

Implementation and Verification of Simple Concrete Biaxial Models Under Monotonic,
Cyclic, and Dynamic Loading

by

Jesus Ramon Salazar Lopez

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

STRUCTURAL ENGINEERING

Department of Civil and Environmental Engineering
University of Alberta

© Jesus Ramon Salazar Lopez, 2019

ABSTRACT

An accurate prediction of the response and strength of concrete elements, which exhibit nonlinear behavior even under moderate loading, is essential for evaluating their safety and serviceability. To describe the nonlinear behavior of this material, continuum damage mechanics has been demonstrated to be effective at developing damage models that can be then implemented in finite element analysis (FEA) platforms. One such platform is the open-source, freely available software OpenSees, which is a FEA software framework for simulation in earthquake engineering (Fenves, 2001).

This project studies the performance of simple, yet accurate biaxial concrete materials, amenable for FE analysis, with the ability to account for stiffness recovery in reversal loading (crack closing), permanent deformations, and low to moderate confinement. Two concrete damage models – the PRM model (Pontiroli, Rouquand, & Mazars, 2010) and the “ μ ” model (Mazars, Hamon, & Grange, 2015)– were implemented in OpenSees to develop new biaxial concrete materials.

The performance of the 2D new biaxial materials implemented in OpenSees is studied by comparing five concrete experimental tests with varying complexity taken from the literature with analytical models built in OpenSees. The experiments consist of 1) plain concrete plates tested under biaxial states of stress (Kupfer, Hilsdorf, & Rüschi, 1969), 2) a simply-supported beam under monotonic loading tested as part of this project, 3) a simply-supported beam under reversal-cyclic loading (Ranjbaran, Rezayfar, & Mirzababai, 2018), 4) a rectangular shear wall under reversal-cyclic loading (Hiotakis, 2004), and 5) a full-scale four-storey building under dynamic, seismic loading (Nagae, et al., 2015). The advantages and limitations of each model are discussed.

ACKNOWLEDGMENTS

My graduate degree was not a solitary pursuit; it was the product of many bright and dedicated people offering their support, patience, and encouragement.

First and foremost, I would like to thank my research supervisor, Dr. Carlos Cruz-Noguez, without the assistance and guidance of whom this project would not have been possible.

Special thanks are due to Dr. Ali Imanpour, Dr. Mustafa Gul, and Dr. William Wenming for serving on the examination committee.

For funding my research, I would like to extend my immense gratitude to Mexico's National Commission of Science and Technology (CONACYT) and the University of Alberta.

I am also indebted to my brilliant research peers at the University of Alberta, who offered ongoing guidance, inspiration, and friendship over the course of my research.

An additional thank you goes out to my proofreader, B., for improving my English, catching my sleep-deprivation-related typos, and encouraging me to stay on schedule.

And finally, I am abundantly appreciative for my family, whose love and faith in my abilities, despite the miles between us, have kept me going even when the end seemed so far away.

TABLE OF CONTENTS

- 1 INTRODUCTION 1**
 - 1.1 Background 1
 - 1.2 Problem Statement 5
 - 1.3 Objectives, Methods, and Scope 6
 - 1.4 Thesis Outline 8

- 2 LITERATURE REVIEW 9**
 - 2.1 Introduction 9
 - 2.2 Behavior of Plain Concrete Under Different States of Stress 10
 - 2.2.1 Behavior of Plain Concrete Under Uniaxial State of Stress 10
 - 2.2.2 Behavior of Plain Concrete Under Biaxial State of Stress 13
 - 2.3 Theoretical Approaches for nonlinear FEA Modelling of Concrete 14
 - 2.3.1 Nonlinear-Elasticity-Based Approach 16
 - 2.3.2 Plasticity-Based Approach..... 21
 - 2.3.3 Rotating and Fixed Angle Shear Theories Based on Total-Strain Approach. 21
 - 2.3.4 Damage-Continuous-Mechanics-Based Approach 30
 - 2.4 Structural Analysis Software Using the Finite Element Method (FEM) 33
 - 2.4.1 VecTor Analysis Group 33
 - 2.4.2 DIANA FEA..... 37
 - 2.4.3 Abaqus FEA..... 38
 - 2.4.4 OpenSees..... 41

- 3 CONCRETE DAMAGE MODELS 42**
 - 3.1 Introduction 42

3.2	Scalar Damage Model.....	43
3.3	PRM Model	45
3.4	“ μ ” Model.....	48
3.5	Influence of Test Parameters (A_c , B_c , A_t , and B_t) and Uniaxial Stress-Strain Response of the Scalar Damage Model, the PRM Model, and the “ μ ” Model	51
4	IMPLEMENTATION OF TWO BIAXIAL CONCRETE MATERIALS IN THE OPENSEES FRAMEWORK	66
4.1	Introduction	66
4.2	OpenSees	66
4.2.1	Materials	68
4.2.2	Elements.....	74
4.3	Coordinate System.....	79
4.4	Material Constitutive Matrix.....	81
4.5	Finite Element Analysis Procedure.....	83
4.6	Introducing a New Multi-dimensional Material –nD Material– to the OpenSees Framework	86
4.6.1	New nDMaterial: Header and C++ Files	91
4.6.2	Class Tags.....	92
4.6.3	Object Broker	92
4.6.4	Model Builder.....	93
5	VALIDATION & DISCUSSION.....	95
5.1	Introduction	95
5.2	Plain Concrete Panels Under Biaxial Stress.....	96
5.2.1	Finite Element Model.....	97
5.2.2	Comparison Between Analytical and Experimental Data	97

5.3	RC Beam Under Monotonical Loading	98
5.3.1	Finite Element Model	100
5.3.2	Comparison Between Analytical and Experimental Data	102
5.4	RC Beam Under Reversal-Cyclic Loading	103
5.4.1	Finite Element Model.....	104
5.4.2	Comparison Between Analytical and Experimental Data	106
5.5	RC Shear Wall Under Reversal-Cyclic Loading	110
5.5.1	Finite Element Model.....	111
5.5.2	Comparison Between Analytical and Experimental Data	116
5.6	Full-Scale Four-Storey RC Building Under Seismic Loading.....	120
5.6.1	Finite Element Model.....	126
5.6.2	Comparison Between Analytical and Experimental Data	128
6	SUMMARY, CONCLUSIONS, AND FUTURE RESEARCH	131
6.1	Summary.....	131
6.2	Conclusions	133
6.3	Recommendations & Future Research	136
	REFERENCES	137
	APPENDIX A: PRM SOURCE CODE	144
	APPENDIX B: “MUMAZARS” (μ) SOURCE CODE	158

LIST OF TABLES

Table 3.1. Damage models parameters	52
Table 5.1. Model parameters for biaxial test	97
Table 5.2. Model parameters for FEM of beam under monotonic loading	100
Table 5.3. Model parameters for FEM of beam under cyclic loading	105
Table 5.4. Model parameters for FEM of shear wall under cyclic loading	114
Table 5.5. Mesh size analysis of the shear wall FEM.....	115
Table 5.6. RC building actual material properties (Nagae, et al., 2015)	125
Table 5.7. Maximum roof displacements of RC building.	130

LIST OF FIGURES

Fig. 2.1. Typical compressive stress-strain curve for concrete (Attard & Mendis, 1993)....	10
Fig. 2.2. Typical tensile stress-strain curve for concrete (Raphael, 1984)	11
Fig. 2.3. Behavior of concrete under cyclic compressive loading (Sinha, Gerstle, & Tulin, 1964)	12
Fig. 2.4. Biaxial strength of concrete (Kupfer, Hilsdorf, & Rüsçh, 1969)	13
Fig. 2.5. Stress-strain relationship of concrete under biaxial compression (Kupfer, Hilsdorf, & Rüsçh, 1969).....	14
Fig. 2.6. Theoretical Approaches for Nonlinear FEA Modelling of RC Structures	15
Fig. 2.7. Stress-strain relationship for concrete in compression (Hognestad, 1951)	18
Fig. 2.8. Stress-strain relationship for concrete in compression (Hognestad, 1955)	18
Fig. 2.9. Stress-strain relationship for unconfined and confined concrete in compression (Kent & Park, 1971).....	20
Fig. 2.10. Stress-strain relationship for tensile concrete in flexure (Kaklauskas, 1999)	21
Fig. 2.11. Decomposition of an RC element into a concrete element and a steel grid element	24
Fig. 2.12. Change of direction of the subsequent cracks	25
Fig. 2.13. Summary of the MCTF (Vecchio & Collins, 1986)	28
Fig. 2.14. Summary of the CMSS (adapted from Mansour & Hsu, 2005).....	30
Fig. 2.15. Stress-strain response of damage models	31
Fig. 2.16. VecTor2 element library (Chak, 2013).....	34
Fig. 2.17. VecTor3 element library (Chak, 2013).....	35
Fig. 2.18. VecTor4 element library (Chak, 2013).....	35
Fig. 3.1. Typical stress-strain relationship of concrete using damage models	43

Fig. 3.2. Uniaxial stress-strain response of the PRM damage model (Mazars & Grange, 2014)	47
Fig. 3.3. μ model variables evolution during a tensile-compression loading path	51
Fig. 3.4. A_c parameter influence in the compressive stress-strain response for the scalar damage model (a), the PRM model (b), and the μ model (c).....	54
Fig. 3.5. B_c parameter influence in the compressive stress-strain response for the scalar damage model (a), the PRM model (b), and the μ model (c).....	55
Fig. 3.6. A_t parameter influence in the tensile stress-strain response for the scalar damage model (a), the PRM model (b), and the μ model (c).....	56
Fig. 3.7. B_t parameter influence in the tensile stress-strain response for the scalar damage model (a), the PRM model (b), and the μ model (c).....	57
Fig. 3.8. Loading path 1: tension-compression	58
Fig. 3.9. Loading path 2: compression-tension	59
Fig. 3.10. Loading path 3: tension-compression-tension-compression	59
Fig. 3.11. Uniaxial stress-strain response for the scalar damage model (a), the PRM model (b), and the μ model (c) under the loading path 1	60
Fig. 3.12. Uniaxial stress-strain response for the scalar damage model (a), the PRM model (b), and the μ model (c) under the loading path 2	62
Fig. 3.13. Uniaxial stress-strain response for the scalar damage model (a), the PRM model (b), and the μ model (c) under the loading path 3	64
Fig. 4.1. Steel 02 material command input parameters.....	69
Fig. 4.2. Steel02 material: parameters of monotonic envelope (OpenSees).....	69
Fig. 4.3. Steel02 material: hysteretic behavior without isotropic hardening (OpenSees)	70
Fig. 4.4. Steel02 material: hysteretic behavior with isotropic hardening in compression (OpenSees).....	70

Fig. 4.5. Steel02 material: hysteretic behavior with isotropic hardening in tension (OpenSees).....	71
Fig. 4.6. Hysteretic material input parameters	72
Fig. 4.7. Hysteretic Material: typical stress-strain relationship.....	72
Fig. 4.8. Concrete01 material input parameters	73
Fig. 4.9. Concrete01 material: typical stress-strain relationship	73
Fig. 4.10. ZeroLength element input parameters	74
Fig. 4.11. Truss element input parameters.....	75
Fig. 4.12. NonlinearBeamColumn element input parameters	76
Fig. 4.13. Interaction of a zero-length element and a beam-column element (OpenSees)...	77
Fig. 4.14. Quad element input parameters	77
Fig. 4.15. ShelMITC4 element input parameters	78
Fig. 4.16. Multilayered-shell element (Lu, Xie, Guan, & Lu, 2015).....	79
Fig. 4.17. Local (x-y) and principal stresses (1-2) coordinate systems	80
Fig. 4.18. Nonlinear analysis procedure	85
Fig. 4.19. PRM material input parameters.....	86
Fig. 4.20. MuMazars material input parameters	86
Fig. 4.21. OpenSees classes hierarchy	88
Fig. 4.22. OpenSees source code solution explorer	89
Fig. 4.23. New internal class defined in the file classTags.h	92
Fig. 4.24. Modifications needed for the file FEM_ObjectBrokerAllClasses.cpp	93
Fig. 4.25. Modifications needed for the file FEM_ObjectBrokerAllClasses.cpp	94
Fig. 5.1. Plain concrete specimen under different biaxial states of stress	96
Fig. 5.2. Biaxial behavior of concrete. Adapted from Kupfer et. al. (1969)	98

Fig. 5.3. RC beam geometry and reinforcement details	99
Fig. 5.4. FEM of RC beam under monotonic loading	101
Fig. 5.5. Load-displacement response of RC beam under monotonic loading	102
Fig. 5.6. Geometry and reinforcement details of RC beam under cyclic loading (Ranjbaran, Rezayfar, & Mirzababai, 2018).....	103
Fig. 5.7. Imposed loading historey (Ranjbaran, Rezayfar, & Mirzababai, 2018).....	104
Fig. 5.8. Test set-up (Ranjbaran, Rezayfar, & Mirzababai, 2018).....	104
Fig. 5.9. FEM of RC beam under cyclic loading	105
Fig. 5.10. FEM deformed shape of a loading cyclic	106
Fig. 5.11. Measured vs calculated load-displacement hysteretic response using the scalar damage model. Adapted from Ranjabaran (2018).....	108
Fig. 5.12. Measured vs calculated load-displacement hysteretic response using the “ μ ” model. . Adapted from Ranjabaran (2018).....	108
Fig. 5.13. Measured vs calculated load-displacement hysteretic response for the (a) SD model and (b) “ μ ” model. First 7 cycles of the calculated response. Adapted from Ranjabaran (2018).....	109
Fig. 5.14. Measured vs calculated load-displacement hysteretic response using the PRM model. Adapted from Ranjabaran (2018).	110
Fig. 5.15. Shear wall reinforcement and geometry specifications	111
Fig. 5.16. FEM of RC Shear Wall (a) and multilayered shell element (b).....	113
Fig. 5.17. Measured vs calculated load-displacement hysteretic response using the scalar damage model.....	117
Fig. 5.18. Measured vs calculated load-displacement hysteretic response using the “ μ ” model.....	117
Fig. 5.19. Measured vs calculated first 12 cycles of the load-displacement hysteretic response using the scalar damage model	118

Fig. 5.20. Measured vs. calculated first 12 cycles of the load-displacement hysteretic response using the “ μ ” model.....	118
Fig. 5.21. Measured vs. calculated load-displacement hysteretic response using the PRM model.....	119
Fig. 5.22. Four-storey building geometry and structural members nomenclature: (a) longitudinal elevation; (b) transverse elevation; and (c) plan view (Nagae, et. al., 2011b)	121
Fig. 5.23. Structural members details (Nagae, et. al., 2011b)	123
Fig. 5.24. Ground motion time history.....	126
Fig. 5.25. FEM of RC building.	127
Fig. 5.26. RC building global base shear response for the 25% Kobe record in (a) longitudinal direction and (b) transverse direction	129
Fig. 5.27. RC building global base shear response for the 50% Kobe record in (a) longitudinal direction and (b) transverse direction	129
Fig. 5.28. RC building global base shear response for the 100% Kobe record in (a) longitudinal direction and (b) transverse direction	129

1 INTRODUCTION

1.1 Background

Reinforced concrete (RC) is one of the most important and widely used materials in infrastructure projects worldwide. The ultimate goal of structural design is to determine the forces in structural members to achieve economical and safe structures. Determining an adequate safety margin in RC buildings requires an accurate prediction of the ultimate capacity of concrete, which is a complex task because this material exhibits nonlinear behavior even under moderate loading. Furthermore, when subjected to multi-axial loads, it displays different strength and stiffness properties from those it exhibited when subjected to uniaxial loading.

In recent decades, the framework of continuum mechanics has been used to create multiple models to describe the nonlinear behavior of concrete. The different models can be categorized into three states of stress: uniaxial, biaxial, and triaxial. Some of the most common formulations to simulate the behaviour of the concrete are the plasticity-based models (Argyris, Faust, & William, 1981), the fracture-based models (Bazant, 1994), the total-strain-based models (Vecchio & Collins, 1986), the damage-based models (Mazars, 1986), or plastic-damage-based models (Lee & Fenves, 1998).

Total-strain based models have provided two reliable and accurate models for concrete under biaxial stress, the Modified Compression Field Theory (MCFT) (Vecchio & Collins, 1986) and the Cyclic Softened Membrane Model (CSMM) (Mansour & Hsu, 2005). These models are capable of predicting the behavior of RC structures under different loading mechanisms with good accuracy, but they require iterative procedures which may cause

convergence problems within the FEA formulations. Most are implemented in proprietary finite element software.

One of the main aspects of concrete analysis is the initiation and propagation of microcracks, which defines damage (Kachanov, 1958), represented as strain softening in structural analysis. Mazars was the first to introduce an isotropic scalar damage-based model with multiple damage variables (1986), which uses a combination of elastic damage mechanics and linear elastic fracture mechanics. This model can be used to predict the nonlinear behavior of concrete elements without requiring complex solution formulations.

Finite element analysis (FEA) has been widely used in recent decades as a tool for analyzing RC structures. In any finite element calculations for reinforced concrete, the basic information required besides loading, geometry, and boundary conditions is the multi-dimensional stress-strain relationship, known as a constitutive relationship. These incorporate suitable aspects of the response, such as cyclic behaviour, pinching effect, residual deformations, that allow the study of reinforced concrete materials subjected to monotonic, cyclic, and dynamic loading (ASCE, 1982). Once the constitutive relationships have been defined, the constitutive, equilibrium, and compatibility equations can be solved using iterative procedures until an acceptable convergence criterion is reached.

In order to avoid the complex formulations for 3D finite element analysis, simplifications can be made when one or two dimensions of an element are considerably larger than the rest. Elements where the longitudinal dimension dominates (beams and columns) can be simplified using uniaxial elements and fiber sections (1D), while elements where two dimensions dominate (plates, walls, slabs) can be simplified using biaxial elements such as plates, membranes, or shell elements (2D).

While it is now relatively straightforward to capture the flexural behaviour of the so-called “line” elements in a building, such as beams and columns through 1D uniaxial materials, it is still difficult to model the flexural response of “plane” elements such as slabs and walls, which require biaxial or triaxial material formulations.

Different analysis models able to account for material or structural non-linearity of 2D concrete-membrane elements are available today. VecTor is a suite of computer programs that has been in continuous development at the University of Toronto over the last two decades. It is capable of performing accurate nonlinear analyses of 2D (VecTor2) and 3D (VecTor3 and VecTor4) shell structural elements under monotonic and dynamic loading, using the MCTF (Vecchio & Collins, 1986). General purpose programs, such as DIANA FEA (TNO, 2016), Abaqus (Abaqus, 2009), and OpenSees (Fenves, 2001) implement non-linear analysis for subassemblies and whole structures under both monotonic and dynamic loading. For concrete, DIANA implements the Maekawa-Fukuura model (2016), which is a combination of the Total Strain Crack Model (Selby & Vecchio, 1997) and the Elasto-Plastic Fracture Model (Maekawa, Takemura, Irawan, & Irie, 1993). Abaqus implements a plastic-damage-based formulation for 3D shell elements. An elastic-damage-based model has been implemented in OpenSees (Lu, Xie, Guan, & Lu, 2015) for 2D shell elements.

Analyzing structures using nonlinear 2D concrete elements with the above programs has certain limitations. Research-oriented program VecTor2 is able to perform accurate nonlinear analysis of shell elements but it is limited to the study of subassemblies, with a limitation on the number of nodes and elements that can be used. Therefore, the analysis of whole structures is not possible. In the case of general-purpose program Abaqus, analysis of whole structures is possible, but the material models used, in general, require a large

number of input material parameters which can only be obtained through specialized testing, not usually available to the average analyst. The complexity of the models for concrete in this type of programs often leads to convergence problems that limit their applicability.

The limitations listed above indicate that there is a need of simple, yet reasonably accurate biaxial material models for concrete that allow the development of large structural models. These can be used to investigate the structures at the system level rather than at the element level, leading to an improved understanding of their structural performance under complex loading.

In a previous study, Garcia (2017) implemented a new OpenSees concrete material using the scalar damage model (Mazars, 1986) and compared the analytical results with experimental data for an RC beam under monotonic loading, an RC shear wall under reverse-cyclic loading, and a full-scale RC building subjected to earthquake loading. This software material accurately predicted the failure load of the beam, but it was not capable of assessing the failure deflection nor the cracking and yielding moments. It exhibited moderate agreement in the hysteretic analysis of the shear wall but failed to predict the energy dissipation capacity and residual displacements. Some parameters of the overall response of the RC building were predicted with reasonable accuracy. The concrete models implemented in this study aim to improve the prediction of these structural parameters.

1.2 Problem Statement

A number of concrete models have been implemented in different software platforms to create concrete materials that describe the nonlinear behavior of 2D elements. Some of them are implemented in software unable to analyze large-scale structures under dynamic loading, while others use proprietary, advanced concrete models that lead to convergence problems when the number of elements is large. Constitutive models for materials such as concrete should have a balance between simplicity and reasonable conformity to the experimental results.

The need for a simple, transparent, open source code concrete material motivated the research by García (2017), who implemented a scalar damage-based model for concrete in the OpenSees framework. Although this new material was based on one of the simplest damage-based models (Mazars, 1986), it provided a reasonable prediction of structural response at the macroscopic level, i.e., force-displacement response. However, it was not capable of accounting for unilateral effects (crack opening and closure), permanent strains, or moderate-high confinement. New materials that can describe these aspects of concrete behavior under types of loading different from monotonic loading need to be implemented in a freely available, open-source framework able to analyze complete structures and not only subassemblies.

Due to its simplicity for modeling, a damage-based approach was selected as a candidate to develop the new materials implemented in this study. Damage-based models allow an explicit solution of the material constitutive, equilibrium, and compatibility equations without needing an iterative procedure to calculate stresses from a given set of strains.

1.3 Objectives, Methods, and Scope

The general objective of this research is to study the performance of biaxial concrete formulations capable of determining stiffness recovery in reversal loading (crack closing), permanent deformations, and moderate confinement at the element and system levels, studying subassemblies (plates, beams and walls) and systems (a complete reinforced-concrete building). Two concrete damage models –PRM model (Pontiroli, Rouquand, & Mazars, 2010) and the “ μ ” model (Mazars, Hamon, & Grange, 2015)– are used to implement two new OpenSees concrete materials. The performance of these material formulations is investigated through the modelling of five structural systems and components and the comparison of the analytical response with the experimental data. The main objective can be subcategorized into three specific objectives, which are presented below, including their methodology:

1. Selection of the formulations most suited to describe biaxial concrete behavior under monotonic, cyclic, and dynamic loadings for both subassemblies and full-scale structures:

- Chapter 2 provides a literature review of the different approaches used to formulate concrete formulations, and discusses some of the most prominent concrete models, including their advantages and disadvantages. It also describes representative FEA frameworks that accurately describe the behavior of concrete structures using 2D elements, with emphasis on the specialized, object-oriented FE framework used for this research: OpenSees.

2. *Implementation of the Mazars' " μ " and PRM concrete damage models as new materials in the OpenSees framework, for their use in plane-stress shell elements:*
 - Formulate the source code for the new material in C++ language by implementing the plain-stress and specific damage model formulations specified in chapters 4 and 3, specifically.
 - Prepare user-defined subroutines for the material formulations suitable for implementation in the OpenSees framework.
3. *Assess the performance of the new materials implemented in OpenSees by comparing analytical data with experimental tests:*
 - Choose specific RC experimental tests from literature whose overall behavior or failure mechanisms can be used to evaluate the proficiency of the new materials to describe the concrete characteristics of unilateral effects, permanent strains, and moderate confinement. A description of each experiment is provided in chapter 5. Focus on both component- and system-level structural systems (i.e., beams, shear walls, and full buildings).
 - Develop FE analysis models of the structural systems in the FE platform OpenSees, making use of the new materials and shell elements.
 - Compare and discuss the results of the analytical models with the experimental tests.

The scope of this thesis is to study the ability of biaxial materials implemented in OpenSees to reproduce the concrete characteristics of lateral effects, permanent deformations, and moderate confinement under biaxial loading. The proficiency of these materials to represent

the behavior of RC elements dominated by shear or with high confinement due to impact loading can be addressed in future work. The variation of the stress through the thickness of the elements was not investigated. Seismic loading was the only dynamic application studied, and nonlinearity under wind loading was not considered.

1.4 Thesis Outline

This thesis consists of 6 chapters, for which the content is summarized below:

- Chapter 1 provides background and states the objectives and scope of this research study.
- Chapter 2 consists of a comprehensive literature review of nonlinear concrete models that have been implemented for FEA. It discusses their advantages and disadvantages when compared with the damage-based models used in this study.
- Chapter 3 introduces the damage-based model's concepts and discusses the Scalar Damage Model (Mazars, 1986), the PRM Model (Pontiroli, Rouquand, & Mazars, Predicting Concrete Behaviour From Quasi-static Loading to Hypervelocity Impact, 2010), and the “ μ ” Model (Mazars, Hamon, & Grange, 2015) formulations, which were used to accomplish objectives (1) and (2).
- Chapter 4 addresses objective (2) and contains a detailed explanation of the procedure needed to implement the new concrete materials in the OpenSees framework.
- Chapter 5 addresses objective (3) by presenting the comparison between experimental tests and analytical models using the new materials.
- Chapter 6 presents the conclusions based on chapter 5 results and discusses future work that could follow this research study.

2 LITERATURE REVIEW

2.1 Introduction

Concrete behaves either as linear or nonlinear material depending on the magnitude and types of stress to which it is subjected. Concrete has a linear-elastic material compressive response under low-stress levels, but for higher levels of stress or for sustained loading it exhibits nonlinear behavior, which has a significant influence on the behavior of RC structures in terms of increased deformation, development of cracks and stiffness degradation (ASCE, 1982).

As a non-homogeneous material, concrete contains a large number of micro-cracks even when subjected to low levels of stress. This is especially noticeable at the interface between the cement matrix and aggregate particles. Micro-cracks propagation, caused by stress, segregation, shrinkage, or thermal expansion in the cement paste, contributes to the nonlinear behavior of RC (Hsu & Mo, 2010).

In recent years, FEA has become an important tool for the analysis of RC structures. Continuous improvement of finite element nonlinear techniques and software has taken place during the last decades, which makes the structural analysis and design of RC structures more feasible. There are several theoretical approaches for creating RC models that define the complex stress-strain behavior under different stress states. The most commonly used are nonlinear elasticity, plasticity, fracture mechanics, damage continuum mechanics, and total-strain theory. Each approach is better suited for particular types of structures or loading mechanisms (Vecchio & Palermo, 2001).

2.2 Behavior of Plain Concrete Under Different States of Stress

2.2.1 Behavior of Plain Concrete Under Uniaxial State of Stress

Typical stress-strain relationships for plain concrete subjected to monotonical-uniaxial-compressive loading are shown in Fig. 2.1. According to Hsu (1963), concrete behaves as a linear-elastic material up to about 30% of its maximum compressive strength (f'_c). At this point, micro-cracks start developing at the cement-coarse interface and concrete begins to soften. Karsan and Jirsa (1969) stated that for stresses up to about 75% of the f'_c , the concrete softening phenomenon increases and the stress-strain curve bends more sharply as it approaches f'_c . Beyond the f'_c , the stress-strain curve enters the descending portion until concrete crushes; this phase is characterized by the presence of macro-cracks.

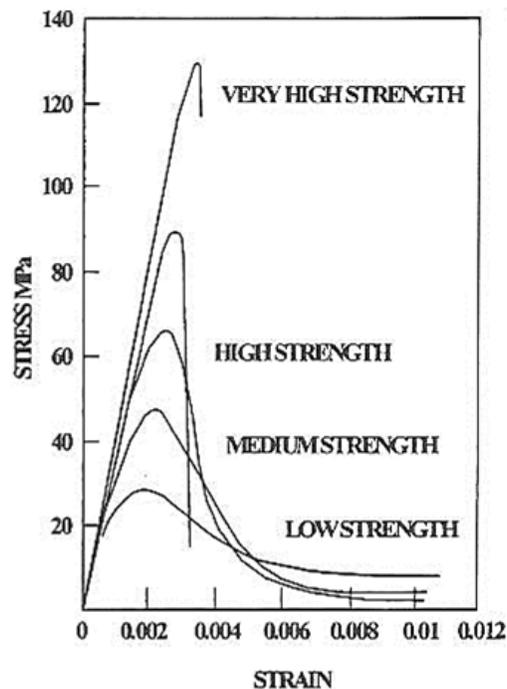


Fig. 2.1. Typical compressive stress-strain curve for concrete (Attard & Mendis, 1993)

The stress-strain curve for concrete under tension is nearly linear-elastic up to the tensile strength (f_t). The magnitude for the tensile strength (f_t) varies from 8% to 15% of the f'_c . The tensile strength is highly sensitive to the type of test, type of aggregate, f'_c , and presence of compressive stresses transverse to the tensile stresses. The tensile strength of concrete can be measured using three types of test: the direct tension test, the splitting tension test, and the modulus of rupture test. The first one is difficult to accomplish and might be subjected to large errors; the second one is the easiest to perform and renders the most reliable results; and the third one provides consistent results in various laboratories and more suitable values for FEA (Raphael, 1984). Typical tension stress-strain curves for concrete are illustrated in Fig. 2.2.

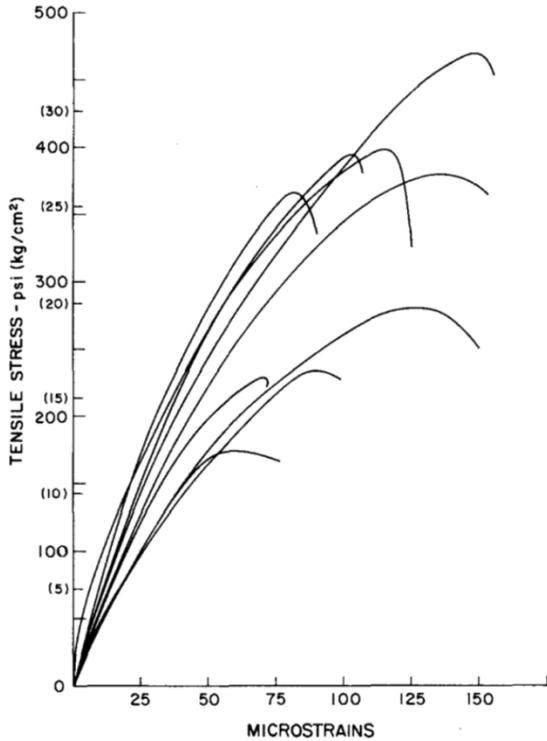


Fig. 2.2. Typical tensile stress-strain curve for concrete (Raphael, 1984)

When concrete is subjected to compressive cyclic loading, it exhibits stiffness and strength degradation (Karsan & Jirsa, 1969). Fig. 2.3 shows that the monotonic stress-strain curve (dotted line) is a reasonable representation of the envelope for the peak values of the cyclic stress-strain curve. Each cycle of loading and unloading corresponds to a hysteresis curve, where the energy dissipated during every cycle is enclosed.

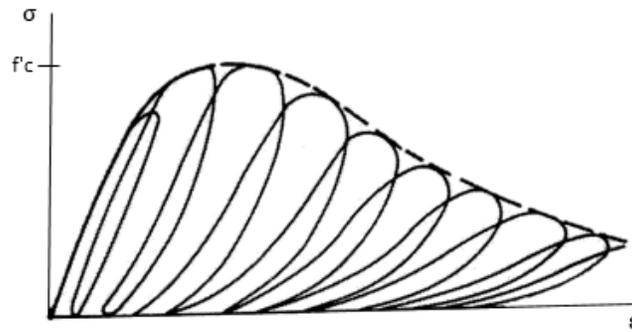


Fig. 2.3. Behavior of concrete under cyclic compressive loading (Sinha, Gerstle, & Tulin, 1964)

High-strength concrete behaves in a more brittle manner than normal strength concrete, having less load-carrying capacity once the peak load is reached (Fig. 2.1). For normal strength concretes, the coarse aggregates and cement paste are stronger than the peak force of the concrete. Failure initiates at the weak cement-coarse interface zone, and the aggregate interlock allows for shear transfer across the cracks, providing the typical post-peak behavior of normal strength concrete. For high-strength concrete, the cement paste and aggregates have a similar strength to the peak force of the concrete. The cracks start at both the cement paste and the coarse aggregates, thus, there is less shear transfer due to aggregate interlock (Attard & Mendis, 1993).

2.2.2 Behavior of Plain Concrete Under Biaxial State of Stress

Fig. 2.4 illustrates the biaxial strength envelope of normal-plain concrete. The ultimate strength under biaxial compression exhibits a strength increase of approximately 25% at a biaxial-strength ratio of 0.5. For an equal biaxial stress with a ratio 1.0, the strength increase is only about 15% (Kupfer, et al., 1969; Liu, et al., 1972; Tasuji, et al., 1978; and Hussein, et al., 2000). Under biaxial compression-tension stress, concrete exhibits a noticeably reduced strength. Under biaxial tension stress, concrete exhibits a constant (Kupfer, et al., 1969; and Hussein, et al., 2000), or a slightly increased (Tasuji, Slate, & Nilson, 1978), tensile strength in comparison with the results obtained for uniaxial tension.

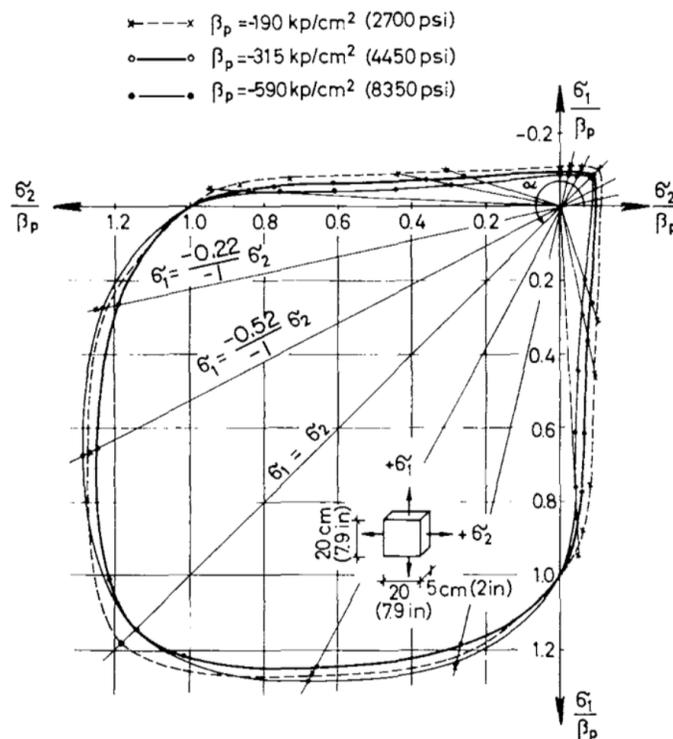


Fig. 2.4. Biaxial strength of concrete (Kupfer, Hilsdorf, & Rüschi, 1969)

Concrete exhibits a different stress-strain behavior when subjected to uniaxial and biaxial states of stress (Fig. 2.5). For biaxial compression, concrete exhibits increased initial stiffness and ductility at the peak stress. The former may be attributed to Poisson's ratio and the latter indicates reduced internal damage as compared to uniaxial loading, because of confinement effects (Kupfer, et al., 1969; and Nelissen, 1972). The failure of concrete under any combination of biaxial loading is related to a maximum-tensile-strain criterion. The fracture surface is orthogonal to the direction of the maximum-tensile strain (Tasuji, Slate, & Nilson, 1978).

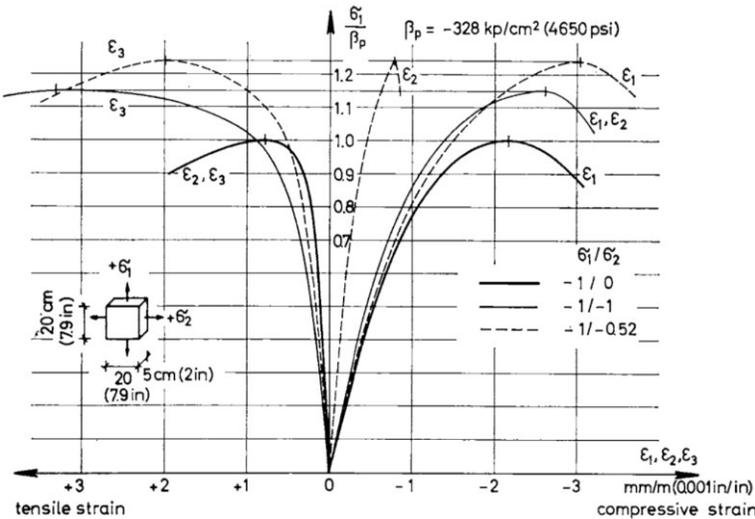


Fig. 2.5. Stress-strain relationship of concrete under biaxial compression (Kupfer, Hilsdorf, & Rüschi, 1969)

2.3 Theoretical Approaches for nonlinear FEA Modelling of Concrete

The nonlinear behavior of RC structures is a complex phenomenon that includes a multiaxial state of stress, strain softening, crack development induced by tensile stress or strain, hysteretic behavior and pinching effect (under cyclic loading), aggregate interlock,

bond slip between concrete and steel reinforcement, and time-dependent behavior such as shrinkage and creep. These phenomena make creating reliable RC constitutive models a challenging task (Chen, Yamaguchi, Kotsovos, & Pan, 1993).

RC models are built using a number of theoretical approaches. Some of the main approaches (Fig. 2.6) are nonlinear elasticity, plasticity, total-strain, and damage continuum mechanics (Vecchio & Palermo, 2001). These approaches are discussed below, including some of the most representative RC models.

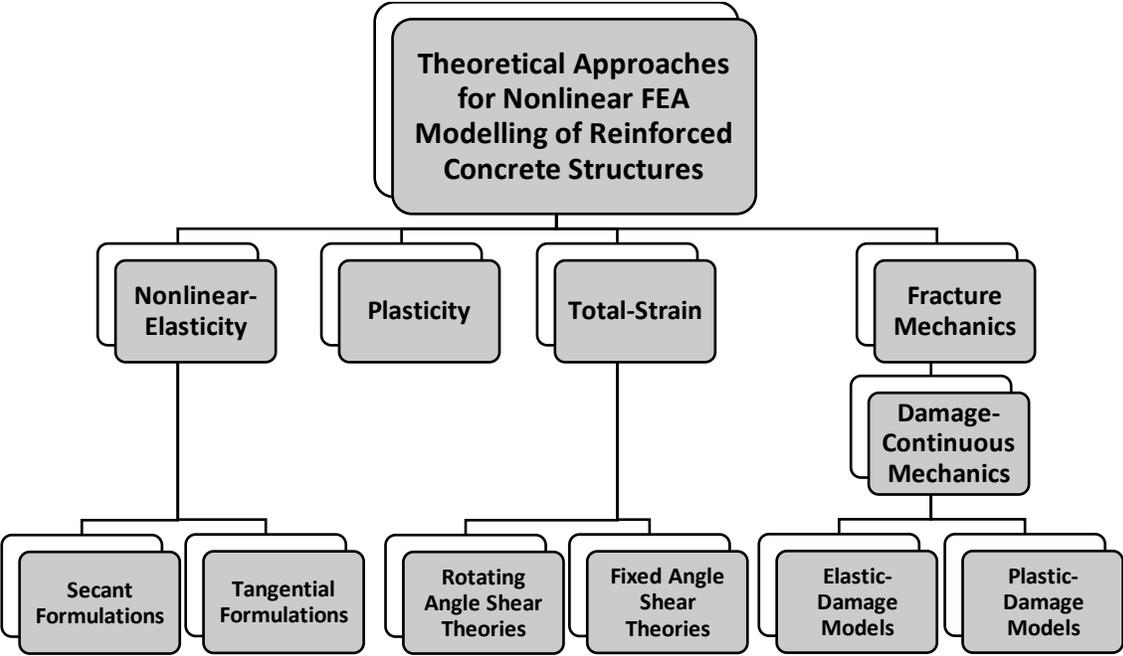


Fig. 2.6. Theoretical Approaches for Nonlinear FEA Modelling of RC Structures

2.3.1 Nonlinear-Elasticity-Based Approach

The earliest models created to represent RC were based on linear behavior. Later on, cracking and compression softening phenomena were incorporated to account for nonlinear behavior. The elastic-based theoretical approach can be divided into two main subcategories to determine the constitutive matrix $[Γ]$: secant (Eq. 2.1) and tangential (Eq. 2.2) stress-strain models (ASCE, 1982), where $\{\sigma\}$ and $\{\varepsilon\}$ are the stress and strain vectors, and $\{d\sigma\}$ and $\{d\varepsilon\}$ are the stress and strain increment vectors.

$$\{\sigma\} = [Γ]\{\varepsilon\} \quad (2.1)$$

$$\{d\sigma\} = [Γ]\{d\varepsilon\} \quad (2.2)$$

The main difference between the two approaches is that for the secant formulation the current state of stress $\{\sigma\}$ is exclusively dependant on the current state of strains $\{\varepsilon\}$ or vice versa, not accounting for deformation-path dependency. Thus, limitations for the secant formulation arise, since the deformation state of these models is independent of the loading path, which is not true for concrete (Chen, Yamaguchi, Kotsovos, & Pan, 1993). Bazant and Tsubaki (1980) stated that the development of microcracks in concrete varies when a different loading path is used. Imran and Pantazopoulou (1996) conducted experimental tests on concrete under triaxial loading, which allowed them to prove that the deformation of concrete is loading path dependant, while the strength of concrete is independent to the loading path. Despite the inherent limitations, secant formulations have been widely used for their simplicity

By contrast, the tangential formulation depends on the current state of strains and the path of stress followed, depending on the deformation history of the material. However, for this

type of formulation concrete behavior becomes anisotropic in the highly nonlinear range near the peak stress, even if the initial behavior was isotropic, which makes the formulations of this type of constitutive model more complex.

For uniaxial models, using one approach over another does not make a substantial difference due to their simplicity; the stress is a function of a single strain value. Three examples of uniaxial RC elasticity-based models are described in the sections that follow. In sections 2.3.1.1 and 2.3.1.2, two of the most commonly used uniaxial compression modes for FEA, the Hognestad model (1951) and the Kent and Park model (1971), are described. In section 2.3.1.3, a uniaxial tension model, the flexural tension-stiffening relationship (Kaklauskas, 1999), is described.

2.3.1.1 Hognestad Model and Modified Hognestad Model

The Hognestad model (Hognestad, A study on combined bending and axial load in reinforced concrete members, 1951) is one of the simplest elastic-based uniaxial models. The model uses a second-order parabola to describe the stress-strain relationship of concrete at any given strain (ε_c). The parabola is defined (Eq. 2.3) using two variables: the concrete compressive strength (f'_c) and the concrete strain at maximum concrete strength (ε_{c0}) defined by Eq. (2.4).

$$f_c = f'_c \left[\frac{2\varepsilon_c}{\varepsilon_{c0}} - \left(\frac{\varepsilon_c}{\varepsilon_{c0}} \right)^2 \right] \quad 0 \leq \varepsilon_c \leq 2\varepsilon_{c0} \quad (2.3)$$

$$\varepsilon_{c0} = \frac{f'_c}{E_c} \quad (2.4)$$

The compressive stress-strain curve of concrete according to the Hognestad model is shown in Fig. 2.7. The stress-strain relationship up to the f'_c is accurately represented, but the post-

peak branch demonstrates symmetry to the ascending branch, which is not true for uniaxial concrete behavior.

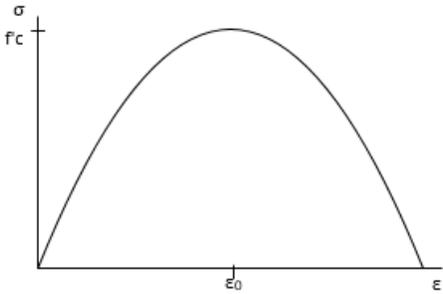


Fig. 2.7. Stress-strain relationship for concrete in compression (Hognestad, A study on combined bending and axial load in reinforced concrete members, 1951)

The modified Hognestad model (Hognestad, 1955) was created to improve the representation of the descending branch of the uniaxial stress-strain concrete curve. The ascending curve is represented with the same equation as the original model (Eq. 2.3), the descending portion is considered linear until the concrete crushes at 85% of f'_c . Fig. 2.8 shows the compressive stress-strain curve of concrete according to the modified Hognestad model.

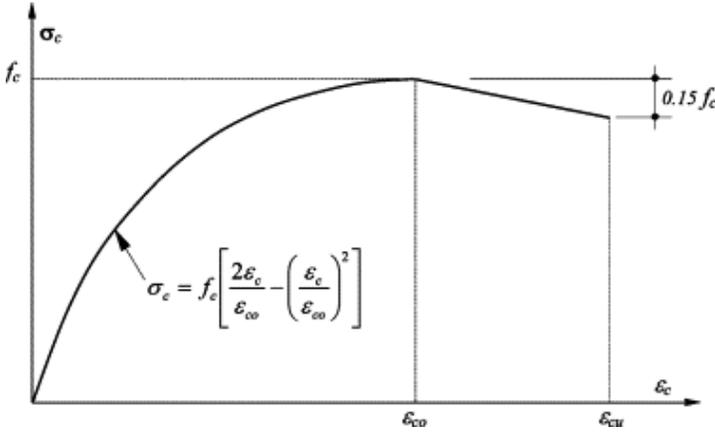


Fig. 2.8. Stress-strain relationship for concrete in compression (Hognestad, 1955)

2.3.1.2 Kent and Park Model

Kent and Park (1971) developed a model to describe the stress-strain relationships of concrete, both unconfined and confined by rectangular hoops. The ascending branch is described by the Eq. (2.3), taken from the Hognestad model, with the difference being that it is limited up to the maximum strength strain (ε_{c0}), which is given a constant value of 0.002. The post-peak branch is assumed to be linear, and its slope is defined by Eq. (2.5) and (2.6) as a function of the peak strength (f'_c), the strain at peak strength (ε_{c0}), the strain at which concrete crushes (ε_u) and the stress equals 50% of the maximum concrete strength (ε_{50u}).

$$f_c = f'_c [1 - Z(\varepsilon_c - \varepsilon_{c0})] \quad \varepsilon_{c0} \leq \varepsilon_c \leq \varepsilon_u \quad (2.5)$$

In which,

$$Z = \frac{0.5}{\varepsilon_{50u} - \varepsilon_{c0}} \quad (2.6)$$

Fig. 2.9 shows the compressive stress-strain curve of concrete according to the Park and Kent model, which accurately describes the ascending and descending branches. However, for the confined case it is conservative because it does not take the strength's increase at the peak of the curve (Mander, Priestly, & Park, 1988) into account. Because of its simplicity and accuracy, this model has been widely used for FEA of concrete structures.

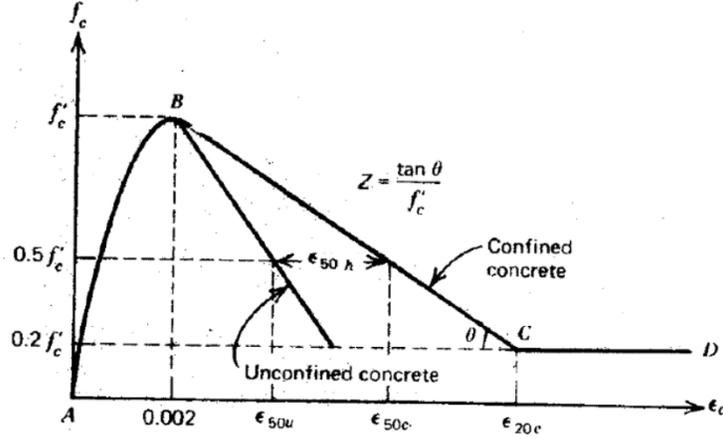


Fig. 2.9. Stress-strain relationship for unconfined and confined concrete in compression (Kent & Park, 1971)

2.3.1.3 Flexural Tension-Stiffening Relationship

A flexural tension-stiffening relationship has been proposed by Kaklauskas (1999). It is called flexural because it has been derived using flexural experimental data. The descending branch of the stress-strain relationship for tensile concrete is described by Eq. (2.7), (2.8), and (2.9), where the tensile stress (σ_t) is described at any current tensile strain (ϵ_t) as a function of the maximum tensile stress or cracking stress (σ_{cr}), the strain at the maximum tensile stress (ϵ_{cr}), and the reinforcement ratio (p).

$$\sigma_t = 0.625 \sigma_{cr} \left(1 - \frac{\bar{\epsilon}_t}{\beta} - \frac{1+0.6\beta}{\beta \bar{\epsilon}_t} \right) \quad (2.7)$$

In which,

$$\bar{\epsilon}_t = \frac{\epsilon_t}{\epsilon_{cr}}; \quad \epsilon_{cr} = \frac{\sigma_{cr}}{E_c} \quad (2.8)$$

$$\beta = 32.8 - 27.6 p + 7.12 p^2 \quad (2.9)$$

Fig. 2.10 shows the stress-strain relationship for tensile concrete in flexure. This model was derived from a number of fitting curves, with the aim of achieving a balance between accuracy and simplicity (Kaklauskas, 1999).

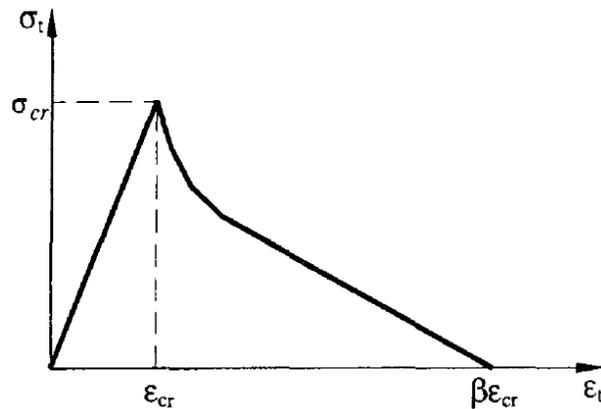


Fig. 2.10. Stress-strain relationship for tensile concrete in flexure (Kaklauskas, 1999)

2.3.2 Plasticity-Based Approach

The strain-stress relationships of concrete can be separated into elastic and inelastic ranges. The recoverable deformation, or elastic range, is well understood and is based on the framework of elasticity theory. The irrecoverable deformation, or inelastic range, is treated within the theory of plasticity. Three basic assumptions exist in the development of the classical theory of plasticity: 1) an initial yield surface, 2) a hardening rule, and 3) a flow rule. The first assumption determines the stress threshold for the onset of plastic deformation; the second regulates the evolution of the subsequent loading surface, which changes its configuration according to the specified hardening rule that describes the plastic stress-strain relationships; and the third defines an incremental plastic stress-strain relationship (Chen & Han, 1988).

Concrete is made of a mortar-aggregate composite that presents voids and microcracks even before loading, thus, it may behave more like a fracture or frictional material than a plastic material. However, the well-established theory behind the classical plasticity approach allows going beyond the elastic range in a theoretically consistent way, making the plasticity-based approach superior to elasticity-based for general concrete behaviors, such as permanent deformations (Han & Chen, 1985).

A number of plasticity-based models have been developed to describe the stress-strain relationship of concrete. An elasto-plastic and fracture model (Maekawa, Takemura, Irawan, & Irie, 1993), used by the DIANA FEA software package to predict concrete behavior for 2D and 3D elements, is described in Section 2.3.2.1.

2.3.2.1 Elasto-Plastic and Fracture Model

A 3D elasto-plastic and fracture model was formulated (Maekawa, Takemura, Irawan, & Irie, 1993) by combining the elasticity, fracture, and plasticity of concrete within the framework of plasticity and fracture laws. Four material functions are proposed for concrete with normal aggregate and strength ranging from 15 MPa to 50 MPa. The fracture function (damage parameter, K) (Eq. 2.10) express the degradation of the shear elastic strain energy of concrete including defects. The equivalent elastic strain parameter (F) (Eq. 2.11) represents the macroscopic intensity of internal stress which advances the damage under an arbitrary level of confinement. The function H (Eq. 2.12) indicates the plastic hardening of the internal plastic element in the damaged concrete, with b being the user-defined correction factor for plastic evolution, which has a default value of 1. The function D (Eq. 2.13) indicates the plastic dilatancy induced by the shear plastic dislocation along the internal defects.

$$K = \exp\left(-\frac{F}{3.25} \left(1 - \exp\left(-\frac{F}{0.8}\right)\right)\right) \quad (2.10)$$

$$F = \frac{\sqrt{2} J_{2e}}{5(0.23\varepsilon_0 - \sqrt{3} I_{1e})} \left(\frac{3\sqrt{3}}{2} \left(\frac{J_{3e}}{J_{2e}}\right)^3 + 6\right) \quad (2.11)$$

$$H = \frac{9}{10} b \varepsilon_0 \left(\frac{J_{2e}}{\varepsilon_0}\right)^3 \quad (2.12)$$

$$D = \frac{-1+2\nu}{\sqrt{3}(1+\nu)} 4K^2 + \frac{\sqrt{2} I_{1e} + 0.38\varepsilon_0}{0.28\varepsilon_0} (1 - 4K^2) \quad (2.13)$$

The constant ε_0 (Eq. 2.14) was adopted with the objective that these material functions would be applicable to normal aggregate and strength concrete.

$$\varepsilon_0 = 1.6 (1 + \nu) \left(\frac{f'_c}{E_0}\right) \quad (2.14)$$

The scalars I_{1e} (Eq. 2.15), J_{2e} (Eq. 2.16), and J_{3e} (Eq. 2.17) are the first, second, and third elastic strain invariants respectively.

$$I_{1e} = \frac{1}{3} \varepsilon_{eii} \quad (2.15)$$

$$J_{2e} = \sqrt{\frac{1}{2} e_{eij} e_{eij}} \quad (2.16)$$

$$J_{3e} = \sqrt[3]{\frac{1}{2} e_{eij} e_{ejk} e_{eki}} \quad (2.17)$$

In which,

$$e_{eij} = \varepsilon_{eij} - I_{1e} \delta_{ij} \quad (2.18)$$

Once the elastic strain vector has been determined, then the so-called fracture parameter (K) is calculated as a function of the invariants of the elastic strain tensor and a number of elastic parameters. Due to damage, it is assumed that the shear modulus will be reduced by a factor K ; i.e. the initial shear modulus G is multiplied by K .

Function K ranges from 1 to 0, where 0 represents the failure of the material and 1 denotes undamaged material (Maekawa, Takemura, Irawan, & Irie, 1993).

2.3.3 Rotating and Fixed Angle Shear Theories Based on Total-Strain Approach

Total-strain models determine the average element stresses as a function of the average strains, using the material constitutive equations. The input for total-strain models is divided into two parts: 1) material properties such as Young's modulus and Poisson's ratio, and 2) the definition of the material behavior in compression, tension, and shear. The constitutive relationships and the compatibility strains depend on the behavior of both concrete and steel. RC 2D elements subjected to in-plane shear and normal stresses can be separated into concrete elements and steel grid elements (Fig. 2.11) (Hsu & Mo, Unified Theory of Concrete Structures, 2010).

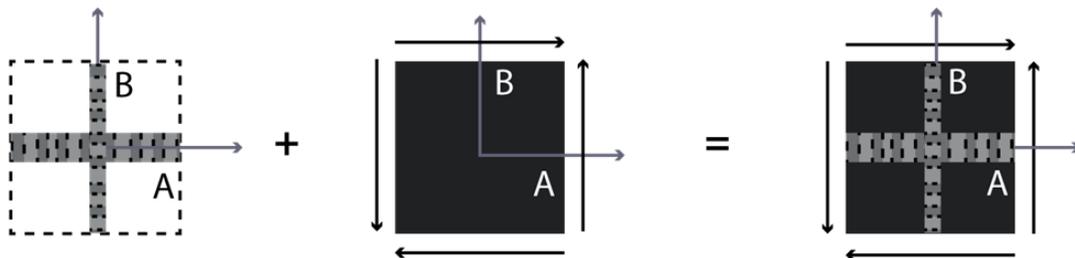


Fig. 2.11. Decomposition of an RC element into a concrete element and a steel grid element

Before cracking, the steel bars have a negligible effect on the behavior of an RC element, thus, the principal stresses in the concrete coincide with the applied principal stresses. When the principal tensile stress reaches the concrete tensile strength threshold, cracks will form in the direction of compressive stresses, activating the steel bars and modifying the principal stresses direction in the RC element (Hsu & Mo, Unified Theory of Concrete Structures, 2010). As a result, the direction of the subsequent cracks deviates from the direction of the initial crack, which occurs between the applied principal stresses and the RC element principal stresses (Fig. 2.12). Consequently, two theories have been developed: the rotating angle shear theories (Vecchio & Collins, 1982), and the fixed angle shear theories (Pang & Hsu, 1996).

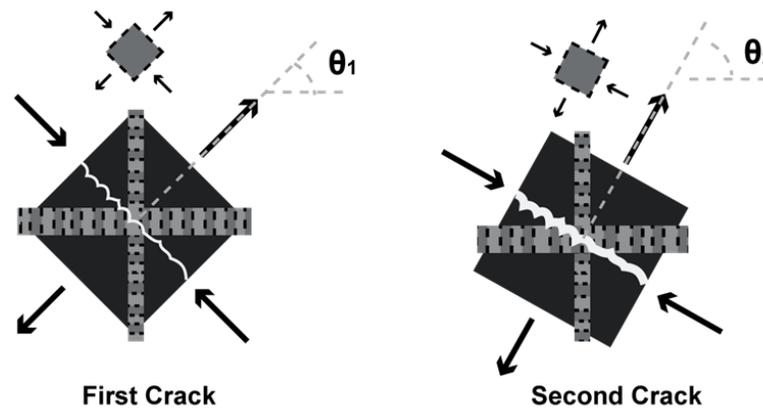


Fig. 2.12. Change of direction of the subsequent cracks

The rotating angle theories assume that the direction of the cracks is perpendicular to the principal tensile stress of the RC element, which will “rotate” after each subsequent crack. The derivations of the equilibrium and compatibility equations are based on the RC element principal stresses’ coordinate system. Conversely, the fixed angle theories assume that the

direction of the cracks is perpendicular to the applied principal tensile stress, and the derivations of the equilibrium and compatibility equations are based on the applied principal stresses' coordinate system (Hsu & Mo, Unified Theory of Concrete Structures, 2010).

Two of the most prominent rotating angle theories and fixed angle shear theories are the modified compression field theory (Vecchio & Collins, 1986) and the cyclic softened membrane model (Mansour & Hsu, 2005), respectively. The formulations of these models are presented in Sections 2.3.3.1 and 2.3.3.2.

2.3.3.1 Modified Compression Field Theory

Vecchio and Collins (1986) formulated the modified compression field theory model based on the rotating angle theory. This model handles cracked concrete as a new material with its own stress-strain characteristics. Equilibrium, compatibility, and stress-strain relationships are formulated based on average stresses and strains. Relationships between external loads and internal strains are achieved by an iterative procedure, for example, the Newton-Raphson method, using a secant stiffness matrix approach (Vecchio & Collins, 1986).

Before cracking, the concrete can be considered a homogenous material, where the principal compressive stresses, the principal tensile stresses, and the shear stress are the same. After cracking, the MCFT accounts for tensile stresses in the diagonal concrete struts. This allows for better agreement with experimental results than the original compression field theory (CFT) (Vecchio & Collins, 1986), in which it is assumed that the struts carry only compressive stresses.

The behavior of a reinforced concrete 2D element is described by solving a 15x15 system of equations, a brief summary is presented in Fig. 2.13. Eq. (2.19) to Eq. (2.23) represent the equilibrium of forces between the applied stresses and the concrete and steel stresses (Fig. 2.13). The geometric conditions (Fig. 2.13) are considered in Eq. (2.24) to Eq. (2.28), where the first three represent the relationships between the horizontal (ε_x) and vertical (ε_z) strains, and the principal compressive (ε_2) and tensile (ε_1) strains distributed in the rotated concrete element as average strains; the last two determine the crack widths (w), considering the principal tensile strain and the spacing between steel reinforcement. The material constitutive relationships (Fig. 2.13) are described by Eq. (2.29) to (2.32), which are built in terms of only principal stresses, thus, avoiding the need for a constitutive model for shear. However, Eq. (2.33) is needed to account for shear stress on cracked surfaces.

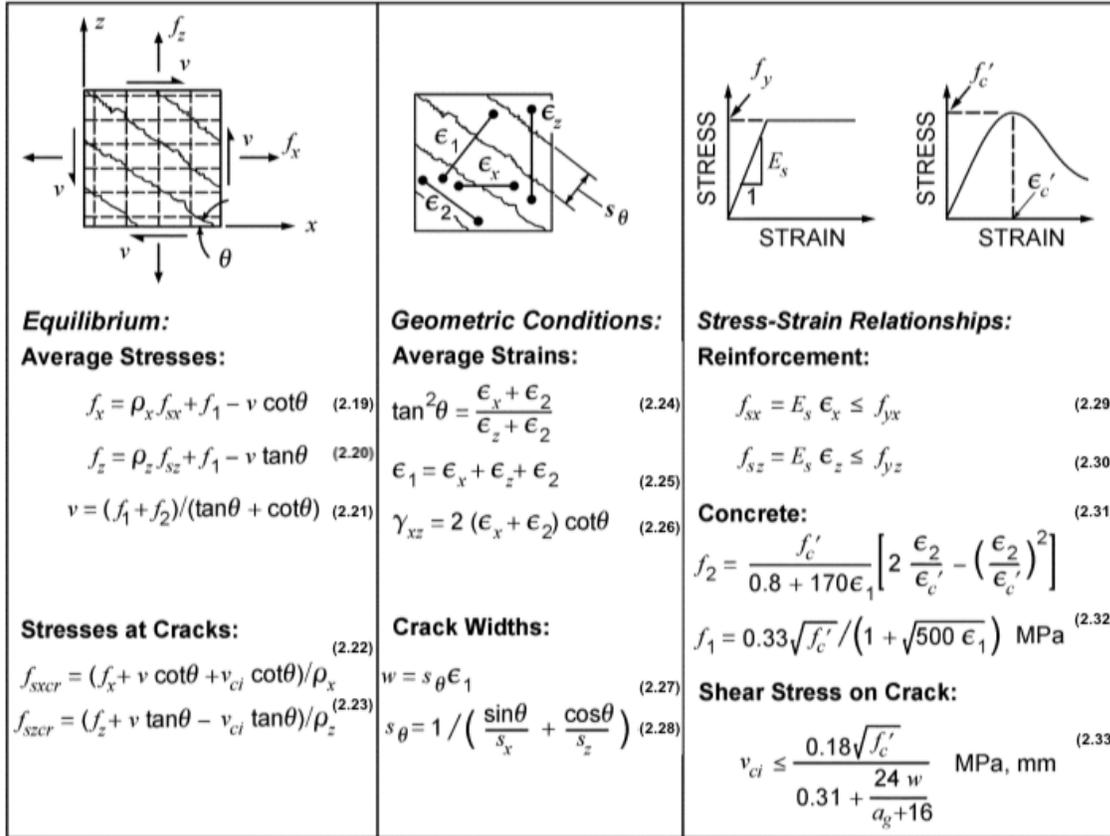


Fig. 2.13. Summary of the MCTF (Vecchio & Collins, 1986)

2.3.3.2 Cyclic Softened Membrane Model

The cyclic softened membrane model (CSMM) was developed by Mansour and Hsu (2005) as an extension of the softened membrane model (SMM) (Hsu & Zhu, 2002). These analytical models describe the behavior of 2D-RC elements under cyclic and monotonic loading, respectively, and are based on fixed angle theories. The CSMM is able to predict hysteretic loops and their pinched shapes, for which it accounts for the stress-strain relationships in the unloading and reloading paths of concrete and steel (Mansour & Hsu,

2005). Like the MCFT, this model is built on stress equilibrium, strain compatibility, and constitutive laws of materials.

A brief summary of this model is presented in Fig. 2.14. The constitutive model for steel accounts for the presence of concrete. The behavior of an embedded steel bar in concrete differs from a bare steel bar due to the transfer of tensile stresses between the bar and the concrete. The constitutive model for steel considers the tension softening after cracking and the softening of concrete compressive strength caused by tensile strains in the orthogonal direction. The compressive and tensile constitutive relationships are determined using equivalent uniaxial strains, which are calculated from the bi-dimensional tensor of strains using the Hsu/Zhu ratios (Mansour & Hsu, 2005).

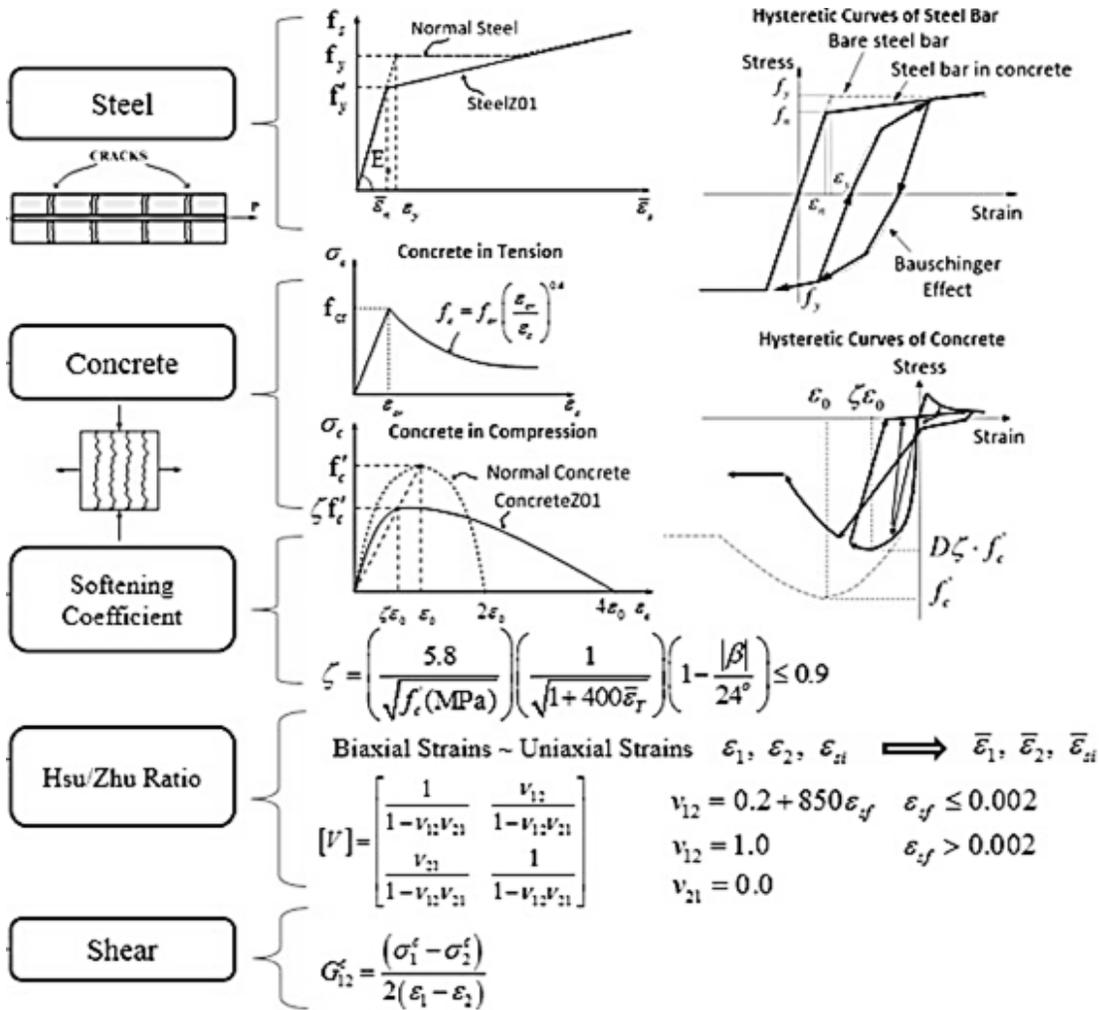


Fig. 2.14. Summary of the CMSS (adapted from Mansour & Hsu, 2005)

2.3.4 Damage-Continuous-Mechanics-Based Approach

Continuum damage mechanics, based on thermodynamics and fracture mechanics, is used to describe the formation and coalescence of microcracks in the concrete, known as damage (Kachanov, 1958). This approach assumes that the concrete stiffness degradation is caused by material damage, which can be represented by the damage variable, D . The damage variable (D) ranges from 0 for the undamaged material to 1 for the complete failure of the

material. The damage variable (D) is calculated using material formulations derived within the framework of thermodynamics. Fig. 2.15 shows the general principle of damage-based models, where the undamaged Young's modulus (E) of the material is 35 GPa. For the first case (E_1) the material is undamaged ($D=0$), being E_1 equal to the initial value of 35 GPa. For the second case (E_2) the damage (D) has a value of 0.82, causing stiffness degradation from an initial Young's modulus of 35 GPa to an E_2 of 6.34 GPa at a strain of 0.004.

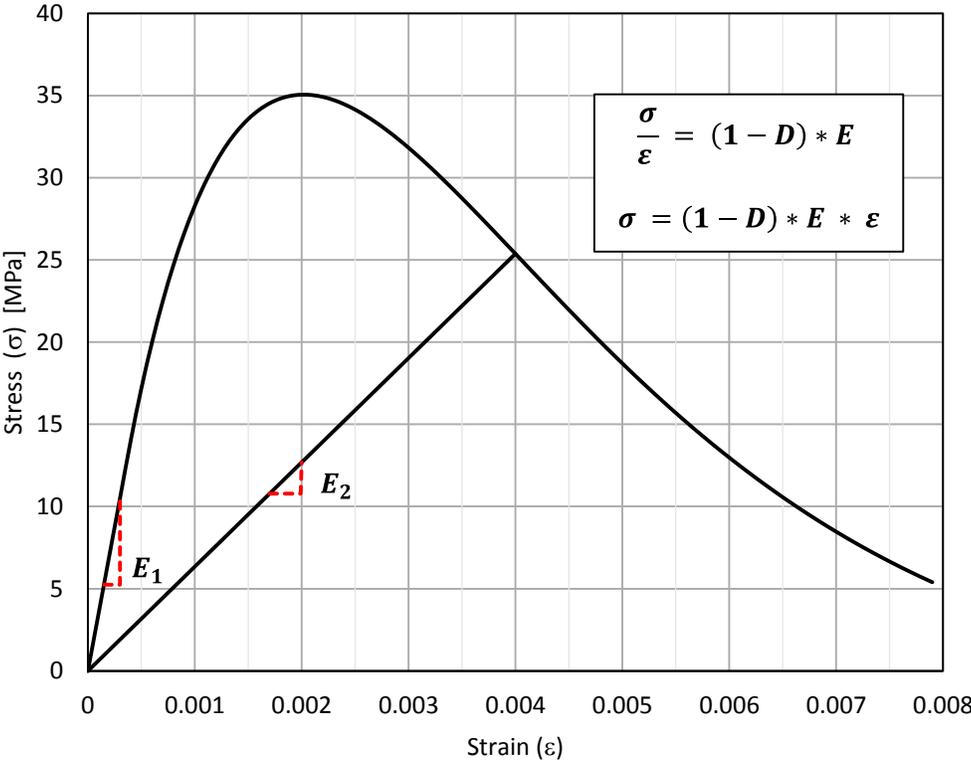


Fig. 2.15. Stress-strain response of damage models

Usually, there are two types of damage variables to describe the damage; one is the scalar damage variable (isotropic), which is related to the collapse of the micropores of the structure; the other is the tensorial damage (anisotropic), which is related to the creation of surfaces in the material caused by decohesion (Lemaitre, 1986).

Two types of concrete-damage models exist: the elastic-damage model and the plastic-damage model. The elastic-damage model describes the inelastic behavior of concrete based only on its stiffness degradation. Most models of this type are unable to account for the permanent deformation of the material after a full unloading cycle. Elastic-damage models consider only a pure extension crack, caused by tensile strains, to describe the behavior of the material in both tension and compression, where the compression is accounted for by tensile strains in the orthogonal direction of the applied load (Mazars, Hamon, & Grange, 2015). The same concept for multiaxial loading was extended by the “ μ ” model (Mazars, Hamon, & Grange, 2015) to describe the behavior of the material under low or moderate confinement, where the extension is allowed in at least one direction.

Plastic-damage models account for high confinement, where extension is not permitted, and thus generate the collapse of the cement porous matrix and shear cracking. Coupling plasticity and damage serves to describe the permanent strain evolution of the material (Ortiz & Popov, 1982; Lee, 1998). However, the PRM model, an elastic-damage based model created from the work of Pontiroli (1995), Rouquand (2005), and Mazars (1986), is able to account for permanent strains.

Isotropic-elastic-damage-based models are used in this project because of the simplicity they exhibit, considering the broad loading mechanism they can be used for. As Gerstle stated (1980), concrete experiment results are significantly scattered, even when identical results are intended, thus the simplicity of the constitutive models should be favoured over achieving perfect similitude of experimental and analytical data. A full discussion of the scalar damage model (Mazars, 1986), the PRM model (Pontiroli, Rouquand, & Mazars,

Predicting Concrete Behaviour From Quasi-static Loading to Hypervelocity Impact, 2010) and the “ μ ” model (Mazars, Hamon, & Grange, 2015) is presented in Chapter 3.

2.4 Structural Analysis Software Using the Finite Element Method (FEM)

The finite element method (FEM) has become a valuable tool for analyzing reinforced concrete structures. During the last 60 years, it has been under constant improvement regarding the constitutive modeling of RC behavior and advanced analysis algorithms (Vecchio & Palermo, 2001). FEM is widely used in professional structural design, for analyzing and designing new structures, and investigating existing structures. As Schlaich (1987) stated, “it has allowed us to design structures we would not have dared in the past.” The advantages and disadvantages of the following FEA software regarding the analysis of a component or a system (full structure) are discussed: VecTor Analysis Group, DIANA FEA, Abaqus FEA, and OpenSees.

2.4.1 VecTor Analysis Group

VecTor is a suite of computer programs developed at the University of Toronto that has been under continuous development over the last two decades. It is capable of performing accurate nonlinear analyses of 2D (VecTor2) and 3D (VecTor3 and VecTor4) shell elements under monotonic and dynamic loading, using rotating angle formulations based on the modified compression field theory (MCFT) and the disturbed stress field model (DSFM).

VecTor2 performs nonlinear finite element analysis of 2D membrane structures. The element library (Fig. 2.16) consists of the following element types: three-node triangular

elements, four-node rectangular elements, four-node quadrilateral elements, two-node truss-bar elements for discrete steel reinforcement, two-node link elements for steel bar slip, and four-node contact elements for contact interactions (Chak, 2013). This program contains 25 concrete material types and 25 steel material types, allowing up to four smeared reinforcement components per material type. However, the program is limited to 6000 elements and 5200 nodes plane structures.

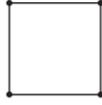
					
Triangular	Rectangular	Quadrilateral	Truss	Link	Contact

Fig. 2.16. VecTor2 element library (Chak, 2013)

VecTor3 was developed in as an analog of VecTor2, but specifically for describing the behavior of 3D solid finite elements, to accurately capture out-of-plane behavior. The element library (Fig. 2.17) includes: six-node wedge elements, eight-node rectangular and isoparametric hexahedral elements, two-node truss-bar elements, and two-node link elements (Chak, 2013). This program contains 45 concrete material types and 15 steel material types, allowing up to four smeared reinforcement components per material type. However, the program is limited to 12,000 elements and 15,000 nodes plane structures. It is also important to note that the 3D feature is only for solid elements, making it useful for modeling solid structures, like dams, but less functional for structures like buildings.

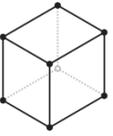
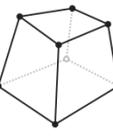
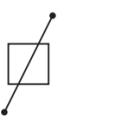
				
Wedge	Regular Hexahedral	Isoparametric Hexahedral	Truss	Link

Fig. 2.17. VecTor3 element library (Chak, 2013)

VecTor4 is used for the analysis of 3D-RC shell and plate elements, capable of accounting for nodal displacements, as well as in-plane rotations. It employs up to 15 layers in a layered element formulation. VecTor4 shell elements are relatively high-power elements, utilizing nine-node layered elements with 42 degrees of freedom. The element library (Fig. 2.18) consists of the following element types: nine-node shell elements, and two-node truss-bar elements. This program contains 20 concrete/steel material types and 15 concrete layers per material type, allowing up to 6 smeared reinforcement components per material type. However, it is limited to 6000 elements and 5200 nodes plane structures.

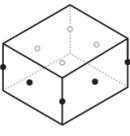
	
Shell	Truss

Fig. 2.18. VecTor4 element library (Chak, 2013)

The VecTor suite also includes a program that specializes in plane frame-related structures (VecTor5) and a program that performs analysis of 2D elements in an axisymmetric fashion (VecTor6), which are unrelated to this research.

VecTor2 and VecTor3 have shown feasible to analyse components of a structure, using an advanced concrete model based on the Modified Field Compression Theory (Vecchio & Collins, 1986). Defining the material properties is a straightforward process where you can obtain them from simple testing. Some of the most significant research that has been done using this software is summarised below. Quasi-static tests on RC shear walls were compared with the FEA model using VecTor2 (Mergos & Beyer, 2013), and showed excellent agreement in the load-displacement response. The punching shear behavior of externally prestressed concrete slabs have been studied (Mostafei, Vecchio, Gauvreau, & Semelawy, 2011), and reasonably accurate correlations were obtained between the VecTor3 analysis results and the test results. Comparisons have been made between beam impact tests and analysis results (Saatci & Vecchio, 2009), in which VecTor2 performed well at predicting displacements, damage levels, and reinforcement strains. The seismic over-strength of RC shear walls of parking structures was investigated by inelastic static pushover analyses and inelastic dynamic response analyses in VecTor2 (Lee & Kuchma, 2007). The shear-flexure interaction response was closely resembled by the analytical model.

However, this computer program suite limits the number of nodes and elements that can be used, and it cannot perform the nonlinear analysis of a full-scale structure or at the system level. VecTor Analysis Group programs also do not allow the modification or addition of new analysis modules to its source code, unlike open-source code.

2.4.2 DIANA FEA

DIANA FEA is a finite element software package that is dedicated to a wide range of problems arising in civil engineering including structural, geotechnical, tunneling, earthquake disciplines, and oil & gas engineering (TNO, 2016). Regarding reinforced concrete nonlinear analysis, DIANA offers a full range of material models for 1D, 2D, and 3D elements, such as smeared crack models with fixed and rotating angle theories and elasto-plastic models. DIANA can perform nonlinear analyses of RC subassemblies and full-scale structures under monotonic, cyclic, and seismic loading.

However, the use of advanced models for nonlinear 2D or 3D concrete elements that can account for permanent deformations has been restricted for subassemblies. A full-scale five-storey RC building was tested on a shake table at the University of California (Pantoli, et al., 2013) to study its performance under seismic loading. A finite element model was built on DIANA (Ebrahimian, Astroza, Conte, Restrepo, & Hutchinson, 2014) to simulate the performance of the mentioned building. The slabs and shear walls were modeled using a simplified version of the MCFT implemented by the user as a built-in subroutine, where the unloading and reloading paths always pass through the origin, preventing permanent deformation in the material. For high-intensity seismic loading, the FE model underestimated the floor displacement and, therefore, the inter-storey drift of the building at the lower levels.

The DIANA FEA Maekawa-Fukura concrete model has been developed as a combination of the elasto-plastic fracture model (Maekawa, Takemura, Irawan, & Irie, 1993), described in Section 2.4.2, the cracked concrete curves (Maekawa et al., 2003), and the three-dimensional extension of the MCFT (Selby & Vecchio, 1997), to make its implementation

in DIANA feasible. The validation of DIANA's material model was performed (Schreppers, 2017) by comparing analytical models with experimental tests taken from literature, which included a shear wall tested under reversed-cyclic loading (Aoyama et al., 1982). The analytical and experimental data for the load-displacement curve showed high accuracy. However, this concrete model has not been used for a full-scale structure.

DIANA FEA shows accurate results when predicting the nonlinear behavior of RC elements or structures under different types of loadings. It provides an excellent user-interface and access to some subroutines. However, the source code is proprietary, limiting the solution algorithms to the ones contained in the software, which may not be suitable for obtaining a solution for complicated models without convergence problems. DIANA FEA is mostly used for structural design, rather than for research purposes. One of the reasons is the relatively high cost of the software, which is less appealing to the research community in contrast to other specialized free research-oriented software that have been widely used, such as VecTor or OpenSees.

2.4.3 Abaqus FEA

Abaqus FEA is a suite of general-purpose software applications for finite element analysis and computer-aided engineering design, through which users can employ user material subroutines to implement their own nonlinear material models (Abaqus, 2009). Abaqus includes three constitutive concrete models: the smeared crack concrete model, the brittle cracking model, and the concrete damaged plasticity model.

The smeared crack concrete model is an elasto-plastic model, controlled by a “compression” yield surface. Cracking is regarded as the most important material behavior.

This material model is best applied when concrete is subjected only to monotonic loading. The brittle cracking model works best in situations where tensile cracking induces the failure of concrete. When compression occurs, the behavior is assumed to be elastic, and thus, dynamic analysis with load reversals is not suitable. The concrete damaged plasticity model is a damage-based material model that can be used when concrete undergoes loading conditions of any kind, including cyclic loading (Abaqus , 2009).

Studies on the performance of the FEM of 2D concrete elements using Abaqus have focused mainly on subassemblies. An RC railway bridge was tested under monotonic loading until failure occurred (Puurula, Enochsson, Sas, & Elfgren, 2015). An analytical model built on Abaqus showed excellent agreement for the load-carrying capacity of the bridge with the experimental data of the pushover performed. The behavior of an RC beam element under the effect of impact vibration was studied using the concrete damage-plasticity model, the results between experimental and analytical data showed good agreement for the midpoint displacements of the beam and the reaction forces on the supports (Ahmed, 2014). The nonlinear cyclic behavior of shear walls with composite steel-concrete was studied by Ali (2013), in which concrete damaged plasticity material was used in solid 3D elements. The predicted load-deformation curves, peak loads, and ultimate strength values exhibited good agreement with the experimental data.

The progressive collapse analysis of a high-rise building was studied using Abaqus (Fu, 2009). Being a general-purpose program places some limitations on the Abaqus preprocessor. For this reason, the model was built on ETABS (CSI, 2018) and then imported to Abaqus using a converter program developed by Fu (2009). The structure consisted of a 20-storey building, the lateral and vertical resisting load system consisted of

an RC shear-wall core and steel frames, and the floor system consisted of an RC slab. The shear walls and the slab were modeled using four-node shell elements, and the material used was the concrete damage plasticity model. However, for the validation, a 2-storey building was used, consisting of two frames in each orthogonal direction, with the ones in the longitudinal direction having two spans and the ones in the transverse direction having one span. The frames consisted of steel, and a concrete slab was placed on the intermedium level, without shear walls. The analytical data of the validation presented good agreement for the joints moment-rotation relationship with the experimental data.

Abaqus offers the analyst the opportunity to implement user-defined subroutines.

However, like DIANA, the source code of most of the solution and analysis modules is proprietary, making it challenging for researchers to analyze or verify the underlying code.

As a result, solution algorithms are constrained to those already present within the software, which may cause convergence problems in some situations. This limitation makes it impossible to analyze previously untapped engineering phenomena, or to analyze the behavior of newly discovered materials.

Similar to DIANA and other FEA programs, Abaqus has shown being feasible of performing FEA of subassemblies using advanced concrete models. However, some of the material parameters are difficult to determine without complex texting, and the theory of such models is not transparent, besides having a proprietary source code for the materials and solution algorithms. Regarding the FEA at the system level or full-structures, convergence issues may arise when using advanced concrete formulations.

2.4.4 OpenSees

Open System for Earthquake Engineering Simulation (OpenSees) is an object-oriented open source-code framework for simulation in earthquake engineering using FEA techniques (Fenves, 2001).

Many advanced finite element techniques appropriate for nonlinear finite element analysis of shell elements, have been implemented in OpenSees. A damage-based concrete material model has been implemented (Lu, Xie, Guan, & Lu, 2015) in OpenSees, which can be used to simulate the behavior of RC plane-stress elements in multilayer-shell elements. A biaxial concrete material model (based on the CSMM) has also been implemented (Zhong, 2005) in OpenSees, and has shown accurate prediction for the behavior of shear walls under reversed-cyclic loading.

These material models have shown being feasible for the FEA of both, subassemblies and full structures for different loading mechanism such as monotonic and dynamic loading. However, these two research groups have made their source code proprietary, so key aspects of the model performance are consequently unknown. After reviewing the main biaxial concrete models for different FEA software, the need of a simple, yet accurate material model is evident.

3 CONCRETE DAMAGE MODELS

3.1 Introduction

Concrete damaged-based models are formulated using continuum damage mechanics and aim to describe the nonlinear behavior of concrete. Many materials, including concrete, can exhibit internal failures at the micro- and macro- scale that are produced by effects such as creep, fatigue, constant load, and chemical reactions. These internal failures are produced in the form of microcracks, and their propagation and coalescence in concrete elements are known as “damage” (Kachanov, 1958).

Concrete is a composite material formed by granulates in a hydrated cement paste or brittle matrix. Damage mechanics is able to describe the interface between the aggregate grains and the cement matrix when they are subjected to loading. Damage mechanics is a simplified strategy for describing the behavior of concrete considering its complex microstructure.

The concrete models used in this research are based on the work done by Mazars through the last three decades. Three models will be discussed: the scalar damage model (Mazars, 1986), the PRM model (Pontiroli, Rouquand, & Mazars, 2010), and the third is the “ μ ” model (Mazars, Hamon, & Grange, 2015). In these concrete damage models it is assumed that the material is elastic, isotropic, and has constant stiffness. The stiffness of the material is modified using a scalar damage variable (D), which ranges from 0 for the undamaged material to 1 for the complete failure of the material. The damage variable (D) is calculated using material formulations derived within the framework of thermodynamics. The micro and macro effects of loading, collapse of micro-voids in the mixture, rearrangement of

concrete particles, and interaction of the cement matrix with the aggregates, are all accounted for by Mazars' models. Fig. 3.1 describes the behavior of stress-strain (σ - ϵ) curves for concrete elements using damage models, where the parameter E represents the Young's modulus of the material.

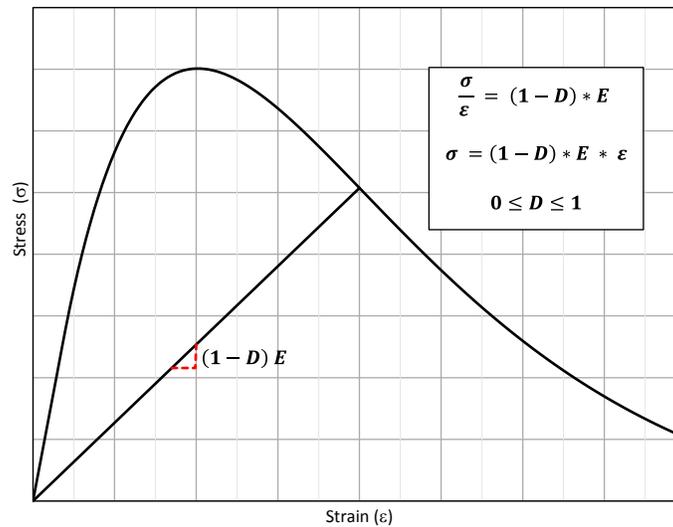


Fig. 3.1. Typical stress-strain relationship of concrete using damage models

3.2 Scalar Damage Model

Mazars (1986) formulated a scalar damage model to predict the triaxial behavior of concrete. This model describes the behavior of concrete as isotropic, elastic-damageable. The damage parameter (D) is calculated by starting from an equivalent strain (ϵ_{eq}), which is the average of the tensile principal strains of the element, meaning that all compressive strains can be represented as tensile strains in the orthogonal direction.

The calculation of the stresses uses elastic theory, reducing the elastic stiffness matrix $[Γ]$ with a damage parameter (D). The parameters $\{\sigma\}$ and $\{\varepsilon\}$ are the stress and the strain vectors, respectively.

$$\{\sigma\} = (1 - D)[Γ]:\{\varepsilon\} \quad (3.1)$$

The total damage of the element is comprised of the weighted sum of the damage caused by tensile stresses and compressive stresses. The modification factor (β) accounts for the presence of shear resistance in the interaction of compression (α_c) and tension (α_t) weights.

$$D = \alpha_t^\beta * D_t + \alpha_c^\beta * D_c; \quad 0 \leq D \leq 1 \quad (3.2)$$

The formulation of the damage in tension (D_t) and compression (D_c) is dependent on the tensile and compressive material parameters (A_c, B_c, A_t, B_t), which are obtained from compression and tensile tests. All the calculations are based on the equivalent strain of the element (ε_{eq}), and damage begins only when the strain surpasses the initial damage strain threshold (ε_{D0}).

$$D_t = 1 - \frac{\varepsilon_{D0} * (1 - A_t)}{\varepsilon_{eq}} - A_t * \exp[-B_t * (\varepsilon_{eq} - \varepsilon_{D0})] \quad (3.3)$$

$$D_c = 1 - \frac{\varepsilon_{D0} * (1 - A_c)}{\varepsilon_{eq}} - A_c * \exp[-B_c * (\varepsilon_{eq} - \varepsilon_{D0})] \quad (3.4)$$

The contribution of the tensile and compressive weights to the total damage are calculated by analyzing each of the principal strains obtained from the positive and negative principal elastic stresses. The weight is only considered if the total strain is tensile, hence the H parameter is used.

$$\alpha_t = \sum_{i=1}^3 H_i \frac{\varepsilon_{ti}(\varepsilon_{ti} + \varepsilon_{ci})}{\varepsilon_{eq}^2} \quad \text{where } H_i = 1 \text{ if } \varepsilon_i = \varepsilon_{ci} + \varepsilon_{ti} \geq 0, \text{ otherwise, } H_i = 0 \quad (3.5)$$

$$\alpha_c = \sum_{i=1}^3 H_i \frac{\varepsilon_{ci}(\varepsilon_{ti} + \varepsilon_{ci})}{\varepsilon_{eq}^2} \quad (3.6)$$

The elastic stiffness matrix $[G]$, and the positive (σ_+) and negative (σ_-) elastic stress vectors are used to calculate the positive (tensile) and negative (compressive) strain vectors, respectively.

$$\{\varepsilon_t\} = [G]^{-1} \cdot \{\sigma_+\} \quad (3.7)$$

$$\{\varepsilon_c\} = [G]^{-1} \cdot \{\sigma_-\} \quad (3.8)$$

The equivalent strain is calculated as the average of the tensile principal strains (ε_i) of the element.

$$\varepsilon_{eq} = \sqrt{\sum_{i=1}^3 \langle \varepsilon_i \rangle^2} \quad (3.9)$$

where $\langle \varepsilon_i \rangle = \varepsilon_i$ if $\varepsilon_i > 0$,

and $\langle \varepsilon_i \rangle = 0$ if $\varepsilon_i < 0$

3.3 PRM Model

The PRM model (2010) is a two scalar damage model formulated from work done by Pontiroli (1995), Rouquand (2005), and Mazars (1986). It improves upon previous models in that it retains the simplicity of an elastic-damage model but also has the capacity to account for crack-closure effects and permanent strains. The variables σ_{ft} and ε_{ft} are the crack closure stress and strain vectors, respectively. The PRM formulations are detailed below.

The constitutive equation of the model occurs in Eq. (3.10)

$$\{\sigma_d\} = (1 - D)[\Gamma]: \{\varepsilon_d\} \quad (3.10)$$

$$\{\sigma_d\} = \{\sigma\} - \{\sigma_{ft}\} \quad (3.11)$$

$$\{\varepsilon_d\} = \{\varepsilon\} - \{\varepsilon_{ft}\} \quad (3.12)$$

Before damage in compression occurs, the crack closure stress and strain are equal to the initial material parameters σ_{ft0} and ε_{ft0} respectively. Once compressive damage (D_c) occurs, the crack closure stress (σ_{ft}) is calculated from D_c as follows:

$$\{\sigma_{ft}\} = \{\sigma_{ft0}\}(1 - D)^2 \quad (3.13)$$

The PRM model uses the same equivalent strain concept as the scalar damage model, and is determined by Eq. (3.14). Where $\langle x_i \rangle$ depends on the sign of the stress in the direction i , being $\langle x_i \rangle = 0$ for compression and $\langle x_i \rangle = (\varepsilon - \varepsilon_{ft})_i$ for tension.

$$\varepsilon_{eq} = \sqrt{\sum_{i=1}^3 \langle x_i \rangle^2} \quad (3.14)$$

The damage evolutions were taken from the original scalar damage model (1984), with the main difference being that the PRM model has different thresholds for tensile damage (ε_{t0}) and compressive damage (ε_{c0}). The damage parameter (D) remains a scalar and is obtained from the calculation of D_c and D_t . The activation factor (α_t) evolves from 0 to 1 depending on the tensor σ_d , where $\alpha_t=1$ if $\sigma_d>0$ and $\alpha_t=0$ if $\sigma_d<0$:

$$D_t = 1 - \frac{\varepsilon_{t0}*(1-A_t)}{\varepsilon_{eq}} - A_t * \exp[-B_t * (\varepsilon_{eq} - \varepsilon_{t0})] \quad (3.15)$$

$$D_{cM} = 1 - \frac{\varepsilon_{c0}*(1-A_c)}{\varepsilon_{eq}} - A_c * \exp[-B_c * (\varepsilon_{eq} - \varepsilon_{c0})] \quad (3.16)$$

$$D_c = \frac{\varepsilon_{c0}*(1-A_c)}{\varepsilon_{eq}} \quad (3.17)$$

$$D = \alpha_t D_t + (1 - \alpha_t) D_c \quad (3.18)$$

Fig. 3.2 shows the uniaxial stress-strain response of the PRM model. The unloading in compression follows the focal point (ϵ_c, σ_c) , and the damage switches from compression to tension at the crack closure point $(\epsilon_{ft}, \sigma_{ft})$.

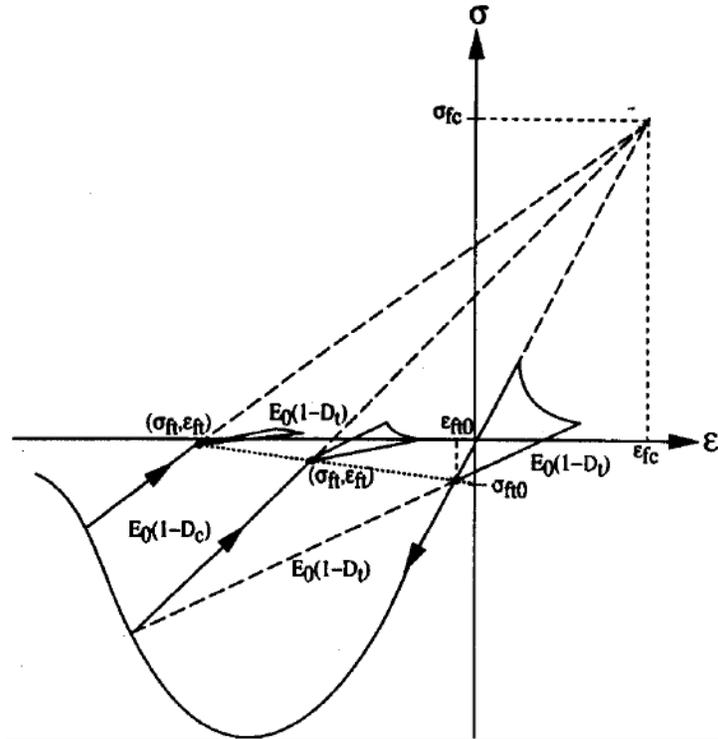


Fig. 3.2. Uniaxial stress-strain response of the PRM damage model (Mazars & Grange, 2014)

3.4 “μ” Model

The “μ” Model (Mazars, Hamon, & Grange, 2015) was created to include the damage effects related to monotonic and cyclic loading that were not incorporated in previous models, such as unilateral effects (crack opening and closing). It has proven capable of describing a broad range of nonlinear behavior: monotonic, cyclic, and dynamic loading. It entails the following assumptions:

- Describes the behavior of concrete as the combination of damage and elasticity.
- The damage behavior is assumed to be isotropic.
- Two damage modes are assumed: cracking (tension) and crushing (compression).
This leads to having two independent equivalent strains: one for tension and another for compression.
- In contrast with the scalar damage model, the effective damage parameter (d) describes the damage on the stiffness, activated either by compressive or tensile loading.
- d is able to describe the unilateral effects (crack opening and closure).

The calculations needed to determine the damage parameters are shown below. The stress vector is obtained using the same Eq. (3.1) as the Mazars’ scalar damage model (1986).

The equivalent strain for cracking (ε_t) and crushing (ε_c) are defined as follows, where ν is Poisson’s ratio:

$$\varepsilon_t = \frac{I_\varepsilon}{2(1-2\nu)} + \frac{\sqrt{J_\varepsilon}}{2(1+\nu)} \quad (3.19)$$

$$\varepsilon_c = \frac{I_\varepsilon}{5(1-2\nu)} + \frac{6\sqrt{J_\varepsilon}}{5(1+\nu)}, \quad (3.20)$$

$$I_\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3 \quad (3.21)$$

$$J_\varepsilon = 0.5[(\varepsilon_1 - \varepsilon_2)^2 + (\varepsilon_2 - \varepsilon_3)^2 + (\varepsilon_3 - \varepsilon_1)^2] \quad (3.22)$$

Y_t and Y_c are the maximum values reached during the loading path, while ε_{0t} and ε_{0c} are the initial threshold of the cracking and crushing equivalent strains, respectively:

$$Y_t = \text{Sup}[\varepsilon_{0t}, \max \varepsilon_t] \quad (3.23)$$

$$Y_c = \text{Sup}[\varepsilon_{0c}, \max \varepsilon_c] \quad (3.24)$$

The damage parameter (d) is directly related to the thermodynamic variables Y_t and Y_c through the Y variable. The triaxial factor (r) evolves from 0 for pure compressive stress to 1 for pure tensile stress:

$$Y = rY_t + (1 - r)Y_c \quad (3.25)$$

$$r = \frac{\sum \langle \bar{\sigma}_i \rangle_+}{\sum |\bar{\sigma}_i|}, \quad \text{where} \quad \{\bar{\sigma}\} = \frac{\{\sigma\}}{(1-d)} = [I]:\{\varepsilon\} \quad (3.26)$$

The damage evolution is defined in accordance with Mazars' scalar damage model, where Y_0 is the initial threshold for Y . A and B define the shape of the effective damage (d), based on the test parameters A_c , B_c , A_t , and B_t .

$$d = 1 - \frac{Y_0 * (1-A)}{Y} - A * \exp[-B * (Y - Y_0)] \quad (3.27)$$

$$Y_0 = r\varepsilon_{0t} + (1 - r)\varepsilon_{0c} \quad (3.28)$$

$$A = A_t(2r^2(1 - 2k) - r(1 - 4k)) + A_c(2r^2 - 3r + 1) \quad (3.29)$$

$$B = r^{(r^2-2r+2)}B_t + (1 - r^{(r^2-2r+2)})B_c \quad (3.30)$$

$$k = \frac{0.5}{A_t} \text{ or } 0.7 \quad (3.31)$$

Fig. 3.3 shows the evolution of the variables when subjected to a tension-compression loading path. It can be observed that the thermodynamic variables Y_t and Y_c evolve independently, and that for the entire loading path they are always increasing. The effective damage (d) is activated by local stress through the variable r , where for uniaxial loading it has a value of 1 for tensile stresses and 0 for compressive stresses, and for biaxial loading, the value ranges from 0 to 1. It can be observed that the damage is equal to zero during ($t_0 - t_1$), until the tensile threshold is reached and the damage starts evolving proportionally to the tensile strain until it reaches its maximum value at t_2 . During the unloading ($t_2 - t_3$), the damage is constant. The damage is equal to zero when the loading switches to compression at t_3 , then it is kept as zero until it reaches the compressive threshold at t_4 . The damage evolves proportionally to the compressive strain until it reaches its maximum value at t_5 , and again is constant during the unloading. If there were an extra tensile cycle after t_6 , the damage would switch to the value it reached at t_2 and would continue being constant until it surpassed the tensile strain at t_2 , where the damage would start increasing again.

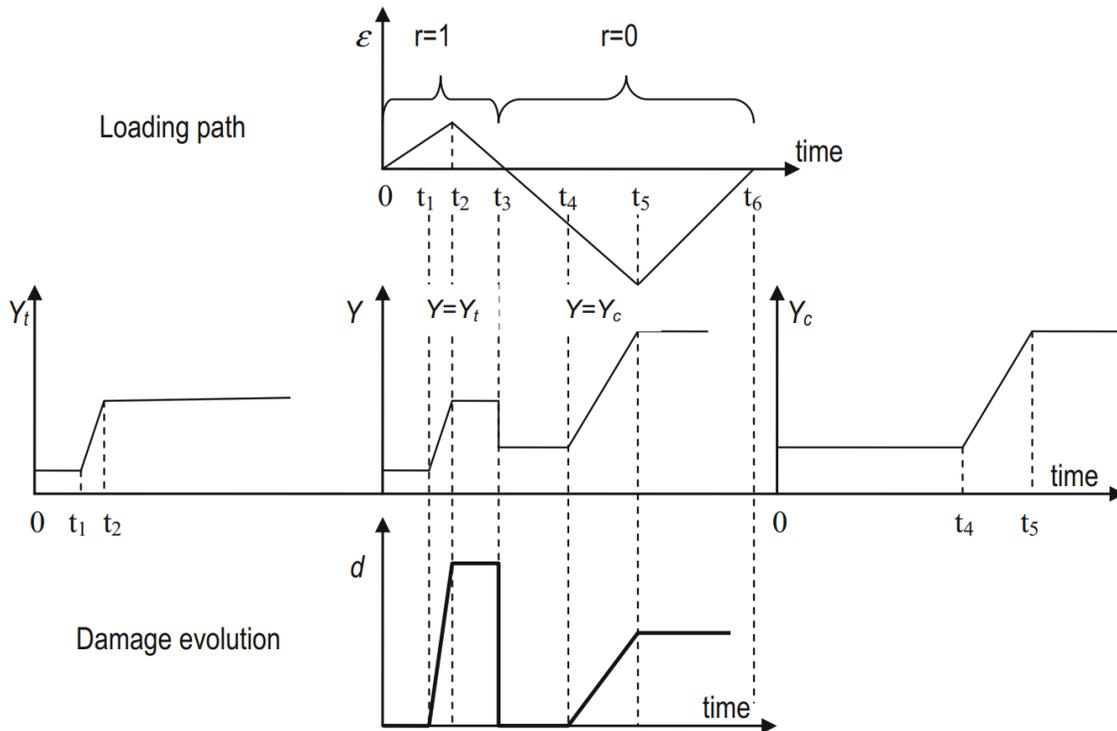


Fig. 3.3. μ model variables evolution during a tensile-compression loading path (Mazars, Hamon, & Grange, 2015)

3.5 Influence of Test Parameters (A_c , B_c , A_t , and B_t) and Uniaxial Stress-Strain Response of the Scalar Damage Model, the PRM Model, and the “ μ ” Model

The characteristics parameters A_c , B_c , A_t , and B_t affect the magnitude of the scalar damage parameter (D) directly. The variables A_c and B_c are obtained from compressive tests, while A_t and B_t come from flexion tests. The quasi-brittle behavior of concrete under tension is reproduced by parameters A_t and B_t , which are adjusted to accurately represent the uniaxial stress-strain curve obtained from a concrete specimen tensile test. The behavior of concrete in compression is reproduced by parameters A_c and B_c , which are adjusted to accurately represent the uniaxial stress-strain curve obtained from a concrete specimen compressive test.

To investigate the influence of the response parameters on the constitutive stress-strain response of concrete, a numerical investigation is performed. The parameters of the scalar damage model, PRM model, and “ μ ” model are calibrated (Table 3.1) to obtain a typical stress-strain relationship for concrete. The concrete properties are arbitrarily chosen as $f'_c = 35$ MPa at a strain value of 0.002, a Young’s modulus of 35 GPa, and a tensile stress-strain relationship with a peak tensile strength (f_t) of 3.5 MPa at a strain value of 0.0001.

Table 3.1. Damage models parameters

Material	E(GPa)	ε_{0c}	$\varepsilon_{0t}/\varepsilon_{D0}$	ε_{fc}	σ_{fc} (MPa)	ε_{ft0}	σ_{ft0} (MPa)
SD Model	35.0	-	1.0e-4	-	-	-	-
PRM Model	35.0	1.0e-4	1.0e-4	5.0e-4	18.1	-3.3e-5	-1.1946
μ Model	35.0	3e-4	1.0e-4	-	-	-	-

Material	A_c	B_c	A_t	B_t	ν
SD Model	1.20	1950.0	0.97	1.0e4	0.18
PRM Model	1.15	1950.0	0.97	1.0e4	0.18
μ Model	1.25	495.0	0.97	1.0e4	0.18

The test parameters A_c (Fig. 3.4), B_c (Fig. 3.5), A_t (Fig. 3.6), and B_t (Fig. 3.7) of each model are varied to exhibit the influence that each one has in the stress-strain response of the material. The test parameters A_c and B_c are modified to 50%, 75%, 125%, and 150% of the original value shown in Table 3.1. The test parameter A_t is modified to the values of 0.0, 0.25, 0.5, 0.8, and 1.0. The test parameter B_t is modified to the values of 1.0e+03, 3.0e+03, 7.0e+03, 2.0e+04, and 1.0e+05. The results show that the parameters have similar effects on the compressive and tensile response of the three models.

Fig. 3.4 shows that when A_c increases, the maximum compressive stress increases, but the material presents a more brittle behavior, and loses its load-carrying capacity at a smaller strain. When A_c decreases, the maximum compressive stress decreases, but the load-carrying capacity is improved, resulting in more ductile behavior. Fig. 3.5 shows that when B_c increases, both the maximum compressive stress and the load-carrying capacity increase. Typical values for A_t ranges from 0.7 to 1.0. Fig. 3.6 shows that A_t controls the residual tensile strength in the concrete; as the value of A_t increases the residual tensile strength increases. A value of 1.0 A_t denotes zero residual tensile strength and a value of 0.0 A_t implies that non load-carrying capacity is lost. Typical values for B_t ranges from 8000 to 21000. When B_t decreases, both the maximum tensile stress and the strain at which it is reached increase (Fig. 3.7).

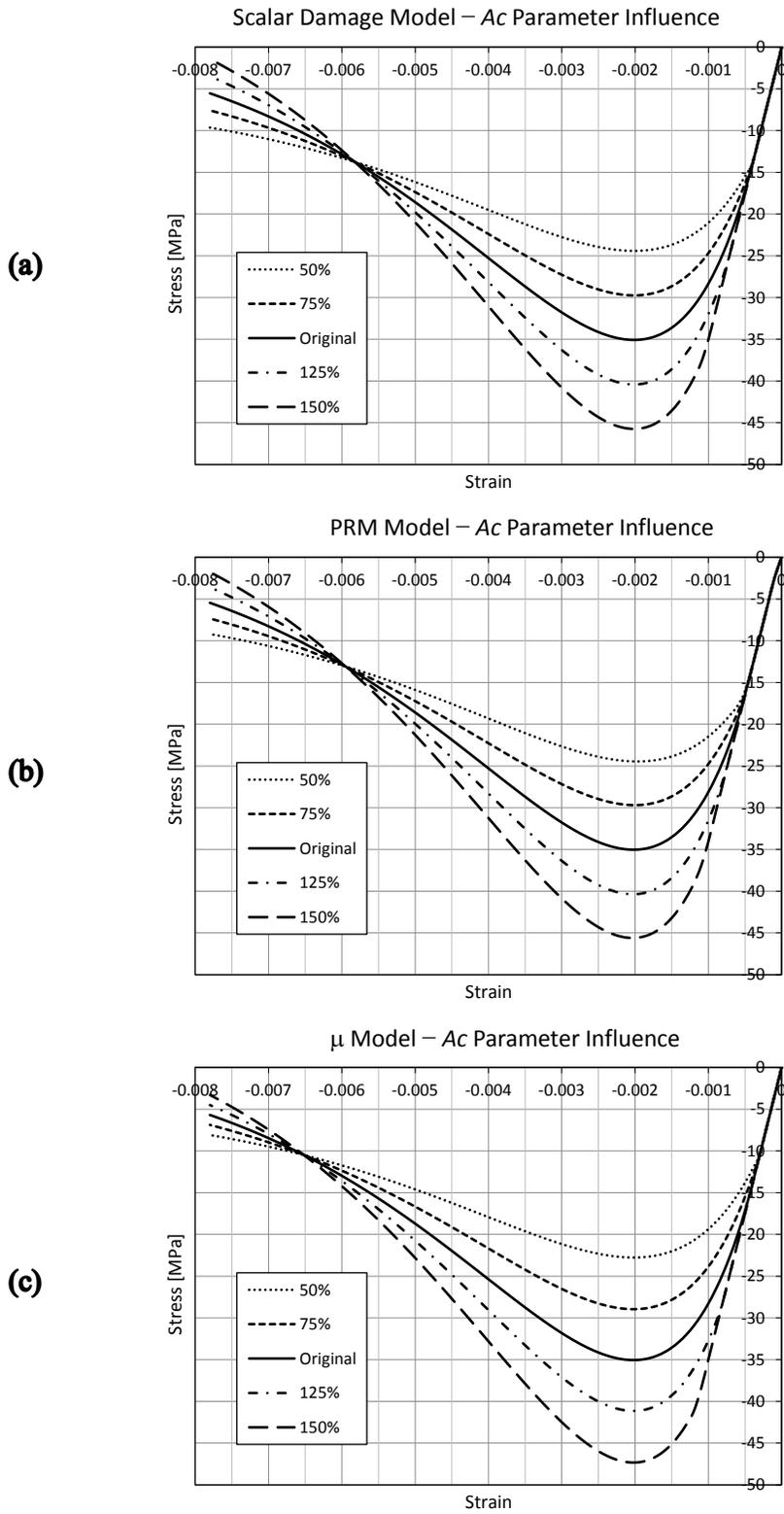


Fig. 3.4. A_c parameter influence in the compressive stress-strain response for the scalar damage model (a), the PRM model (b), and the μ model (c)

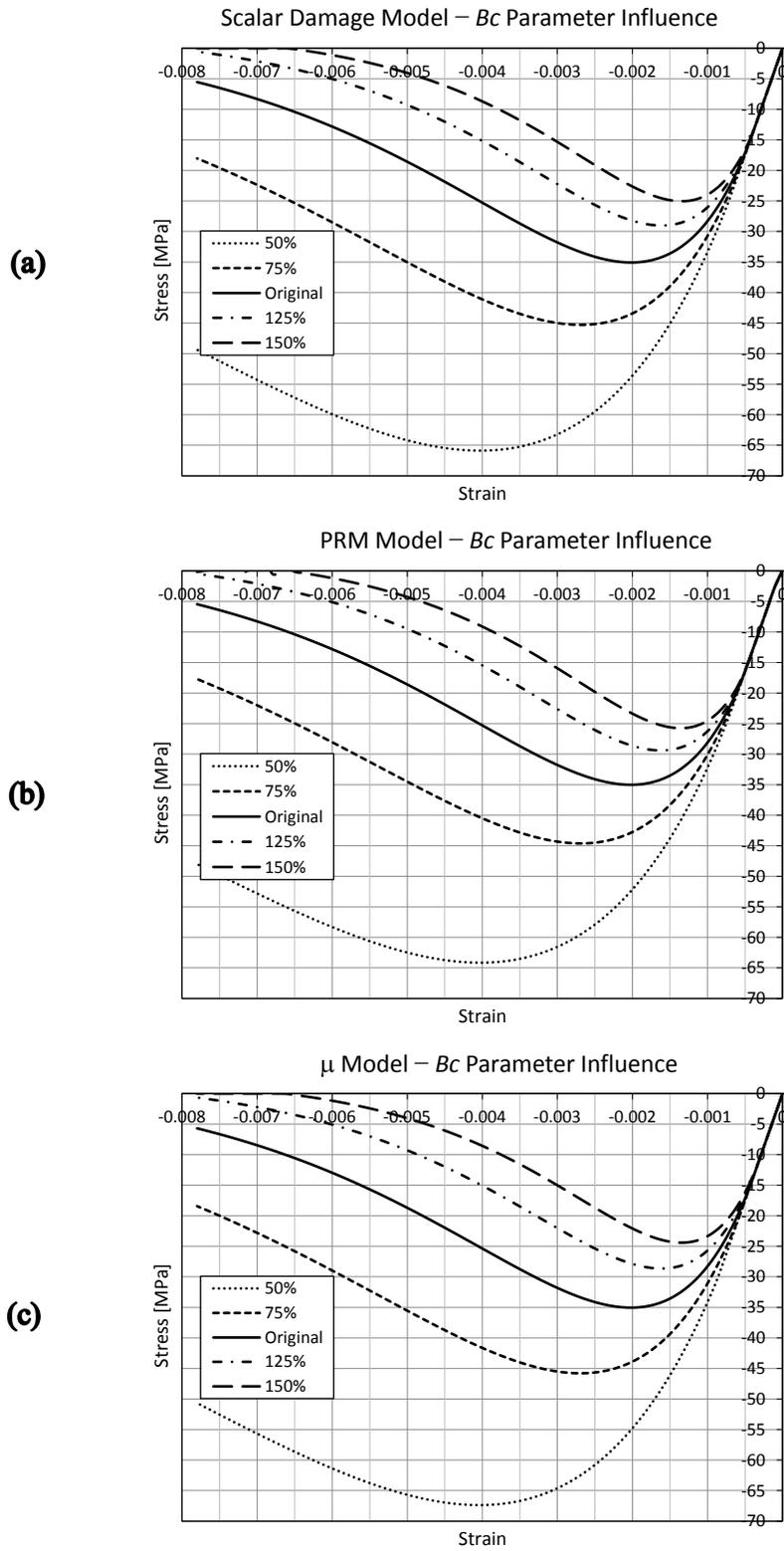


Fig. 3.5. B_c parameter influence in the compressive stress-strain response for the scalar damage model (a), the PRM model (b), and the μ model (c)

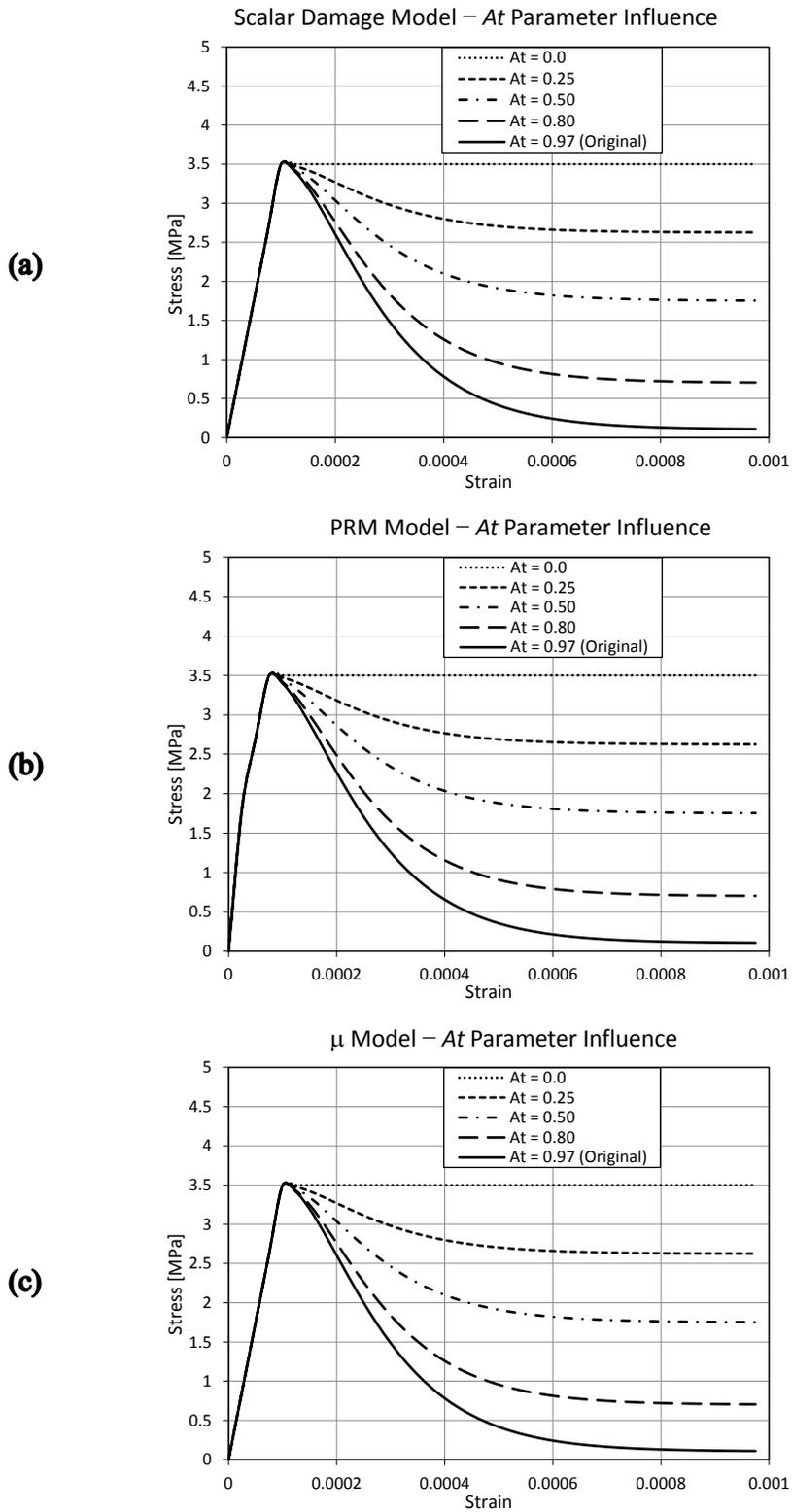


Fig. 3.6. A_t parameter influence in the tensile stress-strain response for the scalar damage model (a), the PRM model (b), and the μ model (c)

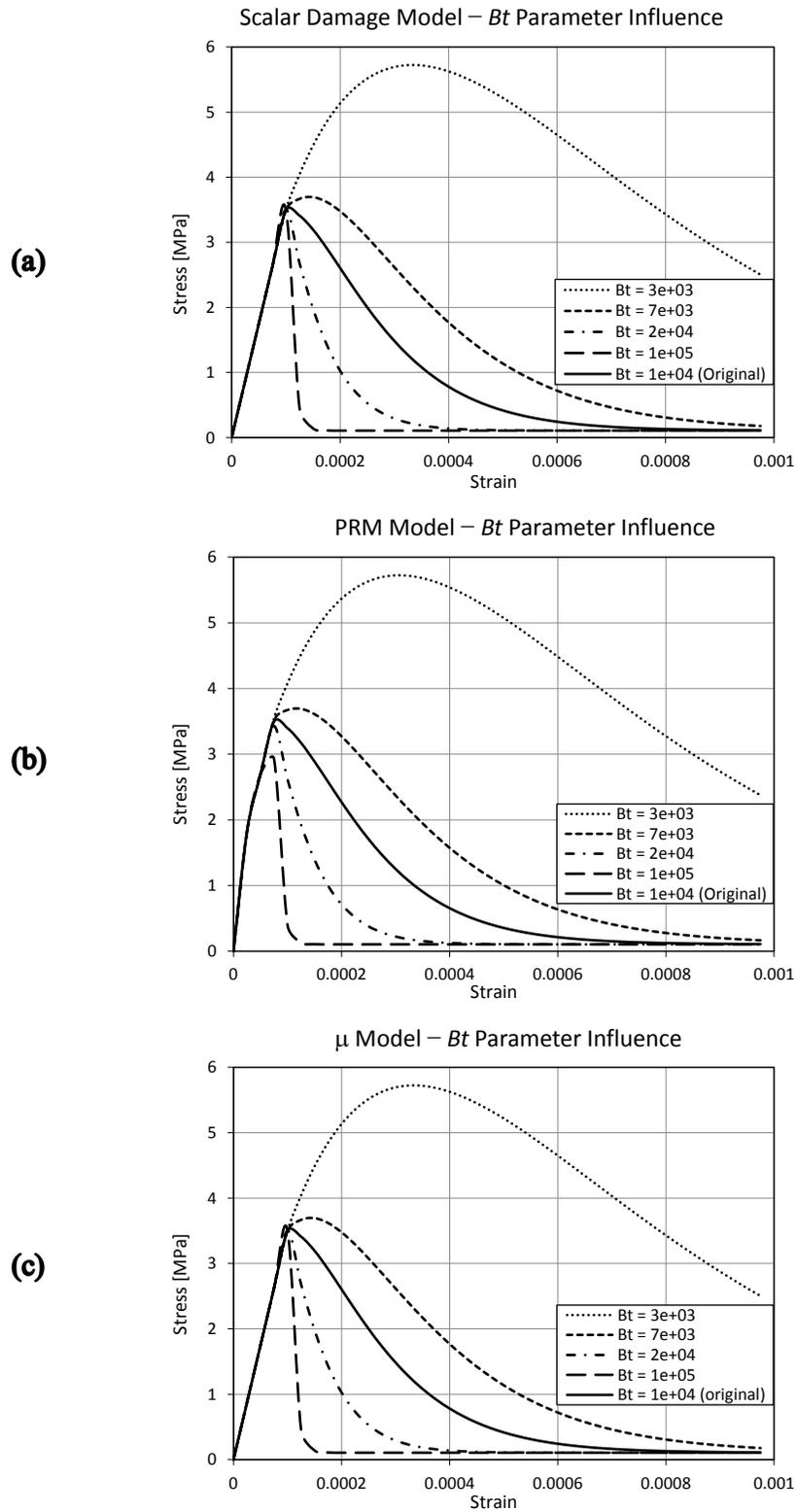


Fig. 3.7. Bt parameter influence in the tensile stress-strain response for the scalar damage model (a), the PRM model (b), and the μ model (c)

To examine the influence of the loading path on the compressive and tensile response of concrete, the uniaxial stress-strain responses for three different loading paths using the scalar damage, the PRM, and the “ μ ” model are presented. The first loading path (Fig. 3.8) consists of a tension-compression path, where the tension loading reaches a strain of 0.0002 before the unloading starts. It is reloaded in compression until a strain of 0.007 is reached.

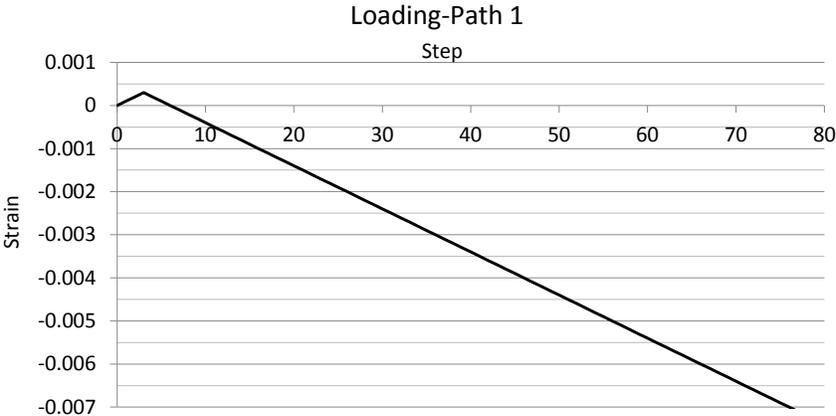


Fig. 3.8. Loading path 1: tension-compression

The second loading path (Fig. 3.9) consists of a compression-tension path, where the compression loading reaches a strain of -0.004 before the unloading starts. It is reloaded in tension until a strain of 0.0005 is reached.

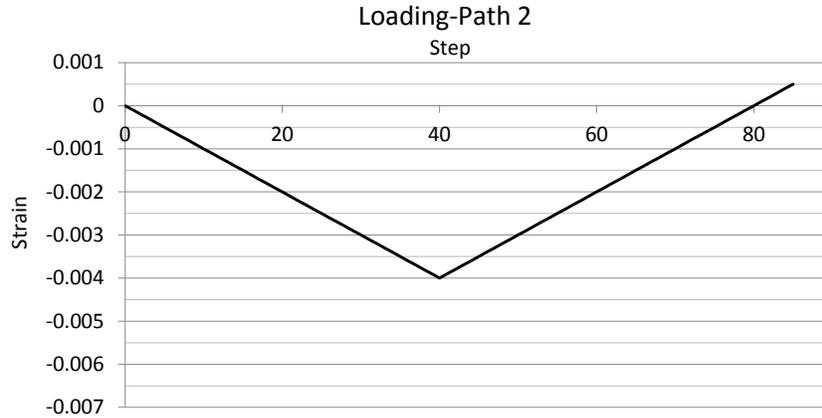


Fig. 3.9. Loading path 2: compression-tension

The third loading path (Fig. 3.10) consists of a tension-compression-tension-compression path, where the tension loading reaches a strain of 0.0003 before the unloading starts. It is reloaded in compression until a strain of -0.003 is reached, and then is unloaded and reloaded in tension until a strain of 0.0005 is reached. Finally, it is unloaded and reloaded in compression until a strain of -0.007 is reached.

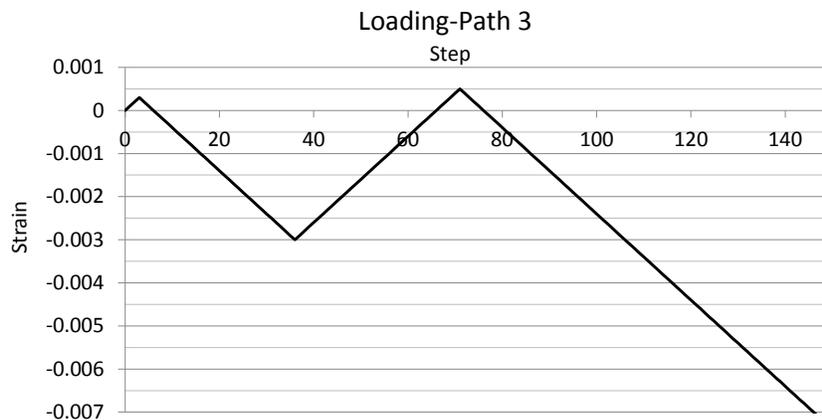
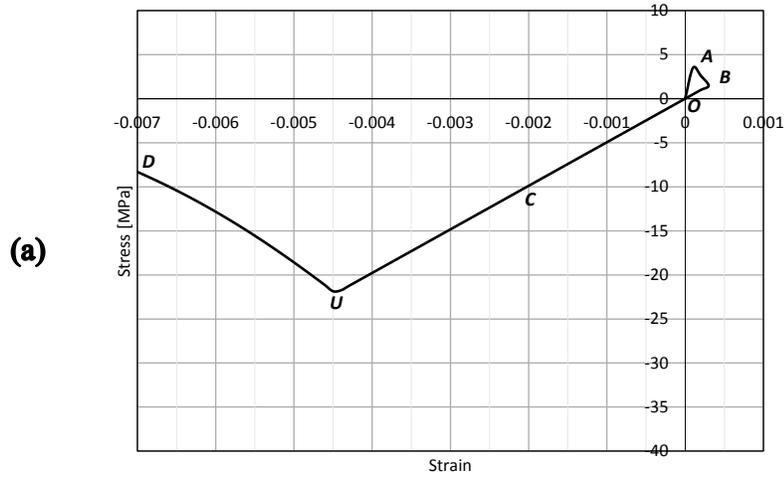


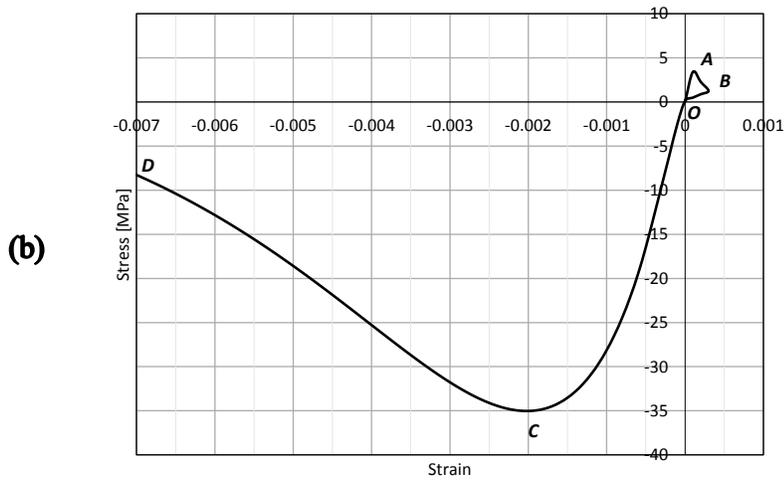
Fig. 3.10. Loading path 3: tension-compression-tension-compression

Loading Path 1

Scalar Damage Model



PRM Model



μ Model

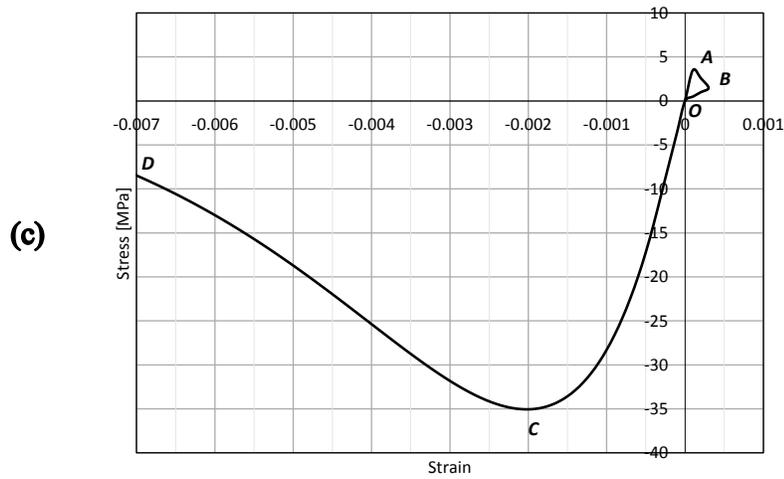


Fig. 3.11. Uniaxial stress-strain response for the scalar damage model (a), the PRM model (b), and the μ model (c) under the loading path 1

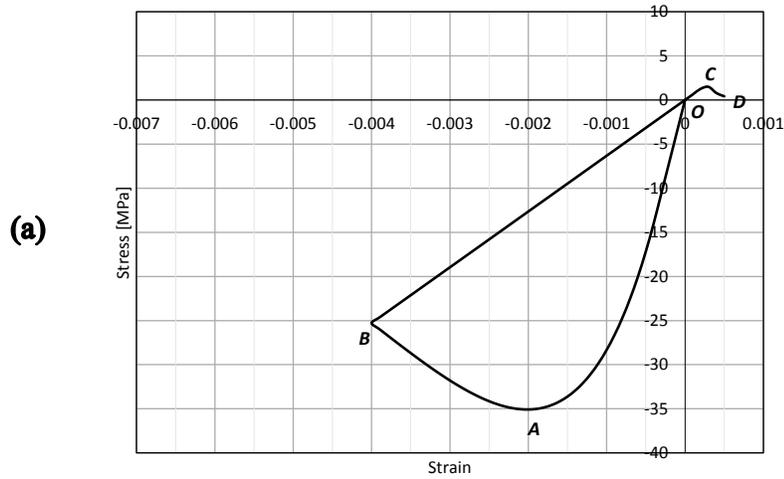
The model parameters, which are calibrated to have a maximum compression strength (f'_c) at a strain of -0.002, and a maximum tensile strength (f_t) at a strain of -0.0001, are presented in Table 3.1. Fig. 3.11 shows the uniaxial stress-strain response of the three models for the loading-path 1 (*O-A-B-O-C-D*). It can be observed that for the first loading portion of the loading *O-A-B*, the three models present the same stress-strain relationship with a peak tensile stress of 3.5 MPa at a strain value of 0.0001 (*B*).

However, notable differences can be observed for the loading in compression *O-C-D*. The scalar damage model (Fig. 3.11a) is unable to separate the evolution of damage into tension and compression; this implies that when it transitions the compression domain, the damage evolution is unrecoverable and the stiffness is held constant until the compressive strain is big enough (*U*) to continue the evolution of the current damage.

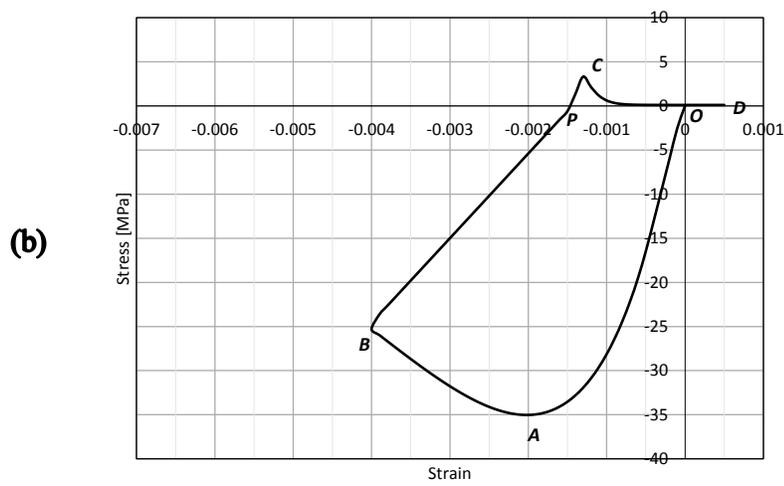
In contrast with the scalar damage model, both the PRM model (Fig. 3.11b) and the “ μ ” model (Fig. 3.11c) are able to account for stiffness recovery when the loading path switches from tension to compression. This is possible because the compressive and tensile damage evolution are independent for both models. This behavior of concrete is explained by the crack-closure phenomenon under loading-unloading cycles.

Loading Path 2

Scalar Damage Model



PRM Model



μ Model

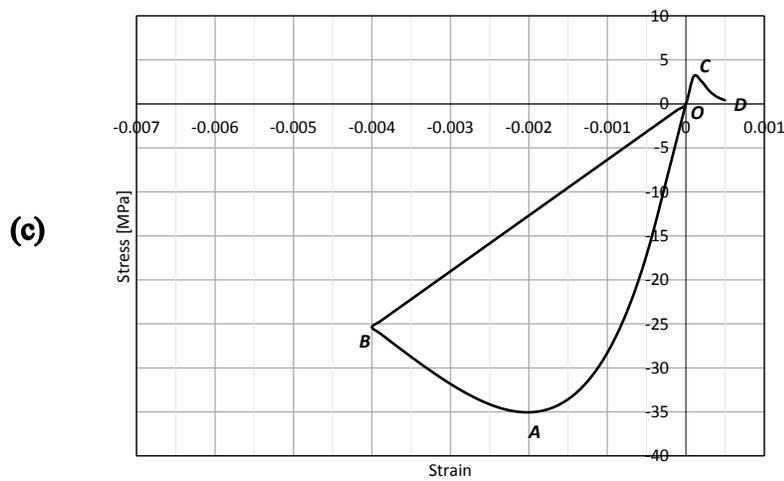


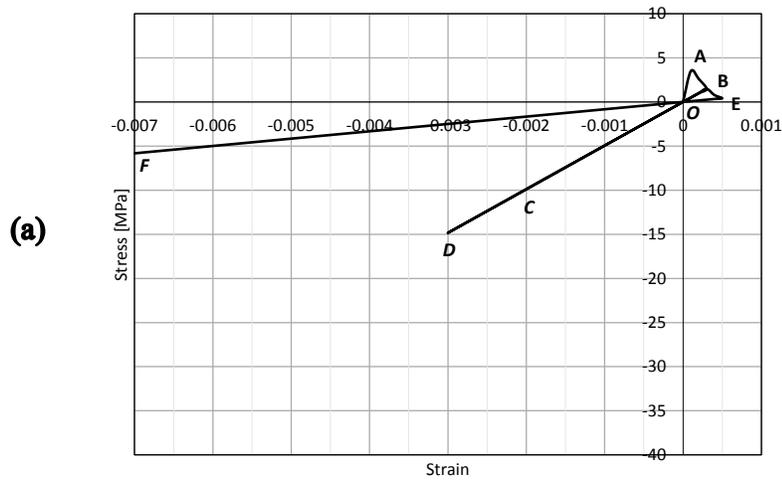
Fig. 3.12. Uniaxial stress-strain response for the scalar damage model (a), the PRM model (b), and the μ model (c) under the loading path 2

The stress-strain response of each model for the loading-path 2 is shown in Fig. 3.12, where the three models follow a different stress-strain evolution path. The three models present the same stress-strain relationship for the first loading portion of the loading ($O-A-B$), having a peak compressive stress of 35 MPa at a strain value of -0.002 (A). Notable differences can be observed for the loading in tension until point D is reached. Just as when it was subjected to the loading-path 1, the scalar damage model (Fig. 3.12a) ($O-A-B-O-C-D$) maintains constant stiffness when the reloading starts.

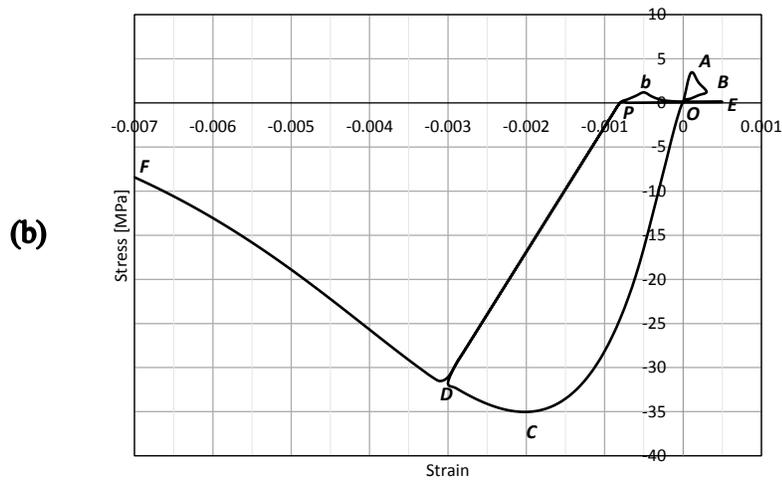
Conversely, the PRM model (Fig. 3.12b) ($O-A-B-P-C-O-D$) and the “ μ ” model (Fig. 3.12c) ($O-A-B-O-C-D$) present stiffness recovery due to crack closing effects, which is possible because the PRM and “ μ ” models have independent compressive and tensile damage evolution. The main difference between the PRM model and the other two models, is that when the PRM model reaches the plastic range under compressive loading, it can reproduce permanent deformations during the unloading-reloading path, which is controlled by the crack closure stress and strain (P). The scalar damage and “ μ ” models do not account for permanent deformations, which is a drawback but results in simpler formulations.

Loading Path 3

Scalar Damage Model



PRM Model



μ Model

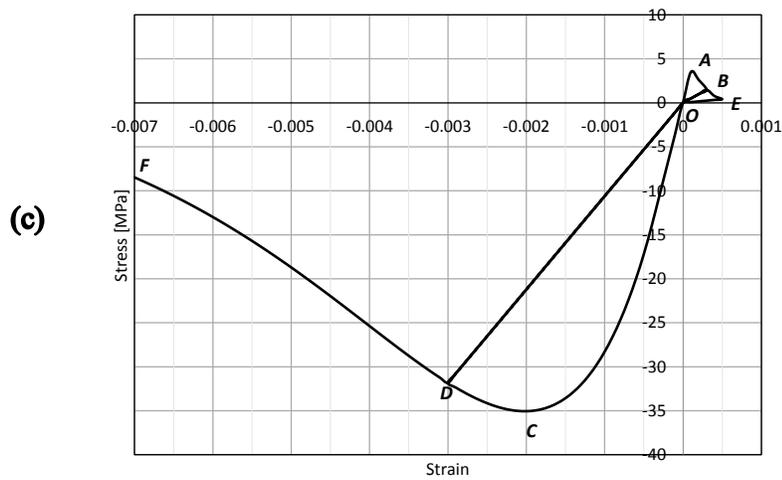


Fig. 3.13. Uniaxial stress-strain response for the scalar damage model (a), the PRM model (b), and the μ model (c) under the loading path 3

The stress-strain response of each model for the loading-path 3 is shown in Fig. 3.13, where the stress-strain response of each model can be observed under two cycles of tensile-compressive loading, where the three models follow a different stress-strain evolution path. As shown in Fig. 3.13a, the scalar damage model (*O-A-B-O-C-D-C-O-B-E-O-F*) should not be used when an element is subjected to reversal-cyclic loading, because it does not separate the damage evolution into tension and compression.

However, the scalar damage model shows a reliable stress-strain response for monotonic loading, and even under cyclic loading when only one of compressive or tensile domains is used. Both the PRM model (Fig. 3.13b) (*O-A-B-O-C-D-p-b-E-p-D-F*) and the “ μ ” model (Fig. 3.13c) (*O-A-B-O-C-D-O-B-E-O-D-F*) have been found to reliably reproduce the uniaxial stress-strain response of concrete under reversal-cyclic loading and monotonic loading, with the main difference being that the PRM model can account for permanent deformations while the “ μ ” model cannot.

4 IMPLEMENTATION OF TWO BIAXIAL CONCRETE MATERIALS IN THE OPENSEES FRAMEWORK

4.1 Introduction

The PRM model (Pontiroli, Rouquand, & Mazars, 2010) and the μ model (Mazars, Hamon, & Grange, 2015) were implemented in the OpenSees framework as 3D plane-stress elements. The OpenSees framework and its main elements and materials used to develop analytical models for validating implemented concrete models are discussed. Additionally, the finite element formulation, the coordinate system, and the material constitutive matrix are discussed. Finally, the steps followed to introduce the concrete models as new materials in OpenSees are detailed.

4.2 OpenSees

The Open System for Earthquake Engineering Simulation (OpenSees), developed within the Network for Earthquake Engineering Simulation (NEES) project, serves as an object-oriented, open source software framework for finite element analysis. The intended users of OpenSees are members of the research community. By using a modular approach in OpenSees, the behaviour of structural and geotechnical systems can be simulated. These modules contain numerical solutions, output recorders, and independently defined model configurations. Because of the flexibility of the modular implementation in OpenSees, researchers can implement new components, such as material models, element types, and solution algorithms, as they are developed (Mazzoni, McKenna, Michael, & Fenves, 2006). Key features of OpenSees include the interchangeability of components and the ability to integrate existing libraries and new components into the framework without changing the

existing code. Core components, which are the abstract base classes, define the minimal interface. It is simple enough to add new component classes with ease, but complex enough to entail all of the essential features.

OpenSees uses Tcl script language to write an input file, which includes the structural model, the analysis type, and the required output recorders. The OpenSees source code is written in C++ language and uses object-oriented programming, which allows users to create new classes or modules as parts of the framework (Mazzoni, McKenna, Michael, & Fenves, 2006).

The main features of OpenSees are as follows:

- The library of materials, elements, and analysis is a powerful tool for numerical simulation of nonlinear systems.
- The OpenSees interface is based on a command that enables the user to create more versatile input files.
- OpenSees is not a black box, which makes it useful for modeling.
- You can create your own material, element, or algorithm and introduce it into OpenSees.
- It is a freely available software framework.
- It was developed for simulation applications in earthquake engineering.
- As open-source software, it has potential as a community code for earthquake engineering.

4.2.1 Materials

There are two types of materials currently available in OpenSees: uniaxial and multi-dimensional (nD materials). The PRM model and the “ μ ” model, discussed in chapter 3, were implemented as plane-stress biaxial concrete materials and new objects of the nD material class. The uniaxial materials used for this research project are the Giuffre-Manegotto-Pinto steel model with isotropic strain hardening (Filippou, Popov, & Bertero, 1983), a hysteretic material for steel, and the Kent-Scott-Park material object for concrete. These three materials are detailed below.

4.2.1.1 Steel Material: Giuffre-Manegotto-Pinto Model with Isotropic Strain Hardening

The steel material selected to model the steel reinforcement of the experimental tests being used to validate this project in the OpenSees framework, was the Steel02 material, which uses the Giuffre-Manegotto-Pinto model with isotropic strain hardening. The input parameters for the Steel02 material command (Fig. 4.1) are the integer tag identifying material ($\$matTag$), the yield strength ($\Fy), the initial elastic tangent ($\$E0$), the strain-hardening ratio ($\$b$), and the parameters to control the transition from elastic to plastic branches ($\$R0$, $\$CR1$, $\$CR2$), with the following recommended values: $\$R0 =$ between 10 and 20, $\$cR1=0.925$, and $\$cR2=0.15$. Fig. 4.2 shows the material parameters of the monotonic envelope, Fig. 4.3 shows the hysteretic behavior without isotropic hardening, Fig. 4.4 shows the hysteretic behavior with isotropic hardening in compression, and Fig. 4.5 shows the hysteretic behavior with isotropic hardening in tension for the Steel02 material (Filippou et al., 1983).

```
uniaxialMaterial Steel02 $matTag $Fy $E0 $b $R0 $cR1
```

Fig. 4.1. Steel 02 material command input parameters

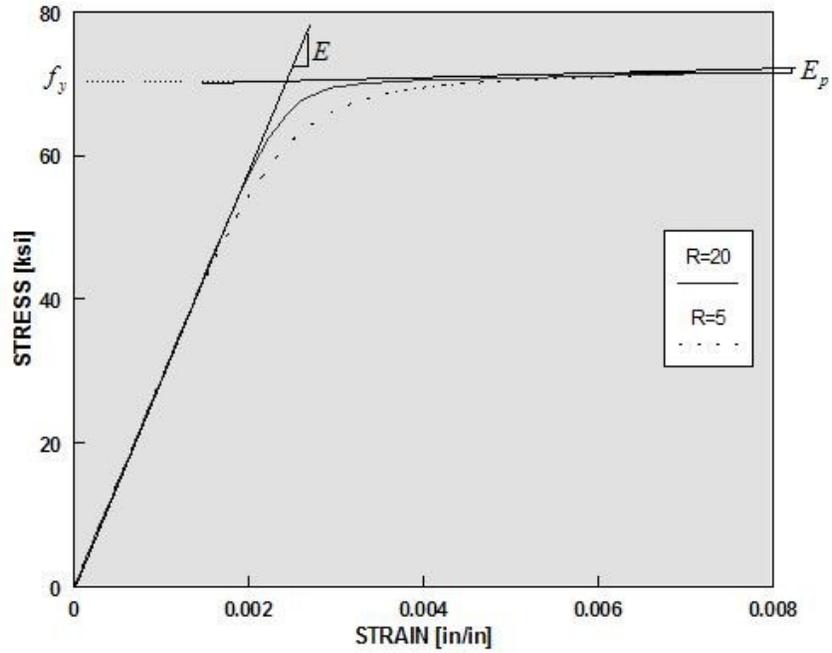


Fig. 4.2. Steel02 material: parameters of monotonic envelope (OpenSees)

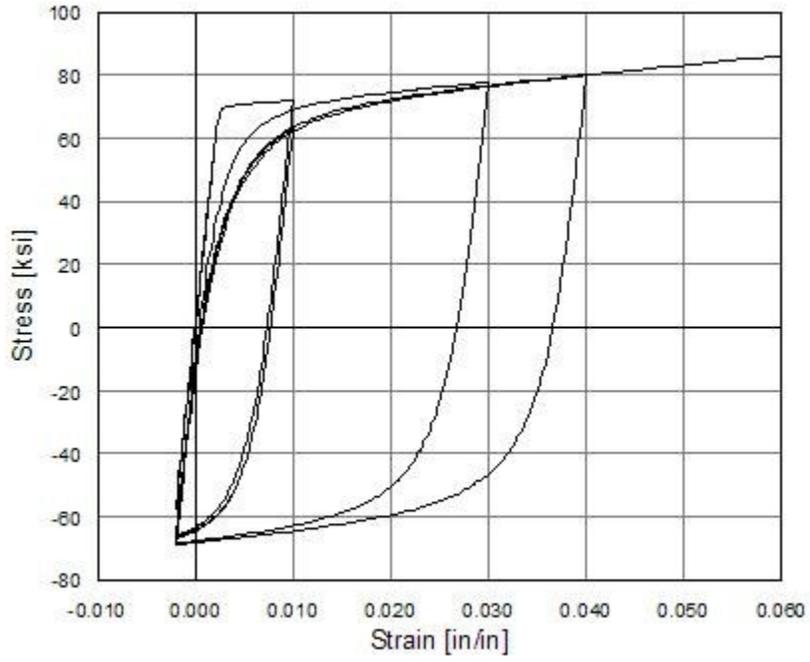


Fig. 4.3. Steel02 material: hysteretic behavior without isotropic hardening (OpenSees)

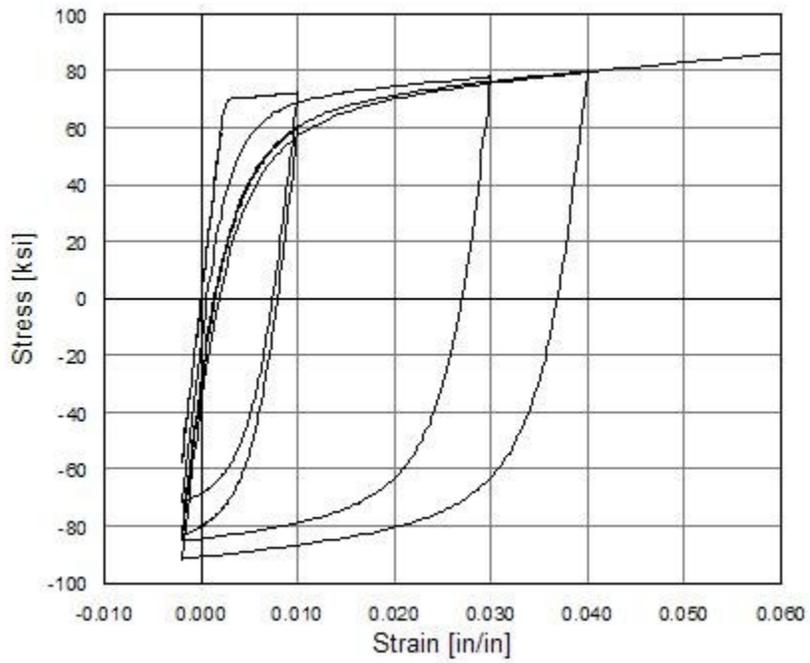


Fig. 4.4. Steel02 material: hysteretic behavior with isotropic hardening in compression (OpenSees)

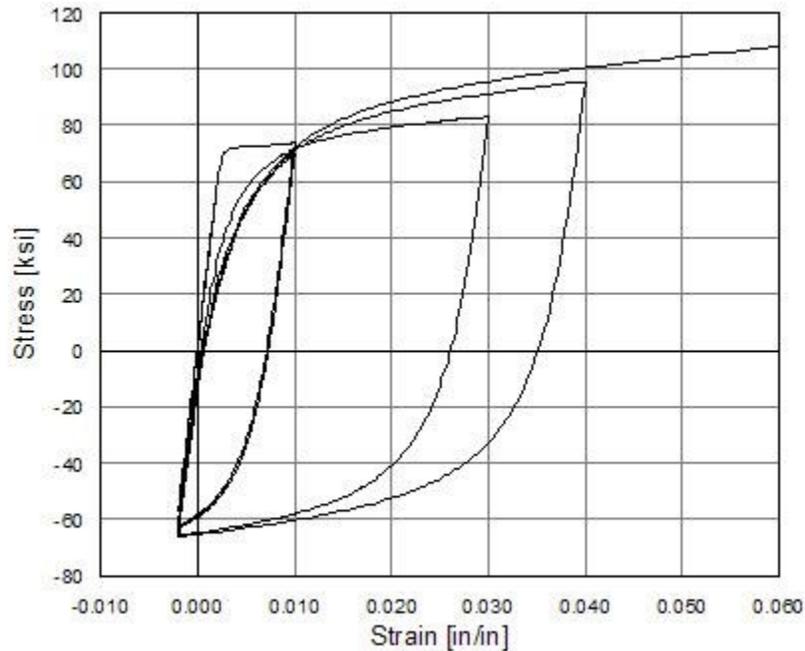


Fig. 4.5. Steel02 material: hysteretic behavior with isotropic hardening in tension (OpenSees)

4.2.1.2 Hysteretic Material

This material command allows the user to construct a uniaxial bilinear hysteretic material object with pinching of force and deformation, degraded unloading stiffness based on ductility, and damage due to energy and ductility. This material command is typically used to model the bond-slip phenomenon presented in RC structures. The input parameters for the Hysteretic Material command (Fig. 4.6) are the integer tag identifying material ($\$matTag$), the stress and strain at the first point of the envelope in the positive direction ($\$s1p$ $\$e1p$), the stress and strain at the second point of the envelope in the positive direction ($\$s2p$ $\$e2p$), the stress and strain at the third point of the envelope in the positive direction (optional) ($\$s3p$ $\$e3p$), the stress and strain at the first point of the envelope in the negative direction ($\$s1n$ $\$e1n$), the stress and strain at the second point of the envelope in

the negative direction (e_2n, s_2n), the stress and strain at the third point of the envelope in the negative direction (optional) (e_3n, s_3n), the pinching factor for strain during reloading (μ_{pinchx}), the pinching factor for stress during reloading (μ_{pinchy}), the damage due to ductility ($\mu_{damage1}$), the damage due to energy ($\mu_{damage2}$), and the power used to determine the degraded unloading stiffness based on ductility (optional, default=0.0) (β). The typical stress-strain relationship for the material is shown in Fig. 4.7.

```

uniaxialMaterial Hysteretic $matTag $s1p $e1p $s2p $e2p <$s3p $e3p> $s1n
  $e1n $s2n $e2n <$s3n $e3n> $pinchX $pinchY $damage1 $damage2 <$beta>

```

Fig. 4.6. Hysteretic material input parameters

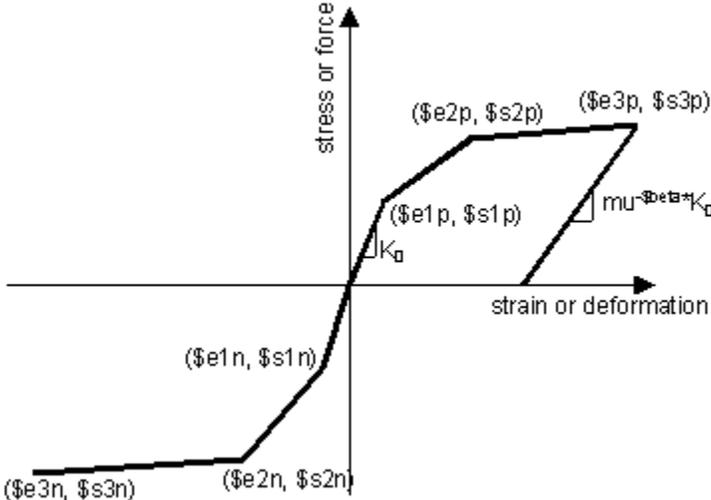


Fig. 4.7. Hysteretic Material: typical stress-strain relationship

4.2.1.3 Concrete material: Kent-Scott-Park

The Concrete01 material command allows the user to construct a uniaxial Kent-Scott-Park concrete material object with degraded linear unloading-reloading stiffness. This material command is based on the work of Karsan-Jirsa and does not take tensile strength into account. The input parameters for the Concrete01 command (Fig. 4.8) are the integer tag identifying material ($\$matTag$), the concrete compressive strength ($\$fpc$), the concrete strain at maximum strength ($\$epsc0$), the concrete crushing strength ($\$fpcu$), and the concrete strain at crushing strength ($\$epsU$). The Young's modulus (E) is considered equal to $2*\$fpc/\$epsc0$, therefore, the concrete strain at maximum strength should be considered equal to $2*\$fpc/E$. The typical stress-strain relationship for the material is shown in Fig. 4.9 (OpenSees).

```
uniaxialMaterial Concrete01 $matTag $fpc $epsc0 $fpcu
```

Fig. 4.8. Concrete01 material input parameters

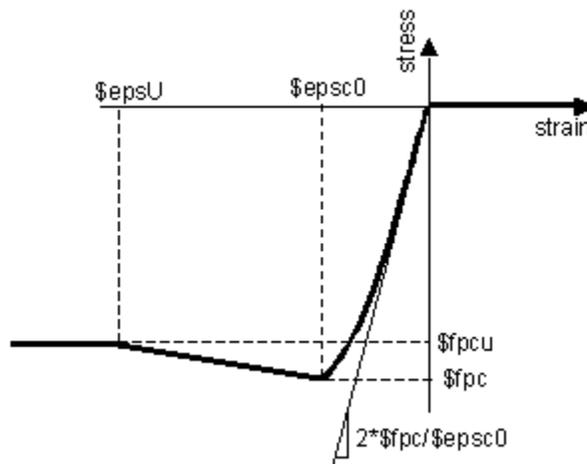


Fig. 4.9. Concrete01 material: typical stress-strain relationship

4.2.2 Elements

The OpenSees framework contains a diverse number of element types, including but not limited to zero-length elements, truss elements, beam-column elements, and quadrilateral elements. This research focuses on the elements used to build the validation models, which are detailed below.

4.2.2.1 Zero-Length Elements: ZeroLength Element

A number of zero-length elements are available in the OpenSees framework, including uniaxial, multi-axial, and contact elements. The primary interest of this project is the ZeroLength element command, which is a uniaxial zero-length element. It is defined by two nodes at the same location, which are connected by multiple UniaxialMaterial objects that represent the force-deformation relationship of the element (OpenSees). A typical application of this element is the simulation of the bond-slip phenomenon.

The input parameters of the ZeroLength command (Fig. 4.10) are the unique element object tag (\$eleTag), the end nodes (\$iNode \$jNode), the tags associated with previously-defined uniaxial materials (\$matTag1 \$matTag2 ...), the material directions 1,2,3 translation along local x,y,z axes, respectively, the 4,5,6 rotation about local x,y,z axes, respectively (\$dir1 \$dir2 ...), and the concrete strain at crushing strength (\$dir1 \$dir2 ...).

```
element zeroLength $eleTag $iNode $jNode -mat $matTag1 $matTag2 ...  
-dir $dir1 $dir2 ...
```

Fig. 4.10. ZeroLength element input parameters

4.2.2.2 *Truss Elements*

There are two types of truss elements in the OpenSees framework: the truss element and the corotational truss element. For both, specifying an area and a uniaxial material is necessary. The only difference is that the corotational truss element considers geometric nonlinearities, while the truss element does not. In this research project we are focusing on the truss element, which was used to model the steel reinforcement of the validation models. The input parameters of the truss element command (Fig. 4.11) are the unique element object tag ($\$eleTag$), the end nodes ($\$iNode$ $\$jNode$), the cross-sectional area of the element ($\$A$), and the tag associated with a previously-defined uniaxial material ($\$matTag$).

```
element truss $eleTag $iNode $jNode $A
```

Fig. 4.11. Truss element input parameters

4.2.2.3 *Beam-Column Elements*

A number of beam-column elements are available in the OpenSees framework, including elastic, inelastic, nonlinear, displacement-based, force-based, and beams with hinge elements. This project is primarily interested in the nonlinearBeamColumn element command. This command is used for constructing a nonlinear beam column element object using the non-iterative or iterative force formulation; it considers the spread of plasticity along the element (OpenSees). The input parameters of the truss element command (Fig. 4.12) are the unique element object tag ($\$eleTag$), the end nodes ($\$iNode$ $\$jNode$), the number of integration points along the element ($\$numIntgrPts$), the tag associated with a

previously-defined section object ($\$secTag$), and the tag associated with a previously-defined coordinate-transformation object ($\$transfTag$).

```
element nonlinearBeamColumn $eleTag $iNode $jNode $numIntgrPts  
$secTag $transfTag
```

Fig. 4.12. NonlinearBeamColumn element input parameters

Fig. 4.13 shows the interaction of a zero-length element and a beam-column element for modelling the bond-slip behavior in RC columns, where both elements are joined by a common node j . Node i is also required to construct the zero-length element, and both nodes i and j have the same coordinates at the base of the column. The beam-column element is defined from node j to node k . For both elements, a previously-defined section element is assigned. Regarding the materials, a previously-defined uniaxial concrete material is assigned to the column-beam element, such as the Concrete01 material command. For the zero-length element, a hysteretic material is assigned.

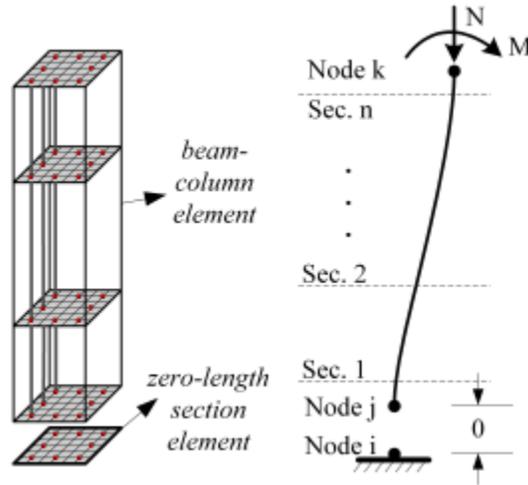


Fig. 4.13. Interaction of a zero-length element and a beam-column element (OpenSees)

4.2.2.4 *Cuadridatelar Elements: Quad Element, ShellMITC4 and Multilayered Shell*

Element

A number of cuadridatelar elements are available in the OpenSees framework. This project is primarily interested in the quad element, the shellMITC4 element, and the multilayered shell element commands. The quad element command is used to construct a four-node quad element object, which uses a bilinear isoparametric formulation (OpenSees). The input parameters of the quad element command (Fig. 4.14) are the unique element object tag ($\$eleTag$), the four nodes defining element boundaries, input in counter-clockwise order around the element ($\$iNode$ $\$jNode$ $\$kNode$ $\$lNode$), the element thickness ($\$thick$), the type of material behavior –plane strain or plane stress– ($\$type$), and the tag associated with a previously-defined material object ($\$matTag$).

```
element quad $eleTag $iNode $jNode $kNode $lNode $thick $type $matTag
```

Fig. 4.14. Quad element input parameters

The ShellMITC4 command is used to construct a shell element object, which uses a bilinear isoparametric formulation in combination with a modified shear interpolation to improve thin-plate bending performance (OpenSees). This element is a 3D element with 6 degrees of freedom and it serves to model plane structures in OpenSees under in- and out-of-plane loading. The input parameters of the ShellMITC4 element command (Fig. 4.15) are the unique element object tag (\$eleTag), the four nodes defining element boundaries, input in counter-clockwise order around the element (\$iNode \$jNode \$kNode \$lNode), and the tag associated with a previously-defined section object (\$secTag).

```
element ShellMITC4 $eleTag $iNode $jNode $kNode $lNode $secTag
```

Fig. 4.15. ShelMITC4 element input parameters

In this project, the section object of the shellMITC4 element is built using the multi-layered shell element (Lu, Xie, Guan, & Lu, 2015). The multi-layered shell element performs the nonlinear analysis of composite-plane elements. It is comprised of a number of layers with different thicknesses and material properties (Fig. 4.16), making it suitable for modelling RC structures, such as beams or shear walls, by having concrete layers and reinforcement steel layers. The steel rebars are smeared into one or more layers, where their thickness depends on the reinforcement ratio, and the orientation depends on the longitudinal and transverse reinforcement steel.

The multi-layered shell element, in which the stresses acting over the thickness of the layers are assumed to be equal to those stresses at the mid-surface of the layer, is built based on the plane-stress theory. Thus, if you subdivide the plane component of an RC

concrete structure into enough layers, the multi-layer shell element can simulate the real stress distribution over the thickness of the shell to a reasonable degree. The in-plane and the out-of-plane behaviour of a plane structure can be represented with biaxial element formulations by using layered shell elements.

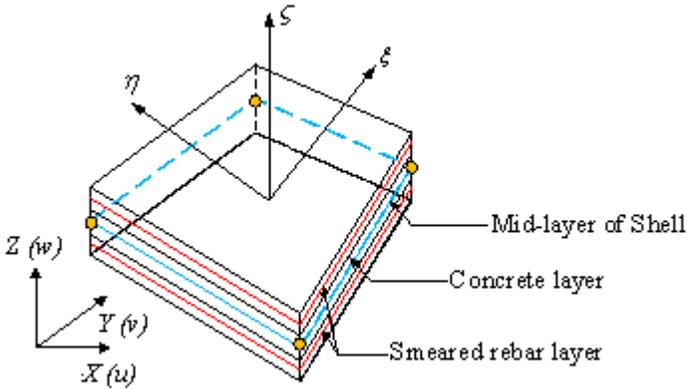


Fig. 4.16. Multilayered-shell element (Lu, Xie, Guan, & Lu, 2015)

4.3 Coordinate System

Two different Cartesian coordinate systems, x-y and 1-2, are defined for plane concrete elements. The local coordinates of the elements are represented by the x-y coordinate system, and the principal directions of the applied stresses are represented by the 1-2 coordinate system. The angle between both coordinate systems is defined by the angle θ (Fig. 4.17).

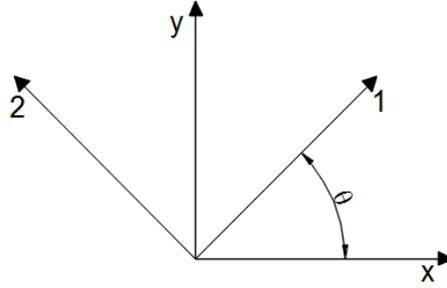


Fig. 4.17. Local (x-y) and principal stresses (1-2) coordinate systems

The stress $\{\sigma\}$ and strain $\{\varepsilon\}$ vectors in the x-y coordinates (local), and in the 1-2 coordinates (principal-stress directions) are denoted by Eq. (4.1) and (4.2), respectively.

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \text{ and } \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \frac{1}{2}\gamma_{xy} \end{Bmatrix} \quad (4.1)$$

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} = 0 \end{Bmatrix} \text{ and } \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \frac{1}{2}\gamma_{12} \end{Bmatrix} \quad (4.2)$$

By using the transformation matrix $T(\theta)$ shown in Eq. (4.3), the stresses and strains can be transformed between the different coordinates.

$$T(\theta) = \begin{bmatrix} \cos^2(\theta) & \sin^2(\theta) & 2\sin(\theta)\cos(\theta) \\ \sin^2(\theta) & \cos^2(\theta) & -2\sin(\theta)\cos(\theta) \\ -\sin(\theta)\cos(\theta) & \sin(\theta)\cos(\theta) & \cos^2(\theta) - \sin^2(\theta) \end{bmatrix} \quad (4.3)$$

The stresses and strains are transformed from the local coordinate (x-y) to the principal stresses coordinate (1-2) by using the transformation matrix as shown in Eq. (4.4) and (4.5), respectively.

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ 0 \end{Bmatrix} = [T(\theta)] \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \quad (4.4)$$

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \frac{1}{2}\gamma_{12} \end{Bmatrix} = [T(\theta)] \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \frac{1}{2}\gamma_{xy} \end{Bmatrix} \quad (4.5)$$

4.4 Material Constitutive Matrix

The material constitutive matrix or material stiffness matrix relates the state of stresses and strains of an element, and it can be expressed in terms of secant or tangent formulations.

The secant material constitutive matrix relates the absolute values of strains and stresses of the element, and the tangent material constitutive matrix relates the increment of the stresses and strains of the element. The damage theories being evaluated for performance in this project use a secant material constitutive matrix approach.

The secant constitutive matrix (Γ) of an isotropic-linear-elastic 3D material is given by Eq. (4.6), and the relationship between the stress and strain components is shown in Eq. (4.7).

The secant constitutive matrix is a function of the concrete initial Young's modulus (E) and the Poisson's ratio (ν).

$$\Gamma = \frac{E}{(1-2\nu)(1+\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & & & \\ \nu & 1-\nu & \nu & & & \\ \nu & \nu & 1-\nu & & & \\ & & & \frac{1-2\nu}{2} & 0 & 0 \\ & & & 0 & \frac{1-2\nu}{2} & 0 \\ & & & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad (4.6)$$

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix} = \frac{E}{(1-2\nu)(1+\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & & & \\ \nu & 1-\nu & \nu & & & \\ \nu & \nu & 1-\nu & & & \\ & & & \frac{1-2\nu}{2} & 0 & 0 \\ & & & 0 & \frac{1-2\nu}{2} & 0 \\ & & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{12} \\ 2\varepsilon_{13} \\ 2\varepsilon_{23} \end{Bmatrix} \quad (4.7)$$

The plane-stress or plane-strain assumption can be considered for plane isotropic-linear-elastic 3D materials. The first is used for situations where the material can freely expand or contract in the thickness direction, and the loads are applied in the plane perpendicular to the thickness direction, and thus stress in the thickness direction is equal to zero. The latter of the two is used for situations where the material is restrained from expanding or contracting in the thickness direction, and the loads are applied in the plane perpendicular to the thickness direction, and thus there is a restraining stress in the thickness direction. The plane-stress assumption suits the structures in which this research is interested, such as RC shear walls.

The secant constitutive matrix (Γ) of a plane isotropic-linear-elastic 3D material is given by Eq. (4.8).

$$\Gamma = \frac{E}{(1+\nu)} \begin{bmatrix} \frac{1}{1-\nu} & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & \frac{1}{1-\nu} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \quad (4.8)$$

The relationship between the stress and strain components is given by Eq. (4.9).

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_{12} \end{Bmatrix} = \frac{E}{(1+\nu)} \begin{bmatrix} \frac{1}{1-\nu} & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & \frac{1}{1-\nu} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ 2\varepsilon_{12} \end{Bmatrix} \quad (4.9)$$

In the plane stress assumption, the material can expand and contract freely in the thickness direction, and the corresponding value of the axial strain in this direction can be obtained by using Eq. (4.10).

$$\varepsilon_3 = -\frac{\nu}{E}(\sigma_1 + \sigma_2) \quad (4.10)$$

The material constitutive matrix relates the concrete principal stresses σ_p and strains ε_p as given by Eq. (4.11), where the damage (D) in the material is calculated using each Mazars's concrete model in Eq. (3.1) to (3.31).

$$\{\sigma_p\} = (1 - D)[\Gamma_c]\{\varepsilon_p\} \quad (4.11)$$

$$[\Gamma_c]_D = (1 - D)[\Gamma_c] \quad (4.12)$$

4.5 Finite Element Analysis Procedure

After using Eq. 4.12 to determine the damaged-material constitutive matrix $[\Gamma_c]_D$, the basic finite element procedure, which depends on the type of element used and is expressed by Eq. (4.13), is used to evaluate the element stiffness matrix. The matrix $[B]$ is dependent on the element displacement functions.

$$[K] = \int [B]^T [\Gamma_c]_D [B] dV \quad (4.13)$$

Several solution schemes exist for performing nonlinear analyses of reinforced concrete structures. Two options include the use of static integrators, as in displacement-controlled

analysis, or dynamic integrators, like the Newmark method. There are also several possible solution algorithms including the Modified Newton method and the Krylov-Newton method. The flow chart (Fig. 4.18) depicting an iterative analysis solution uses the Newton-Raphson method to perform a static analysis using load increment.

The damaged material stiffness matrix $[T_c]_D$ is determined in each iteration, using the concrete damage models described previously. The element stiffness matrix $[K]$ and the element resisting force increment vector $\{\Delta F\}$ are also calculated. Until the convergence criterion is achieved, the element stiffness matrix $[K]$ and the damaged material constitutive matrix $[T_c]_D$ are iteratively refined.

Fig. 4.18 shows the procedure for establishing the damaged material constitutive matrix using the damage-based implemented models, the PRM and “ μ ” models. It should be noted that an iterative procedure is not required to establish the damaged material constitutive matrix, thus these damage models are computationally efficient for finite element analysis. The source code developed in this research obtains the input strains of the element and from these it calculates the corresponding stresses in the material (procedure depicted in the dotted square, Fig. 4.18). The procedure outside the dotted square is previously defined in the OpenSees framework.

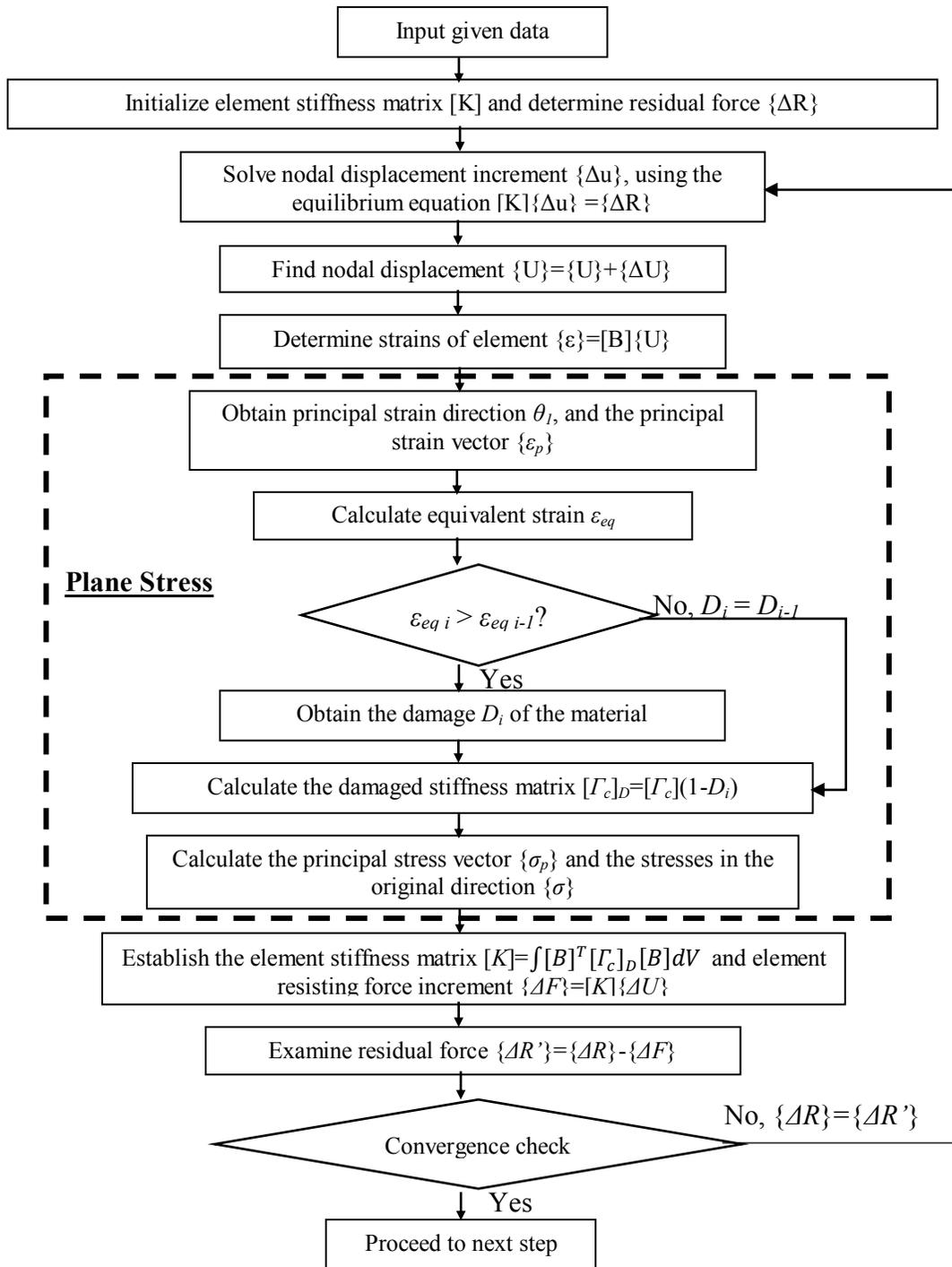


Fig. 4.18. Nonlinear analysis procedure

4.6 Introducing a New Multi-dimensional Material –nD Material– to the OpenSees Framework

Two concrete damage models, the PRM model (Pontiroli, Rouquand, & Mazars, Predicting Concrete Behaviour From Quasi-static Loading to Hypervelocity Impact, 2010) and the “ μ ” model” (Mazars, Hamon, & Grange, 2015), are used to implement two new material classes into the OpenSees framework for the analysis of RC plane structures. The plane stress concrete material parameters for the PRM model and the “ μ ” model are shown in Fig. 4.19 and Fig. 4.20, respectively.

```
nDMaterial PRM $matTag $Ec $sepsD0c $Ac $Bc $sepsD0t $At $Bt $sepsfc
$sigfc $sepsft0 $sigft0 $nu
```

Fig. 4.19. PRM material input parameters

For the PRM nDMaterial (based on the PRM model) the input parameters are the unique material object integer tag (\$matTag), the initial Young’s modulus of the concrete (\$Ec), the initial compressive-damage threshold (\$sepsD0c), the material parameters that define the compressive behaviour of the concrete (\$Ac and \$Bc), the initial tensile-damage threshold (\$sepsD0t), the material parameters that define the tensile behaviour of the concrete (\$At and \$Bt), the focal point followed by the unloading in compression (\$sepsfc, \$sigfc), the initial crack closure stress and strain (\$sepsft0 and \$sigft0), and Poisson’s ratio (\$nu).

```
nDMaterial MuMazars $matTag $Ec $sepsD0c $sepsD0t $Ac $Bc $At $Bt $nu
```

Fig. 4.20. MuMazars material input parameters

For the MuMazars nDMaterial (based on the “ μ ” model) the input parameters are the unique material object integer tag (`$matTag`), the initial Young’s modulus of the concrete (`$Ec`), the initial compressive-damage threshold (`$epsD0c`), the initial tensile-damage threshold (`$epsD0t`), the material parameters that define the compressive behaviour of the concrete (`$Ac` and `$Bc`), the material parameters that define the tensile behaviour of the concrete (`$At` and `$Bt`), and Poisson’s ratio (`$nu`).

Because of the modular and hierarchical nature of the OpenSees framework (Fig. 4.21), new material models can be added independently from the existing elements and materials implementations. Therefore, a new material model can be assigned to an existing element without modifying the element implementation. Fig. 4.22 shows the OpenSees source code solution explorer, wherein all modules are required to produce the executable file (`OpenSees.exe`). These modules have sub-classes of their own.

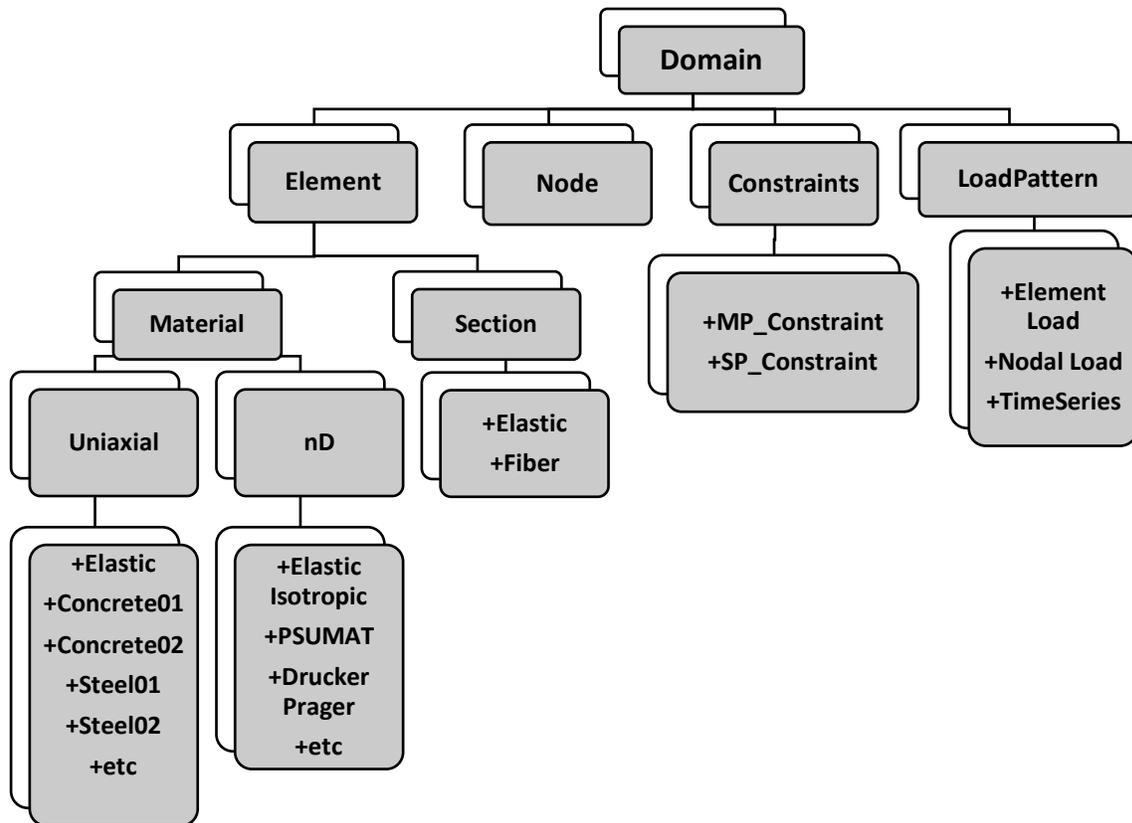


Fig. 4.21. OpenSees classes hierarchy

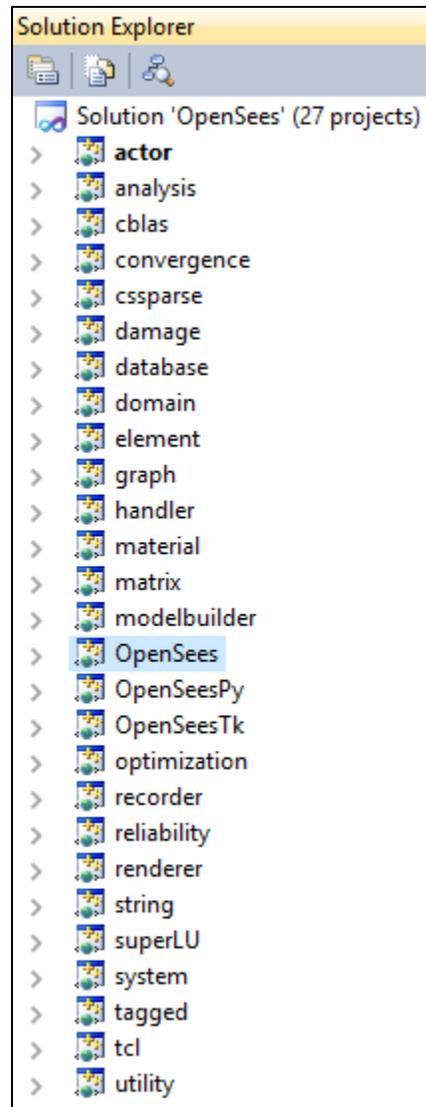


Fig. 4.22. OpenSees source code solution explorer

The minimum steps needed to add a new multi-dimensional material (nDMaterial) are as follows:

1. Download version 8.5.18 of the Tcl/tk from following link:
<http://OpenSees.berkeley.edu/OpenSees/user/download.php>
2. Download and install the software Tortoise SVN from the link below. It is needed to obtain an updated version of the OpenSees source code.
<https://tortoisesvn.net/downloads.html>
3. Obtain Visual Studio Community 2017 and Microsoft NET Framework 4.5. When installing Visual Studio, select the option: “Desktop development with C++” and “Windows 10 SDK (10.0.15063.0) for Desktop C++.”
4. Create a folder in an accessible location with a recognizable name, such as “OpenSeesSourceCode.”
5. Right click on the folder created in step 4 and select “SVN Checkout.” Type “svn://peera.berkeley.edu/usr/local/svn/OpenSees/trunk” in the “URL of repository” box. Make sure you select the option “Fully recursive” for the “Checkout Depth.” Write the folder location in the “Checkout directory.” Select the OpenSees revision number for which the source code is desired. If you select “HEAD” revision, you will obtain the latest revision. For each revision, different existing material classes may be modified for the successful compilation of the executable file.
6. Open **OpenSeesSourceCode\Win64\OpenSees.sln** using Visual Studio Community 2017 to create a 64-bit OpenSees application, or **OpenSeesSourceCode\Win32\OpenSees.sln** to create a 32-bit OpenSees application.

7. Right click on “OpenSees” in the solution explorer and select “Build.” The compilation should occur without errors. The OpenSees.exe application can be found in **OpenSeesSourceCode\Win32\bin** or **OpenSeesSourceCode\Win64\bin**
8. Place the new material header and cpp files in **OpenSeesSourceCode\SRC\material\nD**. Remember that both files must have the same name: newMaterial.cpp and newMaterial.h. See Section 4.6.1 for details.
9. Create a new material tag. See 4.6.2 for details.
10. Modify the class broker, which is responsible for verifying that the class you are calling within the application exists. See 4.6.3 for details.
11. Modify the nD material model builder. See 4.6.4 for details.
12. Right click on “OpenSees” in the solution explorer and select “Rebuild.”

4.6.1 New nDMaterial: Header and C++ Files

The nDMaterial class provides default implementations for the functions used to create a new nDMaterial subclass (material). Classes, functions, and variables of the new material can be declared in the Header File (newMaterial.h). The C++ File (newMaterial.cpp) is used to implement these classes, functions, and variables. It is necessary to include the C and C++ libraries in both files. The C++ source code for the PRM model and the “ μ ” model can be found in Appendix A and B, respectively.

4.6.2 Class Tags

In order for the new material model to communicate with other classes in the OpenSees framework, a new internal class tag needs to be defined in the file

OpenSeesSourceCode\SRC\classTags.h (Fig. 4.23).

```
#define ND_TAG_newMaterial 01
```

Fig. 4.23. New internal class defined in the file classTags.h

4.6.3 Object Broker

The file **FEM_ObjectBrokerAllClasses.cpp** needs to be modified as shown in Fig. 4.24, where the class header needs to be included. This file can be found in

OpenSeesSourceCode\SRC\actor\objectBroker\FEM_ObjectBrokerAllClasses.cpp. The nDMaterial model undergoes modifications that allow for parallel processing and database programming to the new material object. Two modifications are needed: 1) the inclusion of the new material header, 2) the addition of a new case, where if the new material tag is invoked, the new material is returned.

```

1)
#include <newMaterial.h>
.
.
.

2)
NDMaterial*
FEM_ObjectBrokerAllClasses::getNewNDMaterial(int classTag)
{
    switch(classTag) {

case ND_TAG_newMaterial:
    return new newMaterial();
    }
}

```

Fig. 4.24. Modifications needed for the file FEM_ObjectBrokerAllClasses.cpp

4.6.4 Model Builder

The last file that needs to be modified is the TclModelBuilderNDMaterialCommand.cpp, located in

OpenSeesSourceCode\SRC\actor\material\nd\TclModelBuilderNDMaterialComman

d.cpp. This command functions by parsing the material parameters in Tcl script input files and then transferring them to the new material constructor. Three modifications are needed: 1) the inclusion of the new material header, 2) the declaration of an external void/empty function, and 3) the implementation of the void function, which grants access to the new material class when it is specified to OpenSees (Fig. 4.25).

1)

```
#include <newMaterial.h>
```

```
.  
. .  
.
```

2)

```
extern void *OPS_newMaterial(void);
```

```
.  
. .  
.
```

3)

```
else if (strcmp(argv[1], "newMaterial") == 0) {
```

```
    void *theMat = OPS_newMaterial();
```

```
    if (theMat != 0)
```

```
        theMaterial = (NDMaterial *)theMat;
```

```
    else
```

```
        return TCL_ERROR;
```

```
}
```

Fig. 4.25. Modifications needed for the file FEM_ObjectBrokerAllClasses.cpp

5 VALIDATION & DISCUSSION

5.1 Introduction

Two concrete damage models, the PRM model (Pontiroli, Rouquand, & Mazars, 2010) and the “ μ ” model” (Mazars, Hamon, & Grange, 2015), were implemented in OpenSees to simulate the biaxial response of concrete materials. The performance of the new biaxial material formulations was studied by comparing five concrete experimental tests of varying complexity taken from the literature. These consist of: 1) plain concrete plates tested under biaxial stress (Kupfer, Hilsdorf, & Rüsçh, 1969), 2) a simply supported beam under monotonic loading tested as part of this project, 3) a simply supported beam under reversal-cyclic loading (Ranjbaran, Rezayfar, & Mirzababai, 2018), 4) a rectangular shear wall under reversal-cyclic loading (Hiotakis, 2004), and 5) a full-scale, four-storey building under seismic loading (Nagae, et al., 2015). The performance of a biaxial model previously implemented in OpenSees by Garcia (2017), based on the model proposed by Mazars (1986), was also studied. This model is referred as the “scalar” model.

The “ μ ” and scalar models showed the ability to capture the general response of the different types of RC structures subjected to various loading conditions in the study. The “ μ ” model was found to be more suited for cyclic loading and better at predicting the initial cracking strength than the scalar damage model was. However, neither model captured the energy dissipation capabilities of concrete accurately, nor do they attempt to describe the cracking mechanism of concrete.

The PRM model was able to accurately reproduce permanent deformations for simple analytical models, such as concrete membranes, under biaxial or uniaxial loading.

However, for complex analytical models under reversal-cyclic loading (e.g. shear wall), convergence limitations were observed. These limitations could be addressed in future research that improves upon the current algorithm suite for the OpenSees framework.

The advantages and limitations of each material model are discussed in this chapter through the calculated results of the experiments.

5.2 Plain Concrete Panels Under Biaxial Stress

Kupfer (1969) conducted a series of experiments to investigate the biaxial behavior of concrete. He tested concrete specimens of 200 x 200 x 50 mm under biaxial stress, for the regions of biaxial compression (quadrant III), compression-tension (quadrant II and IV), and biaxial tension (quadrant I) (Fig. 5.1). Within each region of state of stress, different horizontal and vertical stress ratios σ_I/σ_{II} were used, and a constant strain rate was maintained while loading the specimens. The material properties were: compression strength (f'_c) = 32.7 MPa, tensile strength (f_t) = 3.2 MPa, and Young’s modulus (E) = 30 GPa.

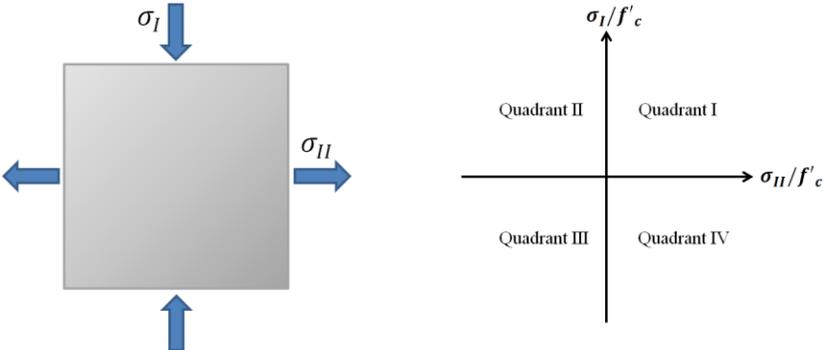


Fig. 5.1. Plain concrete specimen under different biaxial states of stress

5.2.1 Finite Element Model

The biaxial tests conducted by Kupfer were simulated in OpenSees by defining a single quad element with dimensions of 200 x 200 x 50 mm, using the implemented concrete materials for the scalar damage (SD) model (Mazars, 1986), the “ μ ” model (Mazars, Hamon, & Grange, 2015), and the PRM model (Pontiroli, Rouquand, & Mazars, 2010). A displacement-controlled analysis was performed, combining compression and tensile displacements for each of the four quadrants shown in Fig. 5.1. The principal stresses of the element were obtained for each combination and normalized with the compressive strength (f'_c). The model parameters for each material used in the analysis are shown in Table 5.1. For this specific test, the analysis is not mesh-dependant; the same analytical results were obtained when using one, four, and sixteen quad elements.

Table 5.1. Model parameters for biaxial test

Material	E(GPa)	ϵ_{0c}	$\epsilon_{0t}/\epsilon_{D0}$	ϵ_{fc}	σ_{fc} (MPa)	ϵ_{ft0}	σ_{ft0} (MPa)
SD Model	30.0	-	1.0e-4	-	-	-	-
PRM Model	30.0	1.0e-4	1.38e-4	5.0e-4	37.2	-3.3e-5	-0.99
μ Model	30.0	3.0e-4	1.1e-4	-	-	-	-

Material	Ac	Bc	At	Bt	ν
SD Model	1.275	1850.0	1.0	1.0e4	0.21
PRM Model	1.29	1550.0	0.8	1.0e4	0.21
μ Model	1.7	570.5	1.0	1.0e4	0.21

5.2.2 Comparison Between Analytical and Experimental Data

The experimental biaxial behavior of the concrete panels was compared with the analytical biaxial behavior obtained using OpenSees. The model was analyzed using the

implemented concrete materials for the scalar damage (SD) model (Mazars, 1986), the “ μ ” model (Mazars, Hamon, & Grange, 2015), and the PRM model (Pontiroli, Rouquand, & Mazars, 2010). The comparison between experimental and analytical results are shown in Fig. 5.2. It can be observed that the three models are reasonable candidates for describing the biaxial behavior of concrete under the uniaxial-compression, compression-tension, and biaxial-tension ranges. However, while the PRM and the scalar damage model do not predict the response in the biaxial compression domain accurately, the “ μ ” model has a superior performance, including the strength increase on the biaxial compression range due to confinement effects (Kupfer, 1969).

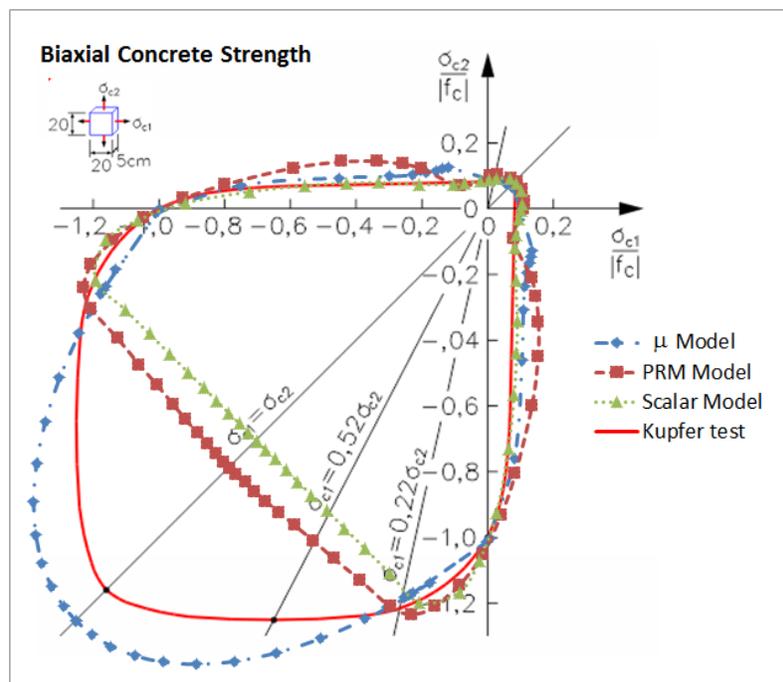


Fig. 5.2. Biaxial behavior of concrete. Adapted from Kupfer et. al. (1969)

5.3 RC Beam Under Monotonical Loading

An under-reinforced RC beam was tested as part of this project. The simply supported beam was subjected to four-point bending loads (monotonic). The concrete properties were

measured as f'_c of 40 MPa, and a Young's modulus (E) of 37.2 GPa. The yield stress (f_y) of the reinforcement steel was measured as 475 MPa, with an E of 183 GPa.

The beam had a rectangular cross-section with dimensions of 150 x 300 mm. The top longitudinal reinforcement consisted of two 10M bars (\varnothing 11.3 mm), and the bottom longitudinal reinforcement consisted of two 15M bars (\varnothing 16 mm). The transverse reinforcement consisted of 10M closed stirrups with a spacing of 220 mm. It was placed only on the first 980 mm starting at each end of the beam. The beam supports (R) were located at 200 mm from each end, and the load (P) was applied at 1/3 of the free span. The failure of the beam was flexure dominated and was due to the crushing of concrete in compression and the buckling of the compressive reinforcement, both of which were located at the middle-top of the beam. The dimensions, loading points, and reinforcement specifications are shown in Fig. 5.3.

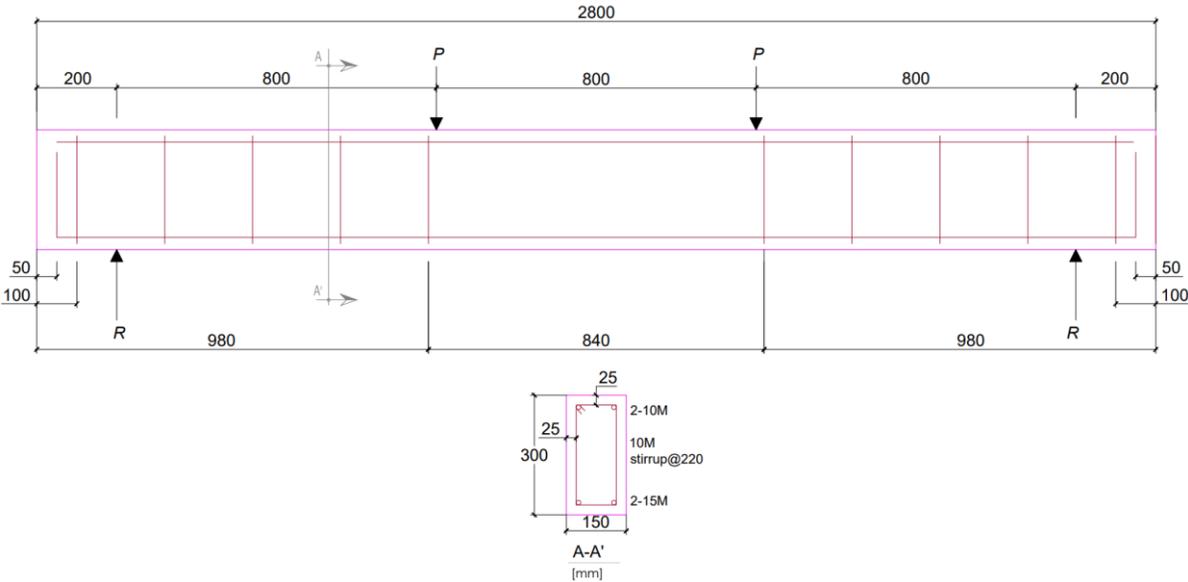


Fig. 5.3. RC beam geometry and reinforcement details

5.3.1 Finite Element Model

The beam was modeled in OpenSees using 312 four-node multilayer shell elements for the concrete. Each element consisted of three layers with a thickness of 50 mm, with a height and width of 50 mm (Fig. 5.4). The concrete materials used were the scalar damage (SD) model (Mazars, 1986), the “ μ ” model (Mazars, Hamon, & Grange, 2015), and the PRM model (Pontiroli, Rouquand, & Mazars, 2010). The concrete parameters for each material are shown in Table 5.2. For the steel reinforcement, 140 truss elements were used. The truss elements were modeled using the Steel02 material, which uses the Giuffre-Menegotto-Pinto steel model with isotropic strain hardening (Filippou, Popov, & Bertero, 1983). The following steel parameters were used: yield strength (F_y) = 475 MPa, the initial elastic tangent (E_0) = 183 GPa, the strain-hardening ratio (b) = 0.015, and the parameters to control the transition from elastic to plastic branches (R_0 , SCR_1 , SCR_2) = 18.5, 0.925, and 0.15, respectively. The pushover analysis was done using a displacement-controlled analysis, with descending displacement applied at nodes 337 and 353. Boundary conditions were introduced at nodes 5 and 49, restricting the vertical displacement, while the middle of the beam was restricted horizontally to allow for symmetry.

Table 5.2. Model parameters for FEM of beam under monotonic loading

Material	E(GPa)	ε_{0c}	$\varepsilon_{0t}/\varepsilon_{D0}$	ε_{fc}	σ_{fc} (MPa)	ε_{ft0}	σ_{ft0} (MPa)
SD Model	37.2	-	5.0e-5	-	-	-	-
PRM Model	37.2	1.3e-4	1.25e-4	5.0e-4	37.2	-3.3e-5	-1.2276
μ Model	37.2	4e-4	5.0e-5	-	-	-	-

Material	Ac	Bc	At	Bt	ν
SD Model	0.73	1065.0	0.97	1.0e4	0.18
PRM Model	0.6	1100.0	0.68	1.0e4	0.18

μ Model	0.645	280.0	0.97	1.0e4	0.18
-------------	-------	-------	------	-------	------

The analysis was performed in two steps: 1) the gravitational load, which in this case was the weight of the beam, was applied at all the top nodes and kept constant; 2) a displacement-controlled analysis was conducted by applying a downward displacement to nodes 337 and 353. The Krylov-Newton algorithm, with current tangent for the iterations, was selected because it is relative lower cost, computationally, for static and quasi-static analyses (Scott & Fenves, 2010). Nodal displacement and corresponding vertical forces were recorded at each converged displacement step.

The adopted mesh (Fig. 5.4), which consisted of square elements of 50 x 50 mm, allowed an adequate modelling of the longitudinal reinforcement steel by placing it in its actual position in comparison with the test. A bigger mesh would not allow for this consistency between the analytical model and the experimental test. A sensitivity analysis showed that a second mesh alternative, consisting of square elements measuring 25 x 25 mm, had a high time-cost (around 1000% slower) and the difference between the calculated results between the first and second mesh options was negligible.

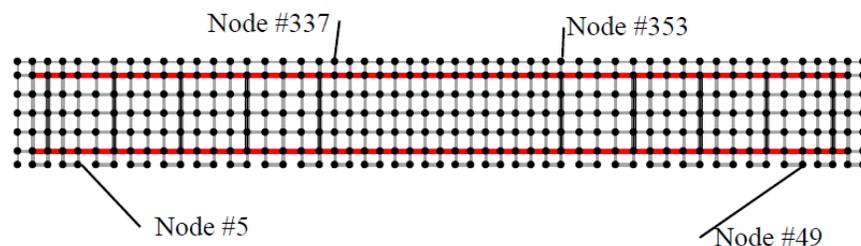


Fig. 5.4. FEM of RC beam under monotonic loading

5.3.2 Comparison Between Analytical and Experimental Data

The comparison between the experimental and analytical results for the load-displacement response of the downward pushover is shown in Fig. 5.5. The model was analyzed using the implemented concrete materials for the scalar damage (SD) model (Mazars, 1986), the “ μ ” model (Mazars, Hamon, & Grange, 2015), and the PRM model (Pontiroli, Rouquand, & Mazars, 2010). The three model materials show an accurate prediction for the load-displacement response, but the PRM and Mazars’ scalar damage models were unable to predict the failure displacement of the beam, whereas the “ μ ” model is able to accurately predict the displacement at which the beam fails. None of the three materials accurately predicted the cracking load, but the “ μ ” model provided a better prediction for the displacement at which the yielding moment occurs. For this experiment, the “ μ ” model performed best, but the three models are all reliable options for predicting the experimental pushover of a simple-supported beam.

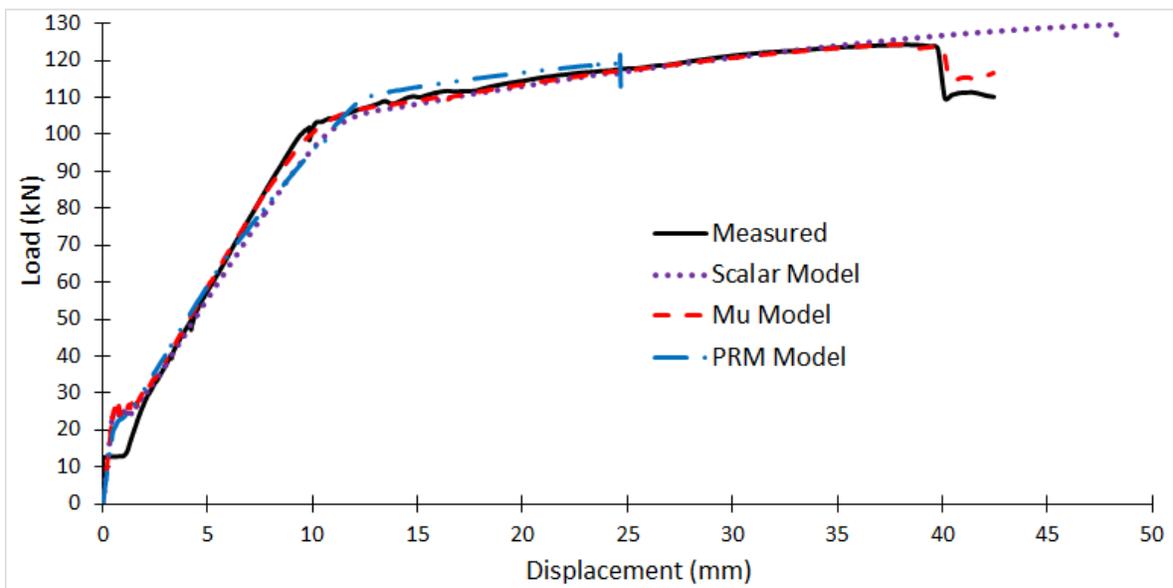


Fig. 5.5. Load-displacement response of RC beam under monotonic loading

5.4 RC Beam Under Reversal-Cyclic Loading

The experimental test consisted of a four-point bending beam subjected to cyclic loading (Ranjbaran, Rezayfar, & Mirzababai, 2018). The compressive stress (f'_c) consisted of 34.69 MPa. The yield stress of the steel (f_y) was specified as 392 MPa, with Young's modulus (E) of 200 GPa. The beam had a rectangular cross-section with dimensions of 250 x 250 mm (Fig. 5.6). The longitudinal reinforcement for both the top and bottom of the beam consisted of three $\text{Ø}10$ bars, whereas the transverse reinforcement consisted of $\text{Ø}8$ bars with a spacing of 15 cm along the whole length of the beam.

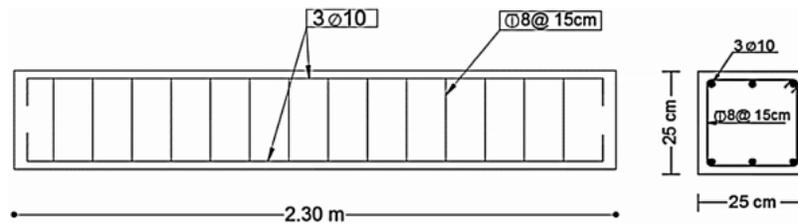


Fig. 5.6. Geometry and reinforcement details of RC beam under cyclic loading (Ranjbaran, Rezayfar, & Mirzababai, 2018)

The displacement-controlled loading was applied at one-quarter of the total length of the beam (both ends). The first two loading cycles were ± 1 mm and ± 2 mm, and the cycles that followed increased by ± 2 mm until 30 mm were reached (Fig. 5.7). The failure of the beam was flexure dominated and was due to the crushing of the concrete external fibers in compression. The test setup is shown in Fig. 5.8.

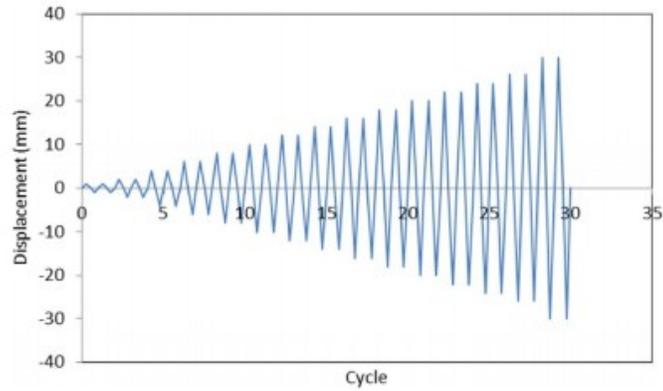


Fig. 5.7. Imposed loading history (Ranjbaran, Rezayfar, & Mirzababai, 2018)

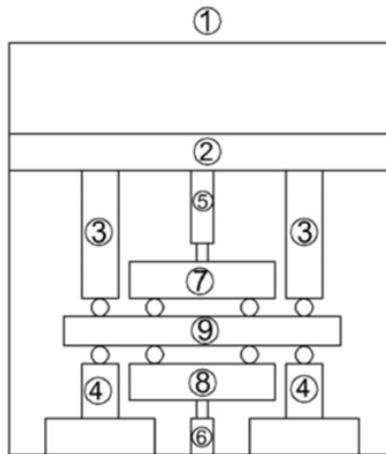


Fig. 5.8. Test set-up (Ranjbaran, Rezayfar, & Mirzababai, 2018)

5.4.1 Finite Element Model

The beam final model was built using 322 four-node multilayer-shell elements for the concrete (Fig. 5.9). Each element consisted of five layers with a thickness of 50 mm. The concrete materials used were the scalar damage (SD) model (Mazars, 1986), the PRM model (Pontiroli, Rouquand, & Mazars, 2010), and the “ μ ” model (Mazars, Hamon, &

Grange, 2015). The concrete parameters for each material are shown in Table 5.3. For the steel reinforcement, 197 truss elements were needed. The truss elements were modeled using the Giuffre- Menegotto-Pinto steel model with isotropic strain hardening (Filippou, Popov, & Bertero, 1983). The adopted mesh was able to accommodate the steel reinforcement in their actual positions within the beam. The middle of the beam was fixed in the horizontal and out of plane direction to account for symmetry. The supports were fixed only in the vertical direction; they were assigned to the nodes 4, 333, 44 and 373.

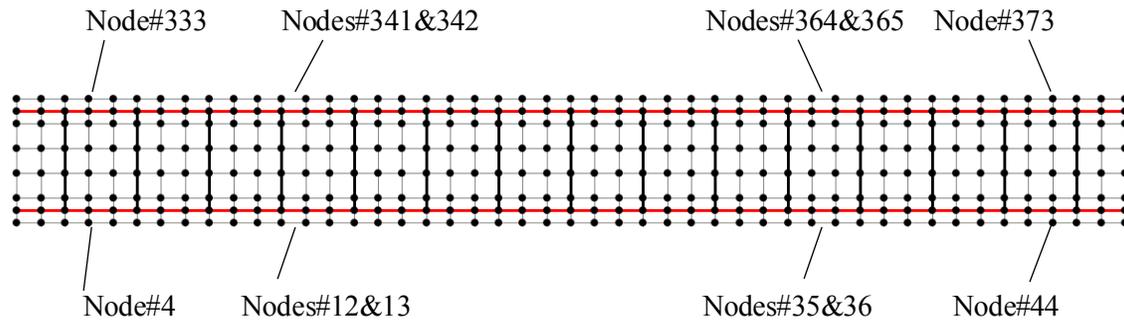


Fig. 5.9. FEM of RC beam under cyclic loading

Table 5.3. Model parameters for FEM of beam under cyclic loading

Material	E(GPa)	ϵ_{0c}	$\epsilon_{0t}/\epsilon_{D0}$	ϵ_{fc}	σ_{fc} (MPa)	ϵ_{ft0}	σ_{ft0} (MPa)
SD Model	34.69	-	5.0e-5	-	-	-	-
PRM Model	34.69	1.0e-4	7.5e-5	5.0e-4	17.3	-3.3e-5	-1.145
μ Model	34.69	3.57e-4	1.14e-5	-	-	-	-

Material	Ac	Bc	At	Bt	ν
SD Model	0.9	1200.0	0.97	1.0e4	0.18
PRM Model	0.9	1850.0	0.8	7.0e3	0.18
μ Model	0.75	302.0	0.97	1.0e4	0.18

The analysis was performed in two steps, 1) the gravitational load, in this case the weight of the beam, was applied to all the top nodes and kept constant, 2) a displacement-controlled analysis was conducted by applying a downwards displacement to nodes 12, 13, 341, 342, 35, 36, 364 and 365. The deformed shape of the beam is shown in Fig. 5.10, where an adequate overall behavior of the beam can be observed for two different cycles. The Krylov-Newton algorithm, with current tangent for the iterations, was selected on the basis that it is less computationally expensive for use in static and quasi-static analyses (Scott and Fenves, 2003). At each converged displacement step, nodal displacement and corresponding vertical forces were recorded. The same mesh size selected for the simple-supported beam under monotonic loading in the previous section was selected for this test because of the similarity between the dimensions of the two beams.

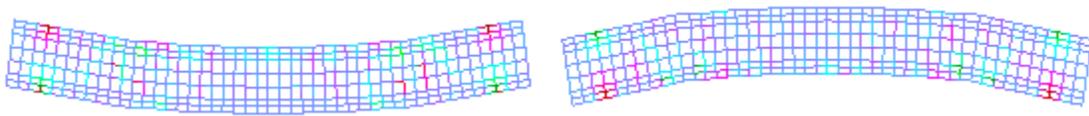


Fig. 5.10. FEM deformed shape of a loading cyclic

5.4.2 Comparison Between Analytical and Experimental Data

The comparison between the experimental and analytical results for the load-displacement hysteretic response of the reversal-cyclic loading for the scalar damage model and the “ μ ” model is shown in Fig. 5.11 and Fig. 5.12, respectively. The ultimate load and ultimate displacement were overestimated by both the scalar damage (SD) model (Fig. 5.11) and the

“ μ ” model (Fig. 5.12). Another important feature to consider is the first cycles of the analysis, where the “ μ ” model shows higher energy dissipation than the scalar damage model, because it can account for stiffness recovery when the cracks close under cyclic loading. Fig. 5.13 shows a snapshot of the first 7 cycles of the experiment, where the “ μ ” model (Fig. 5.13a) shows better agreement than the scalar damage model (Fig. 5.13b) for the overall load-displacement response, including the initial cracking load and the yielding load .

The predictions of the scalar damage (SD) model and the “ μ ” model failed to represent the pinching effect of the concrete, as observed on the first cycles of the hysteretic loops of the experiment. Neither of these concrete models account for permanent strains, which means that the unloading and reloading paths after each cycle will always return to the origin of the stress-strain relationship. The plastic deformations observed in the beam calculated response, are therefore generated from the residual strains in the steel reinforcement.

Although unable to account for permanent deformation, the global results were predicted with moderate agreement.

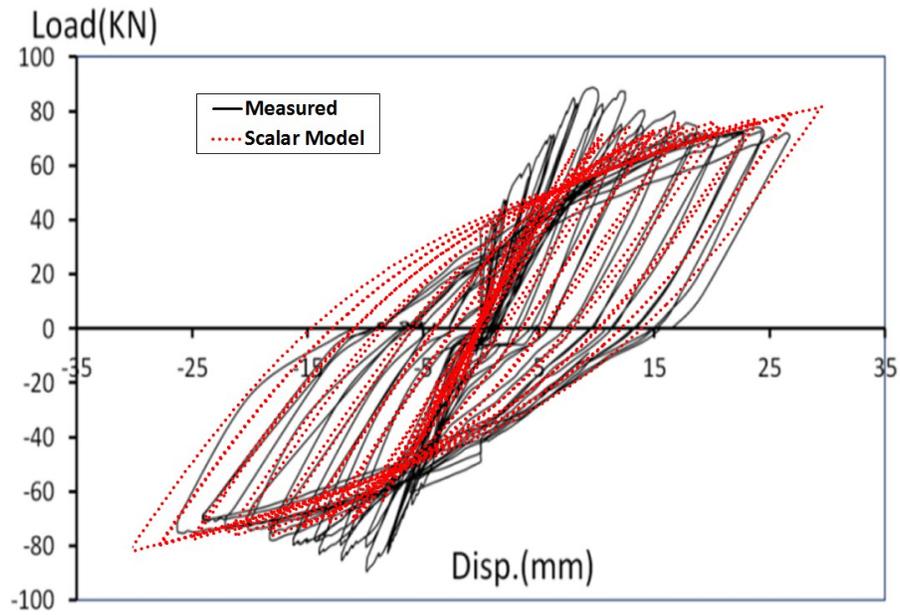


Fig. 5.11. Measured vs calculated load-displacement hysteretic response using the scalar damage model. Adapted from Ranjabaran (2018)

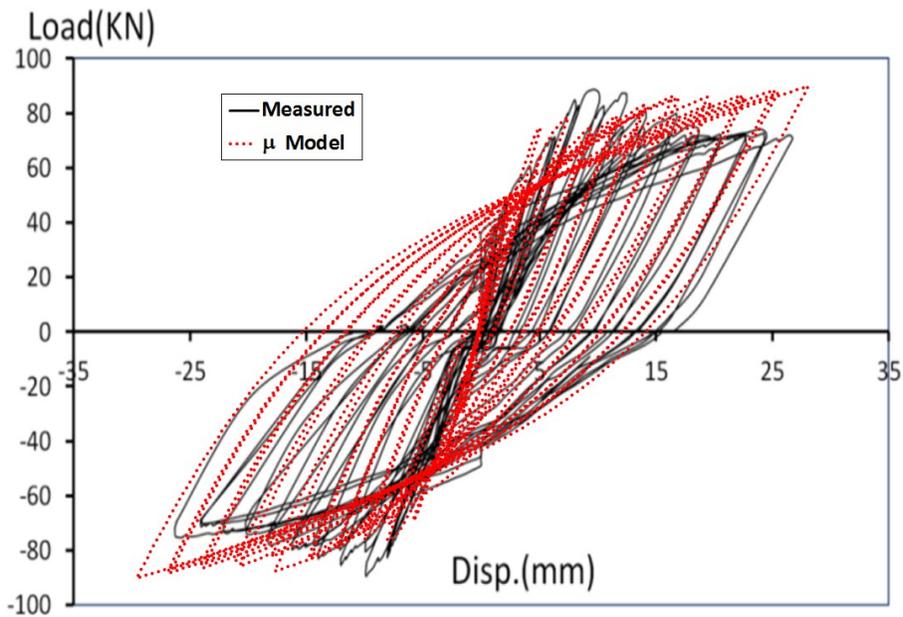


Fig. 5.12. Measured vs calculated load-displacement hysteretic response using the “μ” model. . Adapted from Ranjabaran (2018)

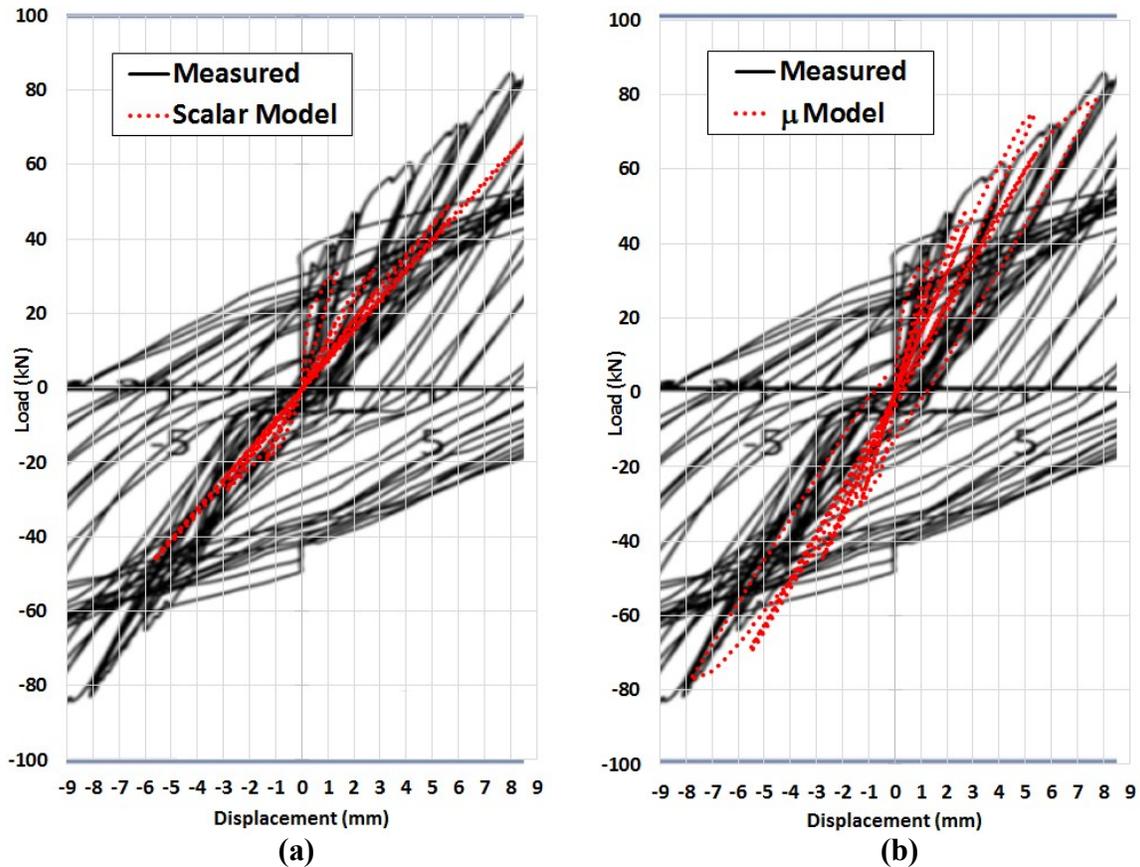


Fig. 5.13. Measured vs calculated load-displacement hysteretic response for the (a) SD model and (b) “ μ ” model. First 7 cycles of the calculated response. Adapted from Ranjabaran (2018)

The comparison between the experimental and analytical results for the load-displacement hysteretic response of the reversal-cyclic loading for the PRM model is shown in Fig. 5.14. This model shows a better agreement for each unloading and reloading cycle of the hysteretic response. However, due to the complex formulations required to account for permanent deformations in two directions, the analysis was unable to achieve convergence for the full loading path using the current solver algorithms available in OpenSees.

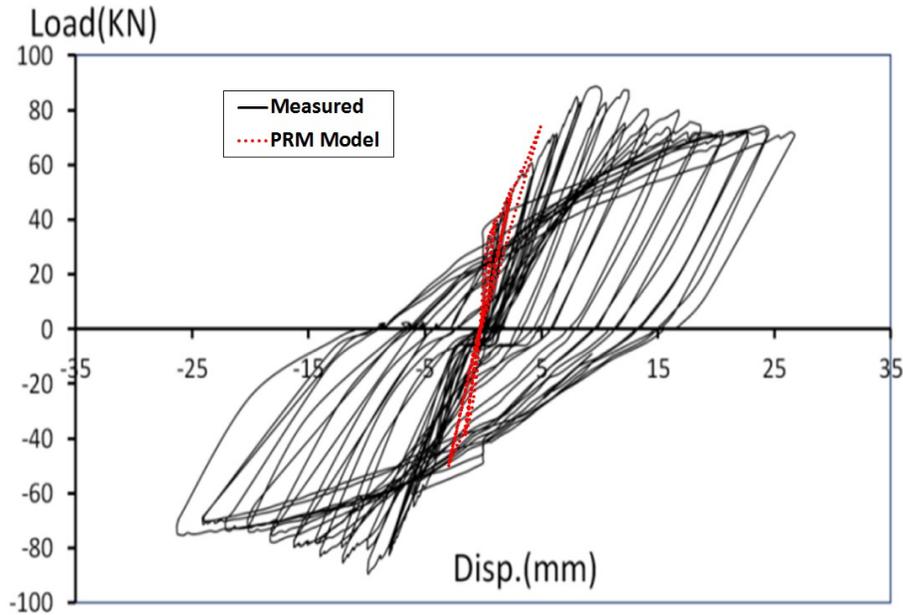


Fig. 5.14. Measured vs calculated load-displacement hysteretic response using the PRM model. Adapted from Ranjabaran (2018).

5.5 RC Shear Wall Under Reversal-Cyclic Loading

The experimental test consisted of a quad shear wall under reversal-cyclic loading (Hiotakis, 2004). The experimental specimen consisted of a heavily reinforced foundation block, a cantilevered shear wall with dimensions of 100 mm thick, 1500 mm wide x 1795 mm high, and a cap beam at the top to distribute the applied load (Fig. 5.15). Reversal-cyclic lateral loading of increasing magnitude was applied to the wall at the cap beam through hydraulic actuators. The vertical reinforcement of the shear wall was composed of six pairs of 10M ($\text{\O} 11.3 \text{ mm}$) bars uniformly distributed along the wall with a spacing of 280 mm including the boundary elements, with a corresponding steel reinforcement ratio (ρ_s) of 0.8%. The horizontal reinforcement of the shear wall consisted of five pairs of 10M

(Ø11.3 mm) bars uniformly distributed and spaced at 400 mm, which corresponded to a steel reinforcement ratio (ρ_s) of 0.5%. The boundary elements had stirrups consisting of 10M (Ø11.3 mm) with a spacing of 80 mm along the height of the wall boundary elements. The concrete compressive cylinder strength (f'_c) at the time of test was 36.2 MPa, and the yield stress of the reinforcing steel bars was 425 MPa.

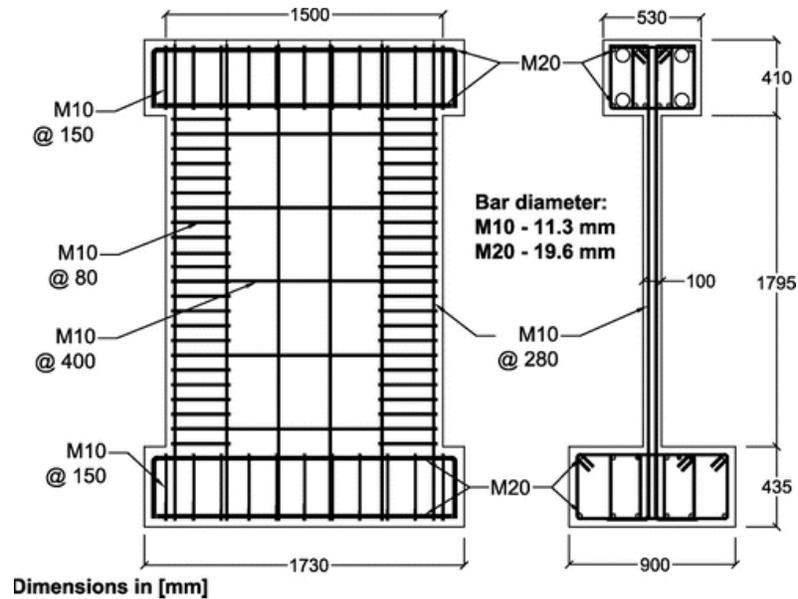


Fig. 5.15. Shear wall reinforcement and geometry specifications

5.5.1 Finite Element Model

The concrete materials used were the scalar damage (SD) model (Mazars, 1986), the PRM model (Pontiroli, Rouquand, & Mazars, 2010), and the “ μ ” model (Mazars, Hamon, & Grange, 2015). The shear wall model was built using 414 four-node rectangular, multilayered shell elements with a size of 80 x 80 mm. The multilayered shell elements were divided into wall boundary elements (shaded area) and wall core elements (Fig. 5.16a). The layers of the multilayered shell elements of the wall core were divided into

concrete layers and smeared steel layers for the horizontal and vertical reinforcement. The thickness of the steel layers (t_s) was calculated using Eq. (5.1), where t_s was calculated as a function of the shear wall thickness (t_c), the reinforcement steel area (A_s), and the concrete area (A_c). The layers of the multilayered shell elements of the wall boundaries were divided into concrete layers only. For the longitudinal steel reinforcement and the stirrups of the boundary elements, 696 truss elements were needed. The steel reinforcement for the multilayered shell elements and the truss elements was modeled using the Giuffre- Menegotto-Pinto steel model with isotropic strain hardening (Menegotto, 1978).

$$\rho_s = \frac{A_s}{A_c} = \frac{t_s}{t_c} \quad (5.1)$$

The multilayer-shell elements of the wall core consisted of 10 layers (Fig. 5.16b). The two outer layers composed the concrete cover with a thickness of 12.5 mm each. Two layers composed the horizontal reinforcement with a thickness of 0.2778 mm each. Two layers composed the vertical reinforcement with a thickness of 0.2381 mm each. The four inner concrete layers had a thickness of 18.4921 mm each. The multilayer-shell elements of the wall boundaries consisted of 8 concrete layers with a thickness of 12.5 mm each. These added up to a total of 100 mm for both types of multilayer-shell elements, which is the same as the thickness of the wall. The parameters for each concrete model are shown in Table 5.4.

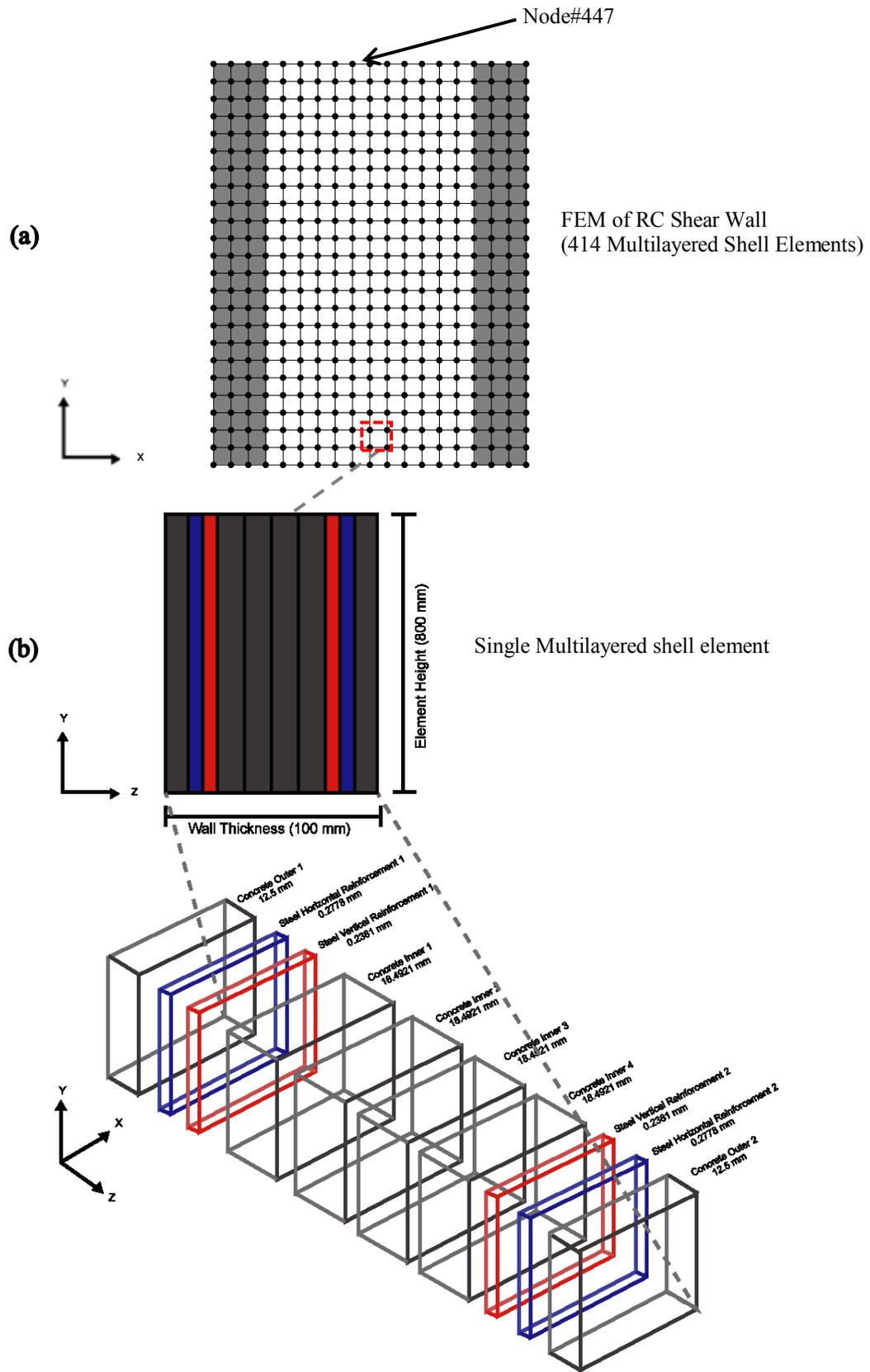


Fig. 5.16. FEM of RC Shear Wall (a) and multilayered shell element (b).

Table 5.4. Model parameters for FEM of shear wall under cyclic loading

Material	E(GPa)	ε_{0c}	$\varepsilon_{0t}/\varepsilon_{D0}$	ε_{fc}	σ_{fc} (MPa)	ε_{ft0}	σ_{ft0} (MPa)
SD Model	36.2	-	1.0e-4	-	-	-	-
PRM Model	36.2	1.0e-4	1.0e-4	5.0e-4	18.1	-3.3e-5	-1.1946
μ Model	36.2	4.0e-4	1.0e-5	-	-	-	-

Material	Ac	Bc	At	Bt	ν
SD Model	1.25	2250.0	0.97	1.0e4	0.18
PRM Model	0.9	1850.0	0.8	7.0e3	0.18
μ Model	1.25	650.0	0.97	1.0e4	0.18

The cap-beam and the foundation block were assumed to be stiffer than the shear wall due to the high amounts of steel reinforcement and their larger cross-sections compared to that of the wall. For simplicity, the bottom and top elements were not modelled and instead were accounted for by combining the horizontal degree of freedoms of all the nodes in the top of the wall, and fixing all the degrees of freedom of the nodes at the base. A cyclic displacement history was applied to the centre top node of the wall (node 447) in the horizontal direction. The horizontal reactions and the base of the wall were recorded.

The load pattern in the finite element model is determined according to the test conditions. The cyclic load was applied as a horizontal nodal displacement at node 447 (Fig. 5.16a), which varied in relation to the displacement control scheme. The Krylov-Newton algorithm, with current tangent for the iterations, was selected for the analysis. The nodal displacement and corresponding horizontal forces were recorded, and the stress and strain of the elements were monitored at each converged displacement step.

A mesh sensitivity analysis was performed using the “ μ ” model to make sure the adopted mesh was adequate (Table 5.5). The eleventh load step in the positive direction was

analyzed, which measured results that showed a maximum shear force (V_f) of 155 kN at a displacement of 6.2 mm. Four different mesh sizes (total number of elements) were selected: 99, 180, 414, and 1656. Each mesh size consisted of elements with dimensions of 160 x 160 mm, 120 x 120 mm, 80 x 80 mm, and 40 x 40 mm, respectively. As shown in Table 5., the mesh size of 414 elements has an acceptable error of 6.45 %, and convergence errors were presented when using elements of half the size.

Table 5.5. Mesh size analysis of the shear wall FEM.

Mesh Size (# of elements)	Individual element dimensions (mm)	V_f (kN)	Error %
99	160 x 160	209	34.8
180	120 x 120	182	17.4
414	80 x 80	165	6.45
1656	40 x 40	Convergence Error	-
Test	-	155	-

5.5.2 Comparison Between Analytical and Experimental Data

The calculated vs. measured global base shear-top displacement response of the shear wall for the scalar damage (SD) model and the “ μ ” model is presented in Fig. 5.17 and Fig. 5.18. The maximum strength and displacement calculated at each cycle with both models show a reasonable correspondence with the measured results. These analyses could not represent the pinching effect of the concrete. Neither of these models account for permanent strains, which means that the unloading and reloading paths after each cycle will always return to the origin of the stress-strain relationship. The plastic deformations observed in the shear wall calculated response are generated from the residual strains in the steel reinforcement. The wall strength after the steel yields is underestimated (Fig. 5.17 and Fig. 5.18), which could be explained by the fact that the cyclic hardening of the steel material (steel02) is not predicting the experimental results correctly.

If we take a closer look at the first cycles of the scalar damage (SD) model and the “ μ ” model, Fig. 5.19 and Fig. 5.20 respectively, the fact that the scalar damage model does not separate the damage into tension and compression causes it to overestimate the force at the first crack, while the “ μ ” model can predict the force that causes the first crack, the overall strength, and the loading and unloading paths of the wall with more accuracy.

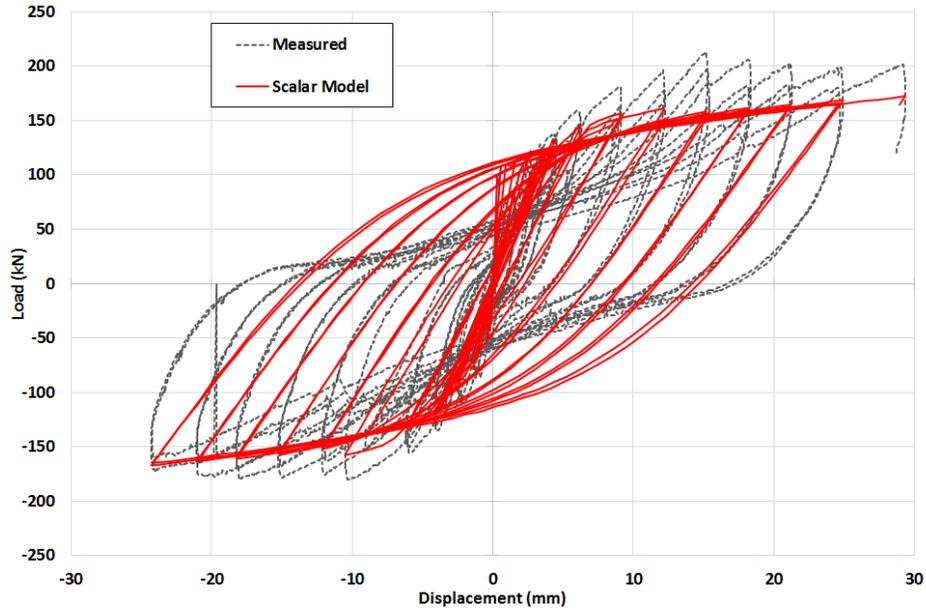


Fig. 5.17. Measured vs calculated load-displacement hysteretic response using the scalar damage model

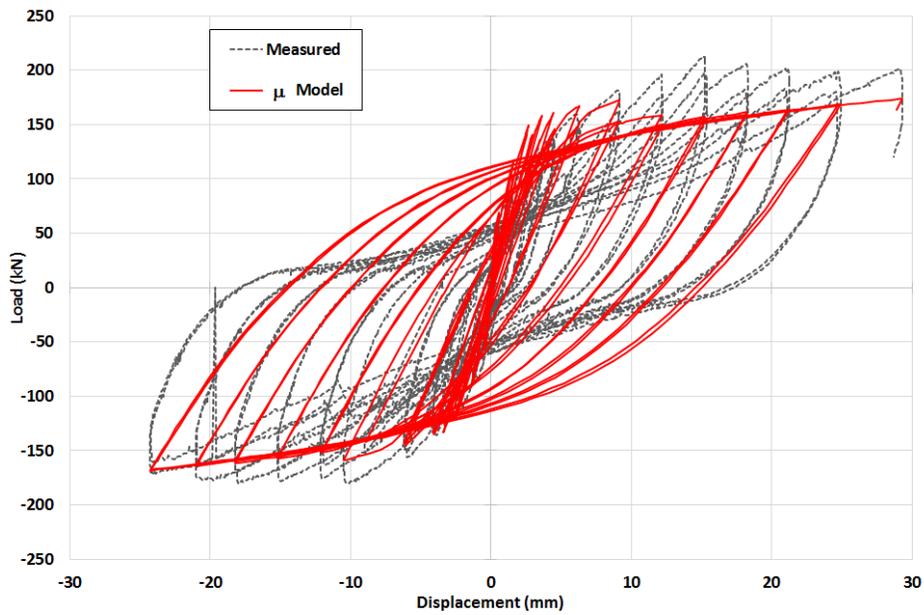


Fig. 5.18. Measured vs calculated scalar load-displacement hysteretic response using the “ μ ” model

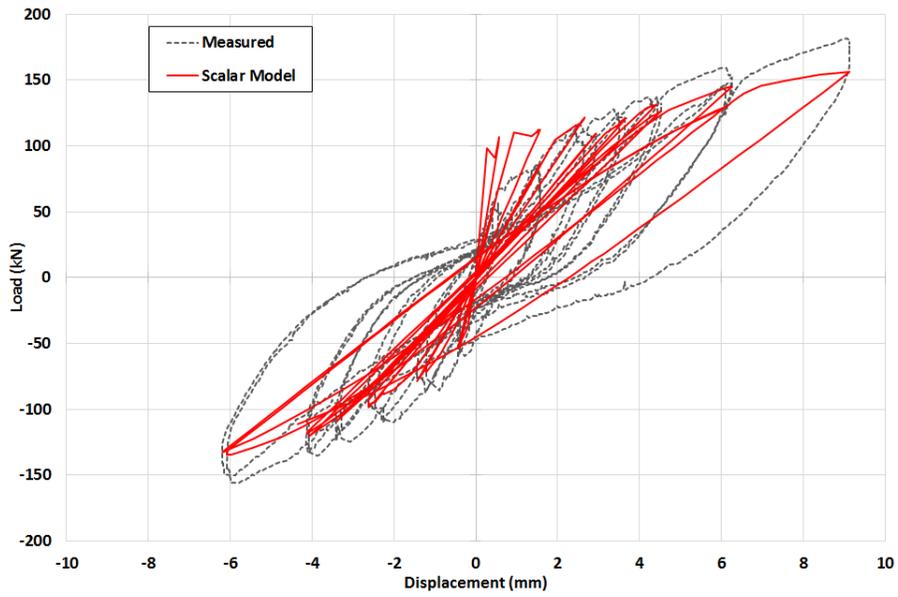


Fig. 5.19. Measured vs calculated first 12 cycles of the load-displacement hysteretic response using the scalar damage model

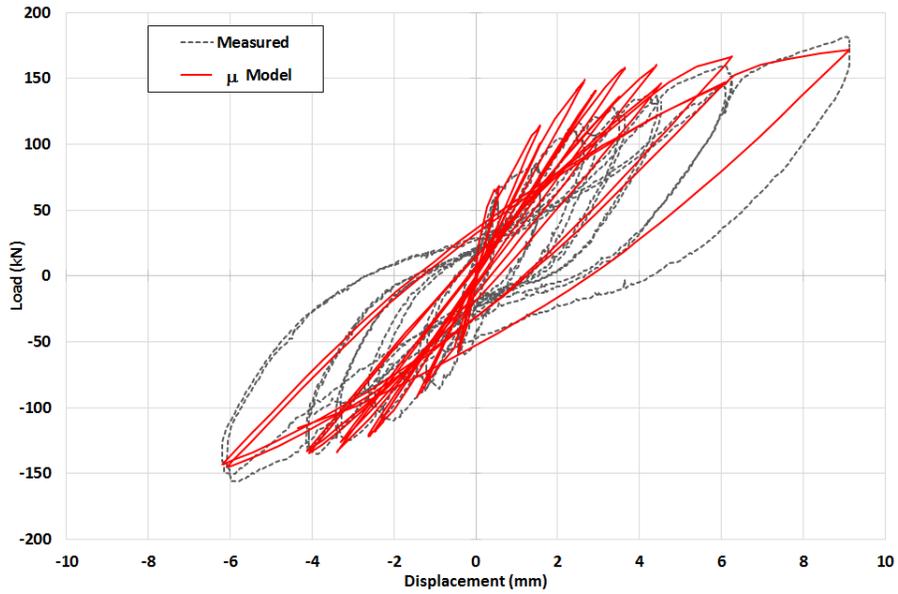


Fig. 5.20. Measured vs. calculated first 12 cycles of the load-displacement hysteretic response using the “μ” model

Fig. 5.21 shows the calculated vs. measured global base shear-top displacement response of the shear wall for the PRM model. In contrast with the “ μ ” and the scalar damage models (Fig. 5.17 and Fig. 5.18), the PRM model can account for the permanent displacement of the concrete, which makes the model capable of representing the pinching effects of the concrete. This model shows a better agreement for each unloading and reloading cycle of the hysteretic response. Unfortunately, due to the complex formulations required to account for permanent deformation in two directions, the analysis was unable to achieve convergence for the full loading path using the current solver algorithms available in OpenSees.

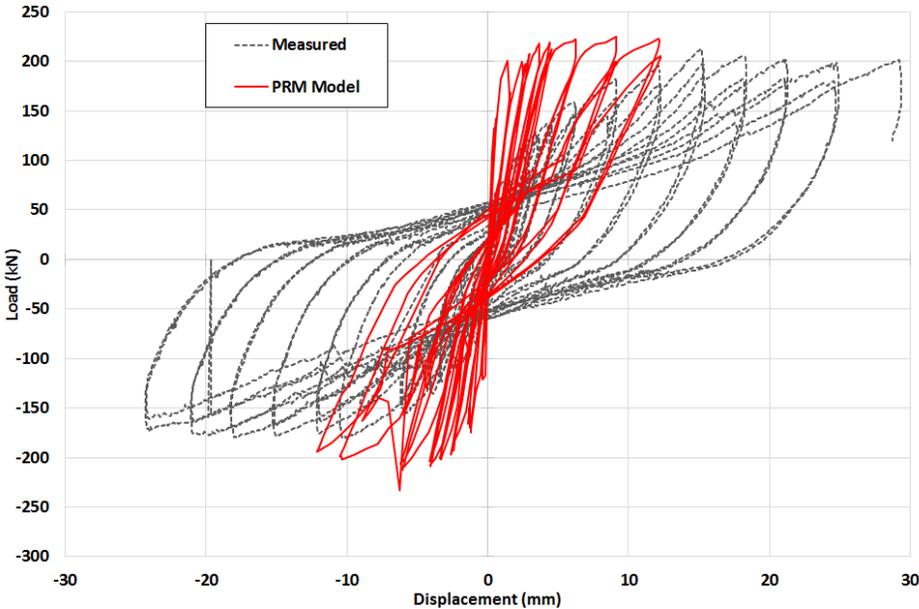


Fig. 5.21. Measured vs. calculated load-displacement hysteretic response using the PRM model

5.6 Full-Scale Four-Storey RC Building Under Seismic Loading

A full-scale, four-storey, reinforced concrete building was tested using the E-Defense shake in 2010 in Japan (Nagae, et al., 2015). The plan view and the longitudinal and transverse elevations can be observed in Fig. 5.22. The storey height was 3 m, and the footprint measured 14.4 m in the longitudinal direction (L) and 7.2 m in the transverse direction (T). The lateral-force resisting system consisted of a two-bay RC moment frame system in the longitudinal direction on axes 1 and 2. In the transverse direction, a pair of multi-storey RC shear walls was incorporated within a moment-resisting frame in exterior axes A and C, and a single-bay moment-resisting frame in the middle axis B. To the knowledge of the author, this is the only full-scale structure tested in the world in which relevant input & output data is available.

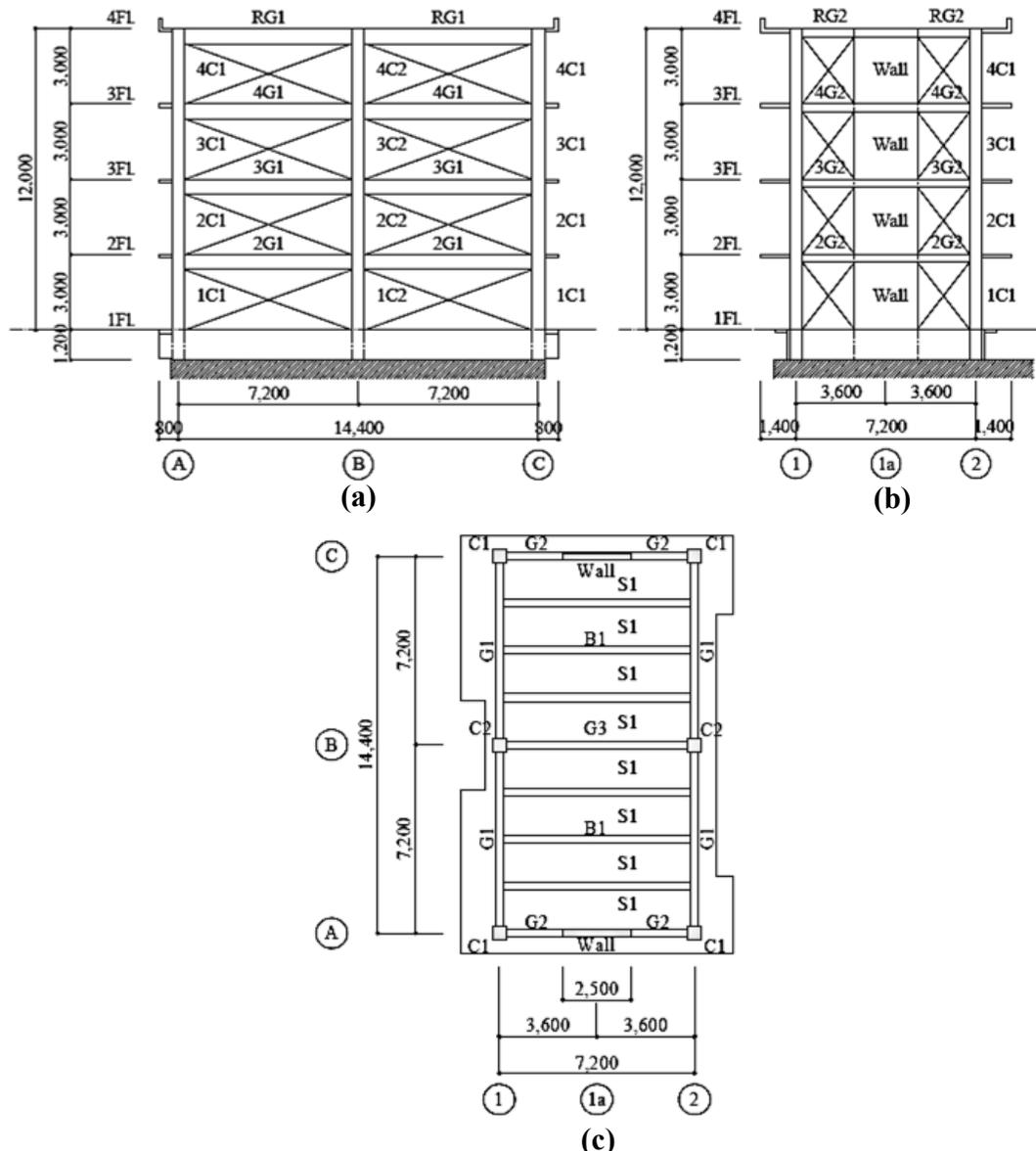


Fig. 5.22. Four-storey building geometry and structural members nomenclature: (a) longitudinal elevation; (b) transverse elevation; and (c) plan view (Nagae, et. al., 2011b)

The cross-sections and reinforcement specifications of each structural member are shown in Fig. 5.23. The columns of the structure have a rectangular cross-section with dimensions of 500 x 500 mm. The girders G1 in the L-direction have a rectangular cross-section with dimensions of 300 x 600 mm. In the T-direction the girders G2 and G3 dimensions are 300 x 300 mm and 300 x 400 mm respectively, both with a rectangular cross-section. The

shear-walls have a rectangular cross-section with dimensions of 250 x 2500 mm. The beams B1 had a rectangular cross-section of 300 x 400 mm. The slab S1 had a thickness of 120 mm. The steel reinforcement of each section varies according to the storey level as shown in Fig. 5.23.

List of Column			
		C1	C2
4FL 3FL	Section		
	B x D	500 x 500	500 x 500
	Rebar	8-D22	10-D22
	Hoop	2,2-D10@100	2,2-D10@100
	Joint	2,2-D10@140	2,2-D10@140
2FL	Section		
	B x D	500 x 500	500 x 500
	Rebar	8-D22	10-D22
	Hoop	2,3-D10@100	2,4-D10@100
	Joint	2,2-D10@140	2,2-D10@140
1FL	Top Section		
	B x D	500 x 500	
	Rebar	8-D22	
	Hoop	2,3-D10@100	
	Joint	2,2-D10@140	
	Bottom Section		
	B x D	500 x 500	500 x 500
	Rebar	10-D22	10-D22
	Hoop	3,4-D10@100	3,4-D10@100
	Joint	2,2-D10@140	2,2-D10@140

List of Wall				
	Wall			
4FL 3FL 2FL	Section			
	B x D	2,500 x 250		
	Rebar	2 x 6-D19	Vertical	D13@300 (W)
	Hoop	2,2-D10@100	Horizontal	(A) D10@125 (W)
			(C) D10@200 (W)	
Joint	2,2-D10@150			
1FL	Section			
	B x D	2,500 x 250		
	Rebar	2 x 6-D19	Vertical	D13@300 (W)
	Hoop	(A) 2,3-D10@80	Horizontal	(A) D10@125 (W)
				(C) D10@200 (W)
	Joint	2,2-D10@150		

List of Slab			
Depth: 130mm			
		Shorter direction	Longer direction
S1	Top	D10@200	D10@250
	Bottom	D10@200	D10@250
CS1	Top	D10,D13@200	D10@250
	Bottom	D10@200	D10@250
CS2	Top	D10@200	D10@250
	Bottom	D10@200	D10@250
CS3	Top	D10,D13@200	D10,D13@200
	Bottom	D10@200	D10@200

List of Girder					
	G1				
	Location	End	Center	End	
RFL 4FL	Section				
	B x D	300 x 600			
	Top	4-D22	3-D22	4-D22	
	Bottom	3-D22	3-D22	3-D22	
	Web	4-D10			
	Stirrup	2-D10@200			
3FL	Section				
	B x D	300 x 600			
	Top	5-D22	3-D22	5-D22	
	Bottom	3-D22	3-D22	3-D22	
	Web	4-D10			
	Stirrup	2-D10@200			
2FL	Section				
	B x D	300 x 600			
	Top	6-D22	3-D22	6-D22	
	Bottom	3-D22	3-D22	3-D22	
	Web	4-D10			
	Stirrup	2-D10@200			

List of Girder				
	G2			
	Location	End	Center	
RFL	Section			
	B x D	300 x 300		
	Top	3-D19	3-D19	
	Bottom	2-D19	3-D19	
	Web	-		
	Stirrup	2-D10@100(KSS785)		
4FL 3FL 2FL	Section			
	B x D	300 x 300		
	Top	3-D19	4-D19	
	Bottom	3-D19	3-D19	
	Web	-		
	Stirrup	2-D10@100(KSS785)		

List of Girder				
	G3			
	Location	End	Center	
4FL 3FL 2FL	Section			
	B x D	300 x 400		
	Top	5-D19	3-D19	
	Bottom	3-D19	4-D19	
	Web	2-D10		
	Stirrup	2-D10@200		
1FL	Section			
	B x D	300 x 400		
	Top	4-D19	3-D19	
	Bottom	3-D19	4-D19	
	Web	2-D10		
	Stirrup	2-D10@200		

List of beam				
	B1			
	Location	End	Center	
All	Section			
	B x D	300 x 400		
	Top	3-D19	3-D19	
	Bottom	4-D19	7-D19	
	Web	2-D10		
	Stirrup	2-D10@200		

Fig. 5.23. Structural members details (Nagae, et. al., 2011b)

The concrete nominal compression strength (f'_c) was 27 MPa, the longitudinal reinforcement had a nominal yield strength of 345 MPa, and the shear reinforcement had a nominal yield strength of 295 MPa. Table 5.6 shows the actual material properties at the time of the test, including the steel yield strength (σ_y) and rupture strength (σ_t), and the concrete compressive strength (σ_b). The area of each reinforcement rebar type is shown as well.

The structure was designed using the seismic provisions of the Japanese Code (AIJ, 1999), and the detailing of the frame nodes was carefully adapted to also comply with the American ACI code (318-11). The estimated weight of the structure accounted for the structural members, the measuring equipment, and the safety steel frames. Table 5.6 shows the weight per floor and components of the structure considered for the dynamic and gravitational analysis. The floors of the building weighed 867 kN for the second floor, 872 kN for the third floor, 867 kN for the fourth floor, and 934 kN for the roof, resulting in a total estimated weight of 5877 kN for the structure.

Table 5.6. RC building actual material properties (Nagae, et al., 2015)

Steel					Concrete			
	Grade	A_{nominal} (mm ²)	σ_y (N/mm ²)	σ_t (N/mm ²)		F_c (N/mm ²)	σ_B (N/mm ²)	Age (Days)
D22	SD345	387	370	555	1st - 2nd floor	27	39.6	91
D19	SD345	287	380	563	2nd - 3rd floor	27	39.2	79
D13	SD295	127	372	522	3rd - 4th floor	27	30.2	65
D10	SD295	71	388	513	4th - roof floor	27	41.0	53
D10*	SD295	71	448	545				
D10*	KSS785	71	952	1055				

The Kobe records for the 1995 Hyogoken-Nanbu earthquake served as the input ground motions for the test. The accelerograms in the North-South, East-West, and up-down directions were provided as input motions for the longitudinal direction, transverse direction, and vertical direction, respectively. Fig. 5.24 illustrates the acceleration records for the seismic motions. During the test, the intensity of the motions was gradually increased to observe the damage process in the structure. The intensity increments for the Kobe record were 25%, 50%, and 100% of the recorded motions for the earthquake.

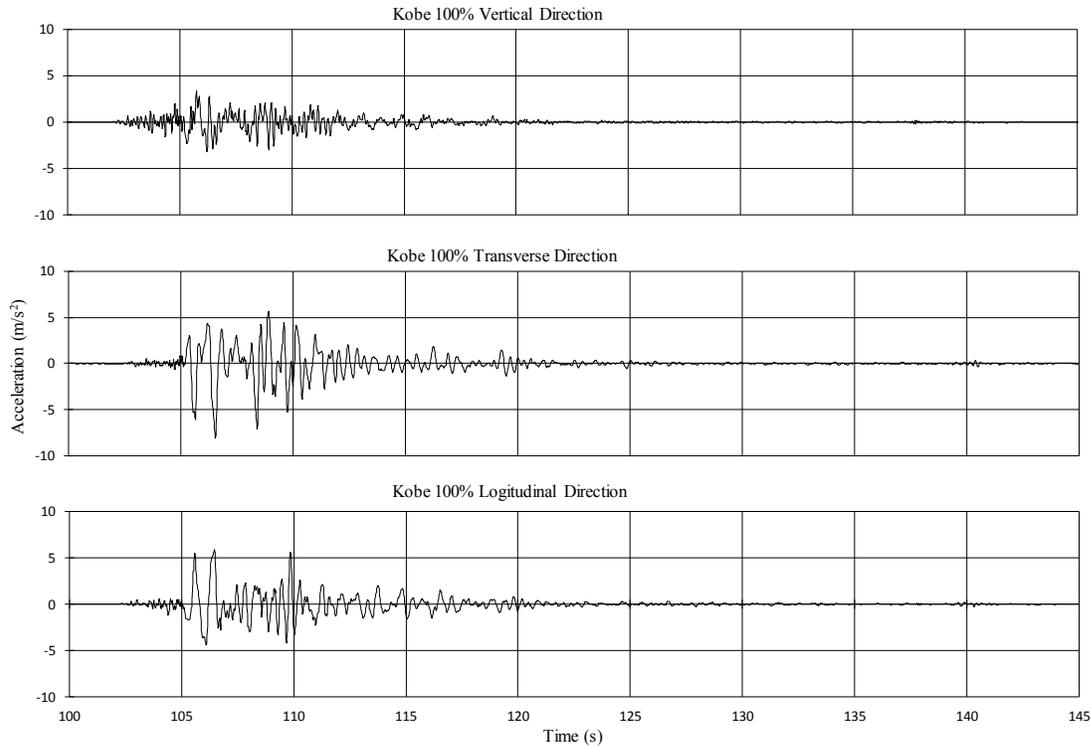


Fig. 5.24. Ground motion time history

5.6.1 Finite Element Model

An analytical model of the full structure was created in OpenSees (Fig. 5.25). The newly implemented “ μ ” model material was assigned to the 3D multilayered shell elements used to model the shear walls. The “ μ ” model was the only concrete material used for 3D elements because it showed better prediction of the cyclic behavior of a shear wall than the scalar damage model. Due to the complexity of the full-structure model, the girders, beams, and columns were modeled using frame elements composed of uniaxial concrete and steel fibers. The actual material parameters were used to model the structure (Table 5.6), and the confinement parameters that provided the stirrups were calculated using

Mander's model (Mander et al., 1988). A rigid diaphragm was assigned at each floor, using a number of multi-point constraint objects. The base of the structure was fixed, but the bond-slip phenomenon at the base of the columns was evaluated using zero-length elements with a hysteretic material that is able to predict this behavior.

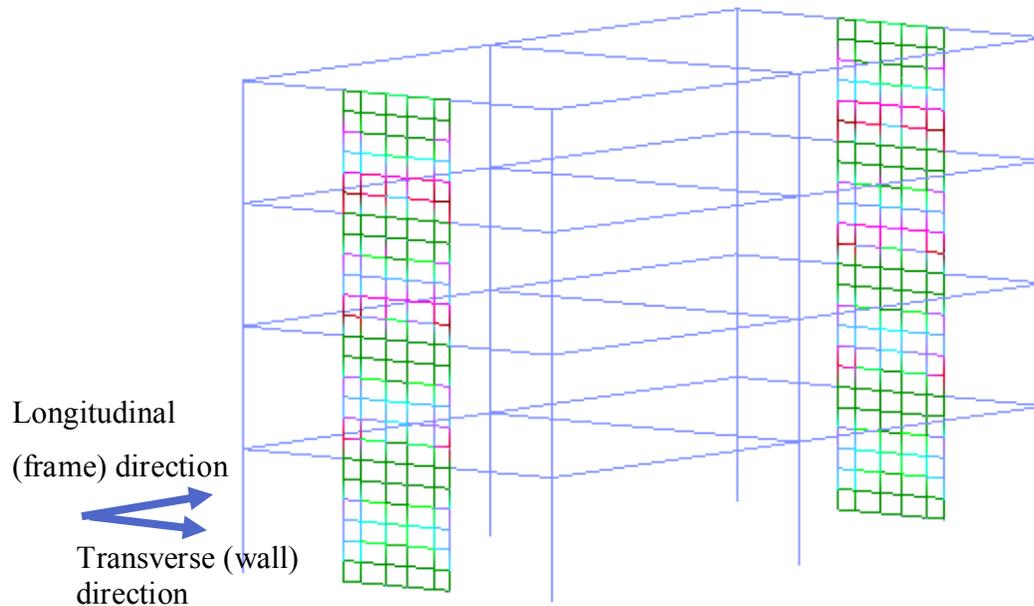


Fig. 5.25. FEM of RC building.

Smearred steel reinforcement was used for the multilayered shell elements of the shear walls. The concrete material used for the frame elements of the girders and columns was the Kent-Scott-Park concrete material (Scott et al., 1982). The steel material used for modelling the reinforcement of all elements was the Giuffre-Menegotto-Pinto steel material with isotropic strain hardening (Menegotto and Pinto, 1973).

The loading pattern was divided into two steps: 1) the gravitational load was applied and kept constant on the horizontal members and on the upper end of the columns and walls, 2) a ground motion displacement loading path was created in each direction and imposed on all of the base nodes. The transient analysis was performed using the Modified Newton

algorithm, with the UmfPack integrator to solve the system of equations. The damping was defined using Rayleigh Reitz damping of 5% in the first three modes of vibration. The nodal displacements of each floor were recorded at each converged step.

5.6.2 Comparison Between Analytical and Experimental Data

The global base-shear force measured and calculated for the 25%, 50%, and 100% Kobe records are presented in Fig. 5.26, Fig. 5.27, and Fig. 5.28, respectively, for both the longitudinal and transverse direction (Fig. 5.22). For the 25% and 50% Kobe records on the transverse (shear wall) direction, both the magnitude of the calculated global base shear response and the wave shape show a very close prediction of the measured results, Fig. 5.26(a) and Fig. 5.27(a), respectively. While the longitudinal or frames direction showed reasonable agreement for the magnitude and wave shape of the first half of the response, for the second half, the magnitude of the global base shear is overestimated (Fig. 5.26b and Fig. 5.27b). This means that the analytical model is dissipating less energy than the experimental test on the longitudinal direction for the 25% and 50% Kobe records. For the 100% Kobe record, wave shape couldn't be predicted for both directions. One explanation for the discrepancy on the energy dissipation between the analytical and experimental results is the inability of the concrete materials to reproduce the permanent deformations. However, the shear walls in the transverse direction, modelled using the “ μ ” model concrete material, showed a better prediction of the global base shear response than the frames in the longitudinal direction, modelled using the Kent-Scott-Park model concrete material.

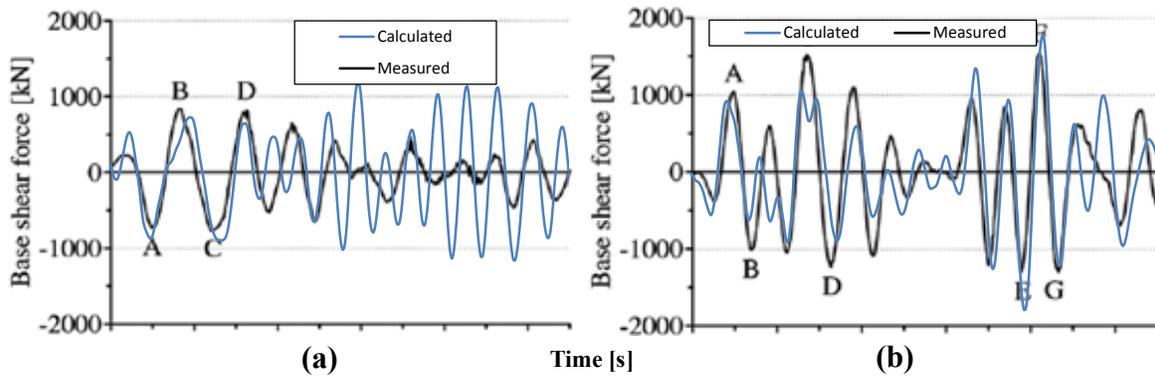


Fig. 5.26. RC building global base shear response for the 25% Kobe record in (a) longitudinal direction and (b) transverse direction

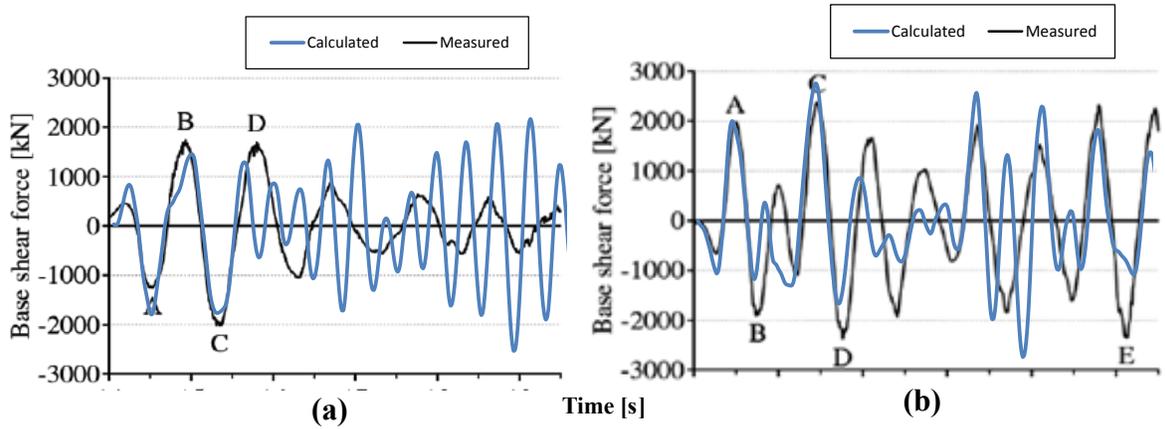


Fig. 5.27. RC building global base shear response for the 50% Kobe record in (a) longitudinal direction and (b) transverse direction

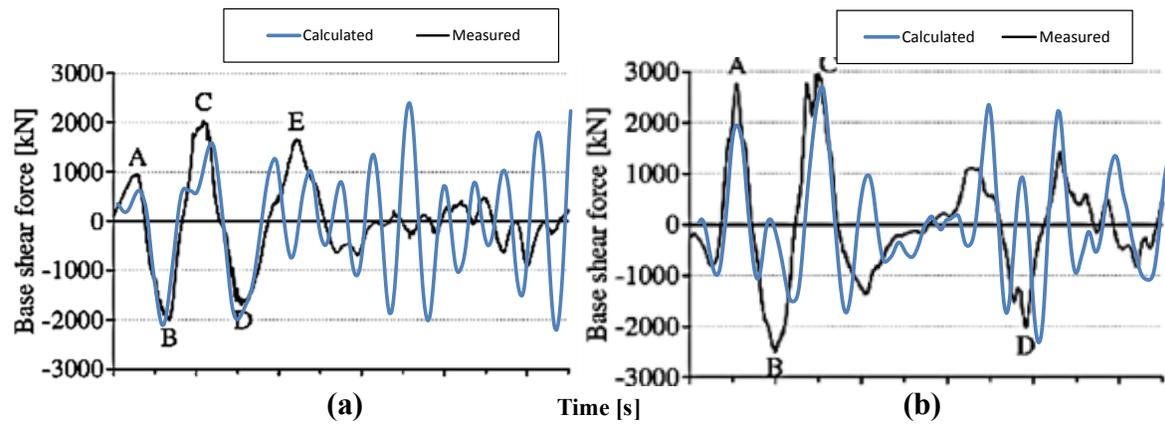


Fig. 5.28. RC building global base shear response for the 100% Kobe record in (a) longitudinal direction and (b) transverse direction

The measured and calculated maximum roof displacements are presented in Table 5.7. For the 25%, the 50%, and the 100% records, the roof displacements are underestimated in both directions. The difference percentage in the longitudinal direction is constant for the different Kobe magnitude records, whereas in the transverse direction the error % increases as the magnitude of the record increases. This can be explained by the fact that this model does not account for bond-slip at the bottom of the shear walls. Similar to the global base shear response, the calculated results had a higher stiffness than the measured results, which translates to less energy dissipation.

Table 5.7. Maximum roof displacements of RC building.

Magnitude of Kobe earthquake record (%)	Maximum roof displacement measured (mm)		Maximum roof displacement calculated (mm)		Difference (%)	
	Longitudinal direction	Transverse direction	Longitudinal direction	Transverse direction	Longitudinal direction	Transverse direction
25	16.9	24.2	8.99	15.97	46.8%	34%
50	122.4	106.9	65.27	62.75	46.7%	41.3%
100	242.7	323.9	129.23	125.22	46.8%	61.3%

The calculated results had greater stiffness than the measured results, as they had smaller deformations and higher strength responses. This can be explained in part by the inability of the model to account for bond-slip at the base of the walls, and to predict shear deformations for the 1D frame elements. A better prediction is observed for the base-shear in the wall direction (transverse), which means that the “ μ ” model is a reliable tool for predicting the behavior of 3D elements for full-scale structures. Despite the limitations discussed, the calculated results present a reasonable prediction for the 25%, 50%, and 100% intensity ground motion, resembling the actual behavior of the structure.

6 SUMMARY, CONCLUSIONS, AND FUTURE RESEARCH

6.1 Summary

The objective of studying the performance of simple biaxial concrete formulations capable of determining stiffness recovery in reversal loading (crack closing), permanent deformations, and moderate confinement at the element and system levels was achieved by the following:

1. The selection of the formulations most suited to describe biaxial concrete behavior under monotonic, cyclic, and dynamic loadings for both subassemblies and full-scale structures:
 - An assessment of different theoretical approaches to build nonlinear FEA models for RC structures was performed. The theoretical approaches consisted of: elasticity, plasticity, total-strain, and damage-continuous mechanics. The behavior of concrete under uniaxial and biaxial states of stress was presented.
 - Elasticity-based models include secant and tangential formulations. The commonly used nonlinear FEA models that are built using this approach are restricted to uniaxial models.
 - Plasticity-based models are able to account for concrete permanent deformations. However, the complexity of these models leads to convergence problems when the number of elements is large.
 - Total-strain based models are reliable and accurate at predicting RC behavior under plane loading. However, they require iterative procedures which may cause convergence problems in FE formulations.

- Isotropic-elastic-damage based models are a simplified strategy for describing the behavior of concrete that account for its complex microstructure. These models describe the inelastic behavior of concrete based only on its stiffness degradation and are simple enough to perform FEA of RC structures at the system level, while still being reasonably accurate.
 - Two isotropic-elastic-damage based models were selected: the PRM model (Pontiroli, Rouquand, & Mazars, 2010) and the “ μ ” model (Mazars, Hamon, & Grange, 2015). These two formulations were described and parametric analyses for the influence of their parameters were presented.
 - The PRM model was formulated from work done by Pontiroli (1995), Rouquand (2005), and Mazars (1986). It has the capacity to account for crack-closure effects under reversal-cyclic loading and permanent deformations.
 - The “ μ ” model was created to include the effects related to monotonic and cyclic loading that were not incorporated in previous models, such as low to moderate confinement and crack opening and closure.
2. The selected concrete models were implemented for their use in plane-stress shell elements in a freely available, open-source framework able to perform nonlinear FEA of complete structures and subassemblies under non-monotonic types of loading: OpenSees.
 3. The performance of the implemented models –the PRM model and the “ μ ” model– was studied by comparing five concrete experimental tests with varying complexity taken from the literature with analytical models built in OpenSees. The performance of a

biaxial model previously implemented in OpenSees by Garcia (2017), based on the scalar damage model (Mazars, 1986), was also studied.

- The first experiment consisted of plain concrete plates tested under biaxial states of stress (Kupfer, Hilsdorf, & Rüsç, 1969).
- The second experiment consisted of a simply-supported beam under monotonic loading tested as part of this project.
- The third experiment consisted of a simply-supported beam under reversal-cyclic loading (Ranjbaran, Rezayfar, & Mirzababai, 2018).
- The fourth experiment consisted of a rectangular shear wall under reversal-cyclic loading (Hiotakis, 2004).
- The fifth experiment consisted of a full-scale four-storey building under dynamic, seismic loading (Nagae, et al., 2015).

6.2 Conclusions

The results presented in this study capture the general response of RC structures subjected to monotonic, cyclic, and dynamic loading in terms of peak displacement and strength. The following conclusions were made:

1. The models currently implemented in FEA are generally unsuitable for analysis at the system level of complex RC structures subjected to dynamic loading without numerical-convergence problems and, if analysis is possible, their source-code is proprietary, preventing scrutiny of the material models implemented on them.
2. Analyses of structures with several degrees of complexity (from the element level to the system level) showed that the simple biaxial concrete material models implemented in

this study showed an acceptable ability to predict a number of structural parameters. Although the models failed to capture certain aspects of the response, the scope of the study was to show how simple models can have a robust performance (so that full structures can be analyzed and modelled) and lead to results with a notable level of accuracy. In particular, to the knowledge of the authors, the prediction of the response of a full-scale structure such as the one analyzed in Chapter 5 had not been attempted in any published, journal-like study.

3. This research provides a tool for seismic assessment, rehabilitation assessment, and performance-based design of RC buildings.
4. The biaxial behavior of concrete under uniaxial-compression, compression-tension, and biaxial-tension state of stresses –using the scalar damage (SD) model, the “ μ ” model, and the PRM model– showed a satisfactory agreement with available test data. While the PRM and scalar damage models do not accurately predict the response in the biaxial compression domain, the “ μ ” model had a superior performance, including the strength increase on the biaxial compression range due to confinement effects.
5. The calculated force-displacement response –using the scalar damage (SD) model, the “ μ ” model, and the PRM model– of a simple-supported RC beam under monotonic loading showed a satisfactory agreement with the measured results. The “ μ ” model had a superior performance at predicting the yielding and failure displacements.
6. The calculated load-displacement hysteretic response –using the scalar damage (SD) model, and the “ μ ” model– of a quad shear wall and a simple-supported beam under reversal-cyclic loading showed a moderate agreement with the measured results. Both models failed to capture the pinching effects. The “ μ ” model had a superior

performance at predicting the initial cracking load, the yielding load, and the energy dissipation.

7. The calculated load-displacement hysteretic response –using PRM model– of a quad shear wall and a simple-supported beam under reversal-cyclic loading showed the ability to capture the pinching effects of the measured results. However, the PRM model failed to perform the full time-history analysis of both experiments because of convergence problems, which can be attributed to the complexity of calculating permanent displacements in a multiaxial fashion.
8. The calculated global base-shear response and the roof displacements –using the “ μ ” model– of a full-scale four-storey RC showed a reasonable agreement with the measured results. A better prediction was observed for the walls direction, which were the only elements modelled using the implemented biaxial concrete model.
9. Isotropic-scalar-damage models provide a rational and appropriate approach to predict the behaviour of concrete plane-stress elements, as they use simple calculations that do not require complex solution algorithms within the element formulations.
10. The freely available, open-source code framework, OpenSees, is a useful tool for the implementation of material models that predict the behavior of concrete under monotonic, cyclic, and dynamic loading.

6.3 Recommendations & Future Research

The following are ideas for future studies based on this research:

1. The development or implementation of new algorithms to solve the FE system of equations of the OpenSees framework to improve the performance of the PRM model and to ensure better and more efficient analysis of structures at the system level.
2. The correlation of the parameters that control the compressive and tensile stress-strain relationship of the implemented models (A_c , B_c , A_t , B_t), and the damage thresholds for the compression and tension domains (ε_{D0c} , ε_{D0t}), with the standard parameters of concrete compression and tension tests (f'_c , ε_0 , f_t).
3. Make parametric analyses to study the influence that the shape of the descending branch of the stress-strain compressive response of the concrete models has on the analysis of subassemblies and full structures.
4. Make parametric analyses to evaluate the best way of modelling full-scale structures in OpenSees at the system level. Investigate the use of link elements, joint elements, and bearing elements. Investigate the influence of using different types of frame elements and uniaxial concrete elements. Investigate the influence of the transient analysis selected. Incorporate the bond-slip phenomenon for the shear walls.
5. Compare the analytical results of the full-scale building obtained using OpenSees with commercial FEA software, such as Abaqus and DIANA, by modelling the same four-storey building.
6. Contact OpenSees to evaluate the possibility of incorporating the implemented models to the OpenSees User Manual and the latest OpenSees revision, thus making them freely available for the research community.

REFERENCES

- Abaqus . (2009, May). *Abaqus Analysis User's Manual*.
<http://abaqusdoc.ucalgary.ca/v6.9/books/usb/default.htm?startat=pt05ch19s06abm38.html>. Retrieved from
<http://abaqusdoc.ucalgary.ca/v6.9/books/usb/default.htm?startat=pt05ch19s06abm38.html>
- Ahmed, A. (2014). Modelling of a reinforced concrete beam subjected to impact vibration using ABAQUS. *International Journal of Civil and Structural Engineering*, 4(3).
- AIJ. (1999). *Design Guidelines for Earthquake Resistant Reinforced Concrete Buildings Based on the Inelastic Displacement Concept*. Tokyo, Japan: Architectural Institute of Japan.
- Ali, A., Kim, D., & Cho, S. (2013, February). Modeling of nonlinear cyclic load behavior of I-shaped composite steel-concrete shear walls of nuclear power plants. *Nuclear Engineering and Technology*, 45(1), 89-98.
- Argyris, J., Faust, G., & William, K. (1981). Finite Element Modeling of Reinforced Concrete Structures. *Advanced Mechanics of Reinforced Concrete*. Delft: IABSE.
- ASCE. (1982). *Finite Element Analysis of Reinforced Concrete*. New York: American Society of Civil Engineers.
- Attard, M., & Mendis, P. (1993). Ductility of High Strength Concrete Columns. *Australian Civil Engineering Transactions*, 295-306.
- Bazant, Z. (1994). Nonlocal Damage Theory Based on Micromechanics of Crack Interaction. *J Eng Mech*, (120),593-617.
- Bazant, Z., & Tsubaki, T. (1980). Total Strain Theory and Path-Dependence of Concrete. *ASCE J Eng Mech Div*, 106(6),1151-1173.
- Chak, I. (2013). *Janus: A Post-Processor for VecTor Analysis Software*. Toronto: Master of Applied Science. Graduate Department of Civil Engineering, University of Toronto.

- Chen, W. F., & Han, D. J. (1988). *Plasticity for Structural Engineers*. New York: Springer-Verlag.
- Chen, W., Yamaguchi, E., Kotsovos, M., & Pan, A. (1993). Constitutive Models. In ASCE/ACI, *Finite Element Analysis of Reinforced Concrete Structures II* (pp. 36-117). New York: American Society of Civil Engineers.
- CSI. (2018). *Computers & Structures, Inc.* Retrieved from www.csiamerica.com
- Ebrahimian, H., Astroza, R., Conte, J. P., Restrepo, J. I., & Hutchinson, T. C. (2014). Experimental Validation of Dynamic Nonlinear FE Model of Full-Scale Five-Storey Reinforced Concrete Building. *9th International Conference on Structural Dynamics*. Porto: EUROLYN.
- Fenves, G. (2001). *Annual Workshop on Open System for Earthquake Engineering Simulation*. Retrieved from <http://opensees.berkeley.edu/>
- Filippou, F. C., Popov, E. P., & Bertero, V. V. (1983). Effects of Bond Deterioration on Hysteretic Behavior of Reinforced Concrete Joints. *Engineering Research Center, Univeristy of California*, Report EERC 83-19.
- Fu, F. (2009). Progressive Collapse Analysis of High-Rise Building with 3-D Finite Element Modeling Method. *Journal of Constructional Steel Research*, (6), 1269-1278.
- Garcia, B. (2017). *Implementation of a Simple Biaxial Concrete Model for Reinforced Concrete Membranes*. Edmonton: Department of Civil and Environmental Engineering, University of Alberta.
- Gerstle, K. H. (1980). Behavior of Concrete under Multiaxial Stress States. *Journal of Engineering Mechanics, ASCE*, 106(6),1383-1403.
- Han, D., & Chen, W. (1985). A Nonuniform Hardening Plasticity Model for Concrete Materials. *Mechanics of Materials*, (4),283-302.

- Hiotakis, S. (2004). *Repair and Strengthening of Reinforced Concrete Shear Walls for Earthquake Resistance Using Externally Bonded Carbon Fibre Sheets and a Novel Anchor System*. Ottawa: Master of Applied Science. Department of Civil and Environmental Engineering, University of Carleton.
- Hognestad, E. (1951). A study on combined bending and axial load in reinforced concrete members. *University of Illinois Engineering Experiment Station*, 43-46.
- Hognestad, E. (1955). Concrete Stress Distribution in Ultimate Strength Design. *J. ACI*, 52(12),455-480.
- Hsu, T., & Mo, Y. (2010). *Unified Theory of Concrete Structures*. Houston: John Wiley and Sons Ltd.
- Hsu, T., & Zhu, R. (2002). Softened Membrane Model for Reinforced Concrete Elements in Shear. *Structural Journal of the American Concrete Institute*, 99(4), 460-469.
- Hsu, T., Slate, F., Sturman, G., & Winter, G. (1963). Microcracking of Plain Concrete and the Shape of the Stress-Strain Curve. *Journal of the American Concrete Institute*, 60(2),209-224.
- Hussein, H., & Marzouk, H. (2000). Behavior of High-Strength Concrete under Biaxial Stresses. *ACI Materials Journal*, 97(M4).
- Imran, I., & Pantazopoulou, J. (1996). Experimental Study of Plain Concrete under Triaxial Stress. *ACI Materials Journal*, 93(M67).
- Kachanov, L. (1958). Time of the rupture process under creep conditions. *Izv. Akad. Nauk SSR Otd.*, (8),26-31.
- Kaklauskas, G. (1999). A New Stress-Strain Relationship for Cracked Tensile Concrete in Flexure. *STATYBA-Civil Engineering*, V(6),1392-1525.
- Karsan, P., & Jirsa, J. (1969, December). Behavior of Concrete under Compressive Loading. *Journal of Structural Division, ASCE*, 95(ST12), 2543-2563.

- Kent, D., & Park, R. (1971). Flexural members with confined concrete. *Journal of the Structural Division, ASCE*, 97(ST7), 1969-1990.
- Kupfer, H., Hilsdorf, H., & Rüschi, H. (1969). Behavior of Concrete Under Biaxial Stresses. *ACI J*, 66(66-62),656-666.
- Lee, H., & Kuchma, D. (2007). Seismic overstrength of shear walls in parking structures with flexible diaphragms. *Journal of Earthquake Engineering*, 11(1), 86-109. doi:10.1080/13632460601033488
- Lee, J., & Fenves, G. (1998). Plastic-Damage Model for Cyclic Loading of Concrete Structures. *J Eng Mech*, (124),892.
- Lemaitre, J. (1986). Local Approach of Fracture. *Engineering Fracture Mechanics*, 25(5/6),523-537.
- Liu, T. C., Nilson, A. H., & Slate, F. O. (1972). Stress-Strain Response and Fracture of Concrete in Uniaxial and Biaxial Compression. *Journal of the American Concrete Institute*, 69(5),291-295.
- Lu, X., Xie, L., Guan, H., & Lu, X. (2015, June). A Shear Wall Element for Nonlinear Seismic Analysis of Super-Tall Buildings Using OpenSees. *Finite Element in Analysis and Design*, 98, 14-25.
- Maekawa, K., Takemura, J., Irawan, P., & Irie, M. (1993). Triaxial Elasto-Plastic and Fracture Model for Concrete. *JSCE*, (18),131-138.
- Mander, J., Priestly, M., & Park, R. (1988). Theoretical Stress-strain Model of Confined Concrete. *Journal of Structural Engineering*, 114(8), 1804-1826.
- Mansour, M., & Hsu, T. (2005). Behavior of Reinforced Concrete Elements under Cyclic Shear: Part 2 - Theoretical Model. *Journal of Structural Engineering, ASCE*, 131(1),54-65.
- Mazars, J. (1986). A Description of Micro and Macroscale Damage of Concrete Structures. *Eng Fract Mech*, (25),729-737.

- Mazars, J., & Grange, S. (2014). Simplified Method Strategies Based on Damage Mechanics for Engineering Issues. *EUROC*, St Anton am Arlberg, Austria.
- Mazars, J., Hamon, F., & Grange, S. (2015). A New 3D Damage Model for Concrete. *Materials and Structures*, 48,3779-3793.
- Mazzoni, S., McKenna, F., Michael, H. S., & Fenves, G. L. (2006). *OpenSees Command Language Manual*. OpenSees.
- Menegotto, M., & Pinto, P. (1973). Method of Analysis for Cyclically Loaded Reinforced Concrete Plane Frames Including Changes in Geometry and Nonelastic Behavior of Elements under Combined Normal Forced and Bending. *IABSE Symposium on Resistance and Ultimate Deformability of Structures Acted on by Well-Defined Repetead Loads*. Lisbon, Portugal.
- Mergos, P., & Beyer, K. (2013). Modelling shear-flexure interaction in equivalent frame models of slender reinforced concrete walls. In *Struct. Design Tall Spec. Build.* Wiley Library. doi:10.1002/TAL.1114
- Mostafei, H., Veccio, F. J., Gauvreau, P., & Semelawy, M. (2011). Punching Shear Behavior of Externally Prestressed Concrete Slabs. *Engineering Structures*, 33(5),1723-1733.
- Nagae, T., Ghannoum, W., Kwon, J., Tahara, K., Fukuyama, K., Matsumori, T., & Shiohara, H. (2015). Design Implications of Large-Scale Shake-Table Test on Four-Storey Reinforced Concrete Building. *ACI Structural Journal*, 112(S12),135-146.
- Nelissen, L. (1972). Biaxial Testing of Normal Concrete. *Heron, the Netherlands*, 18(1).
- Ortiz, M., & Popov, E. P. (1982). Plain Concrete as a Composite Material. *Mechanics of Materials*, 1(2),139-150.
- Pang, B., & Hsu, T. (1996). Fixed Angle Softened Truss Model for Reinforced Concrete. *ACI Structural Journal*, 93(2), 197-207.

- Pantoli, E., Wang, X., Chen, M., Hutchinson, T., Meacham, B., & Park, H. J. (2013). Shake Table Testing of a Full-Scale Five-Storey Building: Performance of the Major Nonstructural Components. *ASCE, Structures Congress 2013: Bridging Your Passion with Your Profession* (pp. 1447-1459). American Society of Civil Engineers.
- Pontioli, C. (1995). *Comportement au souffle des structures en béton armé, analyse expérimentale et modélisation*. France: PhD thesis ENS Cachan.
- Pontioli, C., Rouquand, A., & Mazars, J. (2010). Predicting Concrete Behaviour From Quasi-static Loading to Hypervelocity Impact. *Eur J. Environ Civil Eng*, 14(6-7),703-727.
- Puurula, A., Enochsson, O., Sas, G., & Elfgren, L. (2015). Assesment of the Strengthening of an RC Railway Bridge with CFRP Utilizing a Full-Scale Failure Test and Finite Element Analysis. *Journal of Structural Engineering*, 141(1).
- Ranjbaran, F., Rezayfar, O., & Mirzababai, R. (2018). Experimental Investigation of Steel Fiber-Reinforced Concrete Beams Under Cyclic Loading. *Int J Adv Struc Eng*, 10-49.
- Raphael, J. (1984). Tensile Strength of Concrete. *ACI Journal*, 81(17).
- Rouquand, A. (2005). *Presentation d'un modèle de comportement des géomatériaux, applications au calcul de structures*. C.E.G., report T2005-.
- Saatci, S., & Vecchio, F. J. (2009). Effects of Shear Mechanics on the Impact Behavior of RC Beams. *ACI Structural J*, 106(1),78-86.
- Schlaich, J., Schäfer, K., & Jennewein, M. (1987). Toward a Consistent Design of Structural Concrete. *PCI J.*, 32(3), 74-150.
- Schreppers, G. (2017). *Validation Report Maekawa-Fukuura Model and Cracked Concrete Curves in Total Train Crack Model in Diana*. Netherlands: DIANA FEA.

- Scott, M. H., & Fenves, G. L. (2010). A Krylov Subspace Accelerated Newton Algorithm: Application to Dynamic Progressive Collapse Simulation of Frames. *Journal of Structural Engineering*, 136(5).
- Selby, R., & Vecchio, F. (1997). A Constitutive Model for Analysis of Reinforced Concrete Solids. *Canadian Journal of Civil Engineering*, 24(3),460-470.
- Sinha, B., Gerstle, K., & Tulin, L. (1964). Stress-Strain Relations for Concrete Under Cyclic Loading. *Journal of the American Concrete Institute*, 61(2),195-211.
- Tasuji, M., Slate, F., & Nilson, A. (1978). Stress-Strain Relationship and Fracture of Concrete in Biaxial Loading. *Journal of the American Concrete Institute*, 75(7),306-312.
- TNO. (2016). *DIANA FEA*. Retrieved from dianafea.com
- Vecchio, F., & Collins, M. (1982). *Response of Reinforced Concrete to In-Plane Shear and Normal Stresses*. Toronto, Canada: University of Toronto.
- Vecchio, F., & Collins, M. (1986). The Modified Compression-Field Theory for Reinforced Concrete Elements Subjected to Shear. *ACI Structural Journal*, (83),219-231.
- Vecchio, F., & Palermo, D. (2001). NLFEARC: Look Both Ways Before Crossing. In ACI, *Finite Element Analysis of Reinforced Concrete Structures* (pp. 1-15). Farmington Hills: Bonnie L. Gold.
- Zhong, J. (2005). *Model-Based Simulation of Reinforced Concrete Plane Stress Structures*. Houston: PhD Dissertation. Department of Civil and Environmental Engineering, University of Houston.

APPENDIX A: PRM SOURCE CODE

A.1. Header File (PRM.h)

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include <Vector.h>
#include <Matrix.h>
#include <ID.h>
#include <NDMaterial.h>

class PRM: public NDMaterial{
public :
    PRM( ) ;
    PRM(int tag, double _Ec, double _epsD0c, double _Ac, double _Bc, double
_epsD0t, double _At, double _Bt, double _epsfc, double _sigfc, double _epsft0,
double _sigft0, double _nu) ;

    virtual ~PRM( ) ;

    void setInitials( ) ;

    //make a clone of this material
    NDMaterial *getCopy( ) ;
    NDMaterial *getCopy( const char *type ) ;

    //send back order of strain in vector form
    int getOrder( ) const ;

    //send back order of strain in vector form
    const char *getType( ) const ;

    //swap historey variables
    int commitState( ) ;

    //revert to last saved state
    int revertToLastCommit( ) ;

    //revert to start
    int revertToStart( ) ;

    //get the strain
    int setTrialStrain( const Vector &strainFromElement ) ;

    //send back the strain
    const Vector& getStrain( ) ;

    //send back the stress
    const Vector& getStress( ) ;

    //send back the tangent
```

```

const Matrix& getTangent( ) ;

const Matrix& getInitialTangent( ) ;

//print out data
void Print( OPS_Stream &s, int flag ) ;

int sendSelf(int commitTag, Channel &theChannel);
int recvSelf(int commitTag, Channel &theChannel, FEM_ObjectBroker &theBroker);

private :
Matrix tStrain, cStrain, tStress, cStress, tTangent, cTangent, iniTangent;
Vector strain, stress, tDamCdist, cDamCdist;
double Ec, epsD0c, Ac, Bc, epsD0t, At, Bt, epsfc, sigfc, epsft0, sigft0, nu;
double tDamT, cDamT, tDamC, cDamC, tDam, cDam, tepsmin, cepsmin;

};

```

A.2. C++ File (PRM.cpp)

```
#include <elementAPI.h>
#include <PRM.h>
#include <Channel.h>
#include <FEM_ObjectBroker.h>
#include <MaterialResponse.h>

#include <OPS_Globals.h>
#include <Information.h>
#include <Parameter.h>
#include <string.h>
#include <Vector.h>
#include <math.h>
#include <float.h>
#include <Matrix.h>

#include <iostream>

void *
OPS_PRM(void)
{
    NDMaterial *theMaterial = 0;

    int numArgs = OPS_GetNumRemainingInputArgs();

    if (numArgs != 13) {
        opserr << "Want: nDMaterial PRM tag? Ec? epsD0c? Ac? Bc? epsD0t? At? Bt? epsfc?
sigfc? epsft0? sigft0? nu?" << endl;
        return 0;
    }

    int iData[1];
    double dData[12];

    int numData = 1;
    if (OPS_GetInt(&numData, iData) != 0) {
        opserr << "WARNING invalid integer tag: nDMaterial PRM \n";
        return 0;
    }

    numData = 12;

    if (OPS_GetDouble(&numData, dData) != 0) {
        opserr << "WARNING invalid data: nDMaterial PRM : " << iData[0] << "\n";
        return 0;
    }

    theMaterial = new PRM(iData[0], dData[0], dData[1], dData[2], dData[3], dData[4],
dData[5], dData[6], dData[7], dData[8], dData[9], dData[10], dData[11]);

    return theMaterial;
}
```

```

//null constructor
PRM::PRM( ) :
NDMaterial(0, ND_TAG_PRM ),
tStrain(3, 3), cStrain(3, 3), tStress(3, 3), cStress(3, 3), tTangent(6,6),
cTangent(6,6), strain(3), stress(3), iniTangent(6, 6),
tDamCdist(3)
{ }

//full constructor
PRM::PRM(int tag, double _Ec, double _epsD0c, double _Ac, double _Bc, double
_epsD0t, double _At, double _Bt, double _epsfc, double _sigfc, double _epsft0,
double _sigft0, double _nu) :
NDMaterial( tag, ND_TAG_PRM ),
tStrain(3, 3), cStrain(3, 3), tStress(3, 3), cStress(3, 3), tTangent(6, 6),
cTangent(6, 6), strain(3), stress(3), iniTangent(6, 6), tDamCdist(3),
Ec(_Ec), epsD0c(_epsD0c), Ac(_Ac), Bc(_Bc), epsD0t(_epsD0t), At(_At), Bt(_Bt),
epsfc(_epsfc), sigfc(_sigfc), epsft0(_epsft0), sigft0(_sigft0), nu(_nu)
{

    setInitials();
}

//destructor
PRM::~~PRM( )
{

}

void PRM::setInitials()
{

    if (Ec < 0.0)
        Ec = -Ec;

    if (epsD0c < 0.0)
        epsD0c = -epsD0c;

    if (Ac < 0.0)
        Ac = -Ac;

    if (Bc < 0.0)
        Bc = -Bc;

    if (epsD0t < 0.0)
        epsD0t = -epsD0t;

    if (At < 0.0)
        At = -At;

    if (Bt < 0.0)
        Bt = -Bt;

    if (epsfc < 0.0)
        epsfc = -epsfc;
}

```

```

    if (sigfc < 0.0)
        sigfc = -sigfc;

    if (epsft0 > 0.0)
        epsft0 = -epsft0;

    if (sigft0 > 0.0)
        sigft0 = -sigft0;

    if (nu < 0.0)
        nu = -nu;

    tStrain.Zero();
    cStrain = tStrain;
    strain.Zero();

    tStress.Zero();
    cStress = tStress;
    stress.Zero();

    iniTangent.Zero();
    iniTangent(0, 0) = iniTangent(1, 1) = iniTangent(2, 2) = 1.0 - nu;
    iniTangent(0, 1) = iniTangent(0, 2) = iniTangent(1, 0) = iniTangent(1, 2) =
iniTangent(2, 0) = iniTangent(2, 1) = nu;
    iniTangent(3, 3) = iniTangent(4, 4) = iniTangent(5, 5) = (1.0 - 2.0 * nu) /
2.0;
    iniTangent *= Ec / ((1.0 + nu) * (1.0 - 2.0 * nu));

    tTangent = iniTangent;
    cTangent = tTangent;

    tDam = 0.0;
    cDam = tDam;
    tDamT = 0.0;
    cDamT = tDamT;
    tDamC = 0.0;
    cDamC = tDamC;
    tDamCdist.Zero();
    cDamCdist = tDamCdist;

    tepsmin = 0.0;
    cepsmin = tepsmin;
}

//make a clone of this material
NDMaterial*
PRM::getCopy( )
{
    PRM *clone ; //new instance of this class

    clone = new PRM( this->getTag(), Ec, epsD0c, Ac, Bc, epsD0t, At, Bt, epsfc, sigfc,
epsft0, sigft0, nu);

```

```

    return clone ;
}

//make a clone of this material
NDMaterial*
PRM::getCopy( const char *type )
{
    return this->getCopy( ) ;
}

//send back order of strain in vector form
int
PRM::getOrder( ) const
{
    return 3 ;
}

const char*
PRM::getType( ) const
{
    return PRM ;
}

//swap historey variables
int
PRM::commitState( )
{
    cStrain = tStrain;
    cStress = tStress;
    cTangent = tTangent;
    cDam = tDam;
    cDamT = tDamT;
    cDamC = tDamC;
    cDamCdist = tDamCdist;
    cepsmin = tepsmin;

    return 0;
}

//revert to last saved state
int
PRM::revertToLastCommit( )
{
    return 0;
}

```

```

//revert to start
int
PRM::revertToStart( )
{
    setInitials();

    return 0;
}

//receive the strain
int
PRM::setTrialStrain( const Vector &strainFromElement )
{
    double psi, epseq2, epseq, epseq2M, epseqM, alpha, Trsigd;
    double dtM, dcM, DamT, DamC;
    static Vector sigpos(3), signeg(3), epsft(3), sigft(3), epsp(6), epsd(6),
sigp(6), sigd(6) ;
    static Matrix T(3, 3), Ttrans(3, 3);

    tStrain(0,0) = strainFromElement(0) ;
    tStrain(1,1) = strainFromElement(1) ;
    tStrain(0,1) = strainFromElement(2) ;
    tStrain(1,0) = strainFromElement(2) ;
    tStrain(2, 2) = (-nu) * (tStrain(0, 0) + tStrain(1, 1)) / (1.0 - nu);

    /* ---- Retrieve state variables ---- */
    tDam = cDam;
    tDamT = cDamT;
    tDamC = cDamC;
    tepsmin = cepsmin;
    tTangent = cTangent;

    /* ---- Calculate Principal direction angle for the Local Coordinates ---- */
    psi = (atan(2.0 * tStrain(0, 1) / (tStrain(0, 0) - tStrain(1, 1) + 1.0e-20)))
/ 2.0;
    T.Zero();
    Ttrans.Zero();
    T(0, 0) = T(1, 1) = Ttrans(0, 0) = Ttrans(1, 1) = cos(psi);
    T(0, 1) = Ttrans(1, 0) = sin(psi);
    T(1, 0) = Ttrans(0, 1) = -sin(psi);
    T(2, 2) = Ttrans(2, 2) = 1.0;

    /* ---- Obtain Principal strains ---- */
    static Matrix TransfAid(3, 3);
    TransfAid = T * tStrain * Ttrans;
    epsp.Zero();
    epsp(0) = TransfAid(0, 0);
    epsp(1) = TransfAid(1, 1);
    epsp(2) = TransfAid(2, 2);

    TransfAid.Zero();
    TransfAid(0, 0) = cDamCdist(0);
    TransfAid(1, 1) = cDamCdist(1);

```

```

TransfAid(0, 1) = TransfAid(1, 0) = cDamCdist(2);
TransfAid = T * TransfAid * Ttrans;
tDamCdist(0) = TransfAid(0, 0);
tDamCdist(1) = TransfAid(1, 1);
tDamCdist(2) = TransfAid(0, 1);

/* ---- Permanent strains and stresses----*/
epsft.Zero();
sigft.Zero();
for (int i = 0; i <= 1; i++)
{
    sigft(i) = sigft0;
    epsft(i) = (epsft0 - epsfc * tDamCdist(i)) / (1.0 - tDamCdist(i));
}

static Matrix Dist(3, 3);
Dist.Zero();
Dist(0, 0) = Dist(1, 1) = Dist(2, 2) = 1.0;
Dist(0, 1) = Dist(1, 0) = Dist(0, 2) = Dist(1, 2) = Dist(2, 0) = Dist(2, 1) =
-nu;
epsft = Dist * epsft;
Dist.Zero();
Dist(0, 0) = Dist(1, 1) = 1.0;

sigft = Dist * sigft;

/* ---- Distribution of permanent strains and stresses in the element.*/
epsd = epsp;
for (int i = 0; i <= 2; i++)
{
    epsd(i) -= epsft(i);
}

/* ---- Equivalent strain for Compression (Mazars) and Tension (PRM) ---- */
epseq2 = 0.0;
epseq2M = 0.0;
for (int i = 0; i <= 2; i++)
{
    if (epsd(i) > 0)
        epseq2 += epsd(i) * epsd(i);
    if (epsp(i) > 0)
        epseq2M += epsp(i) * epsp(i);
}
epseq = sqrt(epseq2);
epseqM = sqrt(epseq2M);

/* ---- Minimum strain ---- */
for (int i = 0; i <= 2; i++)
{
    if (epsp(i) < tepsgmin)
        tepsgmin = epsp(i);
}

```

```

/* ---- Alpha factor calculation from Rouquand and Pontiroli ---- */
sigd = tTangent * epsd;
sigp = sigd;
for (int i = 0; i <= 2; i++)
{
    sigp(i) += sigft(i);
}
Trsigd = sigd(0) + sigd(1) + sigd(2);

if (Trsigd >= 0.0)
    alpha = 1.0;
else
{
    for (int i = 0; i <= 2; i++)
    {
        if (sigp(i) > 0)
        {
            sigpos(i) = sigp(i);
            signeg(i) = 0.0;
        }
        else
        {
            sigpos(i) = 0.0;
            signeg(i) = sigp(i);
        }
    }
    alpha = abs((sigpos(0) + sigpos(1) + sigpos(2)) / (signeg(0) +
signeg(1) + signeg(2) + 1e-25));
}

if (alpha < 0.0)
    alpha = 0.0;
if (alpha > 1.0)
    alpha = 1.0;

/* ---- Calculation of Damage scalar parameters (Dt and Dc) ---- */
if (tDamT < 1.0)
{
    if (epseq > epsD0t)
        dtM = 1.0 - (epsD0t * (1.0 - At) / epseq) - At * exp(-Bt *
(epseq - epsD0t));
    else
        dtM = 0.0;

    DamT = dtM;
    if (DamT < 0.0)
        DamT = 0.0;
    if (DamT > 1.0)
        DamT = 1.0;

    if (DamT < cDamT)
        DamT = cDamT;
}
else
    DamT = 1.0;

```

```

tDamT = alpha * DamT;

if (tDamC < 1.0)
{
    if (epseqM > epsD0c)
        dcM = 1.0 - (epsD0c * (1.0 - Ac) / epseqM) - Ac * exp(-Bc *
(epseqM - epsD0c));
    else
        dcM = 0.0;

    DamC = dcM / (1.0 - epsfc / tepsmin);

    if (DamC < 0.0)
        DamC = 0.0;
    if (DamC > 1.0)
        DamC = 1.0;

    if (DamC < cDamC)
        DamC = cDamC;
}
else
    DamC = 1.0;

tDamC = (1.0 - alpha) * DamC;

tDam = tDamT + tDamC;
if (tDam < 0.0)
    tDam = 0.0;
if (tDam > 1.0)
    tDam = 1.0;

tTangent = (1.0 - tDam) * iniTangent;

/* ---- Principal stresses ----*/
sigd = tTangent * epsd;
sigp = sigd;
for (int i = 0; i <= 2; i++)
{
    sigp(i) += sigft(i);
}

if (tDamT < cDamT)
    tDamT = cDamT;
if (tDamC < cDamC)
    tDamC = cDamC;

static Vector DamCdist(3);
DamCdist.Zero();
for (int i = 0; i <= 1; i++)
{
    DamCdist(i) = tDamC * (epsp(i) / tepsmin);
    if (DamCdist(i) > tDamCdist(i))

```

```

        tDamCdist(i) = DamCdist(i);
    }

    /* ---- Obtain state variables in Global coordinates ---- */
    T.Zero();
    Ttrans.Zero();
    T(0, 0) = T(1, 1) = Ttrans(0, 0) = Ttrans(1, 1) = cos(-psi);
    T(0, 1) = Ttrans(1, 0) = sin(-psi);
    T(1, 0) = Ttrans(0, 1) = -sin(-psi);
    T(2, 2) = Ttrans(2, 2) = 1.0;

    TransfAid.Zero();
    TransfAid(0, 0) = sigp(0);
    TransfAid(1, 1) = sigp(1);
    TransfAid(2, 2) = sigp(2);
    tStress = T * TransfAid * Ttrans;

    TransfAid.Zero();
    TransfAid(0, 0) = tDamCdist(0);
    TransfAid(1, 1) = tDamCdist(1);
    TransfAid(0, 1) = TransfAid(1, 0) = tDamCdist(2);
    TransfAid = T * TransfAid * Ttrans;
    tDamCdist(0) = TransfAid(0, 0);
    tDamCdist(1) = TransfAid(1, 1);
    tDamCdist(2) = TransfAid(0, 1);

    strain(0) = tStrain(0,0);
    strain(1) = tStrain(1,1);
    strain(2) = tStrain(0,1);

    stress(0) = tStress(0,0);
    stress(1) = tStress(1,1);
    stress(2) = tStress(0,1);

    return 0;
}

//send back the strain
const Vector&
PRM::getStrain( )
{
    return strain ;
}

//send back the stress
const Vector&
PRM::getStress( )
{
    return stress ;
}

//send back the tangent

```

```

const Matrix&
PRM::getTangent( )
{
    static Matrix theTangent(3, 3);
    theTangent.Zero();
    theTangent(0, 0) = iniTangent(0, 0);
    theTangent(0, 1) = iniTangent(0, 1);
    theTangent(1, 0) = iniTangent(1, 0);
    theTangent(1, 1) = iniTangent(1, 1);
    theTangent(2, 2) = iniTangent(3, 3);

    theTangent *= (1 - cDam);

    return theTangent ;
}

const Matrix&
PRM::getInitialTangent
( )
{
    static Matrix theTangent(3, 3);
    theTangent.Zero();
    theTangent(0, 0) = iniTangent(0, 0);
    theTangent(0, 1) = iniTangent(0, 1);
    theTangent(1, 0) = iniTangent(1, 0);
    theTangent(1, 1) = iniTangent(1, 1);
    theTangent(2, 2) = iniTangent(3, 3);

    return theTangent;
}

//print out data
void
PRM::Print( OPS_Stream &s, int flag )
{
    s << "PRM Material tag: " << this->getTag() << endln ;
    s << " Ec: " << Ec << " ";
    s << " epsD0c: " << epsD0c << " ";
    s << " Ac: " << Ac << " ";
    s << " Bc: " << Bc << " ";
    s << " epsD0t: " << epsD0t << " ";
    s << " At: " << At << " ";
    s << " Bt: " << Bt << " ";
    s << " epsfc: " << epsfc << " ";
    s << " sigfc: " << sigfc << " ";
    s << " epsft0: " << epsft0 << " ";
    s << " sigft0: " << sigft0 << " ";
    s << " nu: " << nu << " ";
}

int

```

```

PRM::sendSelf(int commitTag, Channel &theChannel)
{
    int res = 0, cnt = 0;

    static Vector data(25);

    data(cnt++) = this->getTag();
    data(cnt++) = Ec;
    data(cnt++) = epsD0c;
    data(cnt++) = Ac;
    data(cnt++) = Bc;
    data(cnt++) = epsD0t;
    data(cnt++) = At;
    data(cnt++) = Bt;
    data(cnt++) = epsfc;
    data(cnt++) = sigfc;
    data(cnt++) = epsft0;
    data(cnt++) = sigft0;
    data(cnt++) = nu;
    data(cnt++) = cStrain(0,0);
    data(cnt++) = cStrain(0,1);
    data(cnt++) = cStrain(0,2);
    data(cnt++) = cStrain(1,1);
    data(cnt++) = cStrain(1,2);
    data(cnt++) = cStrain(2,2);
    data(cnt++) = cStress(0,0);
    data(cnt++) = cStress(0,1);
    data(cnt++) = cStress(0,2);
    data(cnt++) = cStress(1,1);
    data(cnt++) = cStress(1,2);
    data(cnt++) = cStress(2,2);

    res = theChannel.sendVector(this->getDbTag(), commitTag, data);
    if (res < 0)
        opserr << "PRM::sendSelf() - failed to send data" << endl;

    return res;
}

int
PRM::recvSelf(int commitTag, Channel &theChannel, FEM_ObjectBroker &theBroker)
{
    int res = 0, cnt = 0;

    static Vector data(25);

    res = theChannel.recvVector(this->getDbTag(), commitTag, data);
    if (res < 0) {
        opserr << "PRM::recvSelf -- could not recv Vector" << endl;
        return res;
    }

    this->setTag(int(data(cnt++)));
    Ec = data(cnt++);
    epsD0c = data(cnt++);
    Ac = data(cnt++);
    Bc = data(cnt++);

```

```
epsD0t = data(cnt++);
At = data(cnt++);
Bt = data(cnt++);
epsfc = data(cnt++);
sigfc = data(cnt++);
epsft0 = data(cnt++);
sigft0 = data(cnt++);
nu = data(cnt++);
cStrain(0,0) = data(cnt++);
cStrain(0,1) = data(cnt++);
cStrain(0,2) = data(cnt++);
cStrain(1,1) = data(cnt++);
cStrain(1,2) = data(cnt++);
cStrain(2,2) = data(cnt++);
cStress(0,0) = data(cnt++);
cStress(0,1) = data(cnt++);
cStress(0,2) = data(cnt++);
cStress(1,1) = data(cnt++);
cStress(1,2) = data(cnt++);
cStress(2,2) = data(cnt++);

setInitials();

return res;
}
```

APPENDIX B: “MUMAZARS” (μ) SOURCE CODE

B.1. Header File (MuMazars.h)

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include <Vector.h>
#include <Matrix.h>
#include <ID.h>
#include <NDMaterial.h>

class MuMazars: public NDMaterial{
public :
    MuMazars( ) ;
    MuMazars(int tag, double _Ec, double _epsD0c, double _epsD0t, double _Ac, double
_Bc, double _At, double _Bt, double _nu) ;

    virtual ~MuMazars( ) ;

    void setInitials( ) ;

    //make a clone of this material
    NDMaterial *getCopy( ) ;
    NDMaterial *getCopy( const char *type ) ;

    //send back order of strain in vector form
    int getOrder( ) const ;

    //send back order of strain in vector form
    const char *getType( ) const ;

    //swap historey variables
    int commitState( ) ;

    //revert to last saved state
    int revertToLastCommit( ) ;

    //revert to start
    int revertToStart( ) ;

    //get the strain
    int setTrialStrain( const Vector &strainFromElement ) ;

    //send back the strain
    const Vector& getStrain( ) ;

    //send back the stress
    const Vector& getStress( ) ;

    //send back the tangent
    const Matrix& getTangent( ) ;
```

```

const Matrix& getInitialTangent( ) ;

//print out data
void Print( OPS_Stream &s, int flag ) ;

int sendSelf(int commitTag, Channel &theChannel);
int recvSelf(int commitTag, Channel &theChannel, FEM_ObjectBroker &theBroker);

private :
Matrix iniTangent, tTangent, cTangent, iniSmallTangent, tangent;
Matrix iniTangentInv, tTangentInv, cTangentInv;
Vector tStrain, cStrain, strain, tStress, cStress, stress;
double nu, Ec, epsD0c, epsD0t, Ac, Bc, At, Bt;
double fac, tDam, cDam, tepseqC, cepseqC, tepseqT, cepseqT, tYt, cYt, tYc,
cYc;
};

```

B.2. C++ File (MuMazars.cpp)

```
#include <elementAPI.h>
#include <MuMazars.h>
#include <Channel.h>
#include <FEM_ObjectBroker.h>
#include <MaterialResponse.h>

#include <OPS_Globals.h>
#include <Information.h>
#include <Parameter.h>
#include <string.h>
#include <Vector.h>
#include <math.h>
#include <float.h>
#include <Matrix.h>
#include <algorithm>
#include <minmax.h>

#include <iostream>

void *
OPS_MuMazars(void)
{
    NDMaterial *theMaterial = 0;

    int numArgs = OPS_GetNumRemainingInputArgs();

    if (numArgs != 9) {
        opserr << "Want: nDMaterial MuMazars tag? Ec? epsD0c? epsD0t? Ac? Bc? At? Bt?
nu?" << endl;
        return 0;
    }

    int iData[1];
    double dData[8];

    int numData = 1;
    if (OPS_GetInt(&numData, iData) != 0) {
        opserr << "WARNING invalid integer tag: nDMaterial MuMazars \n";
        return 0;
    }

    numData = 8;

    if (OPS_GetDouble(&numData, dData) != 0) {
        opserr << "WARNING invalid data: nDMaterial MuMazars : " << iData[0] << "\n";
        return 0;
    }

    theMaterial = new MuMazars(iData[0], dData[0], dData[1], dData[2], dData[3],
dData[4], dData[5], dData[6], dData[7]);

    return theMaterial;
}
```

```

}

//null constructor
MuMazars::MuMazars( ) :
NDMaterial(0, ND_TAG_MuMazars ),
tStrain(6), cStrain(6), tStress(6), cStress(6), strain(3), stress(3),
tTangent(6,6), cTangent(6,6), tangent(3,3), tTangentInv(6,6), cTangentInv(6,6),
iniTangent(6,6), iniSmallTangent(3,3), iniTangentInv(6,6)
{ }

//full constructor
MuMazars::MuMazars(int tag, double _Ec, double _epsD0c, double _epsD0t, double _Ac,
double _Bc, double _At, double _Bt, double _nu) :
NDMaterial( tag, ND_TAG_MuMazars ),
tStrain(6), cStrain(6), tStress(6), cStress(6), strain(3), stress(3),
tTangent(6,6), cTangent(6,6), tangent(3,3), tTangentInv(6,6), cTangentInv(6,6),
iniTangent(6,6), iniSmallTangent(3,3), iniTangentInv(6,6),
Ec(_Ec), epsD0c(_epsD0c), epsD0t(_epsD0t), Ac(_Ac), Bc(_Bc), At(_At), Bt(_Bt),
nu(_nu)
{
    setInitials();
}

//destructor
MuMazars::~MuMazars( )
{
}

void MuMazars::setInitials()
{
    if (Ec < 0.0)
        Ec = -Ec;

    if (epsD0c < 0.0)
        epsD0c = -epsD0c;

    if (epsD0t < 0.0)
        epsD0t = -epsD0t;

    tStrain.Zero();
    cStrain = tStrain;
    strain.Zero();

    tStress.Zero();
    cStress = tStress;
    stress.Zero();

    iniTangent.Zero();
    iniTangent(0,0) = 1.0 - nu;
}

```

```

iniTangent(0,1) = nu;
  iniTangent(0,2) = nu;
  iniTangent(1,0) = nu;
iniTangent(1,1) = 1.0 - nu;
  iniTangent(1,2) = nu;
  iniTangent(2,0) = nu;
iniTangent(2,1) = nu;
  iniTangent(2,2) = 1.0 - nu;
  iniTangent(3,3) = (1.0 - 2.0 * nu) / 2.0;
  iniTangent(4,4) = (1.0 - 2.0 * nu) / 2.0;
  iniTangent(5,5) = (1.0 - 2.0 * nu) / 2.0;
  fac = Ec / ((1.0 + nu) * (1.0 - 2.0 * nu));
  iniTangent *= fac;

  iniTangentInv.Zero();
  iniTangentInv(0,0) = 1.0 / Ec;
iniTangentInv(0,1) = - nu / Ec;
  iniTangentInv(0,2) = - nu / Ec;
  iniTangentInv(1,0) = - nu / Ec;
iniTangentInv(1,1) = 1.0 / Ec;
  iniTangentInv(1,2) = - nu / Ec;
  iniTangentInv(2,0) = - nu / Ec;
iniTangentInv(2,1) = - nu / Ec;
  iniTangentInv(2,2) = 1.0 / Ec;
  iniTangentInv(3,3) = (2.0 * (1.0 + nu)) / Ec;
  iniTangentInv(4,4) = (2.0 * (1.0 + nu)) / Ec;
  iniTangentInv(5,5) = (2.0 * (1.0 + nu)) / Ec;

  iniSmallTangent.Zero();
  iniSmallTangent(0,0) = iniTangent(0,0);
  iniSmallTangent(0,1) = iniTangent(0,1);
  iniSmallTangent(1,0) = iniTangent(1,0);
  iniSmallTangent(1,1) = iniTangent(1,1);
  iniSmallTangent(2,2) = iniTangent(3,3);

  tDam = 0;
  cDam = 0;
  tepseqT = 0;
  cepseqT = 0;
  tepseqC = 0;
  cepseqC = 0;
  tYt = 0;
  cYt = 0;
  tYc = 0;
  cYc = 0;
  tTangent = iniTangent;
  cTangent = tTangent;
  tTangentInv = iniTangentInv;
  cTangentInv = tTangentInv;
  tangent = iniSmallTangent;
}

//make a clone of this material
NDMaterial*
MuMazars::getCopy( )
{

```

```

MuMazars *clone ; //new instance of this class

clone = new MuMazars( this->getTag(), Ec, epsD0c, epsD0t, Ac, Bc, At, Bt, nu);

return clone ;
}

//make a clone of this material
NDMaterial*
MuMazars::getCopy( const char *type )
{
    return this->getCopy( ) ;
}

//send back order of strain in vector form
int
MuMazars::getOrder( ) const
{
    return 3 ;
}

const char*
MuMazars::getType( ) const
{
    return "MuMazars" ;
}

//swap historey variables
int
MuMazars::commitState( )
{
    cStress = tStress;
    cStrain = tStrain;
    cTangent = tTangent;
    cTangentInv = tTangentInv;
    cDam = tDam;
    cepseqT = tepseqT;
    cepseqC = tepseqC;
    cYt = tYt;
    cYc = tYc;

    return 0;
}

//revert to last saved state
int
MuMazars::revertToLastCommit( )

```

```

{
    return 0;
}

//revert to start
int
MuMazars::revertToStart( )
{
    cStrain.Zero();
    tStrain.Zero();
    strain.Zero();
    cStress.Zero();
    tStress.Zero();
    stress.Zero();

    setInitials();

    return 0;
}

//receive the strain
int
MuMazars::setTrialStrain( const Vector &strainFromElement )
{
    /*opserr << "setTrialStrain" << endl;
    getchar();*/

    double psi, k;
    double Ieps, Jeps, r, epst, epsc, Ytep, Ycep, Y, Y0, A, B, tYci, tYti;
    static Vector epsp(6), sigbar(6), sigbarpos(6), sigbarabs(6), sigp(6) ;

    tStrain(0) = strainFromElement(0) ;
    tStrain(1) = strainFromElement(1) ;
    tStrain(3) = strainFromElement(2) ;

    tStrain(2) = (- nu ) * (tStrain(0) + tStrain(1))/(1.0-nu) ;

    /*opserr << "strainFromElement = " << strainFromElement << endl;*/
    /*opserr << "tStrain = " << tStrain << endl;
    /*getchar();*/

    tTangent = cTangent ;
    tTangentInv = cTangentInv;

    /*opserr << "tTangent = " << tTangent << " tTangentInv = " << tTangentInv <<
endl;
    getchar();*/

    psi = (atan (2.0 * tStrain(3) / (tStrain(0) - tStrain(1) + 1e-20))) / 2.0;

    epsp.Zero();

```

```

    epsp(0) = ((tStrain(0) + tStrain(1)) / 2.0) + (((tStrain(0) - tStrain(1)) / 2.0) *
cos (2.0 * psi)) + (tStrain(3) * sin (2.0 * psi)) ;
    epsp(1) = ((tStrain(0) + tStrain(1)) / 2.0) - (((tStrain(0) - tStrain(1)) / 2.0) *
cos (2.0 * psi)) - (tStrain(3) * sin (2.0 * psi)) ;
    epsp(2) = tStrain(2) ;

Ieps = epsp(0) + epsp(1) + epsp(2);
Jeps = 0.5*(pow((epsp(0) - epsp(1)),2) + pow((epsp(1) - epsp(2)),2) + pow((epsp(2)
- epsp(0)),2));
tepsqT = ((Ieps/(2.0*(1.0 - 2.0*nu))) + ((1.0* sqrt(Jeps))/(2.0*(1.0 + nu))));
tepsqC = ((Ieps/(5.0*(1.0 - 2.0*nu))) + ((6.0* sqrt(Jeps))/(5.0*(1.0 + nu))));

sigbar = tTangent * epsp;

sigbarpos.Zero();
sigbarabs.Zero();

if (sigbar(0) >= 0)
{
    sigbarpos(0) = sigbar(0);
}
else
{
    sigbarpos(0) = 0;
}

if (sigbar(1) >= 0)
{
    sigbarpos(1) = sigbar(1);
}
else
{
    sigbarpos(1) = 0;
}

if (sigbar(2) >= 0)
{
    sigbarpos(2) = sigbar(2);
}
else
{
    sigbarpos(2) = 0;
}

sigbarabs(0) = abs(sigbar(0));
sigbarabs(1) = abs(sigbar(1));
sigbarabs(2) = abs(sigbar(2));

r = ((sigbarpos(0) + sigbarpos(1) + sigbarpos(2)) / (sigbarabs(0) + sigbarabs(1) +
sigbarabs(2) + 1.0e-20));

tYti = max(epsD0t,tepsqT);
tYci = max(epsD0c,tepsqC);

```

```

if (tYti < tYt)
    tYt = tYt;
else
    tYt = tYti;

if (tYci < tYc)
    tYc = tYc;

else
    tYc = tYci;

Y = r * tYt + (1.0-r) * tYc;

k = 0.7;

A = At*(2.0*(pow(r,2))*(1.0 - 2.0*k) - r*(1.0 - 4.0*k)) + Ac*(2.0*(pow(r,2))
- 3.0*r + 1.0);
B = (pow(r,((pow(r,2)) - 2.0*r + 2.0)))*Bt + ((1.0 - (pow(r,((pow(r,2)) - 2.0*r
+ 2.0))))*Bc);

Y0 = r * epsD0t + (1.0-r) * epsD0c;

tDam = 1.0 - (Y0 * (1.0 - A) / Y) - A * exp(-B * (Y - Y0)) ;

if ( tDam < 0.0 )
    tDam = 0.0;
if ( tDam > 0.9999 )
    tDam = 0.9999;

tTangent = (1.0 - tDam) * iniTangent ;
tTangentInv = (1.0 / (1.0 - tDam)) * iniTangentInv ;
sigp = tTangent * epsp ;

tStress(0) = 0.5 * (sigp(0) + sigp(1)) + 0.5 * (sigp(0) - sigp(1)) * cos(2.0 * (-
psi)) ;
tStress(1) = 0.5 * (sigp(0) + sigp(1)) - 0.5 * (sigp(0) - sigp(1)) * cos(2.0 * (-
psi)) ;
tStress(3) = -0.5 * (sigp(0) - sigp(1)) * sin(2.0 * (-psi)) ;

tangent(0,0) = tTangent(0,0);
tangent(0,1) = tTangent(0,1);
tangent(1,0) = tTangent(1,0);
tangent(1,1) = tTangent(1,1);
tangent(2,2) = tTangent(3,3);

strain(0) = tStrain(0);
strain(1) = tStrain(1);
strain(2) = tStrain(3);

stress(0) = tStress(0);
stress(1) = tStress(1);
stress(2) = tStress(3);

```

```

    return 0;
}

//send back the strain
const Vector&
MuMazars::getStrain( )
{
    return strain ;
}

//send back the stress
const Vector&
MuMazars::getStress( )
{
    return stress ;
}

//send back the tangent
const Matrix&
MuMazars::getTangent( )
{
    return tangent ;
}

const Matrix&
MuMazars::getInitialTangent
( )
{
    return iniSmallTangent ;
}

//print out data
void
MuMazars::Print( OPS_Stream &s, int flag )
{
    s << "MuMazars Material tag: " << this->getTag() << endln ;
    s << " Ec: " << Ec << " ";
    s << " epsD0c: " << epsD0c << " ";
    s << " epsD0t: " << epsD0t << " ";
    s << " Ac: " << Ac << " ";
    s << " Bc: " << Bc << " ";
    s << " At: " << At << " ";
    s << " Bt: " << Bt << " ";
    s << " nu: " << nu << " ";
}

int

```

```

MuMazars::sendSelf(int commitTag, Channel &theChannel)
{
    int res = 0, cnt = 0;

    static Vector data(21);

    data(cnt++) = this->getTag();
    data(cnt++) = Ec;
    data(cnt++) = epsD0c;
    data(cnt++) = epsD0t;
    data(cnt++) = Ac;
    data(cnt++) = Bc;
    data(cnt++) = At;
    data(cnt++) = Bt;
    data(cnt++) = nu;
    data(cnt++) = cDam;

    int i;
    for (i = 0; i < 6; i++)
        data(cnt++) = cStrain(i);

    for (i = 0; i < 6; i++)
        data(cnt++) = cStress(i);

    res = theChannel.sendVector(this->getDbTag(), commitTag, data);
    if (res < 0)
        opserr << "MuMazars::sendSelf() - failed to send data" << endl;

    return res;
}

int
MuMazars::recvSelf(int commitTag, Channel &theChannel, FEM_ObjectBroker &theBroker)
{
    int res = 0, cnt = 0;

    static Vector data(21);

    res = theChannel.recvVector(this->getDbTag(), commitTag, data);
    if (res < 0) {
        opserr << "MuMazars::recvSelf -- could not recv Vector" << endl;
        return res;
    }

    this->setTag(int(data(cnt++)));
    Ec = data(cnt++);
    epsD0c = data(cnt++);
    epsD0t = data(cnt++);
    Ac = data(cnt++);
    Bc = data(cnt++);
    At = data(cnt++);
    Bt = data(cnt++);
    nu = data(cnt++);
    cDam = data(cnt++);

    setInitials();
}

```

```
int i;
for (i = 0; i < 6; i++)
    cStrain(i) = data(cnt++);

for (i = 0; i < 6; i++)
    cStress(i) = data(cnt++);

return res;
}
```