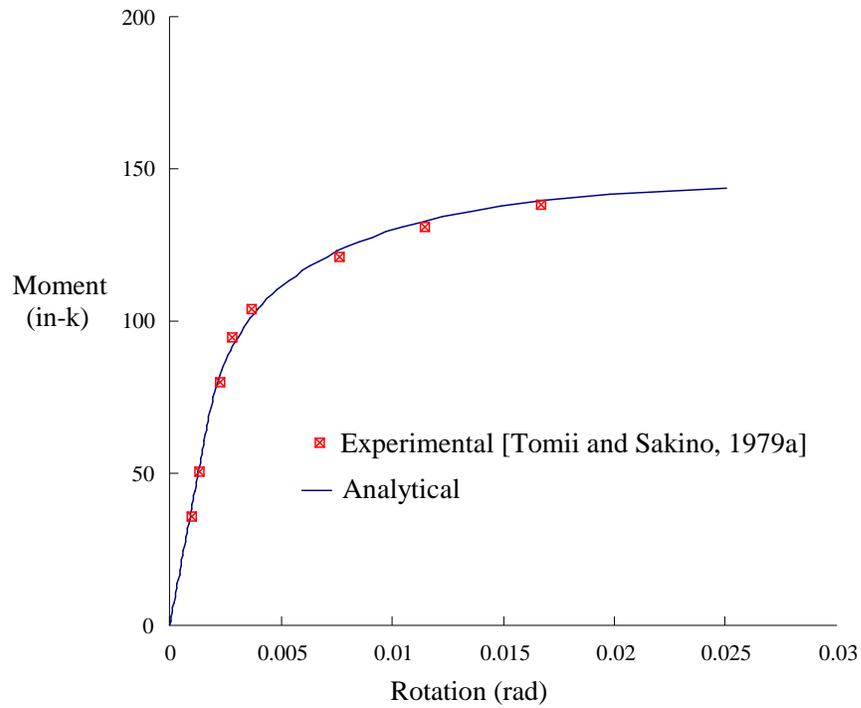


Figure 5.40 Axially-Loaded Beam in Bending (Tom24_5)

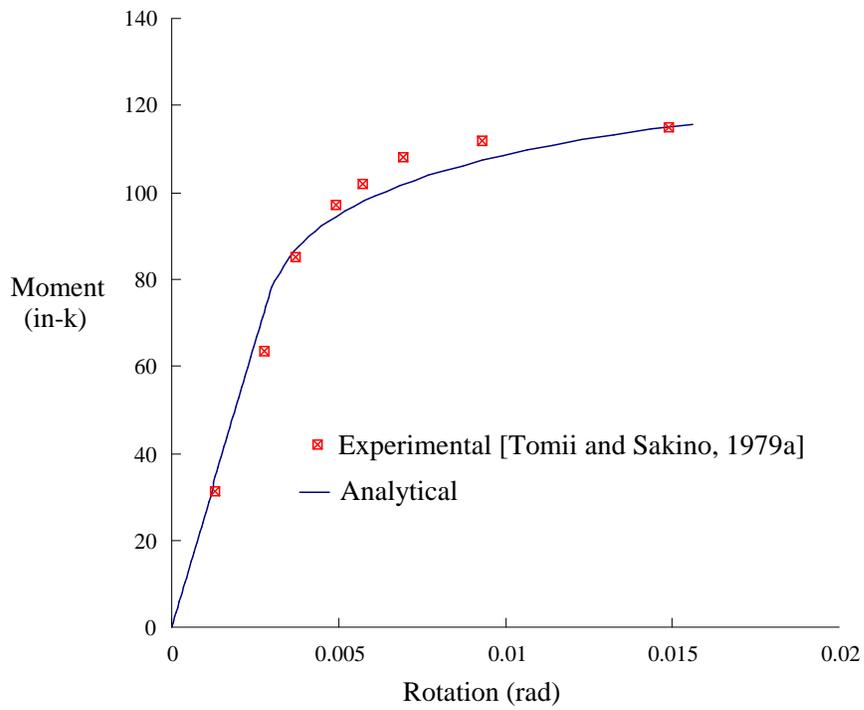


Figure 5.41 Axially-Loaded Beam in Bending (Tom44_2)

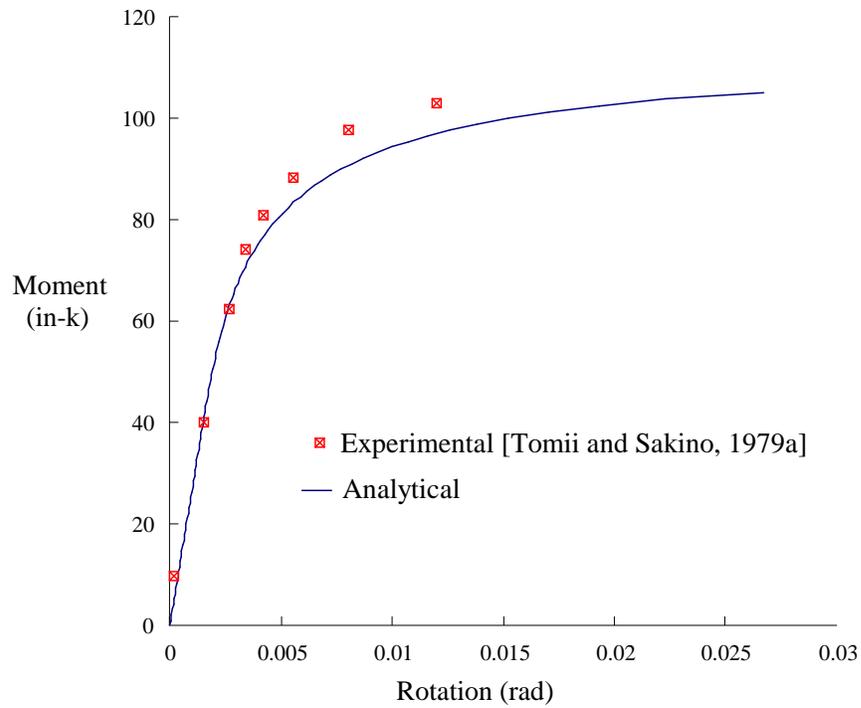


Figure 5.42 Axially-Loaded Beam in Bending (Tom44_4)

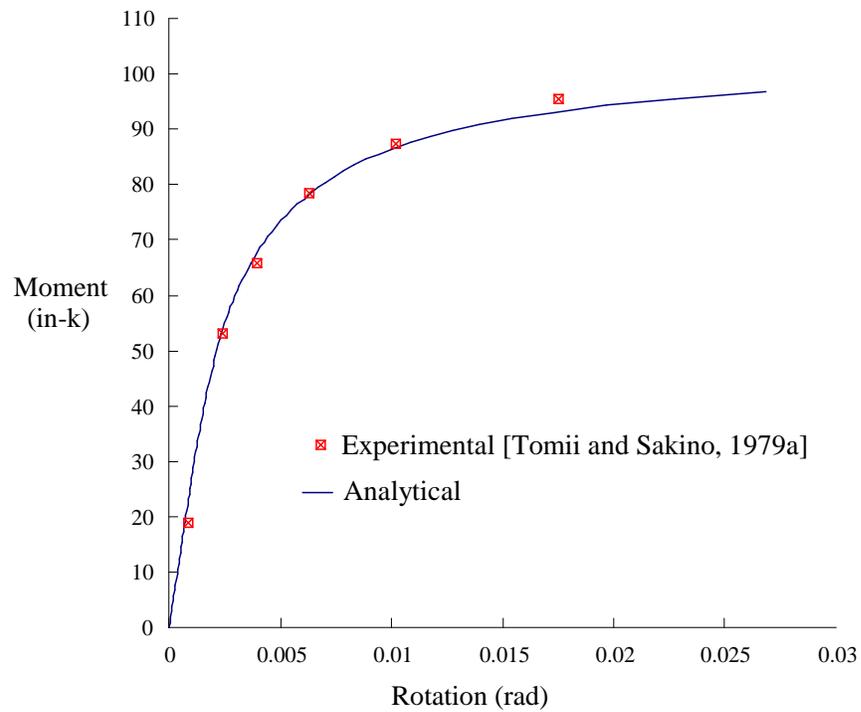


Figure 5.43 Axially-Loaded Beam in Bending (Tom44_5)

5.4 Verification of the Nonlinear Beam-Column Model

The final verification problem consists of a three-dimensional subassembly of steel I-girders framing into a CFT beam-column [Morino et al., 1993]. Figure 5.44 illustrates the structure configuration and the applied loading (Morino's test labeled SCC20 was used for this study). For the cyclic loading test that is examined, a constant axial load, $P = 0.15 \cdot P_o$ ($P_o = 292.5$ k for this specimen), is applied to the CFT beam-column and a constant load, $W = 4.725$ k, is applied to one end of the steel beam in the y-z plane. The cyclic loading, Q , is then applied transversely to the ends of the beams in the x-z plane as shown in Fig. 5.44. The load is cycled by applying anti-symmetric beam loads, Q , for increasing increments of rotation equal to 0.005, 0.01, and 0.02, and 0.03 radians. Two full cycles are performed at each increment in rotation. This CFT beam-column is thus subjected to unsymmetric, cyclic biaxial bending, plus axial force.

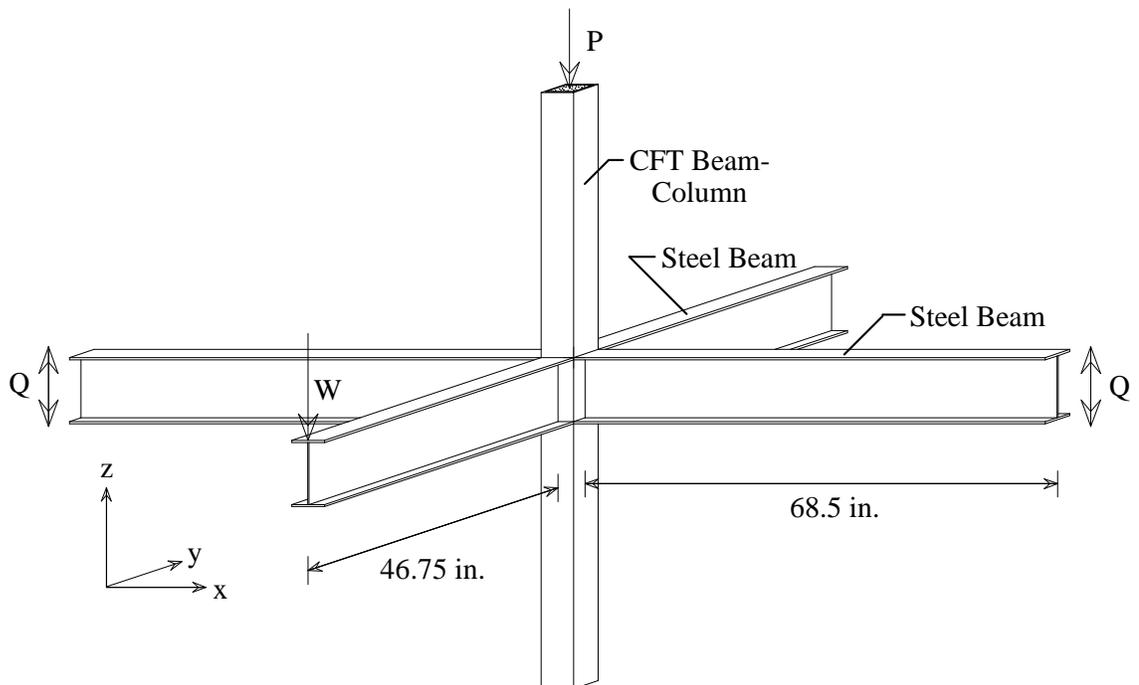


Figure 5.44 Three-Dimensional Subassembly (after Morino et al. [1993])

The CFT beam-column consists of a square tube with cross-section dimensions $4.92 \times 4.92 \times 0.226$ (inches) and material strengths of $f'_c = 2.91$ ksi and $f_y = 57.3$ ksi. The steel I-beams are built-up sections consisting of flanges measuring 4.92×0.354 and a web measuring 9.13×0.236 (all dimensions in inches). The yield strength of the beams is $f_y = 58.0$ ksi. The lengths of the members are given in Figs 5.44 and 5.45.

Figure 5.45 illustrates a schematic representation of the analytical model of the subassembly. The rotation was obtained in a manner analogous to the experimental method [Morino et al., 1993]. The displacements, D_1 and D_2 at locations A and B , respectively, in Fig 5.45 were summed and divided by the length between these points, which is denoted as the length L . This method, like the measurement technique used in the experiment, accounts for rotation of the structure at the connection as well as additional rotation due to beam flexure between the connection and the measuring points, A and B . The beam shear shown in the results is an average of the shears at the ends of each beam. These shears are not equal due to the axial displacement of the column. For the analysis, the structure was divided into three elements per CFT beam-column (to account for geometric nonlinearity with great accuracy), and two elements per beam (one spanning from the connection to the measurement point, point A or B in Fig. 5.45, and the other continuing to the point of the applied shear, Q). The connection is designed to remain fully restrained and the beams are designed to remain elastic throughout the entire analysis. In addition to the boundary conditions shown at the ends of the CFT beam-column in Fig. 5.45, each beam is restrained from translating in its out-of-plane direction (i.e., the beams in the x - z plane--the beams subjected to the applied cyclic loads, Q --are restrained from translating sideways in the x - y plane, and the beams in the y - z plane are restrained from translating in the x - y direction).

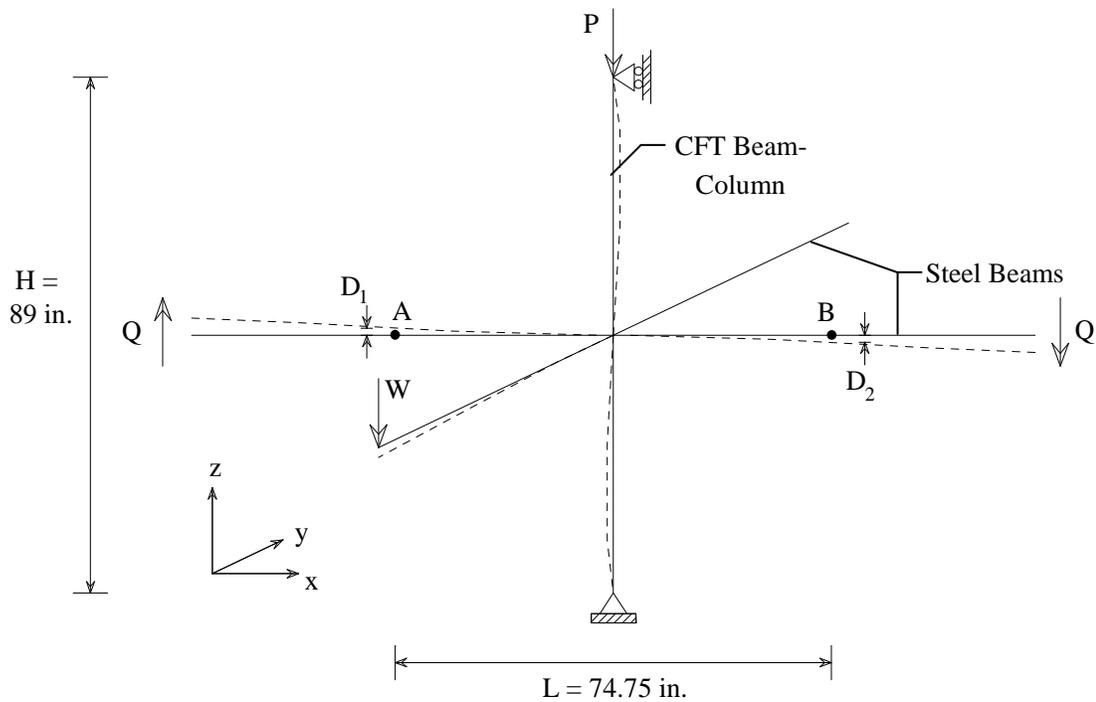


Figure 5.45 3D Subassembly: Analytical Model

The analysis of the structure was performed using the calibration parameters of Table 5.3. Figure 5.46 illustrates the analytical results compared to the experimental results provided by Morino et al. [1993]. The analytical results produce excellent accuracy for the entire hysteresis curve. The stiffness of the analytical curve matches the experimental curve well. The maximum shear at a rotation of 0.02 rad is slightly low in the analytical model (approximately 8.0 %), but the shear at a final rotation of 0.03 rad is nearly identical to the experimental shear for both cycles at this rotation level.

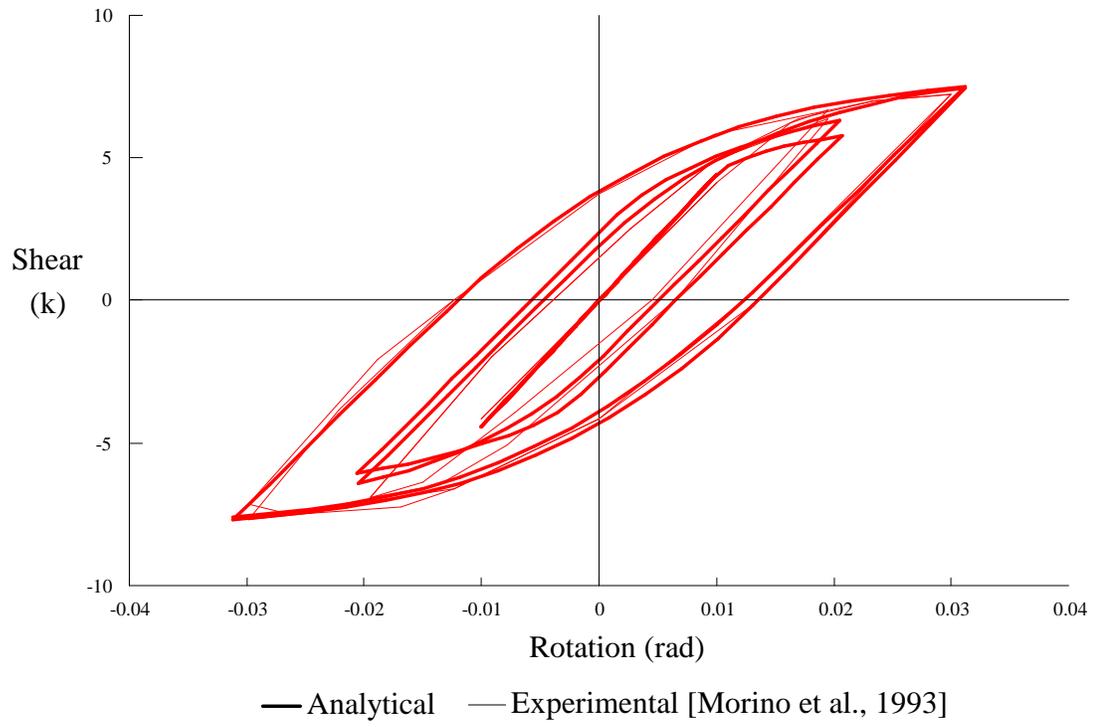


Figure 5.46 3D Subassembly: Shear-Rotation Hysteresis Curve

Chapter 6

Conclusions

The research contained in this work examines the three-dimensional, inelastic behavior of concrete-filled steel tube beam-columns. Analytical methods are developed for determining the cross-section strength of CFTs and the behavior of single member CFT beam-columns and CFT beam-columns contained in composite frame structures. The main impetus of the research focuses on the development of a compact and efficient concentrated plasticity bounding surface model in three-dimensional force-space to analyze the inelastic behavior of CFT beam-columns. This chapter presents a discussion of the results of this research on the behavior of CFTs. Several aspects of the work are highlighted, comments are made regarding the accuracy of the analytical model, and general conclusions are drawn about the behavior of CFTs. Following the conclusions, the final section of this chapter presents suggestions for possible future CFT research topics.

6.1 Conclusions

As background work for the analytical model developed herein, an extensive literature review was undertaken, in which the numerous advantages of CFT members became evident [Gourley and Hajjar, 1993]. CFTs provide a flexible, efficient, and economic alternative to traditional structural members. They offer high strength and stiffness, and exhibit excellent seismic load resistance. The experimental results used for the calibration and verification of the analytical model further demonstrate this highly favorable behavior of CFTs.

The first main topic of this research--the development of an expression for the three dimensional cross-section strength surface of a CFT--introduces uniaxial stress-strain formulations for the steel and the concrete. These formulations are implemented into a fiber element analysis of CFT cross-sections. The comparison of the fiber analysis to experimental moment-curvature-thrust data in Chapter 2 illustrates that these uniaxial stress-strain expressions accurately represent the interactive multiaxial behavior of the two materials due to confinement of the concrete. The use of uniaxial stress-strain relationships for the steel and concrete greatly simplifies the analysis, yet contributes no substantial loss of accuracy in the predicted moment-curvature-thrust behavior of the section.

A second important aspect of the cross-section study concerns the shape of the cross-section strength surface, since this surface forms an integral part of the bounding surface model. Due to the disparity in the compressive and tensile strengths of the concrete, the bending capacity of a CFT is highest when the section is subjected to low to moderate axial compression loads. This behavior results in a cross-section strength surface that is asymmetric about the moment axes (as opposed to strength surfaces for steel members, which have presumed symmetry about all force axes). This research

presents an empirical equation for the three-dimensional CFT cross-section strength surface that models this asymmetry and demonstrates excellent accuracy for a wide range of CFT cross-section sizes and material strengths. This equation, although containing four coefficients represented by cubic polynomials, is easily implemented into the analytical model and is based on only the D/t ratio and the ratio of the concrete strength to the steel strength, f'_c/f_y .

The concentrated plasticity model for CFTs incorporates a number of significant features, particularly with respect to the hardening of the loading and bounding surfaces. First, to model the complicated behavior of CFTs, both the loading surface and the bounding surface harden isotropically and kinematically. The Tseng kinematic hardening method is used in the CFT model as opposed to the Mroz method, which has been used in previous force-space plasticity models for steel beam-columns, since the Tseng method has been shown to work best for non-proportional loading in stress space plasticity. The CFT isotropic hardening approach utilizes a plastic work-based equation, in which the plastic work is normalized by an elastic work value calculated at the onset of initial yield to account for the material and geometric properties of the section.

Cyclic CFT behavior mandates an expansion of the bounding surface to model the strength increase due to cyclic strain hardening and concrete confinement, followed by a contraction of the bounding surface to model the stiffness degradation due to concrete crushing and local buckling of the steel tube. Cyclic CFT behavior also requires a variable κ_2 coefficient, representing the rate of plastification, and a decreasing concrete elastic modulus. The initial and final values of the κ_2 coefficients, the final bounding surface size, and the initial loading surface size are all calibration parameters that are presented as functions of the material strengths and the dimensions of the CFT. In this way, the formulation is generalized for all of the monotonic and cyclic tests examined in

this research, and is thus generally applicable to a wide range of CFT sections and material strengths.

The results of the verification studies of Chapter 5 demonstrate that the bounding surface model may be used with excellent accuracy to model the nonlinear behavior of both monotonically and cyclically-loaded beam-column specimens. The model itself requires a relatively small number of calibrated parameters to provide accuracy over a wide range of steel tube geometries and steel and concrete strengths. The final structure that is analyzed and compared to experimental results--a three-dimensional subassembly consisting of both CFT and steel I-beam elements---demonstrates an accurate representation of both stiffness and strength for this complex cyclically-loaded structure. This study verifies that this compact and efficient CFT finite element is ideal for use in static and transient dynamic analysis of composite CFT frame structures.

6.2 Suggestions for Future Research

The research contained in this work represents an early step in the analysis of CFT member behavior and a number of additional CFT research topics naturally follow this research. This section suggests future topics of research that explore new areas of CFT behavior or expand upon the results presented herein.

The first immediate research need is additional experimental work. A more comprehensive scope of cross-section geometries and material strengths should be investigated. Studies of cross-section strength, monotonic behavior, and cycle behavior--both static and dynamic, warrant further experimental study. The results presented in this work indicate a deficiency of experiments for CFTs with high-strength steel and high-strength concrete. A particularly beneficial, yet perhaps prohibitively expensive, set of

experimental tests might include a study of three-dimensional CFT/steel frames having rigid connections and incorporating a range of concrete strengths and tube thicknesses. Cyclic biaxial lateral loads could be applied to the structure along with gravity loads. Less elaborate tests, especially using high-strength materials, would, however, serve to refine and expand the results presented here.

A number of aspects of the CFT concentrated plasticity formulation warrant further investigation. First, an initial loading surface in the shape of the actual initial yield surface (as discussed in Chapter 5) would provide a more accurate representation of CFT behavior over a wide range of axial load ratios. This modified surface could, by some mathematical formulation, gradually transform into the shape of the bounding surface as the two surfaces approach one another. A second potential modification of the current plasticity model involves computing a vectorial, rather than a scalar, distance between the loading and bounding surfaces to determine the extent of plastic loading in the member. The result of using different kinematic hardening approaches might also be examined. Initial exploration of the difference between the Tseng and Mroz hardening methods produced similar results for both methods, but further studies would be required to state any definitive conclusions regarding the superiority of one method over the other.

A more thorough examination of the local buckling phenomenon in CFTs is also required. A fiber element analysis would work well for examining this characteristic of the infilled steel tube. An accurate local buckling formulation based on the material strengths and the D/t ratio of the CFT section could be implemented into both the cross-section strength formulation (i.e., in the determination of the failure criteria) and the plasticity model (e.g., in establishing the point at which the strength of a cyclically-loaded specimen begins to degrade).

There are a number of additional CFT research topics that should be addressed in the future. Some possibilities include: 1) research on CFT/steel connections, including

topics such as steel/concrete bond at the connection, the development of economic and efficient connections, and the development of an analysis method to model partially-restrained connections; 2) parametric studies of CFT/steel frame structures to develop improved design formulations; and 3) development of an analytical macro model for circular CFTs similar to the one developed in this work for rectangular CFTs.

Appendix A

CFT Experimental Tests

This appendix is composed of three tables which list papers containing experimental test results for rectangular CFTs. Only tests published in sufficient detail to use for calibration are included in the tables, and tests are considered only for CFTs which are completely filled with concrete and make no use of reinforcing bars or shear connectors to improve the concrete/steel bond. Table A.1 contains cross-section tests, i.e., monotonic tests of specimens having L/D ratios less than 10. Tables A.2 and A.3 contain beam-column tests for monotonically- and cyclically-loaded CFT specimens, respectively. The papers in each table are categorized by D/t ratio and concrete strength, f'_c . Tests in which the experimental setup consisted of more than one member (e.g., subassemblies) are noted in the tables. Complete references for each paper are shown in the list of references following the Appendix. Each tabular entry is presented in the following format:

Author(s), Year (Number of Tests)

[L/D; D/t; f'_c , f_y]

Table A.1 Rectangular CFT Cross-Section Tests ($L/D \leq 10$)

	Low D/t (5 - 24)	Medium D/t (24 - 50)	High D/t (50 -)
Low f'_c (2 - 5)	<i>Tomii, Sakino, '79ab</i> (7) [3; 24; 2.7-2.9; 41.4]	<i>Furlong, '67</i> (13) [9.0; 32, 48; 3.4, 4.2; 48] <i>Tomii, Sakino, '79ab</i> (21) [3; 33-44; 3-5.5; 28-49]	
Medium f'_c (5 - 9)		<i>Furlong, '67</i> (4) [7.2; 26; 6.5; 70.3]	
High f'_c (9 -)			

Table A.2 Monotonic Rectangular Beam-Column Tests ($L/D \geq 10$)

	Low D/t (5 - 24)	Medium D/t (24 - 50)	High D/t (50 -)
Low f'_c (2 - 5)	<i>Bridge, '76</i> (4) [10.5, 15; 20, 24; 4.6; 44]	<i>Shakir-Khalil, '90</i> (4) [21-37; 30; 4.4-4.9; 50-56] <i>Matsui et al., '93</i> (16) [12-30; 33.3; 4.6]	
Medium f'_c (5 - 9)	<i>Knowles, Park '69</i> (4) [11, 18.7; 23; 5.9; 47, 58] <i>Bridge, '76</i> (4) [11-20; 20, 24; 5; 37-45] <i>Shakir-Khalil, '89</i> (7) [23, 35; 16, 24; 5.4; 53] <i>Cederwall et al., '90</i> (4) [25; 15; 5.7-6.8; 44-64] <i>Shakir-Khalil, '90</i> (12) [23; 24; 4.9-5.4; 49-53]	<i>Shakir-Khalil, '91</i> (11) [21-32; 30; 5.4-6.2; 48- 53] <i>Cederwall et al., '90</i> (2) [25; 24; 6.8; 44-64]	
High f'_c (9 -)	<i>Cederwall et al., '90</i> (10) [25; 15; 11.6-14.9; 44-64]	<i>Cederwall et al., '90</i> (2) [25; 24; 13.9; 44-64]	

Table A.3 Cyclic Rectangular Beam-Column Tests

	Low D/t (5 - 24)	Medium D/t (24 - 50)	High D/t (50 -)
Low f'_c (2 - 5)	<p><i>Sakino, Tomii, '81</i> (4) [6; 24; 2.9; 42-45]</p> <p><i>Sakino, Ishibashi, '85</i> (4) [2; 24; 3.1-3.7; 42-46]</p> <p><i>Kawaguchi, '91, '93</i> (14) [10; 22, 31; 3.1-3.6; 49]</p> <p><i>Morino et al '93</i> (5--2D and 3D subassemblies) [14.3; 21.3; 2.9; 57]</p>	<p><i>Sakino, Tomii, '81</i> (11) [4, 6; 34-46; 3.5; 42-45]</p> <p><i>Sakino, Ishibashi, '85</i> (8) [2; 34, 45; 2.4-3.7; 42-46]</p> <p><i>Liu, Goel, '88</i> (2) [23, 45; 30; 4; 54, 60]</p>	<p><i>Matsui, Tsuda, '87</i> (6) [5.0; 47-94; 5.7-6.0; 71.5]</p>
Medium f'_c (5 - 9)		<p><i>Matsui, '86</i> (2--2D 1 bay frames) [6.7; 33, 47; 5.4; 42, 60]</p> <p><i>Liu, Goel, '88</i> (4) [23-68; 14, 30; 6-8; 54-60]</p> <p><i>Sugano et al., '92</i> (1) [6.8; 31.3; 5.5; 54]</p>	<p><i>Matsui, '86</i> (1--2D 1- bay frame) [6.7; 68; 5.5; 42]</p>
High f'_c (9 -)			

Appendix B

Key Features of the Computer Implementation

This section summarizes some of the key features of the software implementation of the nonlinear analysis presented in this work. A sample data input file for the program CFTmacro is also presented, along with a description of the salient characteristics of the file.

The program accesses the AISC database of manufactured structural shapes. Therefore, the user need only input a section designation, e.g., W14×53, and the section properties are automatically input. Additionally, each element of the structure may be oriented in any direction in three-dimensional space by specifying the three components of a unit vector, $\{u\}$, in global coordinates. For a steel member this unit vector is oriented parallel to the web at the i -end of the element (Fig. B.1a). For a CFT member, the unit vector lies along the major axis of the element at the i -end, as illustrated in Fig. B.1b. In both elements shown in Fig B.1, the unit vector, $\{u\}$, is given by the global $x - y - z$ coordinates $\{u\} = \{0 \ 0 \ 1\}$.

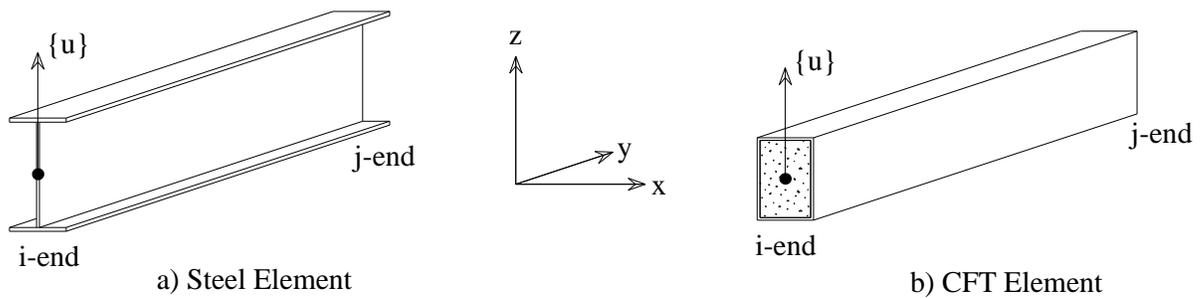


Figure B.1 Unit Vectors Denoting Global Orientation of Steel and CFT Elements

Table B.1 illustrates a sample input file to execute a static cyclic analysis of a two-dimensional portal frame. The program reads the input data in blocks; each block of data is prefaced by a starred keyword (e.g., *A_JTS). The program searches for each keyword, then begins reading data at the first non-blank, non-comment (denoted with a #) line following this keyword until the complete set of data has been read. Therefore, the keywords, each with their corresponding data, may appear in any order in the input file. The keywords *A_START and *A_END signal the beginning and end of the file. All of the units in the input file are in kips and inches to match the units in the AISC database.

The first group of data sets shown in Table B.1 refer to the joints or nodes of the structure. The section *A_JTS defines each node of the structure in global $x - y - z$ coordinates. The next section, *A_JT_RF, specifies the restraint conditions for each nodal degree-of-freedom (DOF)--0 denotes a restrained DOF and 1 denotes a free DOF. The final joint-based category of input for a static analysis is the load data, which is listed under the keyword *A_JT_LOAD. Loads are input on a joint and DOF basis and each load is assigned a load history, which is defined in a subsequent data set and explained below.

The input sections for the CFT and steel elements, *A_CFT_ELS and *A_STL_ELS, respectively, define the connectivity, orientation, and properties of each

element in the structure. The first three numbers denote the element number, i -end, and j -end, respectively, of the element. The next three numbers define the global coordinates of the element orientation vector (see Fig. B.1). The remaining values define the material and section properties of the element. In addition to these parameters, the plasticity parameters are also input on an element by element basis under the headings *A_CFT_PLAST and *A_STL_PLAST. These input sections define the calibrated parameters for the CFT and steel elements (see Chapter 5).

Section *A_LOAD_HIST defines each load history. The user first inputs the number of load histories, the time step, and the total time of the analysis, and then specifies the time versus load fraction coordinate for each point in each load history. The input file shown illustrates non-proportional loading, specifying a constant load (load history 1) and a variable cyclic load (load history 2). These load histories are illustrated in Fig. B.2. Note that the load fraction is the multiple of the joint/DOF load specified under *A_JT_LOAD. For example, for load history 1 (Fig. B.2a), a non-proportional load of -35.7 kips (1.0×-35.7) is applied to joint 3, DOF 2 for the full analysis (0 to 6 sec.). For load history 2 (Fig. B.2b), a cyclically increasing load is applied to joint 3, DOF 1--11810 kips (1.0×11810) at 0.5 sec., -11810 kips (-1.0×11810) at 1.5 sec., 23620 kips (2.0×11810) at 2.5 sec., and so on. For static analyses, the magnitude of time has no influence on the analysis per se; it serves only as a means of incrementing the load. A time history is incorporated in the program to accommodate dynamic analyses, in which accelerations and velocities directly depend upon the value of time.

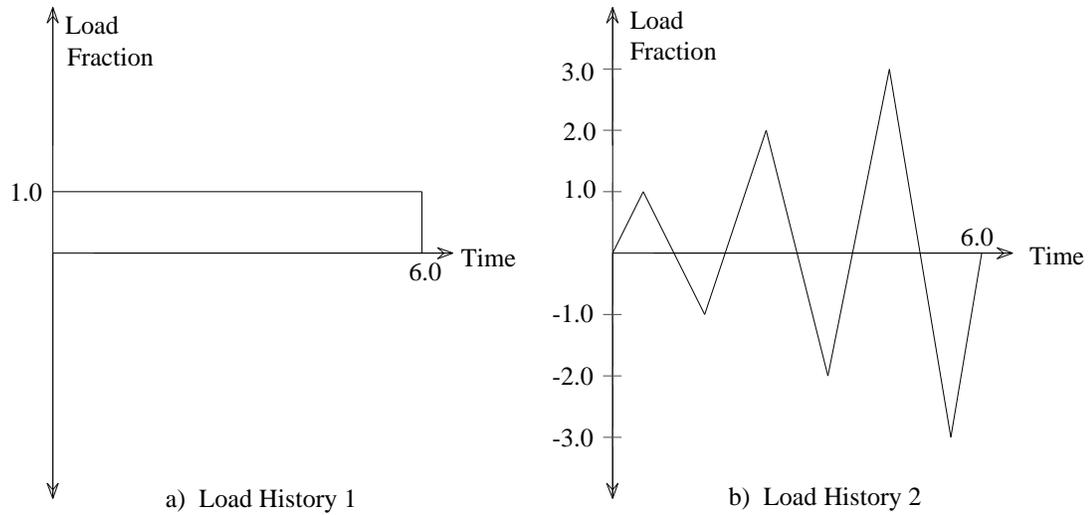


Figure B.2 Sample Load Histories

The section `*A_TOLS` defines the loading and bounding surface tolerance. This value is used in the determination of force point contact with either surface and for force point drift control (Section 4.6). The final section of the input file, `*A_FLAGS`, contains flags to activate or deactivate certain aspects of the program. For example, the user may specify an elastic or second-order elastic analysis. Additionally, either the Mroz or Tseng method of kinematic hardening may be specified for the CFT and steel elements.

Table B.1 Sample Input File for CFTmacro

```

*A_START
#SAMPLE INPUT FILE
#2-D PORTAL FRAME W/ CFT BEAM-COLUMNS (FIXED AT BASE) AND A STEEL BEAM

*A_JTS
#Input of the joint coordinates
#Format: Joint number, x coord, y coord, z coord (inches)

1 0.0 0.0 0.0
2 0.0 100.0 0.0
3 100.0 100.0 0.0
4 100.0 0.0 0.0

*A_JT_RF
#List of joint restraints ( 1 = free, 0 = restrained )
#Format: Joint num., 6 joint directions (x, y, z translation, x, y, z rotation)

1 0 0 0 0 0 0
2 1 1 1 1 1 1
3 1 1 1 1 1 1
4 0 0 0 0 0 0

*A_CFT_ELS
#CFT elements (all data must be in kips and inches)
#Choose either user input format or AISC rectangular tube format
#   line 1: CFT element number, i-end, j-end, i-end unit vector (x, y, z),
#           fy, Es, Gs, fc, Ec, nu_c, wt_c
#   line 2 (user input format): tube depth, tube width, tube thickness
#   line 2 (AISC rectangular tube format): section designation (e.g., TS6x6x1/2)

1 1 2 1.0 0.0 0.0 46.0 29000.0 11500.0 5.0 3500.0 0.2 0.086
6.0 6.0 0.25

2 3 4 1.0 0.0 0.0 46.0 29000.0 11500.0 5.0 3500.0 0.2 0.086
6.0 6.0 0.25

*A_CFT_PLAST
#CFT plasticity coefficients and parameters
#Each CFT element must have separate input, i.e., 3 lines for each element
#   (line 1: elem num and surface radii, line2: initial and final coeffs,
#   line 3: isotropic params):
#   line 1: elem #, init LS rad, fin LS rad, init BS rad, intm BS rad, fin BS rad
#   line 2: pk1, myk1, mzk1, pk2i, myk2i, mzk2i, pk2f, myk2f, mzk2f
#   line 3: ls_iso, bs_iso, k2_iso, ec_iso, intmBS_work

1 0.662 0.1 1.0 1.4 0.503
0.001 0.001 0.001 0.572 0.286 0.286 2.43 1.21 1.21
10.0 50.0 30.0 3000.0 0.02

2 0.662 0.1 1.0 1.4 0.503
0.001 0.001 0.001 0.572 0.286 0.286 2.43 1.21 1.21
10.0 50.0 30.0 3000.0 0.02

*A_STL_ELS
#Steel elements (all data must be in kips and inches)
#Choose either user input format or AISC W-section format
#   line 1: Steel elem #, i-end, j-end, web-vector coords (x, y, z), fy, E, G
#   line 2 (user input format): area, J, Iminor, Imajor, Sy, Sz, Zy, Zz, kpi_s
#   line 2 (AISC W-section format): section designation (e.g., W14x53)

1 2 3 0.0 1.0 0.0 36.0 29000.0 11500.0
W10x45

```

Table B.1 (cont'd)

```
*A_STL_PLAST
#Steel plasticity coefficients and parameters
#Each steel element requires a separate line
#Format (1 line):
#      Element number, 6 coeffs (pk1, myk1, mzk1, pk2, myk2, mzk2),
#      2 radii (init L.S., init B.S.)

1 0.001 0.001 0.001 8.0 11.0 5.0 0.5 1.0

*A_JT_LOADS
#Joint loads
#Format: Joint number, joint direction, load (kips), load history number

2 2 -30.0 1
3 2 -30.0 1
2 1 3.0 2

*A_LOADHIST
#Load history
#
#Format:
#      Number of load histories, time step, total time, output every __ steps
#Format for each load history:
#      First line: Load history number, number of points
#      Successive lines: Load history pts (load fraction, time)
#                        (Note: maximum of 50 (load, time) points per history)

2 0.01 1.0 10

1 2
1.0 0.0
1.0 1.0

2 6
0.0 0.0
0.2 1.0
0.4 -1.0
0.6 2.0
0.8 -2.0
1.0 0.0

*A_TOLS
#Tolerances used throughout the program
#Loading and bounding surface tolerance:

0.0001

*A_FLAGS
#Program flags
#Include geometric nonlinearities? (Y or N)
Y
#Include material nonlinearities? (Y or N)
Y
#Perform force point drift control? (Y or N)
Y
#Include isotropic hardening? (Y or N)
Y
#Use Mroz or Tseng kinematic hardening for steel? (M or T)
Mroz
#Use Mroz or Tseng kinematic hardening for CFTs? (M or T)
Tseng
#Should element masses be applied to joints? (Y or N)
N

*A_END
```

Appendix C

CFTmacro Source Code

This appendix presents a major portion of the source code for the program CFTmacro developed in this work. Table C.1 shows the main driver function followed by an alphabetical listing of selected corollary functions as they appear in the complete program. Although the program has capabilities beyond those utilized in this work, only the routines specifically pertinent to the analyses performed in this work have been included. Therefore the functions relating to dynamic analysis, distributed loading, and member releases, which may appear as function calls in some of the routines in Table C.1, are not shown. Additionally, to prevent undue redundancy, only the functions specific to CFTs are shown; the steel routines, which are nearly identical, have not been included. Functions pertaining to input/output, initialization, memory allocation, skyline storage and solution, and general administration have also been omitted.

Table C.2 contains an alphabetical listing of the header files accessed by the program functions. The files shown contain external variable definitions, macro definitions, and the definitions of all of the data structures that are used throughout the program.

Table C.1 CFTmacro Source Code

```

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****
*/

* FUNCTION- cftmacro

        @(#) CFT MACRO model beam-column program

* ABSTRACT- This program performs a fully nonlinear static or dynamic
analysis of three-dimensional frame structures composed of
concrete- filled steel tube and/or steel elements. Material
nonlinearity is modeled using the bounding surface
concentrated-plasticity approach.
*****/

#include <math.h>          /* C MATH FUNCTIONS          */
#include <stdio.h>         /* C I/O FUNCTIONS         */
#include <string.h>        /* C STRING FUNCTIONS      */
#include <stddef.h>        /* C STANDARD HEADER FILE  */

#include "a_mac.h"         /* MACRO DEFINITIONS       */
#include "a_extern.h"      /* EXTERNAL VARIABLES      */
#include "a_jt.h"          /* DATA STRUCTURE DEFINING JOINT PROPS. */
#include "a_cft_el.h"      /* DATA STRUCTURE FOR CFT MACRO ELEMENT */
#include "a_stl_el.h"      /* DATA STRUCTURE FOR STEEL MACRO ELE.   */
#include "a_param.h"       /* DATA STRUCTURE OF VARIOUS PARAMETERS  */
#include "a_global.h"      /* DATA STRUCTURE OF DOF-BASED VARIABLES  */

void main ( argc, argv )

int  argc;                /* NUMBER OF COMMAND LINE ARGUMENTS      */
char *argv[];             /* COMMAND LINE ARGUMENTS                 */

{
long  status = 1L;        /* RETURN STATUS                          */
long  *diag = ( long * )0; /* INDEX OF K MATRIX DIAGONAL            */
double *kt = ( double * )0; /* SKYLINE GLOBAL K MATRIX               */
A_MODEL_SIZE size;       /* GLOBAL STRUCTURE SIZES                 */
A_TIME time;             /* TIME PARAMETERS                        */
A_NEW_RAPH nr;           /* NEWTON-RAPHSON PARAMETERS             */
A_TOLERANCE tol;        /* PROGRAM TOLERANCES                    */
A_FLAGS flag;           /* PROGRAM FLAGS                          */
A_DYNAMIC dyn;          /* DYNAMIC ANALYSIS PARAMETERS           */
A_ACCEL acclg[ 3L ];     /* ACCELEROGRAM PARAMETERS               */
A_EIGEN *eigen = ( A_EIGEN * )0; /* EIGEN-ANALYSIS DATA                  */
A_LOADHIST *lhist = ( A_LOADHIST * )0; /* LOAD HISTORY PARAMS                   */
A_JT_JOINT *jt = ( A_JT_JOINT * )0; /* SINGLE JOINT                          */
A_CFT_BC *cft = ( A_CFT_BC * )0; /* CFT BEAM-COLUMN                       */
A_STL_BC *stl = ( A_STL_BC * )0; /* STEEL BEAM-COLUMN                    */
A_GLOBAL *global = ( A_GLOBAL * )0; /* DOF-BASED VARIABLES                   */

/*****/
/* DEFINE AND INITIALIZE EXTERNAL VARIABLES */
/*****/

if ( ! a_mem_init_extern() )
{
status = 0L;
printf(
"\nInitialization of external variables failed. Exit program.\n" );
}

/*****/
/* READ FILE NAMES FROM COMMAND LINE */
/*****/

```

```

else if ( ! a_io_filenames( argc, argv, "cftmacro.in", "cftmacro.out",
                             "cftmacro.plot", "cftmacro.max" ) )
    {
        status = 0L;
        printf( "\nFailure to establish file names.  Exit program.\n" );
    }

/*****
/* OPEN AISC SECTION TABLE DATABASE */
*****/

else if ( ! a_io_aisc_open( argc, argv ) )
    {
        status = 0L;
        printf(
            "\nOpening the AISC section database failed.  Exit program.\n" );
    }

/*****
/* OPEN FILES */
*****/

else if ( ! ( A_fp_in = fopen( A_in_file, "r" ) ) ) /* OPEN INPUT FILE */
    {
        status = 0L;
        printf( "\nOpening input file failed.  Exit program.\n" );
    }

else if ( ! ( A_fp_out = fopen( A_out_file, "w" ) ) ) /* OPEN OUTPUT FILE */
    {
        status = 0L;
        printf( "\nOpening output file failed.  Exit program.\n" );
    }

else if ( ! ( A_fp_plot = fopen( A_plot_file, "w" ) ) ) /* OPEN PLOT FILE */
    {
        status = 0L;
        printf( "\nOpening plot file failed.  Exit program.\n" );
    }

/*****
/* INITIALIZE NON-ARRAY VARIABLES */
*****/

else if ( ! a_mem_init( &size, &time, &nr, &tol, &flag, &dyn ) )
    {
        status = 0L;
        printf( "\nInitializaion of variables failed.  Exit program.\n" );
    }

/*****
/* DETERMINE SIZE OF MODEL */
*****/

else if ( ! a_io_model_size( &size ) )
    {
        status = 0L;
        printf( "\nCounting model size failed.  Exit program.\n" );
    }

/*****
/* ALLOCATE DYNAMIC MEMORY FOR MODEL */
*****/

else if ( ! a_mem_alloc_model( size, &lhst, &jt, &cft, &stl ) )
    {
        status = 0L;
        printf( "\nAllocating memory for model failed.  Exit program.\n" );
    }

/*****
/* INITIALIZE STRUCTURE SIZE BASED ARRAYS */
*****/

```

```

else if ( ! a_mem_init_array( size, acclg, lhist, jt, cft, stl ) )
    {
        status = 0L;
        printf( "\nInitialization of arrays failed.  Exit program.\n" );
    }

/*****
/* INPUT STRUCTURAL MODEL, LOADS, ETC. */
*****/

else if ( ! a_io_input_model( &size, &time, &nr, &tol, &flag, &dyn, acclg,
                             lhist, jt, cft, stl ) )
    {
        status = 0L;
        printf( "\nInput of data failed.  Exit program.\n" );
    }

/*****
/* DETERMINE NUMBER OF DOFS IN MODEL AND MAP TO JOINTS AND ELEMENTS */
*****/

else if ( ! a_jt_dof_map( &size, jt, cft, stl ) )
    {
        status = 0L;
        printf( "\nCounting of DOFs failed.  Exit program.\n" );
    }

/*****
/* ALLOCATE DYNAMIC MEMORY FOR DOF-BASED DATA STRUCTURES */
*****/

else if ( ! a_mem_alloc_dof( &diag, size, &eigen, &global ) )
    {
        status = 0L;
        printf( "\nAllocating memory for dofs failed.  Exit program.\n" );
    }

/*****
/* INITIALIZE DOF-BASED ARRAYS */
*****/

else if ( ! a_mem_init_dofarray( diag, size, eigen, global ) )
    {
        status = 0L;
        printf( "\nInitialization of arrays failed.  Exit program.\n" );
    }

/*****
/* CALCULATE TIME- AND LOAD-INDEPENDENT ELEMENT PARAMETERS */
*****/

else if ( ! a_el_calcs( size, flag, cft, stl ) )
    {
        status = 0L;
        printf( "\nPlastic calculations failed.  Exit program.\n" );
    }

/*****
/* SET UP SKYLINE STORAGE, ALLOCATE MEMORY FOR GLOBAL K MATRICES */
*****/

else if ( ! a_eq_skyline( diag, &kt, &size, cft, stl, global ) )
    {
        status = 0L;
        printf( "\nSetup of skyline storage failed.  Exit program.\n" );
    }

/*****
/* ANALYSIS ROUTINE */
*****/

else if ( flag.analysis == STATIC )
    {
        if ( ! a_drv_static( diag, kt, &size, &time, &nr, tol, &flag,

```

```

        acclg, lhist, jt, cft, stl, global ) ) status = 0L;
    }

else if ( ( flag.analysis == DYNAMIC ) || ( flag.analysis == EIGEN ) )
    {
        if ( ! a_drv_dynamic( diag, kt, &size, &time, &nr, tol, &flag, &dyn,
            acclg, eigen, lhist, jt, cft, stl, global ) ) status = 0L;
    }

/*****
/* FREE DYNAMIC MEMORY */
*****/

cfree( diag );
cfree( kt );
cfree( eigen );
cfree( lhist );
cfree( jt );
cfree( cft );
cfree( stl );
cfree( global );

/*****
/* CLOSE FILES */
*****/

if ( A_fp_in ) fclose( A_fp_in );
if ( A_fp_out ) fclose( A_fp_out );
if ( A_fp_plot ) fclose( A_fp_plot );
if ( A_fp_max ) fclose( A_fp_max );

if ( status )
    {
        printf( "\n\nDONE!!!.\n\n" );
    }
else
    {
        printf( "\n\nError encountered / Structure failed.\n\n" );
    }

exit( 1L );
}

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****
*****

* FUNCTION- a_cft_frecovery

* @(#) CFT Force RECOVERY

* CALLED FROM- a_drv_dynamic, a_drv_static

* ABSTRACT- This function computes the incremental forces for the CFT
members based upon the incremental displacements obtained
from the solution procedure.
*****/

#include <math.h> /* C MATH FUNCTIONS */
#include <stdio.h> /* C I/O FUNCTIONS */
#include <string.h> /* C STRING FUNCTIONS */

#include "a_mac.h" /* MACRO DEFINITIONS */
#include "a_extern.h" /* EXTERNAL VARIABLES & PROTOTYPES */
#include "a_jt.h" /* DATA STRUCTURE DEFINING JOINT PROPS */
#include "a_cft_el.h" /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
#include "a_param.h" /* DATA STRUCTURE OF VARIOUS PARAMETERS */
#include "a_global.h" /* DATA STRUCTURE OF DOF-BASED VARIABLES */

long a_cft_frecovery( size, flag, jt, cft, global )

```

```

A_MODEL_SIZE  *size;          /* DATA STRUCTURE OF GLOBAL STRUCT SIZE */
A_FLAGS       *flag;         /* DATA STRUCTURE OF PROGRAM FLAGS */
A_JT_JOINT    jt[];         /* DATA STRUCTURE FOR SINGLE JOINT */
A_CFT_BC      cft[];        /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
A_GLOBAL      global[];     /* DATA STRUCTURE OF DOF-BASED VARIABLES */

{
long   status = 1L;         /* RETURN STATUS */
long   n       = 0L;       /* ELEMENT COUNTER */
long   ctr1    = 0L;       /* COUNTER */
long   ctr2    = 0L;       /* COUNTER */
long   dof     = 0L;       /* DEGREE-OF-FREEDOM COUNTER */
long   j       = 0L;       /* TEMP STORAGE OF CURRENT DOF NUMBER */
double l_tmp   = 0.0;      /* TEMPORARY ELEMENT LENGTH */
double temp_du[ A_CFT_NUM_DOF + 1L ]; /* STORAGE OF GLOBAL ELM DISPLS*/

/*****
/* LOOP OVER ELEMENTS AND RECOVER LOCAL INCREMENTAL MEMBER FORCES */
*****/

for ( n = 1L; n <= size->num_cft_elems; n++ )
{
/*****
/* MAP GLOBAL INCREMENTAL DISPLACEMENTS TO MEMBER DOFS */
*****/

for ( dof = 1L; dof <= A_CFT_NUM_DOF; dof++ )
{
temp_du[ dof ] = 0.0;
j = cft[ n ].mcode[ dof ];
if ( j != 0 )
{
temp_du[ dof ] = global[ j ].dqi;
}
}

/*****
/* ROTATE INCREMENTAL MEMBER DISPLACEMENTS FROM GLOBAL TO LOCAL */
*****/

for ( ctr1 = 1L; ctr1 <= A_CFT_NUM_DOF; ctr1++ )
{
cft[ n ].du[ ctr1 ] = 0.0;

for ( ctr2 = 1L; ctr2 <= A_CFT_NUM_DOF; ctr2++ )
{
cft[ n ].du[ ctr1 ] += temp_du[ ctr2 ] *
cft[ n ].lambda[ ctr1 ][ ctr2 ];
cft[ n ].u2[ ctr1 ] += temp_du[ ctr2 ] *
cft[ n ].lambda[ ctr1 ][ ctr2 ];
}
}

/*****
/* COMPUTE LOCAL INCREMENTAL MEMBER FORCES */
*****/

for ( ctr1 = 1L; ctr1 <= A_CFT_NUM_DOF; ctr1++ )
{
cft[ n ].df_i[ ctr1 ] = 0.0;

for ( ctr2 = 1L; ctr2 <= A_CFT_NUM_DOF; ctr2++ )
{
cft[ n ].df_i[ ctr1 ] += cft[ n ].du[ ctr2 ] *
( cft[ n ].kt[ ctr1 ][ ctr2 ] -
cft[ n ].kr[ ctr1 ][ ctr2 ] );
}
}

/*****
/* FOR 2ND ORDER ELASTIC ANALYSIS, CALC INCR AXIAL FORCES DIRECTLY */
*****/

```

```

if ( ( flag->kg ) && ( ! flag->kp ) )
{
    /* CALCULATE UPDATED MEMBER LENGTH */

    l_tmp = sqrt(
        pow( ( jt[ cft[ n ].j ].co.x - jt[ cft[ n ].i ].co.x ), 2 ) +
        pow( ( jt[ cft[ n ].j ].co.y - jt[ cft[ n ].i ].co.y ), 2 ) +
        pow( ( jt[ cft[ n ].j ].co.z - jt[ cft[ n ].i ].co.z ), 2 ) );

    /* CALCULATE NEW AXIAL FORCES (OVERWRITE ABOVE AXIAL CALCS) */

    cft[ n ].df_i[ 1L ] = - cft[ n ].ea * ( l_tmp / cft[ n ].l - 1L );
    cft[ n ].df_i[ 7L ] =  cft[ n ].ea * ( l_tmp / cft[ n ].l - 1L );
}

/*****
/* COMPUTE END-OF ITERATION FORCES */
*****/

for ( ctrl = 1L; ctrl <= A_CFT_NUM_DOF; ctrl++ )
{
    cft[ n ].f2[ ctrl ] += cft[ n ].df_i[ ctrl ];
}

} /* for ( n = 1L; n <= size->num_cft_elems; n++ ) */

return( status );
}

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****

* FUNCTION- a_cft_plastic

    @(#) calculation of CFT PLASTIC stiffness matrix

* CALLED FROM- a_pl_cft_state

* ABSTRACT- This function calculates the cft plastic stiffness terms
    comprising the kp matrix which is used in turn to calculate
    the plastic reduction matrix.
*****/

#include <math.h>          /* C MATH FUNCTIONS          */
#include <stdio.h>         /* C I/O FUNCTIONS          */
#include <string.h>        /* C STRING FUNCTIONS       */

#include "a_mac.h"         /* MACRO DEFINITIONS        */
#include "a_extern.h"      /* EXTERNAL VARIABLES       */
#include "a_cft_el.h"      /* DATA STRUCTURE FOR CFT BEAM-COLUMN */

long a_cft_plastic_k( n, end, cft )

long n;                    /* CURRENT ELEMENT          */
long end;                  /* CURRENT ELEMENT END      */
A_CFT_BC cft[];           /* DATA STRUCTURE FOR CFT BEAM-COLUMN */

{
long status = 1L;         /* RETURN STATUS           */
long i = 0L;              /* ELEMENT END INDEX       */

/*****
/* CALCULATE INDEX, i, TO REFERENCE PROPER ELEMENT END */
*****/

i = 6L * end;

```

```

/*****
/* COMPUTE PLASTIC AND BOUNDING STIFFNESS TERMS */
*****/

/* AXIAL */
cft[ n ].kpb[ end ].p = cft[ n ].ea / cft[ n ].l *
                      ( cft[ n ].k1[ end ].p + cft[ n ].k2[ end ].p *
                        ( cft[ n ].dist[ end ] / ( cft[ n ].dist_in[ end ] -
                          cft[ n ].dist[ end ] ) ) );
cft[ n ].kpb[ end ].p = cft[ n ].ea / cft[ n ].l *
                      cft[ n ].k1[ end ].p;

/* WEAK AXIS BENDING */
cft[ n ].kpb[ end ].my = cft[ n ].eiy / cft[ n ].l *
                       ( cft[ n ].k1[ end ].my + cft[ n ].k2[ end ].my *
                         ( cft[ n ].dist[ end ] / ( cft[ n ].dist_in[ end ] -
                           cft[ n ].dist[ end ] ) ) );
cft[ n ].kpb[ end ].my = cft[ n ].eiy / cft[ n ].l *
                       cft[ n ].k1[ end ].my;

/* STRONG AXIS BENDING */
cft[ n ].kpb[ end ].mz = cft[ n ].eiz / cft[ n ].l *
                       ( cft[ n ].k1[ end ].mz + cft[ n ].k2[ end ].mz *
                         ( cft[ n ].dist[ end ] / ( cft[ n ].dist_in[ end ] -
                           cft[ n ].dist[ end ] ) ) );
cft[ n ].kpb[ end ].mz = cft[ n ].eiz / cft[ n ].l *
                       cft[ n ].k1[ end ].mz;

return( status );
}

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****

* FUNCTION- a_cft_reduction_k

* @(#) CFT plastic REDUCTION stiffness matrix

* CALLED FROM- a_cft_stiffness

* ABSTRACT- This function computes the CFT element plastic reduction
* stiffness terms. This function is only called if a hinge
* exists at one or both of the element ends.
*****/

#include <math.h>          /* C MATH FUNCTIONS          */
#include <stdio.h>        /* C I/O FUNCTIONS          */

#include "a_mac.h"        /* MACRO DEFINITIONS        */
#include "a_extern.h"     /* EXTERNAL VARIABLES       */
#include "a_cft_el.h"     /* DATA STRUCTURE FOR CFT BEAM-COLUMN */

long a_cft_reduction_k( n, cft )

long n;                  /* CURRENT ELEMENT NUMBER   */
A_CFT_BC cft[];         /* DATA STRUCTURE FOR CFT BEAM-COLUMN */

{
long status = 1L;       /* RETURN STATUS           */
long ctr1 = 0L;        /* COUNTER                 */
long ctr2 = 0L;        /* COUNTER                 */
long ctr3 = 0L;        /* COUNTER                 */
double a = 0.0;        /* STORAGE FOR INVERTING MATRIX */
double b = 0.0;        /* STORAGE FOR INVERTING MATRIX */
double c = 0.0;        /* STORAGE FOR INVERTING MATRIX */
double d = 0.0;        /* STORAGE FOR INVERTING MATRIX */
double inverse[ 3L ][ 3L ]; /* TEMPORARY MATRIX STORAGE */
}

```

```

double kp[ A_CFT_NUM_DOF + 1L ][ A_CFT_NUM_DOF + 1L ]; /* PLASTIC K */
double grad[ A_CFT_NUM_DOF + 1L ][ A_CFT_NUM_DOF + 1L ]; /* GRADIENT */
double gradt[ A_CFT_NUM_DOF + 1L ][ A_CFT_NUM_DOF + 1L ]; /* GRAD TRANSP*/
double temp1[ A_CFT_NUM_DOF + 1L ][ A_CFT_NUM_DOF + 1L ]; /* TEMP MATRIX*/
double temp2[ A_CFT_NUM_DOF + 1L ][ A_CFT_NUM_DOF + 1L ]; /* TEMP MATRIX*/
double temp3[ A_CFT_NUM_DOF + 1L ][ A_CFT_NUM_DOF + 1L ]; /* TEMP MATRIX*/
double temp4[ A_CFT_NUM_DOF + 1L ][ A_CFT_NUM_DOF + 1L ]; /* TEMP MATRIX*/
double temp5[ A_CFT_NUM_DOF + 1L ][ A_CFT_NUM_DOF + 1L ]; /* TEMP MATRIX*/

/*****/
/* INITIALIZE ARRAYS */
/*****/

for ( ctrl = 0L; ctrl < 3L; ctrl++ )
    {
    for ( ctr2 = 0L; ctr2 <= 3L; ctr2++ )
        {
        inverse[ ctrl ][ ctr2 ] = 0.0;
        }
    }

for ( ctrl = 0L; ctrl < A_CFT_NUM_DOF + 1L; ctrl++ )
    {
    for ( ctr2 = 0L; ctr2 < A_CFT_NUM_DOF + 1L; ctr2++ )
        {
        kp[ ctrl ][ ctr2 ] = 0.0;
        grad[ ctrl ][ ctr2 ] = 0.0;
        gradt[ ctrl ][ ctr2 ] = 0.0;
        temp1[ ctrl ][ ctr2 ] = 0.0;
        temp2[ ctrl ][ ctr2 ] = 0.0;
        temp3[ ctrl ][ ctr2 ] = 0.0;
        temp4[ ctrl ][ ctr2 ] = 0.0;
        temp5[ ctrl ][ ctr2 ] = 0.0;
        cft[ n ].kr[ ctrl ][ ctr2 ] = 0.0;
        }
    }

/*****/
/* FILL TEMP kp MATRIX */
/*****/

kp[ 1L ][ 1L ] = cft[ n ].kp[ 0L ].p;
kp[ 5L ][ 5L ] = cft[ n ].kp[ 0L ].my;
kp[ 6L ][ 6L ] = cft[ n ].kp[ 0L ].mz;
kp[ 7L ][ 7L ] = cft[ n ].kp[ 1L ].p;
kp[ 11L ][ 11L ] = cft[ n ].kp[ 1L ].my;
kp[ 12L ][ 12L ] = cft[ n ].kp[ 1L ].mz;

/*****/
/* COMPUTE GRADIENT AND GRADIENT TRANSPOSE ARRAYS */
/*****/

/* GRADIENT */

if ( ( cft[ n ].state[ 0L ] == PL ) && ( cft[ n ].state[ 1L ] == PL ) )
    {
    /* HINGE AT BOTH ENDS */

    a_pl_cft_grad( n, 0L, F1, cft );
    a_pl_cft_grad( n, 1L, F1, cft );

    grad[ 1L ][ 1L ] = cft[ n ].grad[ 0L ].p;
    grad[ 5L ][ 1L ] = cft[ n ].grad[ 0L ].my;
    grad[ 6L ][ 1L ] = cft[ n ].grad[ 0L ].mz;
    grad[ 7L ][ 2L ] = cft[ n ].grad[ 1L ].p;
    grad[ 11L ][ 2L ] = cft[ n ].grad[ 1L ].my;
    grad[ 12L ][ 2L ] = cft[ n ].grad[ 1L ].mz;
    }

else if ( ( cft[ n ].state[ 0L ] == PL ) && ( ( cft[ n ].state[ 1L ] == EL ) ||
( cft[ n ].state[ 1L ] == PL_IN )))
    {
    /* HINGE AT END I */
    a_pl_cft_grad( n, 0L, F1, cft );

```

```

        grad[ 1L ][ 1L ]      = cft[ n ].grad[ 0L ].p;
        grad[ 5L ][ 1L ]      = cft[ n ].grad[ 0L ].my;
        grad[ 6L ][ 1L ]      = cft[ n ].grad[ 0L ].mz;
    }

else if ( ( cft[ n ].state[ 1L ] == PL ) && ( ( cft[ n ].state[ 0L ] == EL ) ||
        ( cft[ n ].state[ 0L ] == PL_IN )))
    {
        /* HINGE AT END J */
        a_pl_cft_grad( n, 1L, F1, cft );

        grad[ 7L ][ 2L ]      = cft[ n ].grad[ 1L ].p;
        grad[ 11L ][ 2L ]     = cft[ n ].grad[ 1L ].my;
        grad[ 12L ][ 2L ]     = cft[ n ].grad[ 1L ].mz;
    }

/* GRADIENT TRANSPOSE */

for ( ctr1 = 1L; ctr1 <= A_CFT_NUM_DOF; ctr1++ )
    {
        for ( ctr2 = 1L; ctr2 <= A_CFT_NUM_DOF; ctr2++ )
            {
                gradt[ ctr2 ][ ctr1 ] = grad[ ctr1 ][ ctr2 ];
            }
    }

/*****
/* CALC. PLASTIC REDUCTION K ( -kt*grad*(gradt*(kt+kp)*grad)^(-1)*gradt*kt ) */
*****/

/** temp1 = gradt * ( kt + kp ) **/
for ( ctr1 = 1L; ctr1 <= A_CFT_NUM_DOF; ctr1++ )
    {
        for ( ctr2 = 1L; ctr2 <= A_CFT_NUM_DOF; ctr2++ )
            {
                temp1[ 1L ][ ctr1 ] += gradt[ 1L ][ ctr2 ] *
                    ( cft[ n ].kt[ ctr2 ][ ctr1 ] +
                      kp[ ctr2 ][ ctr1 ] );
                temp1[ 2L ][ ctr1 ] += gradt[ 2L ][ ctr2 ] *
                    ( cft[ n ].kt[ ctr2 ][ ctr1 ] +
                      kp[ ctr2 ][ ctr1 ] );
            }
    }

/** temp2 = temp1 * grad **/
for ( ctr1 = 1L; ctr1 <= A_CFT_NUM_DOF; ctr1++ )
    {
        for ( ctr2 = 1L; ctr2 <= A_CFT_NUM_DOF; ctr2++ )
            {
                temp2[ 1L ][ ctr1 ] += temp1[ 1L ][ ctr2 ] *
                    grad[ ctr2 ][ ctr1 ];
                temp2[ 2L ][ ctr1 ] += temp1[ 2L ][ ctr2 ] *
                    grad[ ctr2 ][ ctr1 ];
            }
    }

/** inverse = ( gradt * ( kt + kp ) * grad ) ^ (-1) = temp2 ^ (-1) **/
a = temp2[ 1L ][ 1L ];
b = temp2[ 1L ][ 2L ];
c = temp2[ 2L ][ 1L ];
d = temp2[ 2L ][ 2L ];

/* TAKE PROPER INVERSE DEPENDING ON HINGE LOCATION(S) */

if ( ( cft[ n ].state[ 0L ] == PL ) && ( cft[ n ].state[ 1L ] == PL ) )
    {
        /* HINGE AT BOTH ENDS */
        if ( ( a != 0.0 ) && ( ( a * d - b * c ) != 0.0 ) )
            {
                inverse[ 1L ][ 1L ] = ( c * b ) / ( a * ( a * d - b * c ) ) +
                    1 / a;
                inverse[ 1L ][ 2L ] = -b / ( a * d - b * c );
                inverse[ 2L ][ 1L ] = -c / ( a * d - b * c );
                inverse[ 2L ][ 2L ] = a / ( a * d - b * c );
            }
        else

```

```

        {
            printf( "\nDivide-by-zero error in a_cft_reduction_k. \n" );
            status = 0L;
        }
    }

else if ( ( cft[ n ].state[ 0L ] == PL ) && ( cft[ n ].state[ 1L ] == EL ) )
    {
        /* HINGE AT END I */
        if ( a != 0.0 )
            {
                inverse[ 1L ][ 1L ] = 1 / a;
            }
        else
            {
                printf( "\nDivide-by-zero error in a_cft_reduction_k. \n" );
                status = 0L;
            }
    }

else if ( ( cft[ n ].state[ 0L ] == EL ) && ( cft[ n ].state[ 1L ] == PL ) )
    {
        /* HINGE AT END J */
        if ( d != 0.0 )
            {
                inverse[ 2L ][ 2L ] = 1 / d;
            }
        else
            {
                printf( "\nDivide-by-zero error in a_cft_reduction_k. \n" );
                status = 0L;
            }
    }

/** temp3 = kt * grad **/
for ( ctr1 = 1L; ctr1 <= A_CFT_NUM_DOF; ctr1++ )
    {
        for ( ctr2 = 1L; ctr2 <= 2L; ctr2++ )
            {
                for ( ctr3 = 1L; ctr3 <= A_CFT_NUM_DOF; ctr3++ )
                    {
                        temp3[ ctr1 ][ ctr2 ] += cft[ n ].kt[ ctr1 ][ ctr3 ] *
                                                grad[ ctr3 ][ ctr2 ];
                    }
            }
    }

/** temp4 = temp3 * inverse **/
for ( ctr1 = 1L; ctr1 <= A_CFT_NUM_DOF; ctr1++ )
    {
        for ( ctr2 = 1L; ctr2 <= 2L; ctr2++ )
            {
                for ( ctr3 = 1L; ctr3 <= 2L; ctr3++ )
                    {
                        temp4[ ctr1 ][ ctr2 ] += temp3[ ctr1 ][ ctr3 ] *
                                                inverse[ ctr3 ][ ctr2 ];
                    }
            }
    }

/** temp5 = temp4 * gradt **/
for ( ctr1 = 1L; ctr1 <= A_CFT_NUM_DOF; ctr1++ )
    {
        for ( ctr2 = 1L; ctr2 <= A_CFT_NUM_DOF; ctr2++ )
            {
                for ( ctr3 = 1L; ctr3 <= 2L; ctr3++ )
                    {
                        temp5[ ctr1 ][ ctr2 ] += temp4[ ctr1 ][ ctr3 ] *
                                                gradt[ ctr3 ][ ctr2 ];
                    }
            }
    }

/** cft[ n ].kr = temp5 * kt **/
for ( ctr1 = 1L; ctr1 <= A_CFT_NUM_DOF; ctr1++ )
    {

```

```

        for ( ctr2 = 1L; ctr2 <= A_CFT_NUM_DOF; ctr2++ )
            {
                for ( ctr3 = 1L; ctr3 <= A_CFT_NUM_DOF; ctr3++ )
                    {
                        cft[ n ].kr[ ctr1 ][ ctr2 ] +=
                            temp5[ ctr1 ][ ctr3 ] *
                            cft[ n ].kt[ ctr3 ][ ctr2 ];
                    }
            }
    }

return( status );
}

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****

* FUNCTION- a_cft_rotate

        @(#) CFT member force ROTATE from n to n + 1

* CALLED FROM- a_drv_dynamic, a_drv_static

* ABSTRACT- This function takes the forces obtained from the structural
configuration at time n and rotates them to the structural
configuration at time n + 1. The global joint forces are
then calculated. Also, the function which updates the
CFT members (i.e., lengths, etc.) is called in this routine.
*****/

#include <math.h>           /* C MATH FUNCTIONS           */
#include <stdio.h>          /* C I/O FUNCTIONS           */
#include <string.h>         /* C STRING FUNCTIONS        */

#include "a_mac.h"          /* MACRO DEFINITIONS         */
#include "a_extern.h"       /* EXTERNAL VARIABLES        */
#include "a_jt.h"           /* DATA STRUCTURE DEFINING JOINT PROPS */
#include "a_cft_el.h"       /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
#include "a_param.h"        /* DATA STRUCTURE OF VARIOUS PARAMETERS */
#include "a_global.h"       /* DATA STRUCTURE OF DOF-BASED VARIABLES */

long a_cft_rotate( size, nr, flag, lhist, jt, cft, global )

A_MODEL_SIZE *size;        /* DATA STRUCTURE OF GLOBAL STRUCTURE SIZE*/
A_NEW_RAPH *nr;           /* DATA STRUCTURE OF NEWTON-RAPHSON PARAMS*/
A_FLAGS *flag;            /* DATA STRUCTURE OF PROGRAM FLAGS */
A_LOADHIST lhist[];       /* DATA STRUCTURE OF LOAD HISTORY PARAMS */
A_JT_JOINT jt[];         /* DATA STRUCTURE FOR SINGLE JOINT */
A_CFT_BC cft[];          /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
A_GLOBAL global[];        /* DATA STRUCTURE OF DOF-BASED VARIABLES */

{
    long status = 1L;      /* RETURN STATUS */
    long i = 0L;          /* TEMP VARIABLE */
    long j = 0L;          /* TEMP VARIABLE */
    long k = 0L;          /* TEMP VARIABLE */
    long n = 0L;          /* ELEMENT COUNTER */
    long dof = 0L;        /* DOF COUNTER */
    long ctr1 = 0L;       /* COUNTER */
    long ctr2 = 0L;       /* COUNTER */
    double l = 0.0;       /* SHORTENED VARIABLE FOR STEEL MEMBER LENGTH */
    double m2 = 0.0;      /* SHORTENED VARIABLE FOR DOF 2 MULTIPLIER */
    double m6 = 0.0;      /* SHORTENED VARIABLE FOR DOF 6 MULTIPLIER */
    double m8 = 0.0;      /* SHORTENED VARIABLE FOR DOF 8 MULTIPLIER */
    double m12 = 0.0;     /* SHORTENED VARIABLE FOR DOF 12 MULTIPLIER */
    double r2 = 0.0;      /* RATIO OF df_i / f2 FOR DOF 2 */
    double r6 = 0.0;      /* RATIO OF df_i / f2 FOR DOF 6 */
    double r8 = 0.0;      /* RATIO OF df_i / f2 FOR DOF 8 */
    double r12 = 0.0;     /* RATIO OF df_i / f2 FOR DOF 12 */
}

```

```

double wl      = 0.0;          /* INCREMENTAL DISTRIBUTED LOAD          */
double temp_df_i[ A_CFT_NUM_DOF + 1L ]; /* TEMP GLOBAL INCR FORCES          */
double temp_f2[ A_CFT_NUM_DOF + 1L ]; /* TEMP GLOBAL FORCES AT STEP n + 1 */

/*****
/* LOOP OVER ELEMENTS; COMPUTE GLOBAL JOINT FORCES & ROTATED MEMBER FORCES */
*****/

for ( n = 1L; n <= size->num_cft_elems; n++ )
{
/*****
/* UPDATE ELEMENT GEOMETRY, ROTATION MATRICES, ETC. */
*****/

if ( flag->kg )
{
    if ( ! a_cft_update( n, jt, cft, global ) )
    {
        status = 0L;
        printf(
            "\nMember calculations failed. Exit program.\n" );
    }
}

/*****
/* ROTATE LOCAL ELEMENT FORCES TO GLOBAL W/ C2 MATRICES */
*****/

for ( ctr1 = 1L; ctr1 <= A_CFT_NUM_DOF; ctr1++ )
{
    temp_df_i[ ctr1 ] = 0.0;
    temp_f2[ ctr1 ] = 0.0;

    for ( ctr2 = 1L; ctr2 <= A_CFT_NUM_DOF; ctr2++ )
    {
        temp_df_i[ ctr1 ] += cft[ n ].lambda[ ctr2 ][ ctr1 ] *
            cft[ n ].df_i[ ctr2 ];
        temp_f2[ ctr1 ] += cft[ n ].lambda[ ctr2 ][ ctr1 ] *
            cft[ n ].f2[ ctr2 ];
    }
}

/*****
/* ASSEMBLE THE FORCES FOR EACH JOINT DOF */
*****/

for ( dof = 1L; dof <= A_JT_DOF; dof++ )
{
    jt[ cft[ n ].i ].df_i[ dof ] += temp_df_i[ dof ];
    jt[ cft[ n ].j ].df_i[ dof ] += temp_df_i[ dof + 6L ];
}

/*****
/* ROTATE FORCES FROM GLOBAL TO LOCAL W/ C2 MATRICES */
*****/

for ( ctr1 = 1L; ctr1 <= A_CFT_NUM_DOF; ctr1++ )
{
    cft[ n ].df_i[ ctr1 ] = 0.0;
    cft[ n ].f2[ ctr1 ] = 0.0;

    for ( ctr2 = 1L; ctr2 <= A_CFT_NUM_DOF; ctr2++ )
    {
        cft[ n ].df_i[ ctr1 ] += temp_df_i[ ctr2 ] *
            cft[ n ].lambda[ ctr1 ][ ctr2 ];
        cft[ n ].f2[ ctr1 ] += temp_f2[ ctr2 ] *
            cft[ n ].lambda[ ctr1 ][ ctr2 ];
    }
}

/*****
/* COMPUTE INCREMENTAL ADJUSTED FORCES */
*****/

```

```

for ( ctrl = 1L; ctrl <= A_CFT_NUM_DOF; ctrl++ )
    {
        cft[ n ].f1_i[ ctrl ] = cft[ n ].f2_i[ ctrl ];
        cft[ n ].f2_i[ ctrl ] += cft[ n ].df_i[ ctrl ];
    }

} /* for ( n = 1L; n <= size->num_cft_elems; n++ ) */

return( status );
}

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****
*****

* FUNCTION- a_cft_stiffness

        @(#) CFT element STIFFNESS matrix formulation

* CALLED FROM- a_drv_dynamic, a_drv_static

* ABSTRACT- This function calls the subroutines to compute the cft member
        elastic, geometric, and plastic reduction matrices.
*****/

#include <stdio.h> /* C I/O FUNCTIONS */
#include <math.h> /* MATH FUNCTIONS */

#include "a_mac.h" /* MACRO DEFINITIONS */
#include "a_extern.h" /* EXTERNAL VARIABLES */
#include "a_jt.h" /* DATA STRUCTURE FOR JOINTS */
#include "a_cft_el.h" /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
#include "a_param.h" /* DATA STRUCTURE OF VARIOUS PARAMETERS */

long a_cft_stiffness ( n, size, flag, jt, cft )

long n; /* CURRENT ELEMENT NUMBER */
A_MODEL_SIZE *size; /* DATA STRUCTURE OF GLOBAL STRUCT PARAMS*/
A_FLAGS *flag; /* DATA STRUCTURE OF PROGRAM FLAGS */
A_JT_JOINT jt[]; /* DATA STRUCTURE FOR JOINTS */
A_CFT_BC cft[]; /* DATA STRUCTURE FOR CFT BEAM-COLUMN */

{
long status = 1L; /* RETURN STATUS */
long ctrl = 0L; /* COUNTER */
long ctr2 = 0L; /* COUNTER */
long end = 0L; /* COUNTER FOR CURRENT ELEMENT END */

/*****/
/* COMPUTE TANGENT STIFFNESS TERMS */
/*****/

if ( ! a_cft_tangent_k( n, flag, cft ) )
    {
        status = 0L;
        printf( "\nTangent stiffness computation failed. Exit program.\n" );
    }

/*****/
/* ALTER THE TANGENT K MATRIX TO ACCOUNT FOR MEMBER RELEASES */
/*****/

if ( cft[ n ].release )
    {
        a_k_mem_rel( A_CFT_NUM_DOF, cft[ n ].release, cft[ n ].kt );
    }
}

```

```

/*****
/* COMPUTE CFT PLASTIC REDUCTION STIFFNESS TERMS (ONLY IF PLASTIC) */
*****/

if ( flag->kp )
{
for ( ctr1 = 1L; ctr1 <= A_CFT_NUM_DOF; ctr1++ )
    {
        /* ZERO kr FOR ALL ELEMS (THIS IS ESPECIALLY FOR UNLOADING) */
        for ( ctr2 = 1L; ctr2 <= A_CFT_NUM_DOF; ctr2++ )
            {
                cft[ n ].kr[ ctr1 ][ ctr2 ] = 0.0;
            }
    }

if ( ( cft[ n ].state[ 0L ] == PL ) || ( cft[ n ].state[ 1L ] == PL ) )
    {
        if ( ! a_cft_reduction_k( n, cft ) )
            {
                status = 0L;
                printf( "\nPlastic reduction stiffness computation failed." );
                printf( "Exit program.\n" );
            }
    }

} /* if ( flag->kp ) */

return( status );
}

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****

* FUNCTION- a_cft_tangent_k

        @(#) CFT TANGENT stiffness (K) matrix

* CALLED FROM- a_cft_stiffness

* ABSTRACT- This function computes the local CFT element tangent stiffness
        terms (elastic + geometric).
*****/

#include <math.h>          /* C MATH FUNCTIONS          */
#include <stdio.h>        /* C I/O FUNCTIONS          */

#include "a_mac.h"        /* MACRO DEFINITIONS        */
#include "a_extern.h"     /* EXTERNAL VARIABLES & PROTOTYPES */
#include "a_param.h"      /* DATA STRUCTURE OF VARIOUS PARAMETERS */
#include "a_cft_el.h"     /* DATA STRUCTURE FOR CFT BEAM-COLUMN */

long a_cft_tangent_k ( n, flag, cft )

long n;          /* CURRENT ELEMENT NUMBER          */
A_FLAGS *flag;  /* DATA STRUCTURE OF PROGRAM FLAGS */
A_CFT_BC cft[]; /* DATA STRUCTURE FOR CFT BEAM-COLUMN */

{
long status = 1L; /* RETURN STATUS          */
long ctr1 = 0L;  /* COUNTER          */
long ctr2 = 0L;  /* COUNTER          */
long ctr3 = 0L;  /* COUNTER          */
long init1 = 0L; /* COUNTER FOR INITIALIZATIN OF ARRAYS */
long init2 = 0L; /* COUNTER FOR INITIALIZATIN OF ARRAYS */

/*****
/* INITIALIZE kt */
*****/

```

```

for ( init1 = 0L; init1 < A_CFT_NUM_DOF + 1L; init1++ )
    {
        for ( init2 = 0L; init2 < A_CFT_NUM_DOF + 1L; init2++ )
            {
                cft[ n ].kt[ init1 ][ init2 ] = 0.0;
            }
    }

/*****
/* UPDATE CONCRETE ELASTIC STIFFNESS PARAMETERS */
*****/

cft[ n ].gmod_c = cft[ n ].emod_c / ( 2.0 * ( 1 + cft[ n ].nu_c ) );

cft[ n ].eiy = cft[ n ].emod_s * cft[ n ].iy_s +
               cft[ n ].emod_c * cft[ n ].iy_c;
cft[ n ].eiz = cft[ n ].emod_s * cft[ n ].iz_s +
               cft[ n ].emod_c * cft[ n ].iz_c;
cft[ n ].ea = cft[ n ].emod_s * cft[ n ].a_stl +
              cft[ n ].emod_c * cft[ n ].a_conc;
cft[ n ].gj = cft[ n ].gmod_s * cft[ n ].ix;

/*****
/* GENERATE ELASTIC TERMS IN UPPER TRIANGULAR PORTION OF kt */
*****/

cft[ n ].kt[ 1 ][ 1 ] = cft[ n ].ea / cft[ n ].l;
cft[ n ].kt[ 1 ][ 7 ] = - cft[ n ].kt[ 1 ][ 1 ];
cft[ n ].kt[ 2 ][ 2 ] = 12.0 * cft[ n ].eiz / pow( cft[ n ].l, 3 );
cft[ n ].kt[ 2 ][ 6 ] = 6.0 * cft[ n ].eiz / pow( cft[ n ].l, 2 );
cft[ n ].kt[ 2 ][ 8 ] = - cft[ n ].kt[ 2 ][ 2 ];
cft[ n ].kt[ 2 ][ 12 ] = cft[ n ].kt[ 2 ][ 6 ];
cft[ n ].kt[ 3 ][ 3 ] = 12.0 * cft[ n ].eiy / pow( cft[ n ].l, 3 );
cft[ n ].kt[ 3 ][ 5 ] = - 6.0 * cft[ n ].eiy / pow( cft[ n ].l, 2 );
cft[ n ].kt[ 3 ][ 9 ] = - cft[ n ].kt[ 3 ][ 3 ];
cft[ n ].kt[ 3 ][ 11 ] = cft[ n ].kt[ 3 ][ 5 ];
cft[ n ].kt[ 4 ][ 4 ] = cft[ n ].gj / cft[ n ].l;
cft[ n ].kt[ 4 ][ 10 ] = - cft[ n ].kt[ 4 ][ 4 ];
cft[ n ].kt[ 5 ][ 5 ] = 4.0 * cft[ n ].eiy / cft[ n ].l;
cft[ n ].kt[ 5 ][ 9 ] = - cft[ n ].kt[ 3 ][ 5 ];
cft[ n ].kt[ 5 ][ 11 ] = 2.0 * cft[ n ].eiy / cft[ n ].l;
cft[ n ].kt[ 6 ][ 6 ] = 4.0 * cft[ n ].eiz / cft[ n ].l;
cft[ n ].kt[ 6 ][ 8 ] = - cft[ n ].kt[ 2 ][ 6 ];
cft[ n ].kt[ 6 ][ 12 ] = 2.0 * cft[ n ].eiz / cft[ n ].l;
cft[ n ].kt[ 7 ][ 7 ] = cft[ n ].kt[ 1 ][ 1 ];
cft[ n ].kt[ 8 ][ 8 ] = cft[ n ].kt[ 2 ][ 2 ];
cft[ n ].kt[ 8 ][ 12 ] = - cft[ n ].kt[ 2 ][ 6 ];
cft[ n ].kt[ 9 ][ 9 ] = cft[ n ].kt[ 3 ][ 3 ];
cft[ n ].kt[ 9 ][ 11 ] = - cft[ n ].kt[ 3 ][ 5 ];
cft[ n ].kt[ 10 ][ 10 ] = cft[ n ].kt[ 4 ][ 4 ];
cft[ n ].kt[ 11 ][ 11 ] = cft[ n ].kt[ 5 ][ 5 ];
cft[ n ].kt[ 12 ][ 12 ] = cft[ n ].kt[ 6 ][ 6 ];

/*****
/* GENERATE GEOMETRIC TERMS IN UPPER TRIANGULAR PORTION OF kt */
*****/

if ( flag->kg )
    {
        cft[ n ].kt[ 2 ][ 2 ] += 1.2 * cft[ n ].f2[ 7 ] / cft[ n ].l;
        cft[ n ].kt[ 2 ][ 6 ] += cft[ n ].f2[ 7 ] / 10.0;
        cft[ n ].kt[ 2 ][ 8 ] += - 1.2 * cft[ n ].f2[ 7 ] / cft[ n ].l;
        cft[ n ].kt[ 2 ][ 12 ] += cft[ n ].f2[ 7 ] / 10.0;
        cft[ n ].kt[ 3 ][ 3 ] += 1.2 * cft[ n ].f2[ 7 ] / cft[ n ].l;
        cft[ n ].kt[ 3 ][ 5 ] += - cft[ n ].f2[ 7 ] / 10.0;
        cft[ n ].kt[ 3 ][ 9 ] += - 1.2 * cft[ n ].f2[ 7 ] / cft[ n ].l;
        cft[ n ].kt[ 3 ][ 11 ] += - cft[ n ].f2[ 7 ] / 10.0;
        cft[ n ].kt[ 5 ][ 5 ] += 2.0 * cft[ n ].f2[ 7 ] * cft[ n ].l / 15.0;
        cft[ n ].kt[ 5 ][ 9 ] += cft[ n ].f2[ 7 ] / 10.0;
        cft[ n ].kt[ 5 ][ 11 ] += - cft[ n ].f2[ 7 ] * cft[ n ].l / 30.0;
        cft[ n ].kt[ 6 ][ 6 ] += 2.0 * cft[ n ].f2[ 7 ] * cft[ n ].l / 15.0;
        cft[ n ].kt[ 6 ][ 8 ] += - cft[ n ].f2[ 7 ] / 10.0;
        cft[ n ].kt[ 6 ][ 12 ] += - cft[ n ].f2[ 7 ] * cft[ n ].l / 30.0;
    }

```

```

cft[ n ].kt[ 8 ][ 8 ] += 1.2 * cft[ n ].f2[ 7 ] / cft[ n ].l;
cft[ n ].kt[ 8 ][ 12 ] += - cft[ n ].f2[ 7 ] / 10.0;
cft[ n ].kt[ 9 ][ 9 ] += 1.2 * cft[ n ].f2[ 7 ] / cft[ n ].l;
cft[ n ].kt[ 9 ][ 11 ] += cft[ n ].f2[ 7 ] / 10.0;
cft[ n ].kt[ 11 ][ 11 ] += 2.0 * cft[ n ].f2[ 7 ] * cft[ n ].l / 15.0;
cft[ n ].kt[ 12 ][ 12 ] += 2.0 * cft[ n ].f2[ 7 ] * cft[ n ].l / 15.0;

/* HIGHER ORDER GEOMETRIC STIFFNESS TERMS */

if ( flag->order == HIGHORDER )
{
cft[ n ].kt[ 1 ][ 2 ] = ( cft[ n ].f2[ 6 ] + cft[ n ].f2[ 12 ] ) /
                        pow( cft[ n ].l, 2 );
cft[ n ].kt[ 1 ][ 3 ] = - ( cft[ n ].f2[ 5 ] + cft[ n ].f2[ 11 ] ) /
                        pow( cft[ n ].l, 2 );
cft[ n ].kt[ 1 ][ 8 ] = - ( cft[ n ].f2[ 6 ] + cft[ n ].f2[ 12 ] ) /
                        pow( cft[ n ].l, 2 );
cft[ n ].kt[ 1 ][ 9 ] = ( cft[ n ].f2[ 5 ] + cft[ n ].f2[ 11 ] ) /
                        pow( cft[ n ].l, 2 );
cft[ n ].kt[ 2 ][ 4 ] = cft[ n ].f2[ 5 ] / cft[ n ].l;
cft[ n ].kt[ 2 ][ 5 ] = cft[ n ].f2[ 10 ] / cft[ n ].l;
cft[ n ].kt[ 2 ][ 7 ] = - ( cft[ n ].f2[ 6 ] + cft[ n ].f2[ 12 ] ) /
                        pow( cft[ n ].l, 2 );
cft[ n ].kt[ 2 ][ 10 ] = cft[ n ].f2[ 11 ] / cft[ n ].l;
cft[ n ].kt[ 2 ][ 11 ] = - cft[ n ].f2[ 10 ] / cft[ n ].l;
cft[ n ].kt[ 3 ][ 4 ] = cft[ n ].f2[ 6 ] / cft[ n ].l;
cft[ n ].kt[ 3 ][ 6 ] = cft[ n ].f2[ 10 ] / cft[ n ].l;
cft[ n ].kt[ 3 ][ 7 ] = ( cft[ n ].f2[ 5 ] + cft[ n ].f2[ 11 ] ) /
                        pow( cft[ n ].l, 2 );
cft[ n ].kt[ 3 ][ 10 ] = cft[ n ].f2[ 12 ] / cft[ n ].l;
cft[ n ].kt[ 3 ][ 12 ] = - cft[ n ].f2[ 10 ] / cft[ n ].l;
cft[ n ].kt[ 4 ][ 4 ] += cft[ n ].f2[ 7 ] * ( cft[ n ].ip ) /
                        ( ( cft[ n ].a_stl + cft[ n ].a_conc )
                          * cft[ n ].l );
cft[ n ].kt[ 4 ][ 5 ] = cft[ n ].f2[ 12 ] / 6.0 - cft[ n ].f2[ 6 ] / 3.0;
cft[ n ].kt[ 4 ][ 6 ] = cft[ n ].f2[ 5 ] / 3.0 - cft[ n ].f2[ 11 ] / 6.0;
cft[ n ].kt[ 4 ][ 8 ] = - cft[ n ].f2[ 5 ] / cft[ n ].l;
cft[ n ].kt[ 4 ][ 9 ] = - cft[ n ].f2[ 6 ] / cft[ n ].l;
cft[ n ].kt[ 4 ][ 10 ] += - cft[ n ].f2[ 7 ] * ( cft[ n ].ip ) /
                        ( ( cft[ n ].a_stl + cft[ n ].a_conc )
                          * cft[ n ].l );
cft[ n ].kt[ 4 ][ 11 ] = - ( cft[ n ].f2[ 6 ] + cft[ n ].f2[ 12 ] ) / 6.0;
cft[ n ].kt[ 4 ][ 12 ] = ( cft[ n ].f2[ 5 ] + cft[ n ].f2[ 11 ] ) / 6.0;
cft[ n ].kt[ 5 ][ 8 ] = - cft[ n ].f2[ 10 ] / cft[ n ].l;
cft[ n ].kt[ 5 ][ 10 ] = - ( cft[ n ].f2[ 6 ] + cft[ n ].f2[ 12 ] ) / 6.0;
cft[ n ].kt[ 5 ][ 12 ] = cft[ n ].f2[ 10 ] / 2.0;
cft[ n ].kt[ 6 ][ 9 ] = - cft[ n ].f2[ 10 ] / cft[ n ].l;
cft[ n ].kt[ 6 ][ 10 ] = ( cft[ n ].f2[ 5 ] + cft[ n ].f2[ 11 ] ) / 6.0;
cft[ n ].kt[ 6 ][ 11 ] = - cft[ n ].f2[ 10 ] / 2.0;
cft[ n ].kt[ 7 ][ 8 ] = ( cft[ n ].f2[ 6 ] + cft[ n ].f2[ 12 ] ) /
                        pow( cft[ n ].l, 2 );
cft[ n ].kt[ 7 ][ 9 ] = - ( cft[ n ].f2[ 5 ] + cft[ n ].f2[ 11 ] ) /
                        pow( cft[ n ].l, 2 );
cft[ n ].kt[ 8 ][ 10 ] = - cft[ n ].f2[ 11 ] / cft[ n ].l;
cft[ n ].kt[ 8 ][ 11 ] = cft[ n ].f2[ 10 ] / cft[ n ].l;
cft[ n ].kt[ 9 ][ 10 ] = - cft[ n ].f2[ 12 ] / cft[ n ].l;
cft[ n ].kt[ 9 ][ 12 ] = cft[ n ].f2[ 10 ] / cft[ n ].l;
cft[ n ].kt[ 10 ][ 10 ] += cft[ n ].f2[ 7 ] * ( cft[ n ].ip ) /
                        ( ( cft[ n ].a_stl + cft[ n ].a_conc )
                          * cft[ n ].l );
cft[ n ].kt[ 10 ][ 11 ] = cft[ n ].f2[ 6 ] / 6.0 - cft[ n ].f2[ 12 ] / 3.0;
cft[ n ].kt[ 10 ][ 12 ] = cft[ n ].f2[ 11 ] / 3.0 - cft[ n ].f2[ 5 ] / 6.0;
} /* if ( flag->order == HIGHORDER ) */

} /* if ( flag->kg ) */

/*****
/* GENERATE TERMS IN LOWER TRIANGULAR PORTION OF cft[ n ].kt */
*****/

for ( ctrl = 1L; ctrl <= 11L; ctrl++ )
{

```

```

        for ( ctr2 = ( ctr1 + 1L ); ctr2 <= 12L; ctr2++ )
            {
                cft[ n ].kt[ ctr2 ][ ctr1 ] =
                    cft[ n ].kt[ ctr1 ][ ctr2 ];
            }
    }

return( status );
}

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****
*****/

* FUNCTION- a_cft_update

        @(#) CFT member properties UPDATE

* CALLED FROM- a_drv_dynamic, a_drv_static, a_cft_frecovery

* ABSTRACT- This function computes CFT member lengths, direction cosines,
        and transformation matrices
*****/

#include <math.h>                /* C MATH FUNCTIONS                */
#include <stdio.h>               /* C I/O FUNCTIONS                */

#include "a_mac.h"               /* MACRO DEFINITIONS              */
#include "a_extern.h"           /* EXTERNAL VARIABLES & PROTOTYPES */
#include "a_jt.h"               /* DATA STRUCTURE DEFINING JOINT PROPS */
#include "a_cft_el.h"           /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
#include "a_param.h"            /* DATA STRUCTURE OF VARIOUS PARAMETERS */
#include "a_global.h"           /* DATA STRUCTURE OF DOF-BASED VARIABLES */

long a_cft_update ( n, jt, cft, global )

long n;                          /* CURRENT ELEMENT NUMBER          */
A_JT_JOINT jt[];                 /* DATA STRUCTURE FOR JOINT        */
A_CFT_BC cft[];                  /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
A_GLOBAL global[];               /* DATA STRUCTURE OF DOF-BASED VARIABLES */

{
    long status = 1L;             /* RETURN STATUS                   */
    long ctr1 = 0L;              /* COUNTER                          */
    long ctr2 = 0L;              /* COUNTER                          */
    long i = 0L;                 /* INDEX                            */
    double xl = 0.0;             /* ELEMENT LENGTH IN GLOBAL X-DIRECTION */
    double yl = 0.0;             /* ELEMENT LENGTH IN GLOBAL Y-DIRECTION */
    double zl = 0.0;             /* ELEMENT LENGTH IN GLOBAL Z-DIRECTION */
    double cx = 0.0;             /* X-DIRECTION COSINE OF MEMBER      */
    double cy = 0.0;             /* Y-DIRECTION COSINE OF MEMBER      */
    double cz = 0.0;             /* Z-DIRECTION COSINE OF MEMBER      */
    double theta_i = 0.0;        /* MEMBER TORSIONAL ROTATION OF i-END */
    double theta_j = 0.0;        /* MEMBER TORSIONAL ROTATION OF j-END */
    double theta = 0.0;          /* AVERAGE OF INCR. TORSIONAL ROTATIONS */
    double iendx = 0.0;          /* TEMP I-END VECTOR (GLOBAL X COMP) */
    double iendy = 0.0;          /* TEMP I-END VECTOR (GLOBAL Y COMP) */
    double iendz = 0.0;          /* TEMP I-END VECTOR (GLOBAL Z COMP) */
    double outp = 0.0;           /* LENGTH OF OUT-OF-PLANE VECTOR     */
    double outpx = 0.0;          /* GLOBAL X-DIRECTION COMPONENT      */
    double outpy = 0.0;          /* GLOBAL Y-DIRECTION COMPONENT      */
    double outpz = 0.0;          /* GLOBAL Z-DIRECTION COMPONENT      */

/*****
*****/
/* STORE OLD MEMBER LENGTH; COMPUTE NEW */

```

```

/*****/
cft[ n ].l_last = cft[ n ].l;

xl = jt[ cft[ n ].j ].co.x - jt[ cft[ n ].i ].co.x;
yl = jt[ cft[ n ].j ].co.y - jt[ cft[ n ].i ].co.y;
zl = jt[ cft[ n ].j ].co.z - jt[ cft[ n ].i ].co.z;

cft[ n ].l = sqrt( pow( xl, 2 ) + pow( yl, 2 ) + pow( zl, 2 ) );

/*****/
/* COMPUTE DIRECTION COSINES OF MEMBER */
/*****/

cx = xl / cft[ n ].l;
cy = yl / cft[ n ].l;
cz = zl / cft[ n ].l;

/*****/
/* CALCULATE TEMPORARY I-END UNIT VECTOR: {CURRENT I-END VECTOR} + */
/* {RELATIVE MEMBER ROTATION} * {LAST STEP OUT-OF-PLANE VECTOR} */
/*****/

/* CALCULATE RELATIVE LOCAL ROTATION OF MEMBER ENDS */

theta_i = global[ cft[ n ].mcode[ 4 ] ].dqi * cft[ n ].lambda[ 4 ][ 4 ] +
          global[ cft[ n ].mcode[ 5 ] ].dqi * cft[ n ].lambda[ 4 ][ 5 ] +
          global[ cft[ n ].mcode[ 6 ] ].dqi * cft[ n ].lambda[ 4 ][ 6 ];
theta_j = global[ cft[ n ].mcode[ 10 ] ].dqi * cft[ n ].lambda[ 10 ][ 10 ] +
          global[ cft[ n ].mcode[ 11 ] ].dqi * cft[ n ].lambda[ 10 ][ 11 ] +
          global[ cft[ n ].mcode[ 12 ] ].dqi * cft[ n ].lambda[ 10 ][ 12 ];
theta   = ( theta_i + theta_j ) / 2.0;

/* COMPUTE TEMPORARY I-END VECTOR */

iendx = cft[ n ].iend_x + tan( theta ) * cft[ n ].outp_x;
iendy = cft[ n ].iend_y + tan( theta ) * cft[ n ].outp_y;
iendz = cft[ n ].iend_z + tan( theta ) * cft[ n ].outp_z;

/*****/
/* COMPUTE OUT-OF-PLANE VECTOR: {outp} = {c} x {iend} */
/*****/

outpx = ( cy * iendz ) - ( cz * iendy );
outpy = ( cz * iendx ) - ( cx * iendz );
outpz = ( cx * iendy ) - ( cy * iendx );
outp  = sqrt( pow( outpx, 2 ) + pow( outpy, 2 ) + pow( outpz, 2 ) );

cft[ n ].outp_x = outpx / outp;
cft[ n ].outp_y = outpy / outp;
cft[ n ].outp_z = outpz / outp;

/*****/
/* CALCULATE NEW I-END VECTOR AS CROSS PRODUCT: {iend} = {outp} x {c} */
/*****/

cft[ n ].iend_x = cft[ n ].outp_y * cz - cft[ n ].outp_z * cy;
cft[ n ].iend_y = cft[ n ].outp_z * cx - cft[ n ].outp_x * cz;
cft[ n ].iend_z = cft[ n ].outp_x * cy - cft[ n ].outp_y * cx;

/*****/
/* INITIALIZE LAMBDA MATRICES */
/*****/

for ( ctr1 = 0L; ctr1 <= A_CFT_NUM_DOF; ctr1++ )
    {
        for ( ctr2 = 0L; ctr2 <= A_CFT_NUM_DOF; ctr2++ )
            {
                cft[ n ].lambda[ ctr1 ][ ctr2 ] = 0.0;
            }
    }

/*****/
/* COMPUTE LAMBDA (GLOBAL TO LOCAL TRANSFORMATION MATRIX) */

```

```

/*****/
for ( ctrl = 0L; ctrl <= 3L; ctrl++ )
    {
        i = 3L * ctrl;
        cft[ n ].lambda[ 1+i ][ 1+i ] = cx;
        cft[ n ].lambda[ 1+i ][ 2+i ] = cy;
        cft[ n ].lambda[ 1+i ][ 3+i ] = cz;
        cft[ n ].lambda[ 2+i ][ 1+i ] = cft[ n ].iend_x;
        cft[ n ].lambda[ 2+i ][ 2+i ] = cft[ n ].iend_y;
        cft[ n ].lambda[ 2+i ][ 3+i ] = cft[ n ].iend_z;
        cft[ n ].lambda[ 3+i ][ 1+i ] = cft[ n ].outp_x;
        cft[ n ].lambda[ 3+i ][ 2+i ] = cft[ n ].outp_y;
        cft[ n ].lambda[ 3+i ][ 3+i ] = cft[ n ].outp_z;
    }

return( status );
}

/*****/
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****

* FUNCTION- a_drv_static

    @(#) analysis DRiVer--STATIC

* CALLED FROM- cftmacro

* ABSTRACT- This function calls the subroutines which perform the
    non-linear static analysis of the structure based on
    the present time step which is incremented in this
    function.
*****

#include <math.h>           /* C MATH FUNCTIONS           */
#include <stdio.h>          /* C I/O FUNCTIONS           */
#include <string.h>         /* C STRING FUNCTIONS        */

#include "a_mac.h"          /* MACRO DEFINITIONS         */
#include "a_extern.h"       /* EXTERNAL VARIABLES & PROTOTYPES */
#include "sstmatch.h"       /* TOLERANCE COMPARISONS     */
#include "a_jt.h"           /* DATA STRUCTURE DEFINING JOINT PROPS */
#include "a_cft_el.h"       /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
#include "a_stl_el.h"       /* DATA STRUCTURE FOR STEEL BEAM-COLUMN */
#include "a_param.h"        /* DATA STRUCTURE OF VARIOUS PARAMETERS */
#include "a_global.h"       /* DATA STRUCTURE OF DOF-BASED VARIABLES */

long a_drv_static ( diag, kt, size, time, nr, tol, flag, acclg, lhist, jt,
    cft, stl, global )

long diag[];               /* INDEX OF SKYLINE K MATRIX DIAG TERMS */
double kt[];              /* SKYLINE GLOBAL STIFFNESS MATRIX */
A_MODEL_SIZE *size;       /* DATA STRUCTURE OF GLOBAL STRUCT SIZE */
A_TIME *time;             /* DATA STRUCTURE FOR TIME PARAMETERS */
A_NEW_RAPH *nr;          /* DATA STRUCTURE--NEWTON-RAPHSON PARAMS */
A_TOLERANCE tol;         /* DATA STRUCTURE OF PROGRAM TOLERANCES */
A_FLAGS *flag;           /* DATA STRUCTURE OF PROGRAM FLAGS */
A_ACCEL acclg[];         /* DATA STRUCTURE OF ACCELEROGAM PARAMS */
A_LOADHIST lhist[];      /* DATA STRUCTURE OF LOAD HISTORY PARAMS */
A_JT_JOINT jt[];         /* DATA STRUCTURE FOR SINGLE JOINT */
A_CFT_BC cft[];          /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
A_STL_BC stl[];          /* DATA STRUCTURE FOR STEEL BEAM-COLUMN */
A_GLOBAL global[];       /* DATA STRUCTURE OF DOF-BASED VARIABLES */

{
char *calloc();           /* SYSTEM MEMORY ALLOCATOR */
long status = 1L;        /* RETURN STATUS */
long ctr = 0L;           /* COUNTER */
long n = 0L;             /* ELEMENT COUNTER */

```

```

double t      = 0.0;          /* CURRENT TIME */
double displ_y[ 4L ];       /* STORY YIELD DISPLACEMENTS */
double *soln  = ( double * )0; /* VEC PASSED TO AND FROM SOLVER */

/* USER SCREEN INFO */
printf( "\n\n**** Beginning Static Analysis ****" );

/*****
/* ALLOCATE MEMORY FOR SOLUTION VECTOR; INITIALIZE */
*****/

soln      = ( double * ) calloc ( ( unsigned ) ( size->num_dofs + 1L ),
                                ( unsigned ) ( sizeof( *soln ) ) );

for ( ctr = 0L; ctr < size->num_dofs + 1L; ctr++ )
    {
        soln[ ctr ] = 0.0;
    }
for ( ctr = 0L; ctr < 4L; ctr++ )
    {
        displ_y[ ctr ] = 0.0;
    }

/*****
/* CALC LENGTHS, DIR. COSINES, AND ROTATION MATRICES */
*****/

/** CFT ELEMENTS **/
for ( n = 1L; ( n <= size->num_cft_elems ) && ( status ); n++ )
    {
        if ( ! a_cft_update( n, jt, cft, global ) )
            {
                status = 0L;
                printf( "\nCFT element calculations failed.\n" );
            }
    }

/** STEEL ELEMENTS **/
for ( n = 1L; ( n <= size->num_stl_elems ) && ( status ); n++ )
    {
        if ( ! a_stl_update( n, jt, stl, global ) )
            {
                status = 0L;
                printf( "\nSteel element calculations failed.\n" );
            }
    }

/*****
/* ADD ELEMENT LOADS TO JOINTS */
*****/

/** CFT ELEMENTS **/
for ( n = 1L; ( n <= size->num_cft_elems ) && ( status ); n++ )
    {
        if ( ! a_cft_distr( n, flag, jt, cft ) )
            {
                status = 0L;
                printf( "\nCFT element load calcs failed.\n" );
            }
    }

/** STEEL ELEMENTS **/
for ( n = 1L; ( n <= size->num_stl_elems ) && ( status ); n++ )
    {
        if ( ! a_stl_distr( n, flag, jt, stl ) )
            {
                status = 0L;
                printf( "\nSteel element load calcs failed.\n" );
            }
    }

/*****
/* PRINT STRUCTURAL INFORMATION */
*****/

```

```

if ( ! a_io_output( t, displ_y, size, time, nr, flag, acclg, jt, cft, stl,
    global ) && ( status ) )
    {
    status = 0L;
    printf( "\nOutput of structural information failed. \n" );
    }

/*****
/*****          BEGIN STATIC ANALYSIS          *****/
/*****
/*****

t = time->step;

while ( ( t <= ( time->total + .01 * time->step ) ) && ( status ) )
    {
    printf( "\n\n\n***** Time = %6lf *****", t );
    printf( "\n*****\n" );

    nr->conv = 0L;

    /*****
    /* SET BEGINNING OF STEP VALUES */
    /*****

    if ( ! a_nr_init_step( t, size, flag, acclg, lhist, jt, cft, stl,
        global ) && ( status ) )
        {
        status = 0L;
        printf( "\nFailure in step initialization routine.\n" );
        }

    /*****
    /*****          NEWTON-RAPHSON ITERATION          *****/
    /*****

    for ( nr->iter_ct = 1L; ( nr->iter_ct <= nr->iter_max ) &&
        ( ! nr->conv ) && ( status ); nr->iter_ct++ )
        {
        if ( nr->iter_max > 1L )
            {
            printf( "\n*****Iteration %ld*****\n", nr->iter_ct );
            }
        }

/* STIFFNESS CALCULATION */

    /*****
    /* CALCULATE ELEMENT STIFFNESS MATRICES */
    /*****

    /*** CFT ELEMENTS ***/
    for ( n = 1L; ( n <= size->num_cft_elems ) && ( status ); n++ )
        {
        if ( ! a_cft_stiffness( n, size, flag, jt, cft ) )
            {
            status = 0L;
            printf( "\nMember stiffness calcs failed.\n" );
            }
        }

    /*** STEEL ELEMENTS ***/
    for ( n = 1L; ( n <= size->num_stl_elems ) && ( status ); n++ )
        {
        if ( ! a_stl_stiffness( n, size, flag, jt, stl ) )
            {
            status = 0L;
            printf( "\nMember stiffness calcs failed.\n" );
            }
        }

    /*****
    /* ASSEMBLE GLOBAL STIFFNESS MATRIX, kt */
    /*****

```

```

if ( ! a_k_assemble( diag, kt, size, cft, stl ) && ( status ) )
{
    status = 0L;
    printf( "\nAssembly of global matrix failed.\n" );
}

/* GLOBAL MATRICES */

/*****
/* FACTOR GLOBAL STIFFNESS MATRIX */
*****/

if ( status )
{
    printf( "\n** Factoring Global Stiffness Matrix **\n" );

    if ( ! a_eq_factor( diag, kt, size ) )
    {
        status = 0L;
        printf( "\nFactorization of K matrix failed.\n" );
    }
}

/* LOADS */

/*****
/* ASSEMBLE THE GLOBAL INCREMENTAL LOAD VECTOR */
*****/

if ( ! a_load_static( size, nr, lhist, jt, global ) )
{
    status = 0L;
    printf( "Assembly of global load vector failed. \n" );
}

/*****
/* COMPUTE DISTRIBUTED LOAD MULTIPLIERS */
*****/

else if ( ! a_load_distr( size, nr, jt, cft, stl, global ) )
{
    status = 0L;
    printf( "\nCalc of distributed load multipliers failed. \n" );
}

/* SOLVE */

/*****
/* PASS {dr} TO SOLVER; SOLVE THE SYSTEM OF EQUATIONS; GET {dq} */
*****/

for ( ctr = 1L; ( ctr <= size->num_dofs ) && ( status ); ctr++ )
{
    soln[ ctr ] = global[ ctr ].dr;
}

printf( "\n** Solving for Displacements ** \n" );

if ( ! a_eq_solve( diag, kt, soln, size ) )
{
    status = 0L;
    printf( "\nSolution for displacements failed.\n" );
}

for ( ctr = 1L; ( ctr <= size->num_dofs ) && ( status ); ctr++ )
{
    global[ ctr ].dqi = soln[ ctr ];
}

/* UPDATE */

/*****
/* UPDATE NODAL COORDINATES AND DISPLACEMENTS */
*****/

```

```

/*****/

if ( ! a_jt_update( size, nr, flag, jt, global ) )
{
    status = 0L;
    printf( "\nUpdate of joints failed.\n" );
}

/* RECOVER FORCES */

/*****/
/* RECOVER LOCAL MEMBER FORCES */
/*****/

if ( status )
{
    printf( "\n** Recovering Forces **\n" );

    /*** CFT ELEMENTS ***/
    if ( ! a_cft_frecovery( size, flag, jt, cft, global ) )
    {
        status = 0L;
        printf( "\nCFT force recovery failed.\n" );
    }

    /*** STEEL ELEMENTS ***/
    else if ( ! a_stl_frecovery( size, flag, jt, stl, global ) )
    {
        status = 0L;
        printf( "\nSteel force recovery failed.\n" );
    }
}

/*****/
/* UPDATE MEMBERS; ROTATE FORCES */
/*****/

if ( status )
{
    /*** CFT ELEMENTS ***/
    if ( ! a_cft_rotate( size, nr, flag, lhist, jt, cft, global ) )
    {
        /* a_cft_update CALLED INSIDE THIS FUNCTION */
        status = 0L;
        printf( "\nCFT member rotation failed.\n" );
    }

    /*** STEEL ELEMENTS ***/
    else if ( ! a_stl_rotate( size, nr, flag, lhist, jt, stl,
                             global ) )
    {
        /* a_stl_update CALLED INSIDE THIS FUNCTION */
        status = 0L;
        printf( "\nSteel member rotation failed.\n" );
    }
}

/* PLASTICITY */

/*****/
/* DETERMINE THE PLASTICITY STATE AT EACH MEMBER END; UPDATE SURFACES */
/*****/

if ( ( flag->kp ) && ( status ) )
{
    printf( "\n** Plasticity Analysis **\n" );

    /*** CFT ELEMENTS ***/
    if ( ! a_pl_cft_state( size, nr, tol, flag, cft, global ) )
    {
        status = 0L;
        printf( "\nCFT hinge check failed.\n" );
    }

    /*** STEEL ELEMENTS ***/
    else if ( ! a_pl_stl_state( size, nr, tol, flag, stl, global ) )

```

```

        {
            status = 0L;
            printf( "\nSteel hinge check failed.\n");
        }
    }

/* CONVERGENCE CHECK */

    /*****
    /* CHECK FOR ITERATION CONVERGENCE */
    *****/

    if ( status )
        {
            if ( ! a_nr_conv( size, nr, tol, flag, jt, global ) )
                {
                    status = 0L;
                    printf( "\nConvergence check failed.\n" );
                }
        }

    /* for ( nr->iter_ct = 1L; ...; nr->iter_ct++ ) */

    /*****      END NEWTON-RAPHSON ITERATION LOOP      *****/

/* PRINT RESULTS */

    /*****
    /* OUTPUT TIME STEP RESPONSE */
    *****/

    if ( status )
        {
            printf( "\n** Writing Output of Step **\n" );

            if ( ! a_io_output( t, displ_y, size, time, nr, flag, acclg,
                               jt, cft, stl, global ) )
                {
                    status = 0L;
                    printf( "\nOutput of time step failed.\n" );
                }
        }

    /*****
    /* UPDATE MAX AND MIN DISPLS AND FORCES (FOR OUTPUT PURPOSES) */
    *****/

    if ( flag->maxmin )
        {
            if ( ! status )
                {
                    /* STRUCTURE HAS FAILED OR AN ERROR OCCURRED */
                    t = time->total; /* This assures values are printed */
                }

            if ( ! a_io_maxmin( t, size, time, flag, jt, cft, stl, global ) )
                {
                    status = 0L;
                    printf( "\nCalculation of max/min values failed. \n" );
                }
        }

    /*****
    /* INCREMENT TIME; EMPTY INTERNAL BUFFER */
    *****/

    t += time->step;

    fflush( stdout );

    } /* while ( t <= time->total ) */

    /*****      END TIME STEP LOOP      *****/

```

```

/*****
/* FREE DYNAMICALLY-ALLOCATED MEMORY */
*****/

cfree( soln );

return( status );      /* RETURN TO cftmacro.c */
}

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****

* FUNCTION- a_el_calcs

      @(#) miscellaneous ELeMent CALCulationS

* CALLED FROM- cftmacro

* ABSTRACT- This function performs the calculation of variables that are
      independent of the time or load, including plasticity
      variables, and effective CFT stiffness parameters.
*****/

#include <math.h>          /* C MATH FUNCTIONS          */
#include <stdio.h>         /* C I/O FUNCTIONS          */

#include "a_mac.h"         /* MACRO DEFINITIONS        */
#include "a_extern.h"     /* EXTERNAL VARIABLES & PROTOTYPES */
#include "a_cft_el.h"     /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
#include "a_stl_el.h"     /* DATA STRUCTURE FOR STEEL BEAM-COLUMN */
#include "a_param.h"      /* DATA STRUCTURE OF VARIOUS PARAMETERS */

long a_el_calcs( size, flag, cft, stl )

A_MODEL_SIZE size;      /* DATA STRUCTURE OF GLOBAL STRUCT SIZE */
A_FLAGS flag;          /* DATA STRUCTURE OF PROGRAM FLAGS */
A_CFT_BC cft[];       /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
A_STL_BC stl[];       /* DATA STRUCTURE FOR STEEL BEAM-COLUMN */

{
long status = 1L;      /* RETURN STATUS */
long n = 0L;          /* ELEMENT COUNTER */
double na_z = 0.0;    /* MAJOR NEUTRAL AXIS OF CROSS SECTION */
double na_y = 0.0;    /* MINOR NEUTRAL AXIS OF CROSS SECTION */
double dt = 0.0;      /* CFT MAJOR AXIS D/t RATIO */
double bt = 0.0;      /* CFT MINOR AXIS D/t RATIO */
double fcy = 0.0;     /* RATIO OF fc TO fy */
double d_c = 0.0;     /* DEPTH OF CONCRETE */
double b_c = 0.0;     /* WIDTH OF CONCRETE */
double dt_c1 = 0.0;   /* COEFFICIENT c1 USING MAJOR AXIS D/t */
double bt_c1 = 0.0;   /* COEFFICIENT c1 USING MINOR AXIS D/t */
double dt_c2 = 0.0;   /* COEFFICIENT c2 USING MAJOR AXIS D/t */
double bt_c2 = 0.0;   /* COEFFICIENT c2 USING MINOR AXIS D/t */
double dt_c3 = 0.0;   /* COEFFICIENT c3 USING MAJOR AXIS D/t */
double bt_c3 = 0.0;   /* COEFFICIENT c3 USING MINOR AXIS D/t */
double dt_c4 = 0.0;   /* COEFFICIENT c4 USING MAJOR AXIS D/t */
double bt_c4 = 0.0;   /* COEFFICIENT c4 USING MINOR AXIS D/t */

/*****
***** STEEL ELEMENTS *****
*****/

for ( n = 1L; n <= size.num_stl_elems; n++ )
{
/*****
/* COMPUTE POLAR MOMENT OF INERTIA */
*****/

```

```

stl[ n ].i_p = sqrt( pow( stl[ n ].i_y, 2 ) + pow( stl[ n ].i_z, 2 ) );

/*****
/* COMPUTE NOMINAL STRENGTHS */
*****/

stl[ n ].po    = stl[ n ].area * stl[ n ].fy;
stl[ n ].myo   = stl[ n ].z_y * stl[ n ].fy;
stl[ n ].mzo   = stl[ n ].z_z * stl[ n ].fy;
stl[ n ].myy   = stl[ n ].s_y * stl[ n ].fy;
stl[ n ].mzy   = stl[ n ].s_z * stl[ n ].fy;
}

/*****
**** CFT ELEMENTS ****
*****/

for ( n = 1L; n <= size.num_cft_elems; n++ )
{
/*****
/* CALCULATE AREAS, MOMENTS OF INERTIA, AND SECTION MODULI */
*****/

d_c      = cft[ n ].d - 2.0 * cft[ n ].t;      /*CONCR DEPTH*/
b_c      = cft[ n ].b - 2.0 * cft[ n ].t;      /*CONCR WIDTH*/
cft[ n ].a_conc = d_c * b_c;
cft[ n ].a_stl = cft[ n ].d * cft[ n ].b - cft[ n ].a_conc;
cft[ n ].iy_c  = pow( b_c, 3 ) * d_c / 12.0;
cft[ n ].iz_c  = pow( d_c, 3 ) * b_c / 12.0;

if ( ! cft[ n ].section )      /* USER INPUT SECTION */
{
cft[ n ].iz_s = ( cft[ n ].b * pow( cft[ n ].d, 3 ) -
b_c * pow( d_c, 3 ) ) / 12.0;
cft[ n ].iy_s = ( cft[ n ].d * pow( cft[ n ].b, 3 ) -
d_c * pow( b_c, 3 ) ) / 12.0;
cft[ n ].s_z  = cft[ n ].iz_s / ( cft[ n ].d / 2.0 );
cft[ n ].s_y  = cft[ n ].iy_s / ( cft[ n ].b / 2.0 );
cft[ n ].ix   = 4.0 * cft[ n ].t * pow( ( cft[ n ].b -
cft[ n ].t ) * ( cft[ n ].d - cft[ n ].t ), 2 ) /
( 2.0 * ( cft[ n ].b - cft[ n ].t ) +
2.0 * ( cft[ n ].d - cft[ n ].t ) );
cft[ n ].kpi_s = cft[ n ].a_stl * 0.284;
}

cft[ n ].ip    = sqrt( pow( cft[ n ].iy_s, 2 ) + pow( cft[ n ].iz_s, 2 ) );
cft[ n ].gmod_c = cft[ n ].emod_c / ( 2.0 * ( 1 + cft[ n ].nu_c ) );

/*****
/* COMPUTE EFFECTIVE CFT STIFFNESS PARAMETERS */
*****/

cft[ n ].ec_in = cft[ n ].emod_c;
cft[ n ].eiy  = cft[ n ].emod_s * cft[ n ].iy_s +
( cft[ n ].emod_c * cft[ n ].iy_c );
cft[ n ].eiz  = cft[ n ].emod_s * cft[ n ].iz_s +
( cft[ n ].emod_c * cft[ n ].iz_c );

cft[ n ].ea   = cft[ n ].emod_s * cft[ n ].a_stl +
cft[ n ].emod_c * cft[ n ].a_conc;
cft[ n ].gj   = cft[ n ].gmod_s * cft[ n ].ix;

cft[ n ].kpi_c = cft[ n ].wt_c * cft[ n ].a_conc;
cft[ n ].kpi   = cft[ n ].kpi_s + cft[ n ].kpi_c;

/*****
/* COMPUTE NOMINAL STRENGTHS */
*****/

cft[ n ].fct   = 7.5 * sqrt( 1000 * cft[ n ].fc ) / 1000;
cft[ n ].po    = ( cft[ n ].a_stl * cft[ n ].fy ) +
( cft[ n ].a_conc * cft[ n ].fc );
cft[ n ].ptens = ( cft[ n ].a_stl * cft[ n ].fy ) +
( cft[ n ].a_conc * cft[ n ].fct );

```

```

cft[ n ].phi = ( cft[ n ].po - cft[ n ].ptens ) /
              ( 2.0 * cft[ n ].po );
na_z = ( 0.85 * cft[ n ].fc * ( cft[ n ].b * cft[ n ].t - 2.0 *
                                cft[ n ].t * cft[ n ].t ) +
          0.5 * cft[ n ].fct * ( ( cft[ n ].d - cft[ n ].t ) *
                                ( cft[ n ].b - 2.0 * cft[ n ].t ) ) +
          cft[ n ].fy * ( 2.0 * cft[ n ].d * cft[ n ].t ) ) /
( 0.85 * cft[ n ].fc * ( cft[ n ].b - 2.0 * cft[ n ].t ) +
  0.5 * cft[ n ].fct * ( cft[ n ].b - 2.0 * cft[ n ].t ) +
  cft[ n ].fy * ( 4.0 * cft[ n ].t ) );
na_y = ( 0.85 * cft[ n ].fc * ( cft[ n ].d * cft[ n ].t - 2.0 *
                                cft[ n ].t * cft[ n ].t ) +
          0.5 * cft[ n ].fct * ( ( cft[ n ].b - cft[ n ].t ) *
                                ( cft[ n ].d - 2.0 * cft[ n ].t ) ) +
          cft[ n ].fy * ( 2.0 * cft[ n ].b * cft[ n ].t ) ) /
( 0.85 * cft[ n ].fc * ( cft[ n ].d - 2.0 * cft[ n ].t ) +
  0.5 * cft[ n ].fct * ( cft[ n ].d - 2.0 * cft[ n ].t ) +
  cft[ n ].fy * ( 4.0 * cft[ n ].t ) );
cft[ n ].mzo = 0.85 * cft[ n ].fc * ( cft[ n ].b - 2.0 * cft[ n ].t ) *
0.5 * pow( ( na_z - cft[ n ].t ), 2 ) + 0.5 * cft[ n ].fct *
( cft[ n ].b - 2.0 * cft[ n ].t ) * 0.5 * pow( ( cft[ n ].d -
na_z - cft[ n ].t ), 2 ) + cft[ n ].fy * ( 2.0 * cft[ n ].t *
( 0.5 * pow( cft[ n ].d, 2 ) - cft[ n ].d * cft[ n ].t +
pow( na_z, 2 ) + pow( cft[ n ].t, 2 ) - cft[ n ].d * na_z ) +
( cft[ n ].b * cft[ n ].t ) * ( cft[ n ].d - cft[ n ].t ) );
cft[ n ].myo = 0.85 * cft[ n ].fc * ( cft[ n ].d - 2.0 * cft[ n ].t ) *
0.5 * pow( ( na_y - cft[ n ].t ), 2 ) + 0.5 * cft[ n ].fct *
( cft[ n ].d - 2.0 * cft[ n ].t ) * 0.5 * pow( ( cft[ n ].b -
na_y - cft[ n ].t ), 2 ) + cft[ n ].fy * ( 2.0 * cft[ n ].t *
( 0.5 * pow( cft[ n ].b, 2 ) - cft[ n ].b * cft[ n ].t +
pow( na_y, 2 ) + pow( cft[ n ].t, 2 ) - cft[ n ].b * na_y ) +
( cft[ n ].d * cft[ n ].t ) * ( cft[ n ].b - cft[ n ].t ) );
cft[ n ].myy = cft[ n ].s_y * cft[ n ].fy;
cft[ n ].myy = cft[ n ].s_y * cft[ n ].fy;
cft[ n ].mzy = cft[ n ].s_z * cft[ n ].fy;

```

```

/*****
/* COMPUTE COEFFICIENTS FOR MAJOR AND MINOR AXIS D/t RATIOS */
*****/

```

```

dt = cft[ n ].d / cft[ n ].t;
bt = cft[ n ].b / cft[ n ].t;
fcy = cft[ n ].fc / cft[ n ].fy;
dt_c1 = 1.077 - 0.002646 * dt + 0.00002304 * pow( dt, 2 ) - 1.128e-7 *
pow( dt, 3 ) + 0.3745 * fcy - 1.299 * pow( fcy, 2 ) - 0.04193 *
pow( fcy, 3 ) - 0.06913 * dt * fcy + 0.0002339 *
pow( dt, 2 ) * fcy + 0.07542 * dt * pow( fcy, 2 );
bt_c1 = 1.077 - 0.002646 * bt + 0.00002304 * pow( bt, 2 ) - 1.128e-7 *
pow( bt, 3 ) + 0.3745 * fcy - 1.299 * pow( fcy, 2 ) - 0.04193 *
pow( fcy, 3 ) - 0.06913 * bt * fcy + 0.0002339 *
pow( bt, 2 ) * fcy + 0.07542 * bt * pow( fcy, 2 );
dt_c2 = 0.6277 + 0.0259 * dt - 0.0003673 * pow( dt, 2 ) + 1.989e-6 *
pow( dt, 3 ) + 4.496 * fcy - 14.89 * pow( fcy, 2 ) + 22.44 *
pow( fcy, 3 ) + 0.1644 * dt * fcy - 0.0007564 * pow( dt, 2 ) *
fcy - 0.1263 * dt * pow( fcy, 2 );
bt_c2 = 0.6277 + 0.0259 * bt - 0.0003673 * pow( bt, 2 ) + 1.989e-6 *
pow( bt, 3 ) + 4.496 * fcy - 14.89 * pow( fcy, 2 ) + 22.44 *
pow( fcy, 3 ) + 0.1644 * bt * fcy - 0.0007564 * pow( bt, 2 ) *
fcy - 0.1263 * bt * pow( fcy, 2 );
dt_c3 = 0.4204 + 0.08921 * dt - 0.001216 * pow( dt, 2 ) + 0.000005128 *
pow( dt, 3 ) + 4.897 * fcy - 16.51 * pow( fcy, 2 ) + 16.22 *
pow( fcy, 3 ) - 0.1645 * dt * fcy + 0.0007135 * pow( dt, 2 ) *
fcy + 0.1199 * dt * pow( fcy, 2 );
bt_c3 = 0.4204 + 0.08921 * bt - 0.001216 * pow( bt, 2 ) + 0.000005128 *
pow( bt, 3 ) + 4.897 * fcy - 16.51 * pow( fcy, 2 ) + 16.22 *
pow( fcy, 3 ) - 0.1645 * bt * fcy + 0.0007135 * pow( bt, 2 ) *
fcy + 0.1199 * bt * pow( fcy, 2 );
dt_c4 = 0.3456 + 0.009121 * dt - 0.000127 * pow( dt, 2 ) + 4.979e-7 *
pow( dt, 3 ) - 0.3912 * fcy + 4.545 * pow( fcy, 2 ) - 10.3 *
pow( fcy, 3 ) - 0.05924 * dt * fcy + 0.0002449 * pow( dt, 2 ) *
fcy + 0.06592 * dt * pow( fcy, 2 );
bt_c4 = 0.3456 + 0.009121 * bt - 0.000127 * pow( bt, 2 ) + 4.979e-7 *
pow( bt, 3 ) - 0.3912 * fcy + 4.545 * pow( fcy, 2 ) - 10.3 *
pow( fcy, 3 ) - 0.05924 * bt * fcy + 0.0002449 * pow( bt, 2 ) *
fcy + 0.06592 * bt * pow( fcy, 2 );

```

```

        * fcy + 0.06592 * bt * pow( fcy, 2 );

/*****
/* COMPUTE AVERAGE COEFFICIENT VALUES */
*****/

cft[ n ].c1   = ( dt_c1 + bt_c1 ) / 2.0;
cft[ n ].c2   = ( dt_c2 + bt_c2 ) / 2.0;
cft[ n ].c3   = ( dt_c3 + bt_c3 ) / 2.0;
cft[ n ].c4   = ( dt_c4 + bt_c4 ) / 2.0;

}          /* for ( n = 1L; n <= size.num_cft_elems; n++ ) */

return( status );
}

```

```

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****

* FUNCTION- a_jt_update

        @(#) JOINT parameters UPDATE

* CALLED FROM- a_drv_dynamic, a_drv_static

* ABSTRACT- Calculate new nodal coordinates, displacements, velocities,
and accelerations based upon the calculated global
displacements.
*****/

#include <math.h>          /* C MATH FUNCTIONS          */
#include <stdio.h>         /* C I/O FUNCTIONS          */

#include "a_mac.h"         /* MACRO DEFINITIONS        */
#include "a_extern.h"     /* EXTERNAL VARIABLES & PROTOTYPES */
#include "a_jt.h"         /* DATA STRUCTURE DEFINING JOINT PROPS */
#include "a_param.h"      /* DATA STRUCTURE OF VARIOUS PARAMETERS */
#include "a_global.h"     /* DATA STRUCTURE OF DOF-BASED VARIABLES */

long a_jt_update( size, nr, flag, jt, global )

A_MODEL_SIZE *size;      /* DATA STRUCTURE OF GLOBAL STRUCTURE SIZE*/
A_NEW_RAPH *nr;         /* DATA STRUCTURE OF NEWTON-RAPHSON PARAMS*/
A_FLAGS *flag;         /* DATA STRUCTURE OF PROGRAM FLAGS */
A_JT_JOINT jt[];       /* DATA STRUCTURE FOR SINGLE JOINT */
A_GLOBAL global[];     /* DATA STRUCTURE OF DOF-BASED VARIABLES */

{
long status = 1L;      /* RETURN STATUS */
long joint = 0L;      /* JOINT COUNTER */
long k1 = 0L;         /* VARIABLE TO STORE X-TRANSLATION DOF */
long k2 = 0L;         /* VARIABLE TO STORE Y-TRANSLATION DOF */
long k3 = 0L;         /* VARIABLE TO STORE Z-TRANSLATION DOF */
long dof = 0L;       /* DEGREE-OF-FREEDOM COUNTER */

/*****/
/* CALCULATE NEW JOINT COORDINATES */
/*****/

if ( flag->kg )      /* UPDATE COORDS ONLY IF PERFRMING GEOM NONL ANALYSIS */
{
for ( joint = 1L; joint <= size->num_jts; joint++ )
{
k1 = jt[ joint ].jcode[ 1 ];
if ( k1 != 0L )
{
jt[ joint ].co_i.x = jt[ joint ].co.x;
jt[ joint ].co.x += global[ k1 ].dqi;
}
k2 = jt[ joint ].jcode[ 2 ];
if ( k2 != 0L )
{
jt[ joint ].co_i.y = jt[ joint ].co.y;
jt[ joint ].co.y += global[ k2 ].dqi;
}
k3 = jt[ joint ].jcode[ 3 ];
if ( k3 != 0L )
{
jt[ joint ].co_i.z = jt[ joint ].co.z;
jt[ joint ].co.z += global[ k3 ].dqi;
}
}
}

return( status );
}

```

```

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****
*/

* FUNCTION- a_load_static

      @(#) assembly of the LOAD vector -- STATIC portion

* CALLED FROM- a_drv_dynamic, a_drv_static

* ABSTRACT- This function obtains the amount of applied load as a fraction
      of the total load. This is obtained by determining the
      current position on the load history curve and computing the
      load fraction based upon the current time.
*****/

#include <math.h>          /* C MATH FUNCTIONS          */
#include <stdio.h>         /* C I/O FUNCTIONS         */
#include <string.h>       /* C STRING FUNCTIONS      */

#include "a_mac.h"        /* MACRO DEFINITIONS       */
#include "a_extern.h"     /* EXTERNAL VARIABLES & PROTOTYPES */
#include "a_jt.h"        /* DATA STRUCTURE DEFINING JOINT PROPS */
#include "a_param.h"     /* DATA STRUCTURE OF VARIOUS PARAMETERS */
#include "a_global.h"    /* DATA STRUCTURE OF DOF-BASED VARIABLES */

long a_load_static( size, nr, lhist, jt, global )

A_MODEL_SIZE *size;      /* DATA STRUCTURE OF GLOBAL STRUCT SIZE */
A_NEW_RAPH *nr;         /* DATA STRUCTURE OF NEWTON-RAPHSON DATA */
A_LOADHIST lhist[];    /* DATA STRUCTURE OF LOAD HISTORY PARAMS */
A_JT_JOINT jt[];      /* DATA STRUCTURE FOR SINGLE JOINT */
A_GLOBAL global[];     /* DATA STRUCTURE OF DOF-BASED VARIABLES */

{
long status = 1L;      /* RETURN STATUS */
long jnt = 0L;        /* JOINT COUNTER */
long dof = 0L;        /* JOINT DOF COUNTER */
long lhnum = 0L;      /* LOAD HISTORY NUMBER */
long k = 0L;          /* CURRENT GLOBAL DEGREE-OF-FREEDOM */

/* USER SCREEN INFO */
printf( "\n** Assembling Load Vector **\n" );

/*****
/* CALCULATE THE ITERATIVE GLOBAL LOAD VECTOR, global.dr */
*****/

for ( jnt = 1L; jnt <= size->num_jts; jnt++ )
{
for ( dof = 1L; dof <= A_JT_DOF; dof++ )
{
if ( jt[ jnt ].jcode[ dof ] != 0L )
{
k = jt[ jnt ].jcode[ dof ];
lhnum = jt[ jnt ].lhist[ dof ];

/*****
/* COMPUTE TOTAL GLOBAL LOAD VECTOR--1ST ITERATION */
*****/

if ( nr->iter_ct == 1L )
{
global[ k ].r2 = lhist[ lhnum ].ldfrac2 * jt[ jnt ].load[dof] +
lhnum[ 1L ].ldfrac2 * jt[ jnt ].load1[ dof ] +
lhnum[ 2L ].ldfrac2 * jt[ jnt ].load2[ dof ];
jt[ jnt ].r2[ dof ] = global[ k ].r2;
}

/*****

```

```

/* COMPUTE ITERATIVE GLOBAL LOADS */
/*****

global[ k ].dr = global[ k ].r2 + global[ k ].dyn
                + ( jt[ jnt ].ma[ dof ] )
                - ( jt[ jnt ].fl[ dof ] + jt[ jnt ].f2_i[ dof ] );

/* INITIALIZE INCREMENTAL JOINT FORCE--MUST BE 0 ENTERING rotate */

jt[ jnt ].df_i[ dof ] = 0.0;

    } /* if ( jt[ jnt ].jcode[ dof ] != 0L ) */
} /* for ( dof = 1L; dof <= A_JT_DOF; dof++ ) */
} /* for ( jnt = 1L; jnt <= size->num_jts; jnt++ ) */

return( status );
}

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****

* FUNCTION- a_nr_init_step

    @(#) Newton-Raphson iteration--INITialization of first STEP

* CALLED FROM- a_drv_dynamic, a_drv_static

* ABSTRACT- This function sets the beginning-of-step forces and
            displacements for the current time (load) step.
*****

#include <math.h> /* C MATH FUNCTIONS */
#include <stdio.h> /* C I/O FUNCTIONS */
#include <string.h> /* C STRING FUNCTIONS */

#include "a_mac.h" /* MACRO DEFINITIONS */
#include "a_extern.h" /* EXTERNAL VARIABLES */
#include "sstmatch.h" /* TOLERANCE COMPARISONS */
#include "a_param.h" /* DATA STRUCTURE OF VARIOUS PARAMETERS */
#include "a_cft_el.h" /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
#include "a_stl_el.h" /* DATA STRUCTURE FOR STEEL BEAM-COLUMN */
#include "a_global.h" /* DATA STRUCTURE OF DOF-BASED VARIABLES */

long a_nr_init_step( t, size, flag, acclg, lhist, jt, cft, stl, global )

double t; /* CURRENT TIME */
A_MODEL_SIZE *size; /* DATA STRUCTURE OF GLOBAL STRUC. SIZE */
A_FLAGS *flag; /* DATA STRUCTURE OF PROGRAM FLAGS */
A_ACCEL acclg[]; /* DATA STRUCTURE OF ACCELEROGAM PARAMS */
A_LOADHIST lhist[]; /* DATA STRUCTURE OF LOAD HISTORY POINTS */
A_JT_JOINT jt[]; /* DATA STRUCTURE FOR SINGLE JOINT */
A_CFT_BC cft[]; /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
A_STL_BC stl[]; /* DATA STRUCTURE FOR STEEL BEAM-COLUMN */
A_GLOBAL global[]; /* DATA STRUCTURE OF DOF-BASED VARIABLES */

{
long status = 1L; /* RETURN STATUS */
long i = 0L; /* ELEMENT END INDEX */
long j = 0L; /* ELEMENT END INDEX */
long n = 0L; /* ELEMENT COUNTER */
long joint = 0L; /* JOINT COUNTER */
long dof = 0L; /* DOF COUNTER */
long lhnum = 0L; /* LOAD HISTORY NUMBER */
long acclnum = 0L; /* ACCELEROGAM NUMBER */
long coord = 0L; /* VARIABLE DETERMINING CURRENT SEGMENT */
long curve = 0L; /* VARIABLE DETERMINING CURRENT SEGMENT */

```

```

/*****
/* SET BEG-OF-STEP CFT FORCE VALUES */
*****/

for ( n = 1L; n <= size->num_cft_elems; n++ )
  {
    for ( dof = 1L; dof <= A_CFT_NUM_DOF; dof++ )
      {
        cft[ n ].df_i[ dof ] = 0.0;
        cft[ n ].f1_i[ dof ] = 0.0;
        cft[ n ].f2_i[ dof ] = 0.0;
        cft[ n ].f1[ dof ] = cft[ n ].f2[ dof ];
      }
  }

/*****
/* SET BEG-OF-STEP STEEL FORCE VALUES */
*****/

for ( n = 1L; n <= size->num_stl_elems; n++ )
  {
    for ( dof = 1L; dof <= A_STL_NUM_DOF; dof++ )
      {
        stl[ n ].df_i[ dof ] = 0.0;
        stl[ n ].f1_i[ dof ] = 0.0;
        stl[ n ].f2_i[ dof ] = 0.0;
        stl[ n ].f1[ dof ] = stl[ n ].f2[ dof ];
      }
  }

/*****
/* SET BEG-OF-STEP JOINT FORCES */
*****/

for ( joint = 1L; joint <= size->num_jts; joint++ )
  {
    for ( dof = 1L; dof <= A_JT_DOF; dof++ )
      {
        jt[ joint ].df_i[ dof ] = 0.0;
        jt[ joint ].f2_i[ dof ] = 0.0;
        jt[ joint ].f1[ dof ] = jt[ joint ].f2[ dof ];
      }
  }

/*****
/* UPDATE GLOBAL PARAMETERS */
*****/

for ( dof = 1L; dof <= size->num_dofs; dof++ )
  {
    global[ dof ].dr = 0.0;
    global[ dof ].r1 = global[ dof ].r2;
    global[ dof ].q2i = 0.0;
    global[ dof ].q1 = global[ dof ].q2;
    global[ dof ].v1 = global[ dof ].v2;
    global[ dof ].v1_i = global[ dof ].v2;
    global[ dof ].a1 = global[ dof ].a2;
    global[ dof ].a1_i = global[ dof ].a2;
  }

/*****
/* FIND LOCATIONS ON LOAD CURVES AND ACCELEROGRAMS FOR CURRENT TIME STEP */
*****/

/* GET LOCATION ON LOAD HISTORY CURVES */
for ( lhnum = 1L; lhnum <= size->num_lhs; lhnum++ )
  {
    for ( coord = 1L; coord <= lhist[ lhnum ].num_lhpts; coord++ )
      {
        if ( t <= lhist[ lhnum ].time_pt[ coord ] + 0.0000001 )
          {

```

```

        lhist[ lnum ].curve = coord;
        break;
    }
}

/*****
/* CALCULATE THE LOAD FRACTION FOR EACH LOAD HISTORY CURVE */
*****/

for ( lnum = 1L; ( lnum <= size->num_lhs ) && status; lnum++ )
{
    curve = lhist[ lnum ].curve;
    lhist[ lnum ].ldfrac1 = lhist[ lnum ].ldfrac2;

    if ( ! SS_TOL_SAME( ( lhist[ lnum ].time_pt[ curve ] -
                        lhist[ lnum ].time_pt[ curve - 1L ] ), 0.0 ) )
    {
        lhist[ lnum ].ldfrac2 = lhist[ lnum ].load_pt[ curve - 1L ] +
            ( t - lhist[ lnum ].time_pt[ curve - 1L ] ) *
            ( lhist[ lnum ].load_pt[ curve ] -
              lhist[ lnum ].load_pt[ curve - 1L ] ) /
            ( lhist[ lnum ].time_pt[ curve ] -
              lhist[ lnum ].time_pt[ curve - 1L ] );
    }
    else
    {
        status = 0L;
        printf( "\n\nDivide by zero error for load history %ld", lnum);
        printf( "\n\nCheck input under *A_LOADHIST for errors.\n\n" );
    }
}

return( status );
}

```

```

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****

```

```

* FUNCTION- a_pl_cft_dist.c

```

```

    @(#) for CFT element ends: calc. DISTANCE between surfaces

```

```

* CALLED FROM- a_pl_cft_state

```

```

* ABSTRACT- This function calculates the distance between the loading
            surface and the bounding surface by using the Mroz vector.
            If the end became plastic for the first time this step, din is
            calculated; otherwise d is calculated.
*****

```

```

#include <math.h>           /* C MATH FUNCTIONS           */
#include <stdio.h>          /* C I/O FUNCTIONS           */
#include <string.h>         /* C STRING FUNCTIONS        */

#include "a_mac.h"          /* MACRO DEFINITIONS         */
#include "a_extern.h"       /* EXTERNAL VARIABLES        */
#include "a_cft_el.h"       /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
#include "a_param.h"        /* DATA STRUCTURE OF VARIOUS PARAMETERS */

```

```

long a_pl_cft_dist( n, end, tol, cft )

```

```

long n;                    /* CURRENT ELEMENT           */
long end;                  /* CURRENT END: 0L = I-END   */
/*                          1L = J-END           */
A_TOLERANCE tol;          /* DATA STRUCTURE OF PROGRAM TOLERANCES */
A_CFT_BC cft[];           /* DATA STRUCTURE FOR CFT BEAM-COLUMN   */

```

```

{

```

```

long   status = 1L;          /* RETURN STATUS          */
long   ctrl   = 0L;          /* COUNTER                */
long   i       = 0L;          /* FORCE INDEX              */
long   neg     = 1L;          /* AXIAL FORCE MULTIPLIER  */
                                POS = 1L (DO NOT CHANGE SIGN)
                                NEG = -1L (NEGATE FORCE VALUE)
double rho    = 0.0;          /* RATIO OF B.S. RADIUS TO L.S. RADIUS */
double rl    = 0.0;          /* LOADING SURFACE RADIUS */
double rb    = 0.0;          /* BOUNDING SURFACE RADIUS */
double al[ 4 ];              /* VECTOR CONTAINING L.S. CENTROIDS */
double ab[ 4 ];              /* VECTOR CONTAINING B.S. CENTROIDS */
double s[ 4 ];              /* VECTOR OF END OF STEP FORCES */

/*****
/* INITIALIZE ARRAYS */
*****/

for ( ctrl = 0L; ctrl < 4L; ctrl++ )
    {
        al[ ctrl ] = 0.0;
        ab[ ctrl ] = 0.0;
        s[ ctrl ] = 0.0;
    }

/*****
/* CALCULATE FORCE INDEX */
*****/

i = 6L * end;

/*****
/* IF J-END, THEN NEGATE THE FORCE SUCH THAT A COMPRESSIVE LOAD IS POSITIVE */
*****/

( i == 0L ) ? ( neg = POS ) : ( neg = NEG );

/*****
/* SET SHORTENED VARIABLES */
*****/

rho    = cft[ n ].rho[ end ];
rl     = cft[ n ].ls_rad[ end ];
rb     = cft[ n ].bs_rad[ end ];

al[ 1 ] = cft[ n ].ls_cent[ end ].p + cft[ n ].phi * rl * cft[ n ].po;
al[ 2 ] = cft[ n ].ls_cent[ end ].my;
al[ 3 ] = cft[ n ].ls_cent[ end ].mz;

ab[ 1 ] = cft[ n ].bs_cent[ end ].p + cft[ n ].phi * rb * cft[ n ].po;
ab[ 2 ] = cft[ n ].bs_cent[ end ].my;
ab[ 3 ] = cft[ n ].bs_cent[ end ].mz;

s[ 1 ] = cft[ n ].f2[ 1 + i ] * neg;
s[ 2 ] = cft[ n ].f2[ 5 + i ];
s[ 3 ] = cft[ n ].f2[ 6 + i ];

/*****
/* CALCULATE MROZ VECTOR */
*****/

cft[ n ].mroz[ end ].p = ( rho - 1L ) * s[ 1 ] - ( rho * al[ 1 ] - ab[ 1 ] );
cft[ n ].mroz[ end ].my = ( rho - 1L ) * s[ 2 ] - ( rho * al[ 2 ] - ab[ 2 ] );
cft[ n ].mroz[ end ].mz = ( rho - 1L ) * s[ 3 ] - ( rho * al[ 3 ] - ab[ 3 ] );

/*****
/* CALCULATE DISTANCE */
*****/

/* IF THE FORCE POINT IS ON THE B.S., SET d EQUAL TO ZERO */

if ( cft[ n ].surf[ end ] == BS )
    {
        cft[ n ].dist[ end ] = 0.0;
    }

```

```

    }

/* OTHERWISE COMPUTE ACCORDING TO THE MROZ VECTOR */
else if ( cft[ n ].state[ end ] == PL_IN )          /* INITIAL BREACH */
    {
    cft[ n ].dist_in[ end ] = sqrt( pow( cft[ n ].mroz[ end ].p, 2 ) +
                                   pow( cft[ n ].mroz[ end ].my, 2 ) +
                                   pow( cft[ n ].mroz[ end ].mz, 2 ) );
    cft[ n ].dist[ end ]   = cft[ n ].dist_in[ end ] - tol.surf;
    }

else if ( cft[ n ].state[ end ] == PL )           /* END WAS PLASTIC AT THE BEG OF STEP */
    {
    cft[ n ].dist[ end ] = sqrt( pow( cft[ n ].mroz[ end ].p, 2 ) +
                                pow( cft[ n ].mroz[ end ].my, 2 ) +
                                pow( cft[ n ].mroz[ end ].mz, 2 ) );

    if ( cft[ n ].dist[ end ] == cft[ n ].dist_in[ end ] )
        {
        cft[ n ].dist[ end ] = cft[ n ].dist_in[ end ] - tol.surf;
        }
    }

return( status );
}

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****
*****

* FUNCTION- a_pl_cft_drift.c

    @(#) CFT element force point DRIFT control

* CALLED FROM- a_pl_cft_state

* ABSTRACT- This function moves the end of step force point (which may lie
either beyond or beneath the L.S.) back to the loading surface
by moving normal to the axial force axis. The procedure
entails finding the vector that crosses the L.S., then
performing the bisection algorithm on this vector until the
force point is on the L.S. w/in a tol.
Following drift control back to the LS, the shears are
adjusted to equilibrate the moments that may have been alter
by the drift control. Except for torsion, tis procedure
ensures element equilibrium, since axial forces are not altered.

Note: if the force point is beyond the bounding surface, other
functions have already ensured that the two surfaces are
touching. Therefore moving to the L.S. will put the force
point on or w/in a tolerance of the B.S.
*****/

#include <math.h>          /* C MATH FUNCTIONS          */
#include <stdio.h>        /* C I/O FUNCTIONS          */
#include <string.h>       /* C STRING FUNCTIONS       */

#include "a_mac.h"        /* MACRO DEFINITIONS        */
#include "a_extern.h"     /* EXTERNAL VARIABLES       */
#include "a_cft_el.h"     /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
#include "a_param.h"     /* DATA STRUCTURE OF VARIOUS PARAMETERS */

#define IN      0L
#define OUT    1L

long  a_pl_cft_drift( n, end, tol, cft )

long      n;              /* CURRENT ELEMENT          */

```

```

long          end;          /* CURRENT ELEMENT END          */
A_TOLERANCE  tol;          /* DATA STRUCTURE OF TOLERANCES */
A_CFT_BC     cft[];        /* DATA STRUCTURE FOR CFT BEAM-COLUMN */

{
long  status = 1L;         /* RETURN STATUS          */
long  i      = 0L;         /* FORCE INDEX              */
long  done   = 0L;         /* FLAG TO INDICATE COMPLETION OF BISEC */
long  fpt    = 0L;         /* LOCATION OF FORCE PT RELATIVE TO L.S.
                          0L = IN (inside L.S.)
                          1L = OUT (outside L.S.)          */
long  neg     = 1L;         /* AXIAL FORCE MULTIPLIER
                          POS = 1L (DO NOT CHANGE SIGN)
                          NEG = -1L (NEGATE FORCE VALUE)    */
double eqn    = 0.0;       /* VALUE OF SURFACE EQUATION */
double eqn_i  = 0.0;       /* VALUE OF LAST STEP SURFACE EQUATION */
double l1    = 0.0;       /* LOWER AXIAL FORCE PT OF BISECT VEC */
double l2    = 0.0;       /* LOWER Y-MOMENT FORCE PT OF BISECT VEC */
double l3    = 0.0;       /* LOWER Z-MOMENT FORCE PT OF BISECT VEC */
double m1    = 0.0;       /* MID AXIAL FORCE PT OF BISECT VEC */
double m2    = 0.0;       /* MID Y-MOMENT FORCE PT OF BISECT VEC */
double m3    = 0.0;       /* MID Z-MOMENT FORCE PT OF BISECT VEC */
double u1    = 0.0;       /* UPPER AXIAL FORCE PT OF BISECT VEC */
double u2    = 0.0;       /* UPPER Y-MOMENT FORCE PT OF BISECT VEC */
double u3    = 0.0;       /* UPPER Z-MOMENT FORCE PT OF BISECT VEC */
double moment = 0.0;       /* NORMAL OF STRONG AND WK AXIS BENDING */

/***** */
/* CALCULATE FORCE INDEX */
/***** */

i = 6L * end;

/***** */
/* IF J-END, THEN NEGATE THE FORCE SUCH THAT A COMPRESSIVE LOAD IS POSITIVE */
/***** */

( i == 0L ) ? ( neg = POS ) : ( neg = NEG );

/***** */
/* IF DRIFT IS ALONG AXIAL FORCE AXIS, DO NOT PERFORM DRIFT CONTROL */
/***** */

moment = sqrt( pow( cft[ n ].f2[ 5 + i ] / cft[ n ].myo, 2.0 ) +
               pow( cft[ n ].f2[ 6 + i ] / cft[ n ].mzo, 2.0 ) );

if ( moment > tol.surf )
{
/***** */
/* SET INITIAL ENDPOINTS OF BISECTION VECTOR TO F2 FORCE PT AND L.S. CENTROID */
/***** */

u1 = ( neg * cft[ n ].f2[ 1 + i ] - cft[ n ].ls_cent[ end ].p -
      cft[ n ].ls_rad[ end ] * cft[ n ].phi * cft[ n ].po ) /
      ( cft[ n ].ls_rad[ end ] * cft[ n ].po );
u2 = ( cft[ n ].f2[ 5 + i ] - cft[ n ].ls_cent[ end ].my ) /
      ( cft[ n ].ls_rad[ end ] * cft[ n ].myo );
u3 = ( cft[ n ].f2[ 6 + i ] - cft[ n ].ls_cent[ end ].mz ) /
      ( cft[ n ].ls_rad[ end ] * cft[ n ].mzo );

l1 = u1;
l2 = 0.0;
l3 = 0.0;

/***** */
/* DETERMINE WHETHER FORCE POINT IS ON THE INSIDE OR OUTSIDE OF L.S. */
/***** */

/* CALL SURFACE EQN ROUTINE; CALCULATES VALUE OF L.S. EQN FOR l1, l2, AND l3 */

eqn = a_pl_cft_surf( n, cft, u1, u2, u3 );

( eqn < ( 1.0 + tol.surf ) ) ? ( fpt = IN ) : ( fpt = OUT );

```

```

/*****
/* IF THE END-OF-STEP POINT IS INSIDE THE L.S., MOVE OUTWARD ALONG THE VECTOR*/
/* FROM THE CENTROID OF THE L.S. TO THE FORCE POINT IN INCREMENTS EQUAL TO 10*/
/* TIMES THE L.S. TOLERANCE UNTIL THE L.S. IS CROSSED
*/
/*****

if ( fpt == IN )
{
    while ( eqn < ( 1.0 + tol.surf ) )
    {
        u2 += 10.0 * tol.surf * ( u2 - l2 );
        u3 += 10.0 * tol.surf * ( u3 - l3 );

        eqn = a_pl_cft_surf( n, cft, u1, u2, u3 );
    }

/*****
/* PERFORM BISECTION ON VECTOR WITH ENDPOINTS u AND l */
/*****

/* COMPUTE INITIAL MIDPOINT OF BISECTION VECTOR */

m1 = ( l1 + u1 ) / 2.0;
m2 = ( l2 + u2 ) / 2.0;
m3 = ( l3 + u3 ) / 2.0;

eqn = a_pl_cft_surf( n, cft, m1, m2, m3 );

/* CHECK MIDPOINT OF VECTOR AGAINST L.S. IF NOT ON L.S. W/I A TOLERANCE,
/* BISECT THE SEGMENT OF THE VECTOR CROSSING THE L.S AND REPEAT STEPS WITH
/* THE NEW MIDPOINT
*/

while ( ! done )
{
    eqn_i = eqn;

    if ( eqn > ( 1.0 + tol.surf ) )
    {
        u2 = m2;          /* ASSIGN UPPER FORCE POINT TO MIDPT */
        u3 = m3;

        m2 = ( l2 + u2 ) / 2.0;      /* BISECT VECTOR */
        m3 = ( l3 + u3 ) / 2.0;

        eqn = a_pl_cft_surf( n, cft, m1, m2, m3 );
    }
    else if ( eqn < 1.0 )
    {
        l2 = m2;          /* ASSIGN LOWER FORCE POINT OT MIDPT */
        l3 = m3;

        m2 = ( l2 + u2 ) / 2.0;      /* BISECT VECTOR */
        m3 = ( l3 + u3 ) / 2.0;

        eqn = a_pl_cft_surf( n, cft, m1, m2, m3 );
    }
    else
    {
        done = 1L;
    }

    if ( ( ! done ) && ( eqn_i == eqn ) )
    {
        done = 1L;
    }
} /* while ( ! done ) */

/*****
/* COMPUTE NEW f2 AND INCREMENTAL FORCES */
/*****

cft[ n ].f2[ 5 + i ] = m2 * ( cft[ n ].ls_rad[ end ] * cft[ n ].myo ) +

```

```

        cft[ n ].ls_cent[ end ].my;
cft[ n ].f2[ 6 + i ] = m3 * ( cft[ n ].ls_rad[ end ] * cft[ n ].mzo ) +
        cft[ n ].ls_cent[ end ].mz;

cft[ n ].f2_i[ 5 + i ] = cft[ n ].f2[ 5 + i ] - cft[ n ].f1[ 5 + i ];
cft[ n ].f2_i[ 6 + i ] = cft[ n ].f2[ 6 + i ] - cft[ n ].f1[ 6 + i ];

cft[ n ].df_i[ 5 + i ] = cft[ n ].f2_i[ 5 + i ] - cft[ n ].f1_i[ 5 + i ];
cft[ n ].df_i[ 6 + i ] = cft[ n ].f2_i[ 6 + i ] - cft[ n ].f1_i[ 6 + i ];

/*****
/* ADJUST SHEARS TO EQUILIBRATE MOMENTS */
*****/

cft[ n ].f2[ 2L ] = ( cft[ n ].f2[ 6L ] + cft[ n ].f2[ 12L ] ) / cft[ n ].l;
cft[ n ].f2[ 3L ] = ( cft[ n ].f2[ 5L ] + cft[ n ].f2[ 11L ] ) / cft[ n ].l;
cft[ n ].f2[ 8L ] = - cft[ n ].f2[ 2L ];
cft[ n ].f2[ 9L ] = - cft[ n ].f2[ 3L ];

cft[ n ].f2_i[ 2L ] = cft[ n ].f2[ 2L ] - cft[ n ].f1[ 2L ];
cft[ n ].f2_i[ 3L ] = cft[ n ].f2[ 3L ] - cft[ n ].f1[ 3L ];
cft[ n ].f2_i[ 8L ] = cft[ n ].f2[ 8L ] - cft[ n ].f1[ 8L ];
cft[ n ].f2_i[ 9L ] = cft[ n ].f2[ 9L ] - cft[ n ].f1[ 9L ];

cft[ n ].df_i[ 2L ] = cft[ n ].f2_i[ 2L ] - cft[ n ].f1_i[ 2L ];
cft[ n ].df_i[ 3L ] = cft[ n ].f2_i[ 3L ] - cft[ n ].f1_i[ 3L ];
cft[ n ].df_i[ 8L ] = cft[ n ].f2_i[ 8L ] - cft[ n ].f1_i[ 8L ];
cft[ n ].df_i[ 9L ] = cft[ n ].f2_i[ 9L ] - cft[ n ].f1_i[ 9L ];

}      /* if ( moment > tol.surf ) */

return( status );
}

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****
*****

*      FUNCTION- a_pl_cft_grad.c

                @(#) calculation of the CFT loading surface GRADient

*      CALLED FROM- a_pl_cft_state

*      ABSTRACT- This function calculates the gradient to the loading surface
                for the current element end. It changes the flag signaling if
                the end is loading or unloading.

                Note that a positive j-end axial force will be tensile although
                in the plastic formulation a positive force is always
                compressive. Therefore, before computing the gradient, the
                j-end axial force is negated using the 'negate' variable to
                rectify this.
*****
*****

#include <math.h>                /* C MATH FUNCTIONS                */
#include <stdio.h>               /* C I/O FUNCTIONS                */
#include <string.h>              /* C STRING FUNCTIONS            */

#include "a_mac.h"               /* MACRO DEFINITIONS              */
#include "a_extern.h"            /* EXTERNAL VARIABLES            */
#include "a_cft_el.h"            /* DATA STRUCTURE FOR CFT BEAM-COLUMN */

long      a_pl_cft_grad( n, end, force, cft )

long      n;                    /* CURRENT ELEMENT                */
long      end;                  /* CURRENT ELEMENT END (0L = i, 1L = j) */
long      force;                /* FORCES TO USE IN EVAL. OF GRADIENT
                                F1 = 1L (beg of step forces)

```

```

                                F2 = 2L (end of step forces)          */
A_CFT_BC      cft[];          /* DATA STRUCTURE FOR CFT BEAM-COLUMN          */
{
long   status = 1L;          /* RETURN STATUS          */
long   i       = 0L;          /* FORCE INDEX          */
long   neg     = 1L;          /* AXIAL FORCE MULTIPLIER          */
                                POS = 1L (DO NOT CHANGE SIGN)
                                NEG = -1L (NEGATE FORCE VALUE)      */
double product = 0.0;       /* DOT PRODUCT OF GRADIENT AND INCR FORCE*/
                                /* VECTOR, IF < 0, UNLOADING HAS OCCURRED*/
double p       = 0.0;       /* AXIAL FORCE - AXIAL BACK FORCE -
                                PHI * RADIUS * NOM. AXIAL FORCE      */
double my      = 0.0;       /* Y-MOMENT FORCE - Y-MOMENT BACK FORCE      */
double mz      = 0.0;       /* Z-MOMENT FORCE - Z-MOMENT BACK FORCE      */
double po      = 0.0;       /* L.S. RADIUS * NOM. AXIAL FORCE          */
double myo     = 0.0;       /* L.S. RADIUS * NOM. Y-MOMENT FORCE      */
double mzo     = 0.0;       /* L.S. RADIUS * NOM. Z-MOMENT FORCE      */

/*****/
/* CALCULATE FORCE INDEX */
/*****/

i = 6L * end;

/*****/
/* IF J-END, THEN NEGATE THE FORCE SUCH THAT A COMPRESSIVE LOAD IS POSITIVE */
/*****/

( i == 0L ) ? ( neg = POS ) : ( neg = NEG );

/*****/
/* COMPUTE CURRENT GRADIENTS */
/*****/

/* LOADING SURFACE EQUATION:          */
/* F = c1 * ( mz / mzo )^2 + c1 * ( my / myo )^2 + c2 * ( p / po )^2 +          */
/*      c3 * ( mz / mzo )^2 * ( p / po )^2 +          */
/*      c3 * ( my / myo )^2 * ( p / po )^2 +          */
/*      c4 * ( mz / mzo )^2 * ( my / myo )^2          */

if ( force == F1 )
{
/* USE BEGINNING OF ITERATION FORCES */
p = neg * ( cft[ n ].f1[ 1 + i ] + cft[ n ].f1_i[ 1 + i ] ) -
cft[ n ].ls_cent[ end ].p -
cft[ n ].phi * cft[ n ].ls_rad[ end ] * cft[ n ].po;
po = cft[ n ].ls_rad[ end ] * cft[ n ].po;

my = cft[ n ].f1[ 5 + i ] + cft[ n ].f1_i[ 5 + i ] -
cft[ n ].ls_cent[ end ].my;
myo = cft[ n ].ls_rad[ end ] * cft[ n ].myo;

mz = cft[ n ].f1[ 6 + i ] + cft[ n ].f1_i[ 6 + i ] -
cft[ n ].ls_cent[ end ].mz;
mzo = cft[ n ].ls_rad[ end ] * cft[ n ].mzo;
}

else if ( force == F2 )
{
/* USE END OF ITERATION FORCES */
p = neg * cft[ n ].f2[ 1 + i ] - cft[ n ].ls_cent[ end ].p -
cft[ n ].phi * cft[ n ].ls_rad[ end ] * cft[ n ].po;
po = cft[ n ].ls_rad[ end ] * cft[ n ].po;

my = cft[ n ].f2[ 5 + i ] - cft[ n ].ls_cent[ end ].my;
myo = cft[ n ].ls_rad[ end ] * cft[ n ].myo;

mz = cft[ n ].f2[ 6 + i ] - cft[ n ].ls_cent[ end ].mz;
mzo = cft[ n ].ls_rad[ end ] * cft[ n ].mzo;
}

cft[ n ].grad[ end ].p = 2.0 * p / ( po * po ) *

```

```

        ( cft[ n ].c2 + cft[ n ].c3 * pow( my / myo, 2 ) +
          cft[ n ].c3 * pow( mz / mzo, 2 ) );

cft[ n ].grad[ end ].my      = 2.0 * my / ( myo * myo ) *
        ( cft[ n ].c1 + cft[ n ].c3 * pow( p / po, 2 ) +
          cft[ n ].c4 * pow( mz / mzo, 2 ) );

cft[ n ].grad[ end ].mz      = 2.0 * mz / ( mzo * mzo ) *
        ( cft[ n ].c1 + cft[ n ].c3 * pow( p / po, 2 ) +
          cft[ n ].c4 * pow( my / myo, 2 ) );

/*****
/* COMPUTE DENOMINATOR OF NORMAL TO L.S. */
*****/

cft[ n ].norm[ end ] = sqrt( pow( cft[ n ].grad[ end ].p, 2 ) +
                             pow( cft[ n ].grad[ end ].my, 2 ) +
                             pow( cft[ n ].grad[ end ].mz, 2 ) );

/*****
/* CHECK FOR UNLOADING */
*****/

product = cft[ n ].grad[ end ].p * neg * cft[ n ].df_i[ 1 + i ] +
          cft[ n ].grad[ end ].my * cft[ n ].df_i[ 5 + i ] +
          cft[ n ].grad[ end ].mz * cft[ n ].df_i[ 6 + i ];

if ( product < 0.0 )
    {
        cft[ n ].status[ end ] = UNLOAD;
    }
else
    {
        cft[ n ].status[ end ] = LOAD;
    }

return( status );
}

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****
*****

* FUNCTION- a_pl_cft_iso_hard.c

        @(#) Plasticity routine for CFT ISotropic HARDening

* CALLED FROM- a_pl_cft_state

* ABSTRACT- This function performs the isotropic hardening calculations for
movement of the CFT loading surface and bounding surface.
*****/

#include <math.h>          /* C MATH FUNCTIONS          */
#include <stdio.h>        /* C I/O FUNCTIONS          */
#include <string.h>       /* C STRING FUNCTIONS       */

#include "a_mac.h"        /* MACRO DEFINITIONS       */
#include "a_extern.h"     /* EXTERNAL VARIABLES      */
#include "a_cft_el.h"     /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
#include "a_param.h"      /* DATA STRUCTURE OF VARIOUS PARAMETERS */

long a_pl_cft_iso_hard( n, end, cft )

long n;                  /* ELEMENT NUMBER          */
long end;                /* ELEMENT END            */
A_CFT_BC cft[];         /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
{

```

```

long   status = 1L;          /* RETURN STATUS */
long   i       = 0L;        /* FORCE AND STIFFNESS TERM INDEX */
long   ctrl    = 0L;        /* COUNTER */
long   neg     = 1L;        /* AXIAL FORCE MULTIPLIER
                             POS = 1L (DO NOT CHANGE SIGN)
                             NEG = -1L (NEGATE FORCE VALUE) */

double grad[ 7L ];         /* GRADIENT VECTOR STORED IN ARRAY */
double temp    = 0.0;      /* TEMPORARY PRODUCT STORAGE */
double num     = 0.0;      /* {dF/dS}T * [Kt] * {dq} */
double denom   = 0.0;      /* {dF/dS}T * [Kt + Kp] * {dF/dS} */
double dlamba  = 0.0;      /* SCALAR RELATING PLASTIC AND TOTAL DISPLS*/
double dqp_p   = 0.0;      /* INCR OF PLASTIC AXIAL DISPLACEMENT */
double dqp_my  = 0.0;      /* INCR OF PLASTIC Y-MOMENT DISPLACEMENT */
double dqp_mz  = 0.0;      /* INCR OF PLASTIC Z-MOMENT DISPLACEMENT */
double dwork   = 0.0;      /* INCR OF PLASTIC WORK */
double work_norm= 0.0;     /* PLASTIC WORK NORMALIZED BY STRAIN ENERGY*/
double old_lsr = 0.0;      /* BEGINNING OF STEP L.S. RADIUS */
double old_bsr = 0.0;      /* BEGINNING OF STEP B.S. RADIUS */
double eta_ls  = 0.0;      /* AMOUNT OF L.S. MOVEMENT */
double eta_bs  = 0.0;      /* AMOUNT OF B.S. MOVEMENT */
double al[ 4 ];           /* VECTOR CONTAINING L.S. CENTROIDS */
double ab[ 4 ];           /* VECTOR CONTAINING B.S. CENTROIDS */
double ds[ 4 ];           /* VECTOR OF INCREMENTAL FORCES */
double s[ 4 ];           /* VECTOR OF BEGINNING OF STEP FORCES */

/*****
/* INITIALIZE ARRAYS */
*****/

for ( ctrl = 0L; ctrl < 7L; ctrl++ )
    {
        grad[ ctrl ] = 0.0;
    }

for ( ctrl = 0L; ctrl < 4L; ctrl++ )
    {
        al[ ctrl ] = 0.0;
        ab[ ctrl ] = 0.0;
        ds[ ctrl ] = 0.0;
        s[ ctrl ]  = 0.0;
    }

/*****
/* CALCULATE FORCE AND STIFFNESS TERM INDEX */
*****/

i = 6L * end;

/*****
/* IF J-END, THEN NEGATE THE FORCE SUCH THAT A COMPRESSIVE LOAD IS POSITIVE */
*****/

( i == 0L ) ? ( neg = POS ) : ( neg = NEG );

/*****
/* SET SHORTENED VARIABLES */
*****/

al[ 1 ] = cft[ n ].ls_cent[ end ].p + cft[ n ].phi * cft[ n ].ls_rad[ end ] *
          cft[ n ].po;
al[ 2 ] = cft[ n ].ls_cent[ end ].my;
al[ 3 ] = cft[ n ].ls_cent[ end ].mz;

ab[ 1 ] = cft[ n ].bs_cent[ end ].p + cft[ n ].phi * cft[ n ].bs_rad[ end ] *
          cft[ n ].po;
ab[ 2 ] = cft[ n ].bs_cent[ end ].my;
ab[ 3 ] = cft[ n ].bs_cent[ end ].mz;

ds[ 1 ] = cft[ n ].df_i[ 1 + i ] * neg;
ds[ 2 ] = cft[ n ].df_i[ 5 + i ];
ds[ 3 ] = cft[ n ].df_i[ 6 + i ];

s[ 1 ] = ( cft[ n ].fl[ 1 + i ] + cft[ n ].fl_i[ 1 + i ] ) * neg;

```

```

s[ 2 ] = cft[ n ].fl[ 5 + i ] + cft[ n ].fl_i[ 5 + i ];
s[ 3 ] = cft[ n ].fl[ 6 + i ] + cft[ n ].fl_i[ 6 + i ];

/*****
/* CALCULATE INCREMENT OF PLASTIC DISPLACEMENT */
*****/

/* ASSIGN VALUES TO TEMPORARY ARRAYS */

grad[ 1 ]      = cft[ n ].grad[ end ].p;
grad[ 5 ]      = cft[ n ].grad[ end ].my;
grad[ 6 ]      = cft[ n ].grad[ end ].mz;

/* CALCULATE NUMERATOR OF DLAMBDA: num = {dF/dS}T * {dS} */

num = grad[ 1 ] * ds[ 1 ] + grad[ 5 ] * ds[ 2 ] + grad[ 6 ] * ds[ 3 ];

/* CALCULATE DENOMINATOR OF DLAMBDA: denom = {dF/dS}T * Kp * {dF/dS} */

denom = grad[ 1 ] * cft[ n ].kp[ end ].p * grad[ 1 ] +
        grad[ 5 ] * cft[ n ].kp[ end ].my * grad[ 5 ] +
        grad[ 6 ] * cft[ n ].kp[ end ].mz * grad[ 6 ];

/* CALCULATE DLAMBDA AND INCREMENTAL PLASTIC DISPLACEMENTS */

dlambda = num / denom;

dqp_p  = dlambda * grad[ 1 ];
dqp_my = dlambda * grad[ 5 ];
dqp_mz = dlambda * grad[ 6 ];

/*****
/* CALCULATE STEP AND ACCUMULATED PLASTIC WORK */
*****/

dwork  = neg * cft[ n ].df_i[ 1 + i ] * dqp_p +
        cft[ n ].df_i[ 5 + i ] * dqp_my +
        cft[ n ].df_i[ 6 + i ] * dqp_mz;

cft[ n ].pl_work[ end ] += dwork;
work_norm  = cft[ n ].pl_work[ end ] / cft[ n ].el_work;

/*****
/* CALCULATE NEW L.S. AND B.S. RADII */
*****/

old_lsr = cft[ n ].ls_rad[ end ];
old_bsr = cft[ n ].bs_rad[ end ];

cft[ n ].ls_rad[ end ] = cft[ n ].lsr_fn[ end ] - ( cft[ n ].lsr_fn[ end ] -
        cft[ n ].lsr_in[ end ] ) * exp(
        -cft[ n ].ls_iso[ end ] * work_norm );

/* B.S. RADIUS FORMULATION */

if ( work_norm <= cft[ n ].iso_work[ end ] )
    {
    cft[ n ].bs_rad[ end ] =
        cft[ n ].bsr_intm[ end ] - ( cft[ n ].bsr_intm[ end ] -
        cft[ n ].bsr_in[ end ] ) * exp( -cft[ n ].bs_iso[ end ] *
        work_norm );

    cft[ n ].bsr_upd[ end ] = cft[ n ].bs_rad[ end ];
    }

else
    {
    cft[ n ].bs_rad[ end ] =
        cft[ n ].bsr_fn[ end ] - ( cft[ n ].bsr_fn[ end ] -
        cft[ n ].bsr_upd[ end ] ) * pow( 1.1, -cft[ n ].bs_iso[ end ] *
        ( work_norm - cft[ n ].iso_work[ end ] ) );
    }

/* CALCULATE NEW RATIO OF SURFACE RADII */

```

```

cft[ n ].rho[ end ] = cft[ n ].bs_rad[ end ] / cft[ n ].ls_rad[ end ];

/*****
/* CALCULATE UPDATED k2 PARAMETERS, CONCRETE ELASTIC MODULUS */
*****/

cft[ n ].emod_c          = 0.3 * cft[ n ].ec_in + ( 0.689 * cft[ n ].ec_in ) *
                        exp( -cft[ n ].ec_iso[ end ] * work_norm );

cft[ n ].k2[ end ].p    = cft[ n ].k2f[ end ].p - ( cft[ n ].k2f[ end ].p -
cft[ n ].k2i[ end ].p ) *
                        exp( -cft[ n ].k2_iso[ end ] * work_norm );
cft[ n ].k2[ end ].my   = cft[ n ].k2f[ end ].my - ( cft[ n ].k2f[ end ].my -
cft[ n ].k2i[ end ].my ) *
                        exp( -cft[ n ].k2_iso[ end ] * work_norm );
cft[ n ].k2[ end ].mz   = cft[ n ].k2f[ end ].mz - ( cft[ n ].k2f[ end ].mz -
cft[ n ].k2i[ end ].mz ) *
                        exp( -cft[ n ].k2_iso[ end ] * work_norm );

/*****
/* CALCULATE INCREMENTAL SURFACE CENTROID MOVEMENT DUE TO ISO HARD */
*****/

/* CALCULATE DISTANCE TO MOVE CENTROIDS */

eta_ls = 1.0 - ( cft[ n ].ls_rad[ end ] / old_lsr );
eta_bs = 1.0 - ( cft[ n ].bs_rad[ end ] / old_bsr );

/* CALCULATE VECTORS OF CENTROID MOVEMENT */

cft[ n ].dls_iso[ end ].p = eta_ls * ( s[ 1 ] - al[ 1 ] );
cft[ n ].dls_iso[ end ].my = eta_ls * ( s[ 2 ] - al[ 2 ] );
cft[ n ].dls_iso[ end ].mz = eta_ls * ( s[ 3 ] - al[ 3 ] );

cft[ n ].dbs_iso[ end ].p = eta_bs * ( s[ 1 ] - ab[ 1 ] );
cft[ n ].dbs_iso[ end ].my = eta_bs * ( s[ 2 ] - ab[ 2 ] );
cft[ n ].dbs_iso[ end ].mz = eta_bs * ( s[ 3 ] - ab[ 3 ] );

/* COMPUTE SURFACE CENTROIDS AFTER ISOTROPIC HARDENING */

cft[ n ].ls_cent[ end ].p += cft[ n ].dls_iso[ end ].p;
cft[ n ].ls_cent[ end ].my += cft[ n ].dls_iso[ end ].my;
cft[ n ].ls_cent[ end ].mz += cft[ n ].dls_iso[ end ].mz;

cft[ n ].bs_cent[ end ].p += cft[ n ].dbs_iso[ end ].p;
cft[ n ].bs_cent[ end ].my += cft[ n ].dbs_iso[ end ].my;
cft[ n ].bs_cent[ end ].mz += cft[ n ].dbs_iso[ end ].mz;

return( status );
}

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****
* FUNCTION- a_pl_cft_kin_hard
*
* @(#) PLasticity routine for CFT KINematic HARDening
* CALLED FROM- a_pl_cft_state
*
* ABSTRACT- This function performs the kinematic hardening calculations for
movement of the CFT loading surface and bounding surface.
In the input file, the user may specify whether the loading
surface should move in the Mroz direction or the Tseng
direction. Both of these routines are contained herein.
Additionally, if the force point is on the bounding surface,
the surfaces move in the direction of the incremental force
vector.

```

```

*****/
#include <math.h>          /* C MATH FUNCTIONS          */
#include <stdio.h>         /* C I/O FUNCTIONS          */
#include <string.h>       /* C STRING FUNCTIONS       */

#include "a_mac.h"        /* MACRO DEFINITIONS       */
#include "a_extern.h"    /* EXTERNAL VARIABLES      */
#include "a_cft_el.h"    /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
#include "a_param.h"     /* DATA STRUCTURE OF VARIOUS PARAMETERS */

long  a_pl_cft_kin_hard( n, end, tol, flag, cft )

long      n;              /* ELEMENT NUMBER          */
long      end;           /* ELEMENT END             */
A_TOLERANCE  tol;       /* DATA STRUCTURE OF PROGRAM TOLERANCES */
A_FLAGS      *flag;     /* DATA STRUCTURE OF PROGRAM FLAGS      */
A_CFT_BC     cft[];     /* DATA STRUCTURE FOR CFT BEAM-COLUMN   */

{
long  status = 1L;      /* RETURN STATUS          */
long  ctrl  = 0L;      /* COUNTER                */
long  i     = 0L;      /* FORCE INDEX             */
long  done  = 0L;      /* FLAG TO INDICATE COMPLETION OF BISECT */
long  neg   = 1L;      /* AXIAL FORCE MULTIPLIER */
                        /* POS = 1L (DO NOT CHANGE SIGN)        */
                        /* NEG = -1L (NEGATE FORCE VALUE)        */
double r1   = 0.0;     /* LOADING SURFACE RADIUS */
double rb   = 0.0;     /* BOUNDING SURFACE RADIUS */
double rho  = 0.0;     /* RATIO OF B.S. SIZE TO L.S. SIZE     */
double num  = 0.0;     /* {grad}T * df           */
double denom = 0.0;    /* {grad}T * MROZ OR TSENG VECTOR      */
double al[ 4 ];       /* VECTOR CONTAINING L.S. CENTROIDS     */
double ab[ 4 ];       /* VECTOR CONTAINING B.S. CENTROIDS     */
double sl[ 4 ];       /* VECTOR OF f1 FORCES                 */
double ds[ 4 ];       /* VECTOR OF INCREMENTAL FORCES        */

/* TSENG VARIABLES */
double l1   = 0.0;     /* LOWER VALUE OF BISECTION VECTOR     */
double l2   = 0.0;     /* LOWER VALUE OF BISECTION VECTOR     */
double l3   = 0.0;     /* LOWER VALUE OF BISECTION VECTOR     */
double u1   = 0.0;     /* UPPER VALUE OF BISECTION VECTOR     */
double u2   = 0.0;     /* UPPER VALUE OF BISECTION VECTOR     */
double u3   = 0.0;     /* UPPER VALUE OF BISECTION VECTOR     */
double m1   = 0.0;     /* MID VALUE OF BISECTION VECTOR       */
double m2   = 0.0;     /* MID VALUE OF BISECTION VECTOR       */
double m3   = 0.0;     /* MID VALUE OF BISECTION VECTOR       */
double eqn  = 0.0;     /* VALUE OF B.S. EQUATION (BISECTION)  */
double ds_norm = 0.0; /* NORMAL OF THE INCR. FORCE VECTOR     */
double sl_norm = 0.0; /* NORMAL OF THE f1 FORCE VECTOR        */
double incr1 = 0.0;   /* INCREMENTAL VECTOR ADDED TO f1      */
double incr2 = 0.0;   /* INCREMENTAL VECTOR ADDED TO f1      */
double incr3 = 0.0;   /* INCREMENTAL VECTOR ADDED TO f1      */

/*****/
/* INITIALIZE ARRAYS */
/*****/

for ( ctrl = 0L; ctrl < 4L; ctrl++ )
{
    al[ ctrl ] = 0.0;
    ab[ ctrl ] = 0.0;
    sl[ ctrl ] = 0.0;
    ds[ ctrl ] = 0.0;
}

/*****/
/* CALCULATE FORCE INDEX */
/*****/

i = 6L * end;

```

```

/*****
/* IF J-END, THEN NEGATE THE FORCE SUCH THAT A COMPRESSIVE LOAD IS POSITIVE */
*****/

( i == 0L ) ? ( neg = POS ) : ( neg = NEG );

/*****
/* SET SHORTENED VARIABLES */
*****/

rl      = cft[ n ].ls_rad[ end ];
rb      = cft[ n ].bs_rad[ end ];
rho     = cft[ n ].rho[ end ];

al[ 1 ] = cft[ n ].ls_cent[ end ].p + cft[ n ].phi * rl * cft[ n ].po;
al[ 2 ] = cft[ n ].ls_cent[ end ].my;
al[ 3 ] = cft[ n ].ls_cent[ end ].mz;

ab[ 1 ] = cft[ n ].bs_cent[ end ].p + cft[ n ].phi * rb * cft[ n ].po;
ab[ 2 ] = cft[ n ].bs_cent[ end ].my;
ab[ 3 ] = cft[ n ].bs_cent[ end ].mz;

s1[ 1 ] = ( cft[ n ].fl[ 1 + i ] + cft[ n ].fl_i[ 1 + i ] ) * neg;
s1[ 2 ] = cft[ n ].fl[ 5 + i ] + cft[ n ].fl_i[ 5 + i ];
s1[ 3 ] = cft[ n ].fl[ 6 + i ] + cft[ n ].fl_i[ 6 + i ];

ds[ 1 ] = cft[ n ].df_i[ 1 + i ] * neg;
ds[ 2 ] = cft[ n ].df_i[ 5 + i ];
ds[ 3 ] = cft[ n ].df_i[ 6 + i ];

/*****
/* MOVE SURFACES FOR CASE WHERE THE SURFACES TOUCH AND THE FORCE POINT
/* IS ON OR BEYOND THE BS. MOVE BOTH SURFACES IN THE DIRECTION OF
/* THE NORMAL TO THE SURFACES.
*****/

if ( cft[ n ].surf[ end ] == BS )
{
    num      = cft[ n ].grad[ end ].p * ds[ 1 ] +
              cft[ n ].grad[ end ].my * ds[ 2 ] +
              cft[ n ].grad[ end ].mz * ds[ 3 ];

    denom    = cft[ n ].norm[ end ];

    cft[ n ].dls_kin[ end ].p = ( num / denom ) * cft[ n ].grad[ end ].p;
    cft[ n ].dls_kin[ end ].my = ( num / denom ) * cft[ n ].grad[ end ].my;
    cft[ n ].dls_kin[ end ].mz = ( num / denom ) * cft[ n ].grad[ end ].mz;

    cft[ n ].dbs_kin[ end ].p = cft[ n ].dls_kin[ end ].p;
    cft[ n ].dbs_kin[ end ].my = cft[ n ].dls_kin[ end ].my;
    cft[ n ].dbs_kin[ end ].mz = cft[ n ].dls_kin[ end ].mz;
}

/*****
/* KINEMATICALLY HARDEN THE L.S. AND B.S. USING THE MROZ DIRECTION VECTOR */
*****/

else if ( flag->cft_kin == 1L )
{
    /* CALCULATE MROZ VECTOR */

    cft[ n ].mroz[ end ].p = ( rho - 1L ) * s1[ 1 ] - ( rho * al[ 1 ] - ab[ 1 ] );
    cft[ n ].mroz[ end ].my = ( rho - 1L ) * s1[ 2 ] - ( rho * al[ 2 ] - ab[ 2 ] );
    cft[ n ].mroz[ end ].mz = ( rho - 1L ) * s1[ 3 ] - ( rho * al[ 3 ] - ab[ 3 ] );

    /* CALCULATE VECTOR OF INCREMENTAL L.S. CENTROID MOVEMENT DUE TO KIN HARD */

    num      = cft[ n ].grad[ end ].p * ds[ 1 ] +
              cft[ n ].grad[ end ].my * ds[ 2 ] +
              cft[ n ].grad[ end ].mz * ds[ 3 ];

    denom    = cft[ n ].grad[ end ].p * cft[ n ].mroz[ end ].p +
              cft[ n ].grad[ end ].my * cft[ n ].mroz[ end ].my +

```

```

cft[ n ].grad[ end ].mz * cft[ n ].mroz[ end ].mz;

cft[ n ].dls_kin[ end ].p = ( num / denom ) * cft[ n ].mroz[ end ].p;
cft[ n ].dls_kin[ end ].my = ( num / denom ) * cft[ n ].mroz[ end ].my;
cft[ n ].dls_kin[ end ].mz = ( num / denom ) * cft[ n ].mroz[ end ].mz;

/* CALCULATE VECTOR OF INCREMENTAL B.S. CENTROID MOVEMENT DUE TO KIN HARD */

( cft[ n ].kpb[ end ].p == 0.0 ) ? ( cft[ n ].dbs_kin[ end ].p = 0.0 ) :
( cft[ n ].dbs_kin[ end ].p = cft[ n ].kpb[ end ].p / cft[ n ].kpb[ end ].p *
  cft[ n ].dls_kin[ end ].p );

( cft[ n ].kpb[ end ].my == 0.0 ) ? ( cft[ n ].dbs_kin[ end ].my = 0.0 ) :
( cft[ n ].dbs_kin[ end ].my = cft[ n ].kpb[ end ].my / cft[ n ].kpb[ end ].my *
  cft[ n ].dls_kin[ end ].my );

( cft[ n ].kpb[ end ].mz == 0.0 ) ? ( cft[ n ].dbs_kin[ end ].mz = 0.0 ) :
( cft[ n ].dbs_kin[ end ].mz = cft[ n ].kpb[ end ].mz / cft[ n ].kpb[ end ].mz *
  cft[ n ].dls_kin[ end ].mz );

} /* else if ( flag->cft_kin == 1L ) */

/*****
/* KINEMATICALLY HARDEN THE L.S. AND B.S. USING THE TSENG DIRECTION VECTOR */
*****/

else if ( flag->cft_kin == 2L )
{
/* CALCULATE TSENG VECTOR */

/* FIND INTERSECTION OF THE PROJECTION OF INCR FORCE VECTOR WITH THE B.S. */
/* ADD df VECTORS TO f1 UNTIL B.S. IS CROSSED, THEN PERFORM BISECTION */

/* CALCULATE INCREMENTAL VECTOR. BY NORMALIZING THE INCR. FORCE VECTOR AND
MULTIPLYING BY THE s1 NORMAL, THE VECTOR THAT IS ADDED TO THE s1 FORCES IS
OF THE PROPER MAGNITUDE. OTHERWISE, ds MAY BE VERY SMALL AND THOUSANDS OF
INCRS WOULD NEED TO BE ADDED TO s1 TO REACH THE B.S. */

ds_norm = sqrt( pow( ds[ 1 ], 2 ) + pow( ds[ 2 ], 2 ) + pow( ds[ 3 ], 2 ) );
s1_norm = sqrt( pow( s1[ 1 ], 2 ) + pow( s1[ 2 ], 2 ) + pow( s1[ 3 ], 2 ) );
incr1 = s1_norm * ds[ 1 ] / ds_norm;
incr2 = s1_norm * ds[ 2 ] / ds_norm;
incr3 = s1_norm * ds[ 3 ] / ds_norm;

/* SET INITIAL ENDPOINTS OF VECTOR TO BE BIASECTED */

u1 = s1[ 1 ];
u2 = s1[ 2 ];
u3 = s1[ 3 ];

l1 = u1 - ds[ 1 ];
l2 = u2 - ds[ 2 ];
l3 = u3 - ds[ 3 ];

eqn = a_pl_cft_surf( n, cft,
( u1 - ab[ 1 ] ) / ( rb * cft[ n ].po ),
( u2 - ab[ 2 ] ) / ( rb * cft[ n ].myo ),
( u3 - ab[ 3 ] ) / ( rb * cft[ n ].mzo ) );

/* ADD df INCRS. TO f1 UNTIL THE B.S. IS CROSSED */

while ( eqn < ( 1.0 + tol.surf ) )
{
l1 = u1;
l2 = u2;
l3 = u3;

u1 += incr1;
u2 += incr2;
u3 += incr3;

eqn = a_pl_cft_surf( n, cft,
( u1 - ab[ 1 ] ) / ( rb * cft[ n ].po ),

```

```

        ( u2 - ab[ 2 ] ) / ( rb * cft[ n ].myo ),
        ( u3 - ab[ 3 ] ) / ( rb * cft[ n ].mzo ) );
    }

/* BISECT INCR VECTOR ( u - 1 ) THAT CROSSED THE B.S. TO GET INTERSECTING PT. */

m1 = ( l1 + u1 ) / 2.0;
m2 = ( l2 + u2 ) / 2.0;
m3 = ( l3 + u3 ) / 2.0;

eqn = a_pl_cft_surf( n, cft,
                    ( m1 - ab[ 1 ] ) / ( rb * cft[ n ].po ),
                    ( m2 - ab[ 2 ] ) / ( rb * cft[ n ].myo ),
                    ( m3 - ab[ 3 ] ) / ( rb * cft[ n ].mzo ) );

while ( ! done )
    {
    if ( eqn > ( 1.0 + tol.surf ) )
        {
        u1 = m1;          /* ASSIGN UPPER FORCE POINT TO MIDPT */
        u2 = m2;
        u3 = m3;

        m1 = ( l1 + u1 ) / 2.0;      /* BISECT VECTOR */
        m2 = ( l2 + u2 ) / 2.0;
        m3 = ( l3 + u3 ) / 2.0;

        eqn = a_pl_cft_surf( n, cft,
                            ( m1 - ab[ 1 ] ) / ( rb * cft[ n ].po ),
                            ( m2 - ab[ 2 ] ) / ( rb * cft[ n ].myo ),
                            ( m3 - ab[ 3 ] ) / ( rb * cft[ n ].mzo ) );
        }

    else if ( eqn < 1.0 )
        {
        l1 = m1;          /* ASSIGN LOWER FORCE POINT OT MIDPT */
        l2 = m2;
        l3 = m3;

        m1 = ( l1 + u1 ) / 2.0;      /* BISECT VECTOR */
        m2 = ( l2 + u2 ) / 2.0;
        m3 = ( l3 + u3 ) / 2.0;

        eqn = a_pl_cft_surf( n, cft,
                            ( m1 - ab[ 1 ] ) / ( rb * cft[ n ].po ),
                            ( m2 - ab[ 2 ] ) / ( rb * cft[ n ].myo ),
                            ( m3 - ab[ 3 ] ) / ( rb * cft[ n ].mzo ) );
        }

    else
        {
        done = 1L;
        }
    } /* while ( ! done ) */

/* COMPUTE COMPONENTS OF THE TSENG VECTOR */

cft[ n ].tseng[ end ].p = m1 - m1 / rho + ab[ 1 ] / rho - al[ 1 ];
cft[ n ].tseng[ end ].my = m2 - m2 / rho + ab[ 2 ] / rho - al[ 2 ];
cft[ n ].tseng[ end ].mz = m3 - m3 / rho + ab[ 3 ] / rho - al[ 3 ];

/* CALCULATE VECTOR OF INCREMENTAL L.S. CENTROID MOVEMENT DUE TO KIN HARD */
/* (da)kin = [ ( gradT * ds ) / ( gradT * tseng ) ] * ( tseng ) */

num = cft[ n ].grad[ end ].p * ds[ 1 ] +
      cft[ n ].grad[ end ].my * ds[ 2 ] +
      cft[ n ].grad[ end ].mz * ds[ 3 ];

denom = cft[ n ].grad[ end ].p * cft[ n ].tseng[ end ].p +
        cft[ n ].grad[ end ].my * cft[ n ].tseng[ end ].my +
        cft[ n ].grad[ end ].mz * cft[ n ].tseng[ end ].mz;

cft[ n ].dls_kin[ end ].p = ( num / denom ) * cft[ n ].tseng[ end ].p;

```

```

cft[ n ].dls_kin[ end ].my = ( num / denom ) * cft[ n ].tseng[ end ].my;
cft[ n ].dls_kin[ end ].mz = ( num / denom ) * cft[ n ].tseng[ end ].mz;

/* CALCULATE VECTOR OF INCREMENTAL B.S. CENTROID MOVEMENT DUE TO KIN HARD */

cft[ n ].dbs_kin[ end ].p = cft[ n ].kpb[ end ].p / cft[ n ].kp[ end ].p *
    cft[ n ].dls_kin[ end ].p;
cft[ n ].dbs_kin[ end ].my = cft[ n ].kpb[ end ].my / cft[ n ].kp[ end ].my *
    cft[ n ].dls_kin[ end ].my;
cft[ n ].dbs_kin[ end ].mz = cft[ n ].kpb[ end ].mz / cft[ n ].kp[ end ].mz *
    cft[ n ].dls_kin[ end ].mz;

}      /* else if ( flag->cft_kin == 2L )      */

/* COMPUTE SURFACE CENTROIDS AFTER KINEMATIC HARDENING */

cft[ n ].ls_cent[ end ].p += cft[ n ].dls_kin[ end ].p;
cft[ n ].ls_cent[ end ].my += cft[ n ].dls_kin[ end ].my;
cft[ n ].ls_cent[ end ].mz += cft[ n ].dls_kin[ end ].mz;

cft[ n ].bs_cent[ end ].p += cft[ n ].dbs_kin[ end ].p;
cft[ n ].bs_cent[ end ].my += cft[ n ].dbs_kin[ end ].my;
cft[ n ].bs_cent[ end ].mz += cft[ n ].dbs_kin[ end ].mz;

return( status );
}

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****
*/

*      FUNCTION- a_pl_cft_state.c

*      @(#) CFT element end plasticity STATE

*      CALLED FROM- a_drv_dynamic, a_drv_static

*      ABSTRACT- This function determines the state of each CFT element end
*                and calls the appropriate functions to perform the plasticity
*                calculations. The comments below in the code explain this
*                procedure in greater detail.

*                Note that a positive j-end axial force will be tensile although
*                in the plastic formulation a positive force is always
*                compressive. Therefore, before computing the gradient, the
*                j-end axial force is negated using the 'negate' variable to
*                rectify this.
*****

#include <math.h>          /* C MATH FUNCTIONS          */
#include <stdio.h>        /* C I/O FUNCTIONS          */
#include <string.h>       /* C STRING FUNCTIONS       */

#include "a_mac.h"        /* MACRO DEFINITIONS        */
#include "a_extern.h"     /* EXTERNAL VARIABLES       */
#include "a_param.h"      /* DATA STRUCTURE OF VARIOUS PARAMETERS */
#include "a_cft_el.h"     /* DATA STRUCTURE FOR STEEL BEAM-COLUMN */
#include "a_global.h"     /* DATA STRUCTURE OF DOF-BASED VARIABLES */

long  a_pl_cft_state( size, nr, tol, flag, cft, global )

A_MODEL_SIZE  *size;      /* DATA STRUCTURE OF GLOBAL STRUCT SIZE */
A_NEW_RAPH    *nr;        /* DATA STRUCTURE FOR NEWTON-RAPHSON    */
A_TOLERANCE   tol;       /* DATA STRUCTURE FOR PROGRAM TOLERANCES */
A_FLAGS       *flag;      /* DATA STRUCTURE OF PROGRAM FLAGS      */
A_CFT_BC      cft[];     /* DATA STRUCTURE FOR CFT BEAM-COLUMN   */
A_GLOBAL      global[];  /* DATA STRUCTURE OF DOF-BASED VARIABLES */

{
long  status = 1L;      /* RETURN STATUS */

```

```

long   ctr1   = 0L;          /* COUNTER */
long   ctr2   = 0L;          /* COUNTER */
long   end    = 0L;          /* END: 0 = i-end, 1 = j-end */
long   i      = 0L;          /* FORCE INDEX */
long   n      = 0L;          /* ELEMENT COUNTER */
long   neg    = 1L;          /* AXIAL FORCE MULTIPLIER
                             POS = 1L (DO NOT CHANGE SIGN)
                             NEG = -1L (NEGATE FORCE VALUE) */

/*****
/* LOOP OVER CFT ELEMENTS */
*****/

for ( n = 1L; n <= size->num_cft_elems; n++ )
{
/*****
/* IF ELEMENT HAS NOT PLASTIFIED, UPDATE INITIAL ELASTIC ENERGY */
/* I.E., CALCULATE THE STRAIN ENERGY DUE TO AXIAL AND BENDING FORCES */
/* U = Up + Umy + Umz = (P^2*L)/(2EA) +
/*      1/(2EI)*(Myi^2*L+(Myi+Myj)^2*L/3 - Myi*L*(Myi+Myj)) +
/*      1/(2EI)*(Mzi^2*L+(Mzi+Mzj)^2*L/3 - Mzi*L*(Mzi+Mzj)) +
*****/

if ( ( ( cft[ n ].state[ 0L ] == EL ) && ( ! cft[ n ].hinge ) ) ||
      ( cft[ n ].state[ 1L ] == EL ) && ( ! cft[ n ].hinge ) ) )
{
    cft[ n ].el_work =
        ( pow( cft[ n ].f2[ 1 ], 2 ) * cft[ n ].l /
          ( 2.0 * cft[ n ].ea ) ) +
        ( pow( cft[ n ].f2[ 5 ], 2 ) * cft[ n ].l +
          pow( cft[ n ].f2[ 5 ] + cft[ n ].f2[ 11 ], 2 ) *
          cft[ n ].l / 3.0 - cft[ n ].f2[ 5 ] * cft[ n ].l *
          ( cft[ n ].f2[ 5 ] + cft[ n ].f2[ 11 ] ) ) /
          ( 2.0 * cft[ n ].eiz ) +
        ( pow( cft[ n ].f2[ 6 ], 2 ) * cft[ n ].l +
          pow( cft[ n ].f2[ 6 ] + cft[ n ].f2[ 12 ], 2 ) *
          cft[ n ].l / 3.0 - cft[ n ].f2[ 6 ] * cft[ n ].l *
          ( cft[ n ].f2[ 6 ] + cft[ n ].f2[ 12 ] ) ) /
          ( 2.0 * cft[ n ].eiz ) ;
}

/*****
/* CHECK THE PLASTICITY STATE OF EACH ELEMENT END */
*****/

for ( end = 0L; end <= 1L; end++ )
{
/*****
/* CALCULATE FORCE INDEX */
*****/

i = 6L * end;

/*****
/* IF J-END, THEN NEGATE THE FORCE SUCH THAT A COMPRESSIVE LOAD IS POSITIVE */
*****/

( i == 0L ) ? ( neg = POS ) : ( neg = NEG );

/*****
/* END WAS PLASTIC LAST STEP */
/* I. CHECK GRADIENT FOR UNLOADING */
/* A. NEG: SET FLAG TO UNLOADING */
/* 1. CHECK IF RELOADING (IS F2 > L.S.?) */
/* a. NO: SET FLAG TO ELASTIC */
/* b. YES: SET FLAG TO LOADING */
/* DO STEP B. */
/* B. POS: SET FLAG TO LOADING */
/* 1. MOVE SURFACES (KIN & ISO) */
/* 2. DRIFT CONTROL BACK TO L.S. */
/* 3. CALCULATE DISTANCE */

```

```

/*          4.  CALCULATE NEW GRADIENT FOR USE IN Kr          */
/*****/

if ( ( cft[ n ].state[ end ] == PL ) || ( cft[ n ].state[ end ] == PL_IN ) )
{
  cft[ n ].state[ end ] = PL;

  /*****/
  /* CHECK IF HINGE UNLOADED THIS STEP */
  /*****/

  a_pl_cft_grad( n, end, F2, cft );

  /*****/
  /* NEGATIVE GRADIENT:  CHECK IF HINGE TRULY UNLOADED */
  /*****/

  if ( cft[ n ].status[ end ] == UNLOAD )
  {

    /*****/
    /* DETERMINE STATE OF FORCE POINT AND SET FLAGS */
    /*****/

    cft[ n ].ls_eqn[ end ] = a_pl_cft_surf( n, cft,
      ( neg * cft[ n ].f2[ 1+i ] - cft[ n ].ls_cent[ end ].p -
        cft[ n ].phi * cft[ n ].ls_rad[ end ] * cft[ n ].po ) /
      ( cft[ n ].ls_rad[ end ] * cft[ n ].po ),
      ( cft[ n ].f2[ 5 + i ] - cft[ n ].ls_cent[ end ].my ) /
      ( cft[ n ].ls_rad[ end ] * cft[ n ].myo ),
      ( cft[ n ].f2[ 6 + i ] - cft[ n ].ls_cent[ end ].mz ) /
      ( cft[ n ].ls_rad[ end ] * cft[ n ].mzo ) );

    cft[ n ].bs_eqn[ end ] = a_pl_cft_surf( n, cft,
      ( neg * cft[ n ].f2[ 1+i ] - cft[ n ].bs_cent[ end ].p -
        cft[ n ].phi * cft[ n ].bs_rad[ end ] * cft[ n ].po ) /
      ( cft[ n ].bs_rad[ end ] * cft[ n ].po ),
      ( cft[ n ].f2[ 5 + i ] - cft[ n ].bs_cent[ end ].my ) /
      ( cft[ n ].bs_rad[ end ] * cft[ n ].myo ),
      ( cft[ n ].f2[ 6 + i ] - cft[ n ].bs_cent[ end ].mz ) /
      ( cft[ n ].bs_rad[ end ] * cft[ n ].mzo ) );

    if ( cft[ n ].ls_eqn[ end ] >= 1.0 )
    {
      cft[ n ].status[ end ] = LOAD;
      cft[ n ].state[ end ] = PL;
      cft[ n ].surf[ end ] = LS;

      if ( cft[ n ].bs_eqn[ end ] >= 1.0 )
      {
        cft[ n ].surf[ end ] = BS;
      }
    }

    /*****/
    /* HINGE UNLOADED, SET END TO ELASTIC */
    /*****/

    else
    {
      cft[ n ].status[ end ] = UNLOAD;
      cft[ n ].state[ end ] = EL;
      cft[ n ].surf[ end ] = NONE;
      cft[ n ].duct[ end ] = 0.0;
    }

  } /* if ( cft[ n ].status[ end ] == UNLOAD ) */

  /*****/
  /* END IS PLASTIC */
  /*****/

  if ( cft[ n ].status[ end ] == LOAD )
  {

```

```

/* CHECK IF B.S. HAS BEEN BREACHED */
cft[ n ].bs_eqn[ end ] = a_pl_cft_surf( n, cft,
( neg * cft[ n ].f2[ 1+i ] - cft[ n ].bs_cent[ end ].p -
  cft[ n ].phi * cft[ n ].bs_rad[ end ] * cft[ n ].po ) /
( cft[ n ].bs_rad[ end ] * cft[ n ].po ),
( cft[ n ].f2[ 5 + i ] - cft[ n ].bs_cent[ end ].my ) /
( cft[ n ].bs_rad[ end ] * cft[ n ].myo ),
( cft[ n ].f2[ 6 + i ] - cft[ n ].bs_cent[ end ].mz ) /
( cft[ n ].bs_rad[ end ] * cft[ n ].mzo ) );

if ( cft[ n ].bs_eqn[ end ] >= 1.0 )
{
  cft[ n ].surf[ end ] = BS;
}

/* MOVE SURFACES USING KINEMATIC AND ISOTROPIC HARDENING */
if ( flag->iso )
{
  a_pl_cft_iso_hard( n, end, cft );
}

a_pl_cft_kin_hard( n, end, tol, flag, cft );

/* PERFORM FORCE POINT DRIFT CONTROL IF NEEDED */
cft[ n ].ls_eqn[ end ] = a_pl_cft_surf( n, cft,
( neg * cft[ n ].f2[ 1+i ] - cft[ n ].ls_cent[ end ].p -
  cft[ n ].phi * cft[ n ].ls_rad[ end ] * cft[ n ].po ) /
( cft[ n ].ls_rad[ end ] * cft[ n ].po ),
( cft[ n ].f2[ 5 + i ] - cft[ n ].ls_cent[ end ].my ) /
( cft[ n ].ls_rad[ end ] * cft[ n ].myo ),
( cft[ n ].f2[ 6 + i ] - cft[ n ].ls_cent[ end ].mz ) /
( cft[ n ].ls_rad[ end ] * cft[ n ].mzo ) );

if ( ( ( cft[ n ].ls_eqn[ end ] < 1.0 ) ||
( cft[ n ].ls_eqn[ end ] > ( 1.0 + tol.surf ) ) ) &&
( flag->drift ) )
{
  a_pl_cft_grad( n, end, F2, cft );
  a_pl_cft_drift( n, end, tol, cft );
}

/* CALCULATE NEW DISTANCE BETWEEN SURFACES */
a_pl_cft_dist( n, end, tol, cft );

/* CALCULATE PLASTIC STIFFNESS TERMS */
a_cft_plastic_k( n, end, cft );
}

} /* if ( ( cft[ n ].state[ end ] == PL ) || ... */

/*****
/* END WAS ELASTIC LAST STEP */
/* I. CHECK IF LOADING SURFACE WAS BREACHED
/* A. YES: END IS NOW PLASTIC
/* 1. SET STATE FLAG TO PL_IN
/* 2. DRIFT CONTROL BACK TO L.S.
/* 3. CALCULATE INITIAL DISTANCE
/* 4. CALCULATE GRADIENT FOR USE IN Kr
/* B. NO: END REMAINS ELASTIC -- DO NOTHING
*****/

else if ( cft[ n ].state[ end ] == EL )
{

```

```

/*****
/* DETERMINE STATE OF FORCE POINT AND SET FLAGS */
*****/

cft[ n ].ls_eqn[ end ] = a_pl_cft_surf( n, cft,
    ( neg * cft[ n ].f2[ 1+i ] - cft[ n ].ls_cent[ end ].p -
      cft[ n ].phi * cft[ n ].ls_rad[ end ] * cft[ n ].po ) /
    ( cft[ n ].ls_rad[ end ] * cft[ n ].po ),
    ( cft[ n ].f2[ 5 + i ] - cft[ n ].ls_cent[ end ].my ) /
    ( cft[ n ].ls_rad[ end ] * cft[ n ].myo ),
    ( cft[ n ].f2[ 6 + i ] - cft[ n ].ls_cent[ end ].mz ) /
    ( cft[ n ].ls_rad[ end ] * cft[ n ].mzo ) );

/*****
/* IF END BECAME PLASTIC THIS STEP */
*****/

if ( cft[ n ].ls_eqn[ end ] >= 1.0 )
    {
        cft[ n ].state[ end ] = PL_IN;
        cft[ n ].hinge          = YES;

        /* PERFORM FORCE POINT DRIFT CONTROL */

        if ( ( ( cft[ n ].ls_eqn[ end ] < 1.0 ) ||
            ( cft[ n ].ls_eqn[ end ] > ( 1.0 + tol.surf ) ) ) &&
            ( flag->drift ) )
            {
                a_pl_cft_drift( n, end, tol, cft );
            }

        /* CALCULATE THE INITIAL DISTANCE BETWEEN SURFACES */

        a_pl_cft_dist( n, end, tol, cft );

        /* CALCULATE PLASTIC STIFFNESS TERMS */

        a_cft_plastic_k( n, end, cft );

        }

    } /* else if ( cft[ n ].state[ end ] == EL ) */

else
    {
        status = 0L;
        printf( "\nError.  Flag not set to elastic or plastic. \n" );
    }

} /* for ( end = 0L; end <= 1L; end++ ) */

} /* for ( n = 1L; n <= size->num_cft_elems; n++ ) */

return( status );
}

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****

* FUNCTION- a_pl_cft_surf.c

    @(#) CFT loading and bounding SURFace equation calculation

* CALLED FROM- a_pl_cft_state, a_pl_cft_kin_hard, a_pl_cft_drift

* ABSTRACT- This function computes and returns the value of the surface

```

```

equation. It receives normalized values which contain the back
force and the radius of the surface (either L.S. or B.S.)
*****/

#include <math.h>          /* C MATH FUNCTIONS          */
#include <stdio.h>         /* C I/O FUNCTIONS         */
#include <string.h>        /* C STRING FUNCTIONS      */

#include "a_mac.h"         /* MACRO DEFINITIONS       */
#include "a_extern.h"     /* EXTERNAL VARIABLES      */

double a_pl_cft_surf( n, cft, p, my, mz )

long      n;              /* CURRENT ELEMENT NUMBER  */
A_CFT_BC  cft[];         /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
double    p;             /* ( P - a - phi * R * Po ) / ( R * Po ) */
double    my;            /* ( My - a ) / ( R * Myo ) */
double    mz;            /* ( Mz - a ) / ( R * Mzo ) */
/* where R = surf rad, a = surf centroid */
{
double equation = 0.0;    /* VALUE OF THE SURFACE EQUATION RETURNED */

/* COMPUTE VALUE OF SURFACE EQUATION FOR CURRENT FORCE POINT */

equation = cft[ n ].c1 * ( my * my + mz * mz ) +
           cft[ n ].c2 * p * p +
           cft[ n ].c3 * ( my * my * p * p + mz * mz * p * p ) +
           cft[ n ].c4 * my * my * mz * mz;

return( equation );
}

```

Table C.2 CFTmacro Header Files

```

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****
*****

* FUNCTION- a_cft_el.h

        @(#) Concrete-Filled Tube ELeMent data structures

* ABSTRACT- This include file contains the data structure definitions
        pertaining to the CFT macro beam-column finite element
*****

#ifndef A_CFT_EL_H
#define A_CFT_EL_H

#include "a_mac.h"

typedef struct
{
    double      p;          /* AXIAL FORCE */
    double      my;        /* BENDING MOMENT ABOUT Y AXIS */
    double      mz;        /* BENDING MOMENT ABOUT Z AXIS */
} A_CFT_FORCES;

typedef struct
{
    /* ELEMENT AND JOINT NUMBERS, FLAGS */
    long        el;        /* ELEMENT NUMBER */
    long        i;         /* I JOINT */
    long        j;         /* J JOINT */
    long        mcode[A_CFT_NUM_DOF+1L]; /* ELEMENT DOF CODE */
    long        release;   /* FLAG TO SIGNAL MEMBER DOF RELEASE
                            (ALL 12 DOFS STORED BY BIT IN 1 long)
                            0L = NO RELEASE
                            1L = RELEASE */
    long        section;  /* FLAG TO SIGNAL AISC SHAPE READ
                            0L = NO (user input)
                            1L = YES (AISC section) */
    long        ei_status; /* FLAG TO SIGNAL EIC IS UPDATED
                            0L = FULL (full Eic)
                            1L = REDUCED (0.311*Eic) */
    long        hinge;   /* FLAG TO SIGNAL IF HINGE HAS FORMED
                            0L = NO (hinge not yet formed)
                            1L = YES (hinge formed) */

    /* SECTION PROPERTIES */
    double      d;        /* TUBE DEPTH */
    double      b;        /* TUBE WIDTH */
    double      t;        /* TUBE THICKNESS */
    double      a_stl;    /* AREA OF THE STEEL TUBE */
    double      a_conc;   /* AREA OF THE CONCRETE CORE */
    double      area;     /* EFFECTIVE AREA OF CFT */
    double      ix;       /* TORSIONAL CONSTANT */
    double      iy_c;     /* Y-AXIS CONC MOMENT OF INERTIA */
    double      iz_c;     /* Z-AXIS CONC MOMENT OF INERTIA */
    double      iy_s;     /* STL Y-AXIS MOMENT OF INERTIA */
    double      iz_s;     /* STL Z-AXIS MOMENT OF INERTIA */
    double      ip;       /* POLAR MOMENT OF INERTIA */
    double      eiy;      /* Y-AXIS BENDING STIFFNESS */
    double      eiz;      /* Z-AXIS BENDING STIFFNESS */
    double      ea;       /* AXIAL STIFFNESS */
    double      gj;       /* TORSIONAL STIFFNESS */
    double      s_y;      /* Y-AXIS SECTION MODULUS */
    double      s_z;      /* Z-AXIS SECTION MODULUS */

    /* LENGTH AND DIRECTION COSINES */
    double      l;        /* ELEMENT LENGTH */
}

```

```

double      l_last;          /* PREVIOUS ITER ELEMENT LENGTH      */
double      iend_x;         /* GLOBAL X COMP. OF I-END VECT      */
double      iend_y;         /* GLOBAL Y COMP. OF I-END VECT      */
double      iend_z;         /* GLOBAL Z COMP. OF I-END VECT      */
double      outp_x;         /* GLOBAL X COMP. OF OUT-OF-PLANE VECT*/
double      outp_y;         /* GLOBAL Y COMP. OF OUT-OF-PLANE VECT*/
double      outp_z;         /* GLOBAL Z COMP. OF OUT-OF-PLANE VECT*/

/* MATERIAL PROPERTIES */
double      fy;             /* YIELD STRENGTH OF THE STEEL      */
double      fc;             /* CONCRETE STRENGTH                */
double      fct;           /* CONCRETE TENSILE STRENGTH        */
double      emod_s;        /* STEEL ELASTIC MODULUS            */
double      gmod_s;        /* STEEL SHEAR MODULUS              */
double      kpi_s;         /* WEIGHT OF STEEL (k/in)           */
double      ec_in;        /* INITIAL CONCRETE ELASTIC MOD     */
double      emod_c;        /* UPDATED CONCRETE ELASTIC MOD     */
double      nu_c;         /* CONCRETE POISSON'S RATIO         */
double      gmod_c;        /* CONCRETE SHEAR MODULUS           */
double      wt_c;         /* WEIGHT OF CONCR. (k/in^3)        */
double      kpi_c;        /* WEIGHT OF CONCR. (k/in)          */
double      kpi;          /* EFFECTIVE MEMBER WEIGHT           */
double      mass;         /* TOTAL MASS OF MEMBER             */
double      w1;           /* DISTRIB DEAD LOAD ON MEMBER      */
double      w2;           /* DISTRIB LIVE LOAD ON MEMBER      */
double      mult[ A_CFT_NUM_DOF + 1L]; /* DISTRIB LOAD MULT*/

/* LOADING AND BOUNDING SURFACE PROPERTIES */
long        ls[ 2L ];     /* LOADING SURFACE TYPE             */
long        bs[ 2L ];     /* BOUNDING SURFACE TYPE            */
long        state[ 2L ];  /* STATE OF PLASTICITY MODEL        */
                                0L = EL (elastic)
                                1L = PL (plastic)
                                2L = PL_IN (init plast.)
long        status[ 2L ]; /* FLAG TO SIGNAL UNLOADING        */
                                0L = LOAD (loading)
                                1L = UNLOAD (unloading)
long        surf[ 2L ];   /* FLAG - SURFACE EQN TO CHECK     */
                                0L = NONE (elastic)
                                1L = LS (loading surf)
                                2L = BS (bounding surf)
                                3L = BOTH (both surfs)
double      ls_rad[ 2L ]; /* RADIUS OF LOADING SURFACE        */
double      bs_rad[ 2L ]; /* RADIUS OF BOUNDING SURFACE       */
double      ls_eqn[ 2L ]; /* VALUE OF L.S. EQUATION           */
double      bs_eqn[ 2L ]; /* VALUE OF B.S. EQUATION           */
double      rho[ 2L ];    /* RATIO: bs_rad/ls_rad             */
double      dist[ 2L ];   /* DISTANCE BETWEEN SURFACES       */
double      dist_in[ 2L ]; /* INITIAL DISTANCE                 */
double      norm[ 2L ];   /* NORMAL TO LOADING SURFACE        */
A_CFT_FORCES  ls_cent[ 2L ]; /* CENTROID OF L.S.                */
A_CFT_FORCES  bs_cent[ 2L ]; /* CENTROID OF B.S.                */
A_CFT_FORCES  conj[ 2L ]; /* CONJUGATE FORCE POINT ON B.S.    */
A_CFT_FORCES  grad[ 2L ]; /* GRADIENT VECTOR OF FORCE PT.     */

/* KINEMATIC HARDENING PARAMETERS */
A_CFT_FORCES  mroz[ 2L ]; /* DIRECTION OF MROZ VECTOR         */
A_CFT_FORCES  tseng[ 2L ]; /* DIRECTION OF TSENG VECTOR        */
A_CFT_FORCES  dls_kin[ 2L ]; /* LS RAD CHANGE DUE TO KIN HARD    */
A_CFT_FORCES  dbs_kin[ 2L ]; /* BS RAD CHANGE DUE TO KIN HARD    */
A_CFT_FORCES  k1[ 2L ];   /* HARDENING COEFFICIENT            */
A_CFT_FORCES  k2[ 2L ];   /* UPDATED HARDENING COEFFICIENT    */
A_CFT_FORCES  k2i[ 2L ];  /* UPDATED HARDENING COEFFICIENT    */
A_CFT_FORCES  k2f[ 2L ];  /* UPDATED HARDENING COEFFICIENT    */
A_CFT_FORCES  kp[ 2L ];   /* PLASTIC STIFFNESS                */
A_CFT_FORCES  kpb[ 2L ];  /* BOUNDING PLASTIC STIFFNESS       */

/* ISOTROPIC HARDENING PARAMETERS */
double      el_work;      /* MEMBER STRAIN ENERGY            */
double      pl_work[ 2L ]; /* ACCUMULATED PLASTIC WORK         */
double      lsr_in[ 2L ]; /* INITIAL RADIUS OF L.S.           */
double      lsr_fn[ 2L ]; /* FINAL (MAX OR MIN) LS RAD       */
double      bsr_in[ 2L ]; /* INITIAL RADIUS OF B.S.           */
double      bsr_intm[ 2L ]; /* PEAK OF B.S. RAD CURVE          */

```

```

double      bsr_fn[ 2L ]; /* FINAL (MAX OR MIN) BS RAD          */
double      bsr_upd[ 2L ]; /* INITIAL B.S. RAD FOR DESCENT */
double      ls_iso[ 2L ]; /* L.S. SOFTENING PARAMETER     */
double      bs_iso[ 2L ]; /* B.S. SOFTENING PARAMETER     */
double      k2_iso[ 2L ]; /* k2 SOFTENING PARAMETER       */
double      ec_iso[ 2L ]; /* RATE OF CONCR. EMOD DECREASE */
double      iso_work[ 2L ]; /* % OF ELASTIC WORK AT INTM BS  */
A_CFT_FORCES dls_iso[ 2L ]; /* LS RAD CHANGE DUE TO ISO HARD */
A_CFT_FORCES dbs_iso[ 2L ]; /* BS RAD CHANGE DUE TO ISO HARD */

/* PROPERTIES FOR SURFACE EQN AND PLASTIC LENGTH CALCS */
double      c1; /* EQUATION COEFFICIENT */
double      c2; /* EQUATION COEFFICIENT */
double      c3; /* EQUATION COEFFICIENT */
double      c4; /* EQUATION COEFFICIENT */
double      phi; /* CENTROID OF LOADING SURFACE */
double      po; /* NOMINAL AXIAL LOAD CAPACITY */
double      ptens; /* NOMINAL TENSILE CAPACITY */
double      myy; /* Y-AXIS YIELD MOMENT */
double      mzy; /* Z-AXIS YIELD MOMENT */
double      myo; /* NOMINAL Y-MOMENT CAPACITY */
double      mzo; /* NOMINAL Z-MOMENT CAPACITY */

/* MEMBER DISPLACEMENTS */
double      du[A_CFT_NUM_DOF+1L]; /* INCR DISPLS */
double      u2[A_CFT_NUM_DOF+1L]; /* END-OF-STEP DISPLS */
double      rz_y[ 2L ]; /* MAJOR AXIS 'YIELD' ROTATION */
double      duct[ 2L ]; /* ELEMENT END DUCTILITY RATIO */

/* MEMBER FORCES AND DISPLACEMENTS */
double      sratio[ 2L ]; /* STRESS RATIOS */
double      df_i[A_CFT_NUM_DOF+1L]; /* ITERATION FORCES */
double      f1_i[A_CFT_NUM_DOF+1L]; /* BEG-OF-ITER FORCES */
double      f2_i[A_CFT_NUM_DOF+1L]; /* END-OF-ITER FORCES */
double      f1[A_CFT_NUM_DOF+1L]; /* FORCES AT STEP n */
double      f2[A_CFT_NUM_DOF+1L]; /* FORCES AT STEP n + 1 */

/* MAXIMUM AND MINIMUM VALUES */
double      sratio_max[ 2L ]; /* MAX STRESS RATIO */
double      t_sratio[ 2L ]; /* TIME OF MAX STRESS */
double      duct_max[ 2L ]; /* MAX DUCTILITY RATIO */
double      t_duct[ 2L ]; /* TIME OF MAX DUCT */
double      f_max[A_CFT_NUM_DOF+1L]; /* MAX FORCES */
double      f_min[A_CFT_NUM_DOF+1L]; /* MIN FORCES */
double      t_max[A_CFT_NUM_DOF+1L]; /* TIME STEP OF MAX F'S */
double      t_min[A_CFT_NUM_DOF+1L]; /* TIME STEP OF MIN F'S */

/* TRANSFORMATIN MATRIX */
double      lambda[A_CFT_NUM_DOF+1L][A_CFT_NUM_DOF+1L];
/* GLOBAL TO LOCAL TRANSFORM */

/* ELEMENT STIFFNESS MATRICES */
double      kt[ A_CFT_NUM_DOF + 1L ][ A_CFT_NUM_DOF + 1L ];
/* LOCAL ELEMENT TANGENT K */
double      kr[ A_CFT_NUM_DOF + 1L ][ A_CFT_NUM_DOF + 1L ];
/* LOCAL PLASTIC REDUCTION K */

}      A_CFT_BC; /* CFT BEAM-COLUMN ELEMENT */

#endif

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****

* FUNCTION- a_extern.h

*      @(#) Generic EXTERNAL variables

* ABSTRACT- Declaration of generic external variables

```

```

*****/

#ifndef A_EXTERN_H
#define A_EXTERN_H

#include <stdio.h>          /* C I/O FUNCTIONS          */
#include "a_mac.h"         /* MACRO DEFINITIONS       */
#include "a_jt.h"          /* DATA STRUCTURE DEFINING JOINT PROPS */
#include "a_cft_el.h"      /* DATA STRUCTURE FOR CFT BEAM-COLUMN */
#include "a_stl_el.h"     /* DATA STRUCTURE FOR STEEL BEAM-COLUMN */
#include "a_param.h"       /* DATA STRUCTURE OF VARIOUS PARAMETERS */
#include "a_global.h"     /* DATA STRUCTURE OF DOF-BASED VARIABLES */
#include "aisc.h"         /* DATA STRUCTURE FOR AISC DATABASE */
#include "aiscpriv.h"     /* AISC PRIVATE INCLUDE FILE--AISC MACROS*/

#ifndef EXTERN
#define EXTERN extern
#endif

/* FILE POINTERS */

EXTERN FILE* A_fp_in;          /* INPUT FILE POINTER      */
EXTERN FILE* A_fp_out;        /* OUTPUT FILE POINTER     */
EXTERN FILE* A_fp_plot;       /* PLOT FILE POINTER       */
EXTERN FILE* A_fp_max;        /* MAXIMUM VALUE FILE POINTER */
EXTERN FILE* A_fp_accl;       /* ACCELEROGRAM INPUT POINTER */

/* CHARACTER STRINGS */

EXTERN char A_in_file [ A_FILE_NAME_LEN ]; /* INPUT FILENAME */
EXTERN char A_out_file[ A_FILE_NAME_LEN ]; /* OUTPUT FILENAME */
EXTERN char A_plot_file[ A_FILE_NAME_LEN ]; /* PLOT FILENAME */
EXTERN char A_max_file[ A_FILE_NAME_LEN ]; /* MAX VALUE FILENAME */
EXTERN char A_accel_file [ A_FILE_NAME_LEN ]; /* ACCEL INPUT FNAME */

#endif

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****
*/

* FUNCTION- a_global.h

* @(#) data structure of GLOBAL degree-of-freedom-based variables

* ABSTRACT- This include file contains the data structure containing
* global arrays having size equal to the number of degrees-of-
* freedom in the structure. These include global forces,
* displacements, mass, damping, and others.
*****/

#ifndef A_GLOBAL_H
#define A_GLOBAL_H

typedef struct { /* 'global' */
    long col_ht; /* GLOBAL K MATRIX COLUMN HEIGHTS */

    /* LOADS */
    double dr; /* INCREMENTAL GLOBAL LOAD VECTOR */
    double r1; /* BEGINNING OF STEP LOAD VECTOR */
    double r2; /* END OF STEP LOAD VECTOR */
    double dyn; /* DYNAMIC LOAD (ma + NEWMARK EFFECTS) */

    /* DISPLACEMENTS */
    double dqi; /* GLOBAL ITERATIVE DISPLACEMENTS */
    double dq1; /* GLOBAL 1ST ITERATION DISPLACEMENTS */
    double q1; /* BEGINNING OF STEP DISPLACEMENTS */
    double q2; /* END OF STEP DISPLACEMENTS */
    double q2i; /* END OF ITERATION DISPLACEMENT */

    /* MASS AND DAMPING */

```

```

double mass;          /* GLOBAL MASS MATRIX (DIAGONAL)          */
double cv1;           /* PRODUCT OF DAMPING MATRIX AND v1          */
double cvli;          /* PRODUCT OF DAMPING MATRIX AND vli         */

/* ACCELERATIONS AND VELOCITIES */
double a1;            /* BEGINNING OF STEP ACCELERATION           */
double a1_i;          /* BEGINNING OF ITERATION ACCELERATION      */
double a2;            /* END OF STEP ACCELERATION                  */
double v1;            /* BEGINNING OF STEP VELOCITY               */
double v1_i;          /* BEGINNING OF ITERATION VELOCITY          */
double v2;            /* END OF STEP VELOCITY                     */

}      A_GLOBAL;      /* GLOBAL STRUCTURAL PARAMETERS */

#endif

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****

*      FUNCTION- a_jt.h

          @(#) Joint data structures

*      ABSTRACT- This include file contains the data structures pertaining to
          the joints
*****/

#ifndef A_JT_H
#define A_JT_H

#include "a_mac.h"          /* MACRO DEFINITIONS          */

typedef struct {
double      x;              /* X COORDINATE                */
double      y;              /* Y COORDINATE                */
double      z;              /* Z COORDINATE                */
}      A_JT_COORD;          /* GLOBAL CARTESIAN COORDINATES */

typedef struct {
long      jt;                /* JOINT NUMBER                 */
long      jcode[ A_JT_DOF + 1L ]; /* DOFS MAPPED ONTO LOCAL JT   */
long      rf[ A_JT_DOF + 1L ]; /* VECTOR OF JOINT RESTRAINTS  */
/* 0 = restrained, 1 = free          */

/* LOADS */
long      lhist[ A_JT_DOF + 1L ]; /* ASSIGNED LOAD HISTORY NUM */
double    load[ A_JT_DOF + 1L ]; /* TOTAL JOINT LOAD            */
double    load1[ A_JT_DOF + 1L ]; /* TOTAL JOINT LOAD (LHIST 1) */
double    load2[ A_JT_DOF + 1L ]; /* TOTAL JOINT LOAD (LHIST 2) */
double    r2[ A_JT_DOF + 1L ]; /* TOTAL END OF STEP LOAD      */
double    ma[ A_JT_DOF + 1L ]; /* GROUND ACCELERATION LOAD    */
double    dyn[ A_JT_DOF + 1L ]; /* JOINT LOAD DUE TO DYNAMICS  */

/* FORCES */
double    df_i[ A_JT_DOF + 1L ]; /* INCREMENTAL JOINT FORCE     */
double    f2_i[ A_JT_DOF + 1L ]; /* JOINT FORCE AT INCR n + 1   */
double    f1[ A_JT_DOF + 1L ]; /* JOINT FORCE AT STEP n       */
double    f2[ A_JT_DOF + 1L ]; /* JOINT FORCE AT STEP n + 1   */

/* MASS AND DAMPING */
double    mass[ A_JT_DOF + 1L ]; /* JOINT LUMPED MASS           */
double    damp[ A_JT_DOF + 1L ]; /* JOINT LUMPED DAMPING        */

/* MAX AND MIN DISPLACEMENTS */
double    u_max[ A_JT_DOF + 1L ]; /* MAXIMUM DISPLACEMENT       */
double    u_min[ A_JT_DOF + 1L ]; /* MINIMUM DISPLACEMENT       */
double    t_max[ A_JT_DOF + 1L ]; /* TIME OF MAXIMUM DISPL       */
double    t_min[ A_JT_DOF + 1L ]; /* TIME OF MINIMUM DISPL       */

/* COORDINATES */

```

```

        A_JT_COORD    co;                /* JOINT COORDINATES          */
        A_JT_COORD    co_i;             /* INITIAL JOINT COORDINATES */
    }        A_JT_JOINT;    /* JOINT DATA STRUCTURE */

#endif

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****

* FUNCTION- a_mac.h

        @(#) MACro definitions

* ABSTRACT- This include files includes all the macros in the macro
        model program
*****/

#ifndef A_MAC_H
#define A_MAC_H

/*****/
/* DEFINE JOINT AND ELEMENT DOFS */
/*****/

#define A_JT_DOF            6L
#define A_CFT_NUM_DOF_END  6L
#define A_CFT_NUM_DOF      12L
#define A_STL_NUM_DOF_END  6L
#define A_STL_NUM_DOF      12L

/*****/
/* DEFINE SIZE OF DATA ARRAYS */
/*****/

#define A_NUM_ACCLG_PTS    4200L
#define A_NUM_LH_PTS      50L

/*****/
/* DEFINE MISCELLANEOUS MACROS */
/*****/

#define PI                3.1415926536
#define POS                1L
#define NEG                -1L
#define NO                 0L
#define YES                1L
#define STATIC             1L
#define DYNAMIC            2L
#define EIGEN              3L
#define LOWORDER           0L
#define HIGHORDER          1L
#define SMALL              0.000000001

/*****/
/* CHARACTER STRING MACROS */
/*****/

#define A_FILE_NAME_LEN    65L
/* LENGTH OF FILE NAMES          */
#define A_KEYWORD_LEN      16L
/* LENGTH OF KEYWORDS            */
#define A_DESIG_LEN        20L
/* LENGTH OF SHAPE DESIGNATION  */

/*****/
/* CONSTANT ACCELERATION MACROS */
/*****/

#define A_BETA              0.25

```

```

#define A_GAMMA                0.5

/*****
/* CONCRETE STIFFNESS MACROS */
*****/

/* STATUS OF THE CONCRETE STIFFNESS (FULL OR REDUCED DUE TO CRACKING) */
#define FULL                    0L
#define REDUCED                  1L

/*****
/* PLASTICITY FLAG MACROS */
*****/

/* TYPE OF KINEMATIC HARDENING: MROZ OR TSENG */
#define MROZ                     1L
#define TSENG                     2L

/* STATE: ELASTIC, PLASTIC, OR PLASTIC FOR FIRST STEP */
#define EL                        0L
#define PL                        1L
#define PL_IN                      2L

/* HINGE STATUS:  LOADING OR UNLOADING */
#define LOAD                      0L
#define UNLOAD                    1L

/* SURFACE FORCE PT LIES ON:  NONE, LOADING SURF, BOUNDING SURF, OR BOTH */
#define NONE                      0L
#define LS                        1L
#define BS                        2L
#define BOTH                      3L

/* FORCES TO USE:  BEGINNING OF STEP FORCES, END OF STEP FORCES */
#define F1                        1L
#define F2                        2L

#endif

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****
*****

* FUNCTION- a_param.h

      @(#) various structural PARAMETERS

* ABSTRACT- This include file contains the data structure for the
      the size of the structural model, the global stiffness matrix,
      time, and tolerances.
*****

#ifndef A_PARAM_H
#define A_PARAM_H

#include "a_mac.h"

typedef struct          {          /* 'accel' */
    long   num_pts;          /* NUMBER OF ACCELEROGRAM POINTS          */
    long   curve;           /* CURRENT PORTION OF ACCELEROGRAM          */
    double ag;              /* CURRENT GLOBAL BASE ACCELERATION          */
    double angle;          /* ORIENTATION OF EQKE LOADING          */
    double a_pt[ A_NUM_ACCLG_PTS + 1L ]; /* ACCLGRM ACCEL COORDS          */
    double t_pt[ A_NUM_ACCLG_PTS + 1L ]; /* ACCLGRM TIME COORDS          */
    }          A_ACCEL;          /* ACCELEROGRAM DATA          */

typedef struct {          /* 'dyn' */
    long   d_option;          /* INPUT DAMPING OPTION          */
    double a[ 8L ];          /* VECTOR OF INTEGRATION CONSTANTS          */
    double damp;             /* DAMPING RATIO          */
    double dcoeff_k;         /* STIFFNESS PROPORTIONAL DAMPING COEF          */
}

```

```

double dcoeff_m; /* MASS PROPORTIONAL DAMPING COEFF. */
} A_DYNAMIC; /* DYNAMIC ANALYSIS DATA */

typedef struct { /* 'eigen' */
double freq; /* MODAL FREQUENCY */
double period; /* MODAL PERIOD */
} A_EIGEN; /* EIGENANALYSIS DATA */

typedef struct { /* 'flag' */
long analysis; /* TYPE OF ANALYSIS (1 = STATIC,
2 = DYNAMIC, 3 = EIGEN) */
long test_num; /* CODE FOR 'cftmacro.plot' OUTPUT */
long elem_wt; /* ADD ELEMENT WEIGHTS AS JOINT LOADS? */
long stl_kin; /* STEEL KINEMATIC HARDENING METHOD */
long cft_kin; /* CFT KINEMATIC HARDENING METHOD */
long kg; /* INCLUDE GEOMETRIC STIFFNESS? */
long order; /* ORDER OF Kg (0 = LOW, 1 = HIGH) */
long kp; /* INCLUDE MATERIAL NONLINEARITIES? */
long scaleback; /* SCALE BACK CURRENT STEP */
long iso; /* INCLUDE ISOTROPIC HARDENING? */
long drift; /* PERFORM FORCE PT DRIFT CONTROL? */
long echo_input; /* ECHO SELECTED INPUT PARAMETERS? */
long suppress; /* SUPPRESS OUTPUT OF SELECT PARAMS? */
long maxmin; /* PRINT MAX AND MIN F'S AND DISPLS? */
} A_FLAGS; /* PROGRAM FLAGS */

typedef struct { /* 'lhist' */
long num_lhpts; /* NUMBER OF LOAD HISTORY POINTS */
long curve; /* CURRENT SEGMENT OF LOAD-HIST CURVE */
double ldfrac1; /* % TOTAL APPLIED LOAD LAST STEP */
double ldfrac2; /* % TOTAL APPLIED LOAD CURRENT STEP */
double time_pt[ A_NUM_LH_PTS ]; /* LOAD HISTORY TIME POINTS */
double load_pt[ A_NUM_LH_PTS ]; /* LOAD HISTORY LOAD POINTS */
} A_LOADHIST; /* LOAD HISTORY PARAMETERS */

typedef struct { /* 'nr' */
long conv; /* FLAG TO SIGNAL CONVERGENCE
0L = SOLUTION HAS NOT CONVERGED
1L = SOLUTION CONVERGED */
long iter_ct; /* CURRENT ITERATION NUMBER */
long iter_max; /* MAXIMUM NUMBER OF ITERATIONS */
double dq_last; /* TOTAL INCR DISPL FOR LAST ITERATION */
double dq_curr; /* TOTAL INCR DISPL FOR CURRENT ITER. */
} A_NEW_RAPH; /* NEWTON-RAPHSON PARAMETERS */

typedef struct { /* 'size' */
long num_jts; /* NUMBER OF JOINTS IN THE STRUCTURE */
long num_dofs; /* NUMBER OF GLOBAL DEGREES-OF-FREEDOM */
long num_elems; /* NUMBER OF ELEMENTS IN THE MODEL */
long num_cft_elems; /* NUMBER OF CFT MACRO ELEMENTS */
long num_stl_elems; /* NUMBER OF STEEL MACRO ELEMENTS */
long num_jt_loads; /* NUMBER OF JOINT LOADS */
long num_lhs; /* NUMBER OF LOAD HISTORIES */
long max_lhpts; /* MAX NUMBER OF LOAD HISTORY POINTS */
long num_acclgms; /* NUMBER OF ACCELEROGRAMS */
long max_acclpts; /* MAX NUMBER OF ACCELEROGRAM POINTS */
long num_sky; /* NUMBER OF SKYLINE K TERMS */
long num_modes; /* NUMBER OF MODES DESIRED */
} A_MODEL_SIZE; /* SIZE OF STRUCTURAL MODEL */

typedef struct { /* 'time' */
long prstep; /* PRINT RESULTS EVERY __ STEPS */
long count; /* OUTPUT COUNTER */
double a_step; /* SCALED BACK (APPLIED) TIME STEP */
double x_step; /* UNAPPLIED PORTION OF INPUT T STEP */
double step; /* INPUT TIME STEP */
double total; /* TOTAL AMOUNT OF TIME */
} A_TIME; /* TIME PARAMETERS */

typedef struct { /* 'tol' */
double conv; /* NEWTON-RAPHSON CONVERGENCE TOL. */
double eigen; /* EIGENSOLVER CONVERGENCE TOL. */
double surf; /* LOADING AND BOUNDING SURFACE TOLS. */
} A_TOLERANCE; /* PROGRAM TOLERANCES */

```

#endif

```

/*****
*****
* DEPARTMENT OF CIVIL ENGINEERING - UNIVERSITY OF MINNESOTA *
*****
*/

* FUNCTION- a_stl_el.h

        @(#) Steel Element data structures

* ABSTRACT- This include files contains the data structure definitions
        pertaining to the steel macro beam-column finite element
*****/

#ifndef A_STL_EL_H
#define A_STL_EL_H

#include "a_mac.h"

typedef struct {
    double    p;           /* AXIAL FORCE */
    double    my;          /* BENDING MOMENT ABOUT Y AXIS */
    double    mz;          /* BENDING MOMENT ABOUT Z AXIS */
} A_STL_FORCES;

typedef struct {
    /* ELEMENT AND JOINT NUMBERS */
    long      el;          /* ELEMENT NUMBER */
    long      i;           /* I JOINT */
    long      j;           /* J JOINT */
    long      mcode[A_STL_NUM_DOF+1L]; /* ELEMENT DOF CODE */
    long      release;     /* FLAG TO SIGNAL MEMBER RELEASE
                                0L = RELEASE
                                1L = RELEASE */

    /* SECTION PROPERTIES */
    double    area;        /* GROSS AREA OF STEEL SECTION */
    double    i_x;         /* TORSIONAL CONSTANT */
    double    i_y;         /* Y-AXIS MOMENT OF INERTIA */
    double    i_z;         /* Z-AXIS MOMENT OF INERTIA */
    double    i_p;         /* POLAR MOMENT OF INERTIA */
    double    s_y;         /* Y-AXIS SECTION MODULUS */
    double    s_z;         /* Z-AXIS SECTION MODULUS */
    double    z_y;         /* Y-AXIS PLASTIC MODULUS */
    double    z_z;         /* Z-AXIS PLASTIC MODULUS */

    /* LENGTH AND DIRECTION COSINES */
    double    l;           /* ELEMENT LENGTH */
    double    l_last;     /* PREVIOUS ITER ELEMENT LENGTH */
    double    iend_x;      /* GLOBAL X COMP. OF I-END VECT */
    double    iend_y;      /* GLOBAL Y COMP. OF I-END VECT */
    double    iend_z;      /* GLOBAL Z COMP. OF I-END VECT */
    double    outp_x;      /* GLOBAL X COMP. OF OUT-OF-PLANE VECT*/
    double    outp_y;      /* GLOBAL Y COMP. OF OUT-OF-PLANE VECT*/
    double    outp_z;      /* GLOBAL Z COMP. OF OUT-OF-PLANE VECT*/

    /* MATERIAL PROPERTIES */
    double    fy;          /* YIELD STRENGTH OF THE STEEL */
    double    emod;        /* ELASTIC MODULUS */
    double    gmod;        /* SHEAR MODULUS */
    double    kpi;         /* WEIGHT OF STEEL MEMBER */
    double    mass;        /* TOTAL MASS OF MEMBER */
    double    w1;          /* DISTRIB DEAD LOAD ON MEMBER */
    double    w2;          /* DISTRIB LIVE LOAD ON MEMBER */
    double    mult[ A_STL_NUM_DOF + 1L]; /* DISTRIB LOAD MULT*/

    /* PLASTICITY PARAMETERS */

```

```

long      ls[ 2L ];      /* LOADING SURFACE TYPE          */
long      bs[ 2L ];      /* BOUNDING SURFACE TYPE        */
long      state[ 2L ];   /* STATE OF PLASTICITY MODEL
                        0L = EL (elastic)
                        1L = PL (plastic)
                        2L = PL_IN (init plast.) */
long      status[ 2L ]; /* FLAG TO SIGNAL UNLOADING
                        0L = LOAD (loading)
                        1L = UNLOAD (unloading) */
long      surf[ 2L ];    /* FLAG - SURFACE EQN TO CHECK
                        0L = NONE (elastic)
                        1L = LS (loading surf)
                        2L = BS (bounding surf)
                        3L = BOTH (both surfs) */
double    ls_rad[ 2L ]; /* RADIUS OF LOADING SURFACE    */
double    bs_rad[ 2L ]; /* RADIUS OF BOUNDING SURFACE   */
double    ls_eqn[ 2L ]; /* VALUE OF L.S. EQUATION      */
double    bs_eqn[ 2L ]; /* VALUE OF B.S. EQUATION      */
double    rho[ 2L ];    /* RATIO: bs_rad/ls_rad        */
double    dist[ 2L ];    /* DISTANCE BETWEEN SURFACES   */
double    dist_in[ 2L ]; /* INITIAL DISTANCE            */
double    norm[ 2L ];    /* NORMAL TO LOADING SURFACE    */
A_STL_FORCES ls_cent[ 2L ]; /* CENTROID OF L.S.           */
A_STL_FORCES bs_cent[ 2L ]; /* CENTROID OF B.S.           */
A_STL_FORCES con_j[ 2L ]; /* CONJUGATE FORCE POINT ON B.S. */
A_STL_FORCES grad[ 2L ]; /* GRADIENT VECTOR OF FORCE PT. */

/* KINEMATIC HARDENING PARAMETERS */
A_STL_FORCES mroz[ 2L ]; /* DIRECTION OF MROZ VECTOR    */
A_STL_FORCES tseng[ 2L ]; /* DIRECTION OF TSENG VECTOR   */
A_STL_FORCES dls_kin[ 2L ]; /* LS RAD CHANGE DUE TO KIN HARD */
A_STL_FORCES dbs_kin[ 2L ]; /* BS RAD CHANGE DUE TO KIN HARD */
A_STL_FORCES k1[ 2L ]; /* HARDENING COEFFICIENT      */
A_STL_FORCES k2[ 2L ]; /* HARDENING COEFFICIENT      */
A_STL_FORCES kp[ 2L ]; /* PLASTIC STIFFNESS          */
A_STL_FORCES kpb[ 2L ]; /* BOUNDING PLASTIC STIFFNESS  */

/* NOMINAL STRENGTHS */
double    po;           /* NOMINAL AXIAL LOAD CAPACITY */
double    myy;          /* Y-AXIS YIELD MOMENT         */
double    mzy;          /* Z-AXIS YIELD MOMENT         */
double    myo;          /* NOMINAL Y-MOMENT CAPACITY   */
double    mzo;          /* NOMINAL Z-MOMENT CAPACITY   */

/* MEMBER DISPLACEMENTS */
double    du[ A_STL_NUM_DOF + 1L ]; /* INCR DISPLS                */
double    u2[ A_STL_NUM_DOF + 1L ]; /* END-OF-STEP DISPLS         */
double    rz_y[ 2L ]; /* MAJOR AXIS 'YIELD' ROTATION */
double    duct[ 2L ]; /* ELEMENT END DUCTILITY RATIO */

/* MEMBER FORCES */
double    sratio[ 2L ]; /* STRESS RATIOS                */
double    df_i[A_STL_NUM_DOF+1L]; /* ITERATION FORCES            */
double    f1_i[A_STL_NUM_DOF+1L]; /* BEG-OF-ITER FORCES         */
double    f2_i[A_STL_NUM_DOF+1L]; /* END-OF-ITER FORCES         */
double    f1[A_STL_NUM_DOF+1L]; /* FORCES AT STEP n           */
double    f2[A_STL_NUM_DOF+1L]; /* FORCES AT STEP n + 1      */

/* MAXIMUM AND MINIMUM VALUES */
double    sratio_max[ 2L ]; /* MAX STRESS RATIO            */
double    t_sratio[ 2L ]; /* TIME OF MAX STRESS          */
double    duct_max[ 2L ]; /* MAX DUCTILITY RATIO         */
double    t_duct[ 2L ]; /* TIME OF MAX DUCT            */
double    f_max[A_STL_NUM_DOF+1L]; /* MAX FORCES                  */
double    f_min[A_STL_NUM_DOF+1L]; /* MIN FORCES                  */
double    t_max[A_STL_NUM_DOF+1L]; /* TIME STEP OF MAX F'S        */
double    t_min[A_STL_NUM_DOF+1L]; /* TIME STEP OF MIN F'S        */

/* TRANSFORMATION MATRIX */
double    lambda[ A_STL_NUM_DOF + 1L ][ A_STL_NUM_DOF + 1L ]; /* GLOBAL TO LOCAL TRANSFORM */

/* ELEMENT STIFFNESS MATRICES */
double    kt[ A_STL_NUM_DOF + 1L ][ A_STL_NUM_DOF + 1L ];

```

```

                                /* LOCAL ELEMENT TANGENT K          */
double      kr[ A_STL_NUM_DOF + 1L ][ A_STL_NUM_DOF + 1L ];
                                /* LOCAL PLASTIC REDUCTION K      */
                                A_STL_BC;      /* STEEL BEAM-COLUMN ELEMENT */
#endif
```

Appendix D

List of Symbols

- $a_{BS}^p, a_{BS}^{my}, a_{BS}^{mz}$ = coordinates of bounding surface backforce vector in force-space
 $\{a_{BS}\}$ = unnormalized bounding surface backforce vector
 $\{A_{BS}\}$ = unnormalized bounding surface centroidal vector
 a_i = constant coefficients for quadratic form of cross-section strength surface equation
 $a_{LS}^p, a_{LS}^{my}, a_{LS}^{mz}$ = coordinates of loading surface backforce vector in force-space
 $\{a_{LS}\}$ = unnormalized loading surface backforce vector
 $\{A_{LS}\}$ = unnormalized loading surface centroidal vector
 A_c = area of concrete
 A_i = area of fiber element i
 A_s = area of steel
 A_t = total area of CFT
 b = width of rectangular CFT
 b_i = coefficients for cubic form of cross-section strength surface equation
 c_i = coefficients of cross-section strength surface equation
CFT = concrete-filled steel tube
 $\{da_{LS}\}, \{da_{BS}\}$ = incremental loading and bounding surface centroid movement
 $\{da_{LS}\}_{iso}, \{da_{BS}\}_{iso}$ = incremental surface centroid movement due to isotropic hardening
 $\{da_{LS}\}_{kin}, \{da_{BS}\}_{kin}$ = incr. surface centroid movement due to kinematic hardening
 $\{dq\}_i, \{dq\}_j, \{dq\}$ = i -end, j -end, and total incremental element displacement vectors
 $\{dq_e\}$ = incremental elastic element displacements
 $\{dq_p\}$ = incremental plastic element displacements
 dR_{LS}, dR_{BS} = incremental change in loading surface and bounding surface radii

dW_p = incremental plastic work
 $\{dS\}_i, \{dS\}_j, \{dS\}$ = i -end, j -end, and total incremental element force vectors
 $\{dS^*\}$ = incr. force vector relating plastic stiffness with plastic displacement vector
 $d\lambda_i, d\lambda_j$ = magnitude of incremental plastic displacement at element ends i and j
 $\{d\lambda\}$ = vector of incremental plastic displacement magnitudes
 D = depth of rectangular CFT
 DOF = degree-of-freedom
 e = eccentricity of applied load
 e_y = eccentricity of applied load in y -direction
 e_z = eccentricity of applied load in z -direction
 $E \cdot A$ = axial rigidity
 $(E \cdot A)_{cft}$ = effective axial rigidity of CFT section
 E_c = concrete modulus of elasticity
 $(E_c)_{new}$ = updated value of concrete modulus of elasticity
 $E \cdot I$ = flexural rigidity
 $(E \cdot I)_{cft}$ = effective flexural rigidity of CFT section
 E_s = steel modulus of elasticity
 E_{sh} = steel strain hardening modulus
 f = normalized loading or bounding surface function
 f_c = concrete stress (compression)
 f'_c = characteristic 28-day concrete cylinder strength
 f_{cl} = final concrete compressive stress
 f_{ct} = concrete stress (tension)
 f_r = concrete rupture (tensile) strength
 f_s = longitudinal stress in steel tube
 f_u = ultimate strength of steel tube
 f_y = yield strength of steel tube
 F = total internal cross-section force; transverse force
 $\{F\}$ = conjugate vector to $\{F'\}$ on loading surface
 $\{F'\}$ = vector at intersection of bounding surface and incremental force vector extension
 $G \cdot J$ = torsional rigidity
 $(G \cdot J)_{cft}$ = torsional rigidity of CFT section
 G_s = shear modulus of elasticity of steel
 i = cross-section fiber element; finite element end
 I_c = moment of inertia of concrete
 I_s = moment of inertia of steel
 j = finite element end
 J_s = torsional moment of inertia of steel

$[k^e]$ = total element stiffness matrix
 $[k_e^e]$ = element elastic stiffness matrix
 $(k_e)_p, (k_e)_{my}, (k_e)_{mz}$ = elastic stiffness terms--axial, minor and major axis bending
 $[k_g^e]$ = element geometric stiffness matrix
 $[k_p^e]$ = element plastic stiffness matrix
 $(k_p)_p, (k_p)_{my}, (k_p)_{mz}$ = plastic stiffness terms--axial, minor and major axis bending
 $(k_p^b)_p, (k_p^b)_{my}, (k_p^b)_{mz}$ = bounding stiffness terms--axial, minor and major axis bending
 $[k_r^e]$ = element plastic reduction stiffness matrix
 $[k_T^e]$ = element elastic tangent stiffness matrix
L = member length; element length
 m_y = normalized minor axis (y-direction) bending moment, M_y/M_{y0}
 m_z = normalized major axis (z-direction) bending moment, M_z/M_{z0}
M = resultant bending moment
 M_o = ultimate bending moment in presence of no axial load
 M_y = minor axis (y-direction) bending moment
 M_{yld} = yield moment
 M_{y0} = ultimate minor axis bending moment in presence of no axial load
 M_z = major axis (z-direction) bending moment
 M_{z0} = ultimate major axis bending moment in presence of no axial load
n = parameter for ascending branch of concrete stress-strain curve
 n_i = i^{th} exponent of cross-section strength surface equation
 $\{n\}_i, \{n\}_j$ = gradient vectors at i - and j -end of element
N = parameter for strain hardening curve of steel stress-strain curve
 $\{N\}$ = element gradient vector
NPr = non-proportional applied loading
p = normalized axial load
P = applied axial load; axial force
 P_{co} = ultimate axial strength of concrete core of CFT
 P_e = Euler buckling load
 P_{eq} = axial strength of CFT in tension using cross-section strength surface equation
 P_o = ultimate axial load in the presence of no bending
Pr = proportional applied loading
 P_{so} = ultimate axial strength of steel tube of CFT
 P_{tens} = axial strength of CFT in tension
Q = applied shear
 Q_{max}, Q_{max}^i = maximum cyclic shears
R = rotation

R_{BS} = bounding surface "radius"
 $(R_{BS})_{init}$, $(R_{BS})_{int}$, $(R_{BS})_{final}$ = initial, intermediate, and final bounding surface radii
 $(R_{BS})_{new}$, $(R_{BS})_{old}$ = updated and previous step bounding surface radii
 R_{LS} = loading surface "radius"
 $(R_{LS})_{init}$, $(R_{LS})_{final}$ = initial and final loading surface radii
 $(R_{LS})_{new}$, $(R_{LS})_{old}$ = updated and previous step loading surface radii
 $\{S\}$ = normalized force vector
 S = section modulus of steel tube
 S_1 , S_2 = force axes in two-dimensional force-space
 $\{S\}$ = end-of-step force vector
 $\{S'\}$ = conjugate force vector to $\{S\}$
 SRC = steel-reinforced concrete
 t = thickness of steel tube
 t_{final} = time at end of load step
 u = displacement
 $\{u\}$ = unit vector denoting element orientation
 U_T , U_p , U_{my} , U_{mz} = elastic strain energy--total, axial, minor and major axis bending
 V = transverse applied load
 W_p = total accumulated plastic work
 x_n = distance from top fiber of cross-section to neutral axis
 y_i = y-distance from fiber element i to neutral axis
 z_i = z-distance from fiber element i to neutral axis
 $\{\alpha_{LS}\}$ = normalized loading surface centroid
 $\{\alpha_{BS}\}$ = normalized bounding surface centroid
 α = angle of load eccentricity (0° = major axis bending, 90° = minor axis bending)
 β = concrete flexural rigidity reduction factor
 δ = member mid-height deflection; distance between loading and bounding surfaces
 δ_{in} = initial distance between loading and bounding surfaces
 Δ = member end deflection
 $\Delta\varepsilon$ = fiber analysis strain increment
 $\Delta\phi$ = fiber analysis curvature increment
 $\Delta\theta$ = fiber analysis increment in neutral axis orientation
 ε_c = concrete strain (compression)
 ε_{ct} = concrete strain (tension)
 ε_{lb} = strain at which local buckling of the steel tube occurs
 ε_{max} = maximum strain in the steel
 ε_o = concrete strain at f'_c
 ε_s = steel strain
 ε_{sh} = strain at the onset on strain hardening of the steel

ϵ_{top} = strain at top fiber of cross-section
 ϵ_y = strain at yield stress of steel tube
 η_{LS}, η_{BS} = magnitude of centroid movements due to isotropic hardening
 θ = angle between neutral axis and centroidal axis of cross-section; rotation
 φ = normalized axial force coordinate of CFT surface centroid
 φ_{calc} = calculated normalized axial force coordinate of CFT surface centroid
 φ_{fa} = fiber analysis normalized axial force coordinate of CFT surface centroid
 $\{\Phi_{BS}\}$ = unnormalized bounding surface offset vector
 $\{\Phi_{LS}\}$ = unnormalized loading surface offset vector
 $\{\vartheta\}$ = Tseng kinematic hardening vector
 κ_1, κ_2 = calibrated plasticity coefficients
 $(\kappa_2)_{init}, (\kappa_2)_{final}, (\kappa_2)_{new}$ = initial, final, and updated values of κ_2
 $[\Lambda]$ = diagonal matrix of magnitude of bounding surface translation
 ξ_{Ec} = rate of concrete elastic modulus decrease
 ξ_{κ_2} = rate of change in κ_2
 ξ_{LS}, ξ_{BS} = isotropic softening rates of loading and bounding surfaces
 ρ_{ds} = distance between surfaces in Tseng kinematic hardening formulation
 σ_1, σ_2 = stress axes in two-dimensional stress-space
 σ_i = stress in fiber element i
 $\{\tau\}$ = temporary vector in Tseng kinematic hardening formulation
 $\{v\}$ = Mroz kinematic hardening vector
 ϕ = curvature
 Ω_p = normalized accumulated plastic work
 $(\Omega_p)_{intm}$ = normalized plastic work at point of maximum bounding surface radius
 $\{ \}$ = vector
 $[\]$ = matrix
 $\|\{ \}\|$ = magnitude of given vector, $\{ \}$

List of References

American Institute of Steel Construction, Inc. (AISC LRFD) [1993]. "Load and Resistance Factor Design Specification for Structural Steel Buildings," Second Edition, AISC, Inc., Chicago, Illinois, 1993.

American Institute of Steel Construction, Inc. (AISC LRFD) [1994]. "Load and Resistance Factor Design Manual for Structural Steel Buildings," Second Edition, AISC, Inc., Chicago, Illinois, 1994.

American Institute of Steel Construction, Inc. (AISC) [1994]. "Structural Steel Shape Availability," Modern Steel Construction, Vol. 34, No. 1, January 1994, pp. 17-23.

Argyris, J. H., Hilpert, O., Malejannakis, G. A., and Scharpf, D. W. [1979]. "On the Geometrical Stiffness of a Beam in Space," Computer Methods in Applied Mechanics and Engineering, Vol. 20, 1979, pp. 105-131.

Armen, H. [1978]. "Assumptions, Models, and Computational Methods for Plasticity," Computers and Structures, Vol. 10, 1978, pp. 161-174.

Azizinamini, A., and Prakash, B. [1993]. "A Tentative Design Guideline for a new Steel Beam Connection Detail to Composite Tube Columns," Engineering Journal, AISC, Vol. 30, 1993, pp. 108-115.

Bathe, K.-J., and Bolourchi, S. [1979]. "Large Displacement Analysis of Three-Dimensional Beam Structures," International Journal for Numerical Methods in Engineering, Vol. 14, 1979, pp. 961-986.

Bathe, K.-J. [1982]. Finite Element Procedures in Engineering Analysis, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

- Bazant, Z. P. [1982]. "Mathematical Models of Nonlinear Behavior and Fracture of Concrete," Nonlinear Numerical Analysis of Reinforced Concrete, The Winter Annual Meeting of the American Society of Mechanical Engineers, Phoenix, Arizona, 14-19 November, 1982, ASME, New York, 1982, pp. 1-26.
- Bode, H. [1976]. "Columns of Steel Tubular Sections Filled with Concrete Design and Applications," Acier Stahl, No. 11/12, 1976, pp. 388-393.
- Bridge, R. Q. [1976]. "Concrete Filled Steel Tubular Columns," Report No. R283, School of Civil Engineering, University of Sydney, Sydney, Australia, 1976.
- Bridge, R. Q., and Webb, J. [1993]. "Thin Walled Circular Concrete Filled Steel Tubular Columns," Composite Construction in Steel and Concrete II, Easterling, W. S., and Roddis, W. M. (eds.), ASCE, New York, 1993, pp. 634-649.
- Bull Moose Tube Company (BMTC) [1994]. "Connection Guide for Hollow Structural Sections," February 1994.
- Cai, S.-H. [1987]. "Ultimate Strength of Concrete-Filled Tube Columns," Composite Construction in Steel and Concrete, ASCE, New York, 1987, pp. 702-727.
- Cederwall, K., Engstrom, B., and Grauers, M. [1990]. "High-Strength Concrete Used in Composite Columns," Second International Symposium on Utilization of High-Strength Concrete, Hester, W. T. (ed.), Berkeley, California, May 1990, pp. 195-214.
- Chaboche, J. L. [1986]. "Time-Independent Constitutive Theories for Cyclic Plasticity," International Journal of Plasticity, Vol. 2, No. 2, 1986, pp. 149-188.
- Chen, W. F., and Chen C. H. [1973]. "Analysis of Concrete-Filled Steel Tubular Beam-Columns," Memoires, IABSE, 33(II), 1973, pp. 37-52.
- Chen, W. F., and Atsuta, T. [1976, 1977]. Theory of Beam-Columns, Vol.1: In-Plane Behavior and Design, Vol. 2: Space Behavior and Design, McGraw-Hill, New York, Vol. 1, 1976, Vol. 2, 1977.
- Collins, M. P., Mitchell, D., and MacGregor, J. G. [1993]. "Structural Design Considerations for High-Strength Concrete," Concrete International, May 1993, pp. 27-34.
- Council on Tall Buildings and Urban Habitat [1979]. Structural Design of Tall Steel Buildings, Monograph on the Planning and Design of Tall Buildings, Vol. SB, ASCE, New York, 1979, pp. 671-680.

- Cusson, D., and Paultre, P. [1993]. "Experimental Study of High-Strength Concrete Columns Confined by Rectangular Ties," Third International Symposium on Utilization of High-Strength Concrete, Lillehammer, Norway, 20-24 June 1993, pp. 136-145.
- Dafalias, Y. F., and Popov, E. P. [1975]. "A Model of Nonlinearly Hardening Materials for Complex Loading," Acta Mechanica, Vol. 21, 1975, pp. 173-191.
- Dafalias, Y. F., and Popov, E. P. [1976]. "Plastic Internal Variables Formalism of Cyclic Plasticity," Journal of Applied Mechanics, Vol. 48, 1976, pp. 645-650.
- Dafalias, Y. F. [1986]. "Bounding Surface Plasticity. I: Mathematical Foundation and Hypoplasticity," Journal of Engineering Mechanics, ASCE, Vol. 112, 1986, pp. 966-987.
- Dafalias, Y. F. [1992]. "Bounding Surface Plasticity Model for Steel under Cyclic Loading," Stability and Ductility of Steel Structures under Cyclic Loading, Fukumoto, Y., and Lee, G. C. (eds.), CRC Press, Boca Raton, Florida, 1992, pp. 25-36.
- Drucker, D. C. [1951]. "A More Fundamental Approach to Plastic Stress-Strain Relations," Proceedings of the First US National Congress of Applied Mechanics, Chicago, Illinois, 1951.
- Duan, L., and Chen, W.-F. [1990]. "Design Interaction Equations for Cylindrical Tubular Beam-Columns," Journal of Structural Engineering, ASCE, Vol. 116, No. 7, July 1990, pp.1794-1812.
- El-Tawil, S., Sanz-Picon, C. F., and Deierlein, G. G. [1993]. "Applying the Fiber Element Technique to Composite Beam-Columns," ACI Structural Journal, July 1993, submitted for publication.
- Fardis, M. N., Alibe, B., and Tassoulas, J. L. [1983]. "Monotonic and Cyclic Constitutive Law for Concrete," Journal of Engineering Mechanics, ASCE, Vol. 109 (Proceedings Paper 17871), 1983, pp. 516-536.
- Furlong, R. W. [1967]. "Strength of Steel-Encased Concrete Beam Columns," Journal of the Structural Division, ASCE, Vol. 93, No. ST5, 1967, pp. 113-124.
- Galambos, T. V. [1988]. Guide to Stability Design Criteria for Metal Structures, Fourth ed., John Wiley & Sons, Inc., New York, 1988, 786 pp.
- Gardner, N. J., and Jacobson, E. R. [1967]. "Structural Behavior of Concrete Filled Steel Tubes," Journal of the ACI, Vol. 64, No. 11, 1967, pp. 404-413.
- Gardner, N. J. [1968]. "Use of Spiral Welded Steel Tubes in Pipe Columns," Journal of the ACI, Vol. 65, No. 11, 1968, pp. 937-942.

Ge, H. B., and Usami, T. [1994]. "Strength Analysis of Concrete-Filled Thin-Walled Steel Box Columns," Journal of Constructional Steel Research, Vol. 30, No. 3, 1994, pp. 259-281.

Godfrey, K. A. Jr. [1987]. "Concrete Strength Record Jumps 36%," Civil Engineering, October 1987, pp. 84-88.

Goel, S. C., and Yamanouchi, H. (eds.) [1993]. "Proceedings of the 1992 US-Japan Workshop on Composite and Hybrid Structures," Berkeley, California, 10-12 September 1992, US-Japan Cooperative Research Program, Research Report, Department of Civil and Environmental Engineering, University of Michigan, 1993.

Gourley, B. C., and Hajjar, J. F. [1993]. "A Synopsis of Studies of the Monotonic and Cyclic Behavior of Concrete-Filled Steel Tube Beam-Columns," Report No. ST-93-5, Department of Civil and Mineral Engineering, University of Minnesota, Minneapolis, Minnesota, October 1993, 101 pp.

Griffis, L. G. [1992]. "Composite Frame Construction," Constructional Steel Design: An International Guide, Dowling, P. J., Harding, J. E., and Bjorhovde, R. (eds.), Elsevier, London, U.K., 1992, pp. 523-553

Hilmy, S. I. [1984]. "Adaptive Nonlinear Dynamic Analysis of Three-Dimensional Steel Framed Structures with Interactive Computer Graphics," Report No. 84-8, Department of Structural Engineering, Cornell University, Ithaca, New York, June 1984, 291 pp.

Hilmy, S. I., and Abel, J. F. [1985]. "A Strain-Hardening Concentrated Plasticity Model for Nonlinear Dynamic Analysis of Steel Buildings," NUMETA 85, Numerical Methods in Engineering: Theory and Applications, Vol. 1, 1985, pp. 305-314.

Hilmy, S. I., Hajjar, J. F., and Abel, J. F. [1987]. "Practical Aspects of Implementation of Force-Space Bounding-Surface Plasticity for Dynamics of Steel Frames," International Conference on Constitutive Laws for Engineering: Theory and Applications, Desai, C. S. (ed.), University of Arizona, Tucson, Arizona, Elsevier, New York, January 1987.

Ichinohe, Y., Matsutani, T., Nakajima, M., Ueda, H., and Takada, K. [1991]. "Elasto-Plastic Behavior of Concrete Filled Steel Circular Columns," Proceedings of the Third International Conference on Steel-Concrete Composite Structures, Wakabayashi, M. (ed.), Fukuoka, Japan, 26-29 September 1991, pp. 131-136.

Kawaguchi, J., Morino, S., Atsumi, H., and Yamamoto, S. [1991]. "Strength Deterioration Behavior of Concrete-Filled Steel Tubular Beam-Columns Under Repeated Horizontal Loading," Proceedings of the Third International Conference on Steel-Concrete Composite Structures, Wakabayashi, M. (ed.), Fukuoka, Japan, 26-29 September 1991, pp. 119-124.

Kawaguchi, J., Morino, S., Atsumi, H., and Yamamoto, S. [1993]. "Strength Deterioration Behavior of Concrete-Filled Steel Tubular Beam-Columns," Composite Construction in Steel and Concrete II, Easterling, W. S., and Roddis, W. M. (eds.), ASCE, New York, 1993, pp. 825-839.

Kitada, T., Yoshida, Y., and Nakai, H. [1987]. "Fundamental Study on Elastoplastic Behavior of Concrete Encased Steel Short Tubular Columns," Memoirs of the Faculty of Engineering, Osaka City University, Vol. 28, 1987, pp. 237-253.

Kitada, T., and Nakai, H. [1991]. "Experimental Study on Ultimate Strength of Concrete-Filled Square Steel Short Members Subjected to Compression or Torsion," Proceedings of the Third International Conference on Steel-Concrete Composite Structures, Wakabayashi, M. (ed.), Fukuoka, Japan, 26-29 September 1991, pp. 137-142.

Kitada, T. [1992]. "Ductility and Ultimate Strength of Concrete-Filled Steel Members," Stability and Ductility of Steel Structures under Cyclic Loading, Fukomoto, Y., and Lee, G. C. (eds.), CRC Press, Boca Raton, Florida, 1992, pp. 139-148.

Knowles, R. B., and Park, R. [1969]. "Strength of Concrete Filled Steel Tubular Columns," Journal of the Structural Division, ASCE, Vol. 95, No. ST12, 1969, pp. 2565-2587.

Knowles, R. B., and Park, R. [1970]. "Axial Load Design for Concrete Filled Steel Tubes," Journal of the Structural Division, ASCE, Vol. 96, No. ST10, 1970, pp. 2125-2153.

Krieg, R. D. [1975]. "A Practical Two Surface Plasticity Theory," Journal of Applied Mechanics, ASME, Vol. 47, 1975, pp. 641-646.

Lie, T. T., and Stringer, D. C. [1994]. "Calculation of the Fire Resistance of Steel Hollow Structural Columns Filled with Plain Concrete," Canadian Journal of Civil Engineering, Vol. 21, No. 3, June 1994, pp. 382-385.

Liu, Z., and Goel, S. C. [1988]. "Cyclic Load Behavior of Concrete-Filled Tubular Braces," Journal of Structural Engineering, Vol. 114, No. 7, 1988, pp.1488-1506.

Lundberg, J. E. [1993]. "The Reliability of Composite Columns and Beam-Columns," M.S. thesis. Structural Engineering Report 93-2, Department of Civil and Mineral Engineering, University of Minnesota, Minneapolis, Minnesota, 1993.

Masuo, K., Adachi, M., Kawabata, K., Kobayashi, M., and Konishi, M. [1991]. "Buckling Behavior of Concrete Filled Circular Steel Tubular Columns Using Light-Weight Concrete," Proceedings of the Third International Conference on Steel-Concrete Composite Structures, Wakabayashi, M. (ed.), Fukuoka, Japan, 26-29 September 1991, pp. 95-100.

Matsui, C. [1986]. "Strength and Deformation Capacity of Frames Composed of Wide Flange Beams and Concrete Filled Square Steel Tubular Columns," Proceedings of the Pacific Structural Steel Conference, Vol. 2, Auckland, New Zealand, 4-8 August 1986, pp. 169-181.

Matsui, C., and Tsuda, K. [1987]. "Strength and Behavior of Concrete-Filled Steel Square Tubular Columns with Large Width-Thickness Ratios," Proceedings of Pacific Conference on Earthquake Engineering, Vol. 2, Wairakei, New Zealand, 5-8 August 1987, pp. 1-9.

Matsui, C., Tsuda, K., and Din, H. Z. E. [1993]. "Stability Design of Slender Concrete Filled Steel Square Tubular Columns," Proceedings of the Fourth East Asia-Pacific Conference on Structural Engineering and Construction, Vol. 1, Seoul, Korea, 20-22 September 1993, pp. 317-322.

McDowell, D. L. [1985]. "A Two Surface Model for Transient Nonproportional Cyclic Plasticity: Part 1 Development of Appropriate Equations," Journal of Applied Mechanics, Transactions of the ASME, Vol. 52, June 1985, pp. 298-302.

McDowell, D. L. [1987]. "Simple Experimentally Motivated Cyclic Plasticity Model," Journal of Engineering Mechanics, ASCE, Vol. 113, No. 3, March 1987, pp. 378-397.

Morino, S., Kawaguchi, J., Yasuzaki, C., and Kanazawa, S. 1993]. "Behavior of Concrete-Filled Steel Tubular Three-Dimensional Subassemblages," Composite Construction in Steel and Concrete II, Easterling, W. S., and Roddis, W. M. (eds.), ASCE, New York, 1993, pp. 726-741.

Mroz, Z. [1967]. "On the Description of Anisotropic Workhardening," Journal of Mechanical Physics and Solids, Vol. 15, 1967, pp. 163-175.

Nakai, H., Kurita, A., and Ichinose, L. H. [1991]. "An Experimental Study on Creep of Concrete Filled Steel Pipes," Proceedings of the Third International Conference on Steel-Concrete Composite Structures, Wakabayashi, M. (ed.), Fukuoka, Japan, 26-29 September 1991, pp. 55-60.

Neogi, P. K., Sen, H. K., and Chapman, J. C. [1969]. "Concrete-Filled Tubular Steel Columns Under Eccentric Loading," Structural Engineer, Vol. 47, No. 5, 1969, pp. 187-195.

Orbison, J. G. [1982]. "Nonlinear Static Analysis of Three-Dimensional Steel Frames," Ph.D. dissertation, Cornell University, Ithaca, New York, May 1982.

Orbison, J. G., McGuire, W., and Abel, J. F. [1982]. "Yield Surface Applications in Nonlinear Steel Frame Analysis," Computer Methods in Applied Mechanics and Engineering, Vol. 33, 1982, pp. 557-573.

Phillips, A., and Weng, G. J. [1975]. "An Analytical Study of an Experimentally Verified Hardening Law," Journal of Applied Mechanics, Vol. 42, Series E, 1975, pp. 375-378.

Phillips, A., and Lee, C.-W. [1979]. "Yield Surfaces and Loading Surfaces. Experiments and Recommendations," International Journal of Solids and Structures, Vol 15, 1979, pp. 715-729.

Popovics, S. [1973]. "A Numerical Approach to the Complete Stress-Strain Curve of Concrete," Cement and Concrete Research, Vol. 3, No. 5, May 1973, pp. 583-599.

Prager, W. [1956]. "A New Method of Analyzing Stress and Strains in Work-Hardening Solids," Journal of Applied Mechanics, Vol. 23, 1956, pp. 493.

Priestley, M. J. N., Seible, F., Xiao, Y., and Verma, R. [1994]. "Steel Jacket Retrofitting of Reinforced Concrete Bridge Columns for Enhanced Shear Strength--Part 1: Theoretical Considerations and Test Design," ACI Structural Journal, Vol. 91, No. 4, July-August 1994, pp. 394-405.

Prion, H. G. L., and Boehme, J. [1989]. "Beam-Column Behavior of Steel Tubes Filled With High Strength Concrete," Proceedings of the Fourth International Colloquium, SSRC, New York, 1989, pp. 439-450.

Prion, H. G. L., and McLellan, A. B. [1992]. "Connecting Steel Beams to Concrete-Filled Steel Columns," ASCE Structures Congress '92, San Antonio, Texas, 13-16 April 1992, ASCE, New York, 1992, pp. 918-921.

Richard, R., and Abbot, B. [1975]. "Versatile Elastic-Plastic Stress-Strain Formula," Journal of the Engineering Mechanics Division, ASCE, Vol. 101 EM 4, August 1975, pp. 511-515.

Ricles, J. M., and Popov, E. P. [1994]. "Inelastic Link Element for EBF Seismic Analysis," Journal of Structural Engineering, Vol. 120, No. 2, February 1994, pp. 441-463.

Sakino, K., and Tomii, M. [1981]. "Hysteretic Behavior of Concrete Filled Square Steel Tubular Beam-Columns Failed in Flexure," Transactions of the Japan Concrete Institute, Vol. 3, 1981, pp. 439-446.

Salmon, C. G., and Johnson, J. E. [1990]. Steel Structures: Design and Behavior, Third Edition, Harper Collins Publishers, Inc., New York, 1990.

Sanz-Picon, C. F. [1992]. "Behavior of Composite Column Cross Sections Under Biaxial Bending," M.S. thesis, Cornell University, Ithaca, New York, 1992.

Schiller, P. H., Hajjar, J. F., and Gourley, B. C. [1994]. "Expressions for the Elastic Rigidity of Rectangular Concrete-Filled Steel Tube Beam-Columns," Report No. ST-94-2, Department of Civil and Mineral Engineering, University of Minnesota, Minneapolis, Minnesota, 1994, 15 pp.

Sfakianakis, M. G., and Fardis, M. N. [1991]. "Bounding Surface Model for Cyclic Biaxial Bending of RC Sections," Journal of Engineering Mechanics, ASCE, Vol. 117, No. 12, December 1991, pp. 2748-2769.

Shakir-Khalil, H., and Zeghiche, Z. [1989]. "Experimental Behavior of Concrete-Filled Rolled Rectangular Hollow Section Columns," The Structural Engineer, Vol. 67, 1989, pp. 345-353.

Shakir-Khalil, H., and Mouli, M. [1990]. "Further Tests on Concrete-Filled Rectangular Hollow-Section Columns," The Structural Engineer, Vol. 68, No. 20, October 1990, pp. 405-413.

Shakir-Khalil, H. [1991]. "Tests on Concrete-Filled Hollow Section Columns," Proceedings of the Third International Conference on Steel-Concrete Composite Structures, Wakabayashi, M. (ed.), Fukuoka, Japan, 26-29 September 1991, pp. 89-95.

Sherman, D. R. [1992]. "Tubular Members," Constructional Steel Design: An International Guide, Dowling, P. J., Harding, J. E., and Bjorhovde, R. (eds.) Elsevier Science Publishers Ltd., London, 1992, pp. 91-104.

Structural Stability Research Council (SSRC), Task Group 20 [1979]. "A Specification for the Design of Steel-Concrete Composite Columns," Engineering Journal, AISC, Vol. 16, No. 4, 1979, pp. 101-115.

Sugano, S., Nagashima, T., and Kei, T. [1992]. "Seismic Behavior of Concrete-Filled Tubular Steel Columns," ASCE Structures Congress '92, San Antonio, Texas, 13-16 April 1992, ASCE, New York, 1992, pp. 914-917.

Tomii, M., Matsui, C., and Sakino, K. [1973]. "Concrete-Filled Steel Tube Structures," Proceedings of the National Conference on the Planning and Design of Tall Buildings, ASCE-IABSE, Tokyo, Japan, August 1973, pp. 55-72.

- Tomii, M., and Sakino, K. [1979a]. "Experimental Studies on the Ultimate Moment of Concrete Filled Square Steel Tubular Beam-Columns," Transactions of the Architectural Institute of Japan, No. 275, January 1979, pp. 55-63.
- Tomii, M., and Sakino, K. [1979b]. "Elasto-Plastic Behavior of Concrete Filled Square Steel Tubular Beam-Columns," Transactions of the Architectural Institute of Japan, No. 280, June 1979, pp. 111-120.
- Tomii, M., and Sakino, K. [1979c]. "Experimental Studies on Concrete Filled Square Steel Tubular Beam-Columns Subjected to Monotonic Shearing Force and Constant Axial Force," Transactions of the Architectural Institute of Japan, No. 281, July 1979, pp. 81-90.
- Tomii, M. [1991]. "Ductile and Strong Columns Composed of Steel Tube, Infilled Concrete, and Longitudinal Steel Bars," Proceedings of the Third International Conference on Steel-Concrete Composite Structures, Special Volume, Fukuoka, Japan, 26-29 September 1991, pp. 39-66.
- Tseng, N. T., and Lee, G. C. [1983]. "Simple Plasticity Model of Two-Surface Type," Journal of Engineering Mechanics, ASCE, Vol. 109, No. 3, June 1983, pp. 795-810.
- Tsuji, B., Nakashima, M., and Morita, S. [1991]. "Axial Compression Behavior of Concrete Filled Circular Steel Tubes," Proceedings of the Third International Conference on Steel-Concrete Composite Structures, Wakabayashi, M. (ed.), Fukuoka, Japan, 26-29 September 1991, pp. 19-24.
- Vecchio, F. J., and Collins, M. P. [1986]. "The Modified Compression Field Theory for Reinforced Concrete Elements Subjected to Shear," ACI Structural Journal, Vol. 83, No. 2, March-April 1986, pp. 219-231.
- Virdi, P. J., and Dowling, K. S. [1980]. "Bond Strength In Concrete Filled Steel Tubes," Periodica, IABSE, August 1980, pp. 125-139.
- Weaver, W. Jr., and Gere, J. M. [1990]. Matrix Analysis of Framed Structures, Third Edition, Van Nostrand Reinhold, New York, 1990, 546 pp.
- Webb, J. [1993]. "High-Strength Concrete: Economics, Design, and Ductility," Concrete International, Vol. 15, No. 1, January 1993, pp. 27-32.
- White, D. W., and Hajjar, J. F. [1991]. "Application of Second-Order Elastic Analysis in LRFD: Research to Practice," Engineering Journal, AISC, Vol. 28, No. 4, 1991, pp. 133-148.

- White, D. W., and Chen, W. F. (eds.) [1993]. Plastic Hinge Based Methods for Advanced Analysis and Design of Frames: An Assessment of the State-of-the-Art, Structural Stability Research Council, March 1993, 299 pp.
- Yamagouchi et al. [1989]. "Studies on Concrete-Filled Tubular Steel Columns," Proceedings of Annual Meeting, Architectural Institute of Japan, 1989 (in Japanese).
- Yamanouchi, H., Mahin, S. A., Goel, S. C., and Nishiyama, I. [1993]. "US-Japan Cooperative Earthquake Research Program on Composite and Hybrid Structures," Proceedings of the 1992 US-Japan Workshop on Composite and Hybrid Structures, US-Japan Cooperative Research Program, 1993.
- Yang, B. L., Dafalias, Y. F., and Herrmann, L. R. [1985]. "A Bounding Surface Plasticity Model for Concrete," Journal of Engineering Mechanics, ASCE, Vol. 113, No. 3, 1985, pp. 359-380.
- Yoder, P. J., and McDowell, D. L. [1989]. "Bounding Surface Plasticity: Stress Space Versus Strain Space," Acta Mechanica, Vol. 77, 1989, pp. 13-45.
- Zhang, X., Jiang, W., and Wang, D. [1991]. "The Carrying Capacity of Concrete Filled Steel Tube Short Columns," Proceedings of the Third International Conference on Steel-Concrete Composite Structures, Wakabayashi, M. (ed.), Fukuoka, Japan, 26-29 September 1991, pp. 85-94.
- Zhao, Y. [1993]. "Modeling of Inelastic Cyclic Behavior of Members, Connections, and Joint Panels of Steel Frames," Ph.D. dissertation, Cornell University, Ithaca, New York, 1993.
- Zhong, S.-T., and Miao, R.-Y. [1987]. "Stress-Strain Relationship and Strength of Concrete Filled Tubes," Composite Construction in Steel and Concrete, ASCE, New York, 1987, pp. 773-785.
- Ziegler, H. [1959]. "A Modification of Prager's Hardening Rule," Quarterly of Applied Mathematics, Vol 17, 1959, pp. 55-60.