THE UNIVERSITY *of* EDINBURGH

Nonlinear thermomechanical analysis of structures using OpenSees

Jian Jiang

THE UNIVERSITY
of EDINBURGH

A thesis submitted for the degree of Doctor of Philosophy

School of Engineering

The University of Edinburgh

2012

Declaration

This thesis and the research discussed has been completed solely by Jian Jiang under the supervision of Prof. A.S. Usmani and Dr. Luke Bisby.

Where other sources are quoted full references are given.

Jian Jiang

The University of Edinburgh, December 2012

Abstract

The behaviour of heated structures is strongly governed by thermal induced deformation and degradation of material properties. This thesis presents an augmentation of the software framework OpenSees to enable thermomechanical analysis of structures. The developments contributed to OpenSees are tested by series of benchmark cases and experimental results.

OpenSees is an object-oriented, open source software framework developed at UC Berekeley for providing an advanced computational tool to simulate non-linear response of structural frames to earthquakes. OpenSees was chosen to be extended to enable the modelling of structures in fire. The development of this capability involved creating new thermal load classes to define the temperature distribution in structural members and modifying existing material classes to include temperature dependent properties according to Eurocodes. New functions were also added into the existing corotational beam/column element (2D and 3D) to apply temperature related loads. A new geometrically nonlinear shell element was created (based on the existing linear MITC4 shell element in OpenSees) using total Lagrangian formulation. Appropriate thermal load, material and section classes were also developed for enabling thermomechanical analysis using the nonlinear shell element.

A number of benchmark tests were carried out to verify the performance of the new developments implemented in OpenSees. The benchmark tests involved subjecting beams and plates to a range of through depth temperature gradients with OpenSees results compared against closed form solutions. Further verification was also carried out by comparing OpenSees results with ABAQUS results.

The extended OpenSees framework was also used to model experiments such as two plane steel frames at elevated temperatures, the Cardington Restrained Beam Test and the Cardington Corner Test and an earthquake damaged reinforced concrete (RC) frame subjected to a subsequent fire. The existing DruckerPrager material class in OpenSees was used to the model concrete in the composite floor in the Cardington tests and in the RC frame. The pinching material available in OpenSees was used to model the beams and columns in the RC frame to consider the cyclic degradation of strength and stiffness during

the increasing cyclic displacements imposed on the RC frame before the fire. In all cases the results from OpenSees show good agreement with test data.

Acknowledgements

I would like to acknowledge the support and help from my supervisor Prof. Asif Usmani and Dr. Luke Bisby.

Thanks to my family, friend and colleagues at the University of Edinburgh, in particular our team members Jian Zhang, P. Kotsovinos and P. Khazaeinejad.

Publications

The following papers have been produced as a result of this research:

Journal papers

Jian Jiang and Asif Usmani. Modelling of steel frame structures in fire using opensees. (Accetpted and available online)

Jian Jiang, Payam Khazaeinejad and Asif Usmani. Nonlinear analysis of plates in fire using OpenSees. (Submitted to Engineering Structures)

Asif Usmani, Jian Zhang, Jian Jiang, Yaqiang Jiang and Ian May. Using OpenSees for structures in fire, 3(1): 57-70, 2012.

Conference Papers

Mariyana A. Ab-Kadir, Jian Zhang, Jian Jiang, Asif S. Usmani, Martin Gillie, Umesh K. Sharma and Pradeep Bhargava. Modelling of an earthquake damaged RC frame subjected to fire. Proceedings of 7th International Conference on Structures in Fire, Zurich, 479-489, 2012.

J. Jiang, A.S. Usmani. Thermo-mechanical analysis of steel frame structures using OpenSees. Proceedings of the 19th UK Conference of the Association for Computational Mechanics in Engineering, Edinburgh, 209-212, 2011.

J. Jiang, A.S. Usmani, J. Zhang, P. Kotsovinos. Numerical analysis of structures in fire using OpenSees. Proceedings of International Conference on Applications of Structural Fire Engineering, Prague, 319-323, 2011.

A.S. Usmani, J. Zhang, J. Jiang, Y.Q. Jiang. Using OpenSees for structures in fire. Proceedings of 6th International Conference on Structures in Fire, Michigan, 1089-1094,

2010.

J. Zhang, A.S. Usmani, J. Jiang, Y.Q. Jiang, P. Kotsovinos, I. May. Development and application of OpenSees for a RC frame in fire. Proceedings of the 19th UK Conference of the Association for Computational Mechanics in Engineering, Edinburgh, 233-236, 2011.

J. Zhang, A.S. Usmani, J. Jiang, Y.Q. Jiang, P. Kotsovinos, I. May. Using OpenSees for an RC frame in fire. Proceedings of International Conference on Applications of Structural Fire Engineering, Prague, 122-126, 2011.

Content

Declaration ..... **ii**

Abstract ..... **iii**

Acknowledgements ..... **v**

Publications ..... **vi**

Content ..... **viii**

List of Figures ..... **xiv**

List of Tables ..... **xxi**

Chapter 1 Introduction **1**

1.1 Background to the project 2

1.2 Aims of this research 2

1.3 Outline of thesis chapters 3

Chapter 2 Overview of Geometrically Nonlinear Shell Elements and Computational Modelling of Steel Framed Composite Structures in Fire **7**

2.1 Introduction 8

2.2 Literature review of nonlinear shell elements 9

2.2.1 Early conforming elements 11

2.2.2 Methods to bypass C1 continuity 12

2.2.3 Methods to alleviate shear locking 14

2.2.4 Drilling degrees of freedom 19

2.2.5 Geometrically nonlinear flat shell elements 22

2.3 Composite slabs in fire 23
2.3.1 Experimental research 25
2.3.2 Numerical simulation 37
2.3.3 Analytical methods 45
2.4 Performance based design 46

Chapter 3 OpenSees Class Hierarchy and Workflow 48

3.1 Introduction 49
3.2 Introduction to OpenSees 49
3.3 Class Hierarchy 50
3.3.1 Modeling class 51
3.3.2 Finite Element Model classes 51
3.3.3 Analysis class 52
3.3.4 Numerical classes 55
3.4 Steel material 56
3.5 Concrete material 57
3.5.1 Uniaxial concrete material 57
3.5.2 Drucker-Prager concrete material 62

Chapter 4 Nonlinear Corotational Beam-column Element 65

4.1 Introduction 66
4.2 General corotational beam element 66
4.2.1 Corotational framework 67
4.2.2 Local linear Euler-Bernoulli beam 72
4.2.3 Local nonlinear Euler-Bernoulli beam 75
4.2.4 Local Timoshenko beam 77
4.3 Stiffness method theory for OpenSees beam elements 79

4.3.1 Section state determination 80
4.3.2 Element state determination 81
4.3.3 Structure state determination 85
4.4 Force method theory for beam element 88
4.4.1 Section state determination 88
4.4.2 Element state determination 88
4.4.3 Structural state determination 90

Chapter 5 Thermomechanical Analysis of Framed Structures in OpenSees 92

5.1 Introduction 93
5.2 Existing mechanical analysis in OpenSees 93
5.3 Extended thermomechanical analysis of framed structures in OpenSees 95
5.3.1 Predictor 96
5.3.2 Corrector 98
5.4 Implementation of thermomechanical analysis in OpenSees 100
5.4.1 Thermal load class 100
5.4.2 Modified material classes 102
5.4.3 Section class 110
5.4.4 Element class 113
5.5 Benchmark testing of developed codes in OpenSees 115
5.5.1 Key features of thermomechanical response for a structural member 116
5.5.2 Restrained beam under thermal expansion 125
5.5.3 Single beam with finite boundary conditions 126

Chapter 6 Nonlinear Total Lagrangian Shell Element 137

6.1 Introduction 138

6.2 Geometrically nonlinear shell element following Total Lagrangian Formulation 138
6.2.1 Kinematics 138
6.2.2 Stress-strain relation 143
6.2.3 Element stiffness matrix 144

Chapter 7 Thermomechanical Analysis of Shell Structures using OpenSees 146

7.1 Introduction 147
7.2 Extendend thermomechanical analysis of plate structures in OpenSees . 147
7.2.1 Thermal load class 147
7.2.2 3D material class 148
7.2.3 Section class 152
7.2.4 Shell element class 153
7.3 Benchmark testing of developed codes in OpenSees 155
7.3.1 One-way bending of plate 155
7.3.2 Rectangular plate 159
7.3.3 Plate example to test DruckerPragerThermal 163

Chapter 8 Validation using Experimental data 168

8.1 Introduction 169
8.2 Steel frame test 169
8.3 Cardington restrained beam test 171
8.3.1 Geometric model 173
8.3.2 Material properties 174
8.3.3 Results 174
8.4 Cardington corner test 179

8.5 Roorkee RC frame subjected to fire following simulated seismic damage 183
8.5.1 Layout of the test RC frame 183
8.5.2 OpenSees model 188
8.5.3 Results 190

Chapter 9 Conclusions and Further Work 199
9.1 Introduction 200
9.2 Summary and conclusions 200
9.3 Further work 202

Reference 204
Appendix A Uniaxial material at elevated temperature 232
A.1 Steel01Thermal 232
A.2 Concrete02Thermal 234
A.3 Pinching4ThermalSteel 236
A.4 Pinching4ThermalConcrete 238
Appendix B FiberSection2dThermal 244
Appendix C DispBeamColumn2dThermal 248
Appendix D Scripts of restrained beam 251
Appendix E Scripts of beam with finite end restraint 253
Appendix F DruckerPragerThermal 257
Appendix G Scripts of plate subjected to thermal load 261
G.1 Cylindrical plate 261
G.2 Square plate 265
G.3 DruckerPragerThermal material 269

Appendix H Material of Cardington test model in OpenSees..... 275

Appendix I Materials of Roorkee RC frame test in OpenSees..... 276

List of Figures

Figure 2.1: Typical composite floor…….....24

Figure 2.2: General view of Cardington test building.…….....…26

Figure 2.3: Floor layout of Cardington test…..…….....26

Figure 2.4: Locations of six Cardington fire tests…….……28

Figure 2.5: British Steel restrained beam test…..…….....29

Figure 2.6: Post-test deformation of restrained beam test …….…….....29

Figure 2.7: Maximum vertical displacement and temperature in the beam…..……29

Figure 2.8: Plane frame test…..…….…….....30

Figure 2.9: Maximum vertical displacement and temperature in the beam…..……30

Figure 2.10: Schematic of BRE corner test……...31

Figure 2.11: General arrangement of BRE corner test……...……...31

Figure 2.12: Maximum vertical displacement and temperature in the beam.....31

Figure 2.13: Schematic of BS corner test…….....32

Figure 2.14: Internal view of the test compartment……...…….....32

Figure 2.15: Front view of compartment during early fire development stage…....33

Figure 2.16: Internal view of Post-test deformation of BS corner test…….....33

Figure 2.17: Front view of post-test deformation of BS corner test…….....33

Figure 2.18: Maximum vertical displacement and temperature in the beam…….....34

Figure 2.19: Schematic of BRE large compartment fire test……...…….34

Figure 2.20: Maximum vertical displacement and temperature in the beam…….....35

Figure 2.21: Internal view of the compartment.……...…….35

Figure 2.22: Maximum vertical displacement and temperature in the beam……36

Figure 2.23: Model of composite slabs with uniform thickness……...38

Figure 2.24: Composite slabs consisting of beam and slab elements……...39

Figure 2.25: Layout of Composite beam-slab system……...……40

Figure 2.26: Layout of composite slabs……41

Figure 2.27: Finite element representation of the composite beam……..……42

Figure 2.28: An orthotropic slab element model……...43

Figure 3.1: Hierarchy diagram of finite element model class……...51

Figure 3.2: Class diagram of Analysis framework……....52

Figure 3.3: Work flow of Analysis class…….....53

Figure 3.4: Access of AnalysisModel object to other objects……..……....53

Figure 3.5: Class diagram of Integrator……...…….……....54

Figure 3.6: Methods in EquiSolnAlgo object……...……....55

Figure 3.7: Procedure of the solution of the system of equations…….……....56

Figure 3.8: Stress-strain relationship of Steel01 in OpenSees……..…….....…....57

Figure 3.9: Stress-strain relationship of Concrete02 in OpenSees…….……...58

Figure 3.10: Definition of Pinching4 uniaxial material model…….61

Figure 3.11: Unloading stiffness degradation…….…….....61

Figure 3.12: Reloading stiffness degradation……...…….....61

Figure 3.13: Strength degradation……..……...62

Figure 4.1: Reference and corotational configuration of a typical beam element…67

Figure 4.2: Beam section subdivided into fibres……..……....80

Figure 4.3: Forces and displacements defined in the local coordinate system.....82

Figure 4.4: Flow chart of element state determination for stiffness method……....85

Figure 4.5: Flow chart of structural state determination…….….87

Figure 4.6: Element and section state determination for flexibility-based element.90

Figure 5.1: Static analysis program in OpenSees…….....94

Figure 5.2 Newton-Raphson method in OpenSees…….....94

Figure 5.3: Function "getResistingForce()" in DispBeamColumn2d class……....94

Figure 5.4: Function “getTangentStiff()” in DispBeamColumn2d class…….....95

Figure 5.5: Function “update()” in DispBeamColumn2d class…….....95

Figure 5.6: A general section divided into n fibres……..……....97

Figure 5.7: Flow chart of thermal-mechanical analysis in OpenSees……..99

Figure 5.8: Class hierarchy of the thermal load class…….100

Figure 5.9: Functions for Beam2DThermalAction class……...…….102

Figure 5.10: Modified material class in OpenSees……...…….….102

Figure 5.11: Stress-strain relationship for carbon steel at elevated temperature....103

Figure 5.12: Reduction factors for mechanical properties of carbon steel at elevated temperature…….103

Figure 5.13: Variation of the thermal elongation strain of steel at elevated temperature…….104

Figure 5.14: Functions for Steel01Thermal……...……105

Figure 5.15: Implementation of the function “getThermalElongation()”……...105

Figure 5.16: Implementation of the function “setThermalTangentAndElongation()”……..106

Figure 5.17: Stress-strain relationship of concrete under compression……..107

Figure 5.18: Mechanical properties of concrete under compression at elevated temperature…….107

Figure 5.19: Stress-strain relation of concrete under compression at elevated temperature…….108

Figure 5.20: Variation of the thermal elongation strain of concrete at elevated temperature…….109

Figure 5.21: Functions for FiberSection2dThermal class…….……..111

Figure 5.22: Implementation of the function “getTemperatureStress()”……....112

Figure 5.23: Functions for DispBeamColumn2dThermal…….....114

Figure 5.24: Implementation of function “addLoad()”…….....115

Figure 5.25: Schematic of a beam model with finite end restraints……...….117

Figure 5.26: Temperature deflection responses for combination of ε_T and ε_ϕ (Usmani et al. 2001) ……..118

Figure 5.27: Pinned-end beam subjected to UDL and ΔT…….....….118

Figure 5.28: Mid-span deflection of pinned-end beam subjected to UDL and ΔT.120

Figure 5.29: Axial restraining force of pinned-end beam subjected to UDL and ΔT……...121

Figure 5.30: Pinned-end beam subjected to UDL and T,y…….....….121

Figure 5.31: Mid-span deflection of pinned-end beam subjected to UDL and T,y123

Figure 5.32: Axial restraining force of pinned-end beam subjected to UDL and T,y……..123

Figure 5.33: Pinned-end beam subjected to UDL, ΔT and T,y…….123

Figure 5.34 Mid-span deflection of pinned-end beam subjected to UDL, ΔT and T,y……...125

Figure 5.35: Schematic of restrained beam subjected to uniform temperature rises…….125

Figure 5.36: Horizontal displacement of midpoint of the beam against temperature…….126

Figure 5.37: Schematic of a single beam with translational and rotational spring at the end……...…….126

Figure 5.38: Schematic of beams with different translational end restraint……...127

Figure 5.39: Horizontal displacement of beams with different translational end restraints…….....128

Figure 5.40: Vertical mid-span deflection of beams with different translational end restraints…….....128

Figure 5.41: End rotation of beams with different translational end restraints…...128

Figure 5.42: Mid-span vertical deflection contributions from different causes…..129

Figure 5.43: Axial force in beams with different translational end restraints…….130

Figure 5.44: Moment at mid-span of beams with different translational end restraints…….....130

Figure 5.45: Schematic of beams with different rotational end restraint……....131

Figure 5.46: Vertical deflection of beams with different rotational end restraints.132

Figure 5.47. End rotation of beams with different rotational end restraint…….132

Figure 5.48: Axial force of beams with different rotational end restraint……...133

Figure 5.49: Moment at mid-span of beams with various end rotational stiffness.133

Figure 5.50: Reaction Moment at the left end of beam for fixed and spring end restraints……..…….....133

Figure 5.51: Schematic of a single beam with translational and rotational spring at the end……134

Figure 5.52: Horizontal displacement of beams with translational and rotational restraints…….134

Figure 5.53: Vertical mid-span deflection of beams with translational and rotational restraints…….135

Figure 5.54: End rotation of beams with translational and rotational restraints….135

Figure 6.1 Geometry of a four-node shell element in the x, y plane……...139

Figure 7.1. Hierarchy for thermal load class in OpenSees……..……...148

Figure 7.2 Temperature distribution of the shell element in OpenSees……..…148

Figure 7.3 Hierarchy for 3D temperature dependent material class in OpenSees..149

Figure 7.4: Functions for DruckerPragerThermal……...…150

Figure 7.5: Implementation of the function "setThermalTangentAndElongation()"……152

Figure 7.6: Functions for MembranePlateFiberSectionThermal class……153

Figure 7.7: Functions for DispBeamColumn2dThermal…….…154

Figure 7.8: Schematic view of cylindrical bending of a rectangular plate……..155

Figure 7.9: Results of bending of uniformly heated cylindrical plate……....158

Figure 7.10: Horizontal reaction distribution along the edge in the y direction….158

Figure 7.11: Small and large deflections of rectangular plate ……...163

Figure 7.12: Plate subjected to edge compression……..……...165

Figure 7.13: Vertical displacement of plate subjected to compressive UDL…..…165

Figure 7.14: Deflection at centre of plate subjected to UDL……..165

Figure 7.15: Deflection at the centre of plate subjected to UDL and uniform temperature rise……..166

Figure 7.16: Deflection at the centre of plate subjected to thermal gradient alone.166

Figure 8.1: Schematic of the tested steel frames (mm) ……..170

Figure 8.2: Temperature dependent stress-strain curves of steel 37 used in test…170

Figure 8.3: Comparison between predicted and test deflection results ……..171

Figure 8.4: Location of the restrained beam test……...……..172

Figure 8.5: Layout and dimensions of the restrained beam……...….173

Figure 8.6: Geometric model of the restrained beam test ……..……174

Figure 8.7: Mesh of Cardington restrained beam model in OpenSees……..….174

Figure 8.8: Temperature distribution in the composite slab of restrained beam test……..177

Figure 8.9: Deformed shape of Cardington restrained beam model in OpenSees..177

Figure 8.10: Mid-span deflection of the restrained beam against temperature…...178

Figure 8.11: Horizontal displacement of the column against temperature…….178

Figure 8.12: Schematic of BS corner test……...…….179

Figure 8.13: Geometric model of the restrained beam test.……180

Figure 8.14: Mesh of Cardington corner test model in OpenSees ……180

Figure 8.15: Temperature distribution of the corner test.……...……181

Figure 8.16: Deformation shape of Cardington restrained beam model in OpenSees……182

Figure 8.17: Mid-span deflection of the secondary beam against temperature.….182

Figure 8.18: Horizontal displacement of the columns against temperature.…..…183

Figure 8.19: Building configuration and the frame sub-assemblage.…..……...185

Figure 8.20: Schematic of the test RC frame (all dimensions in mm)..…..……186

Figure 8.21: Details of a typical column……...…..……187

Figure 8.22: Detailing of slab reinforcement…..……..188

Figure 8.23: Details of beam reinforcement…..…….188

Figure 8.24: 3D model of the test frame…..…….……189

Figure 8.25: Mesh of the beams, columns and slab…..…….……190

Figure 8.26: Proposed loading history of the test frame…...…..……...190

Figure 8.27: Hysteretic curve of the test results…...…..……...191

Figure 8.28: Hysteretic curve of OpenSees results…...…..……..191

Figure 8.29: Comparison of base shear-roof displacement relationship…...……192

Figure 8.30: deformed shape of the RC frame under cyclic load in OpenSees…..192

Figure 8.31: Vertical reaction force (Z direction) of columns under cyclic load…192

Figure 8.32: Vertical reaction force (Z direction) of columns against peak displacement……...193

Figure 8.33: Fire test of the pre-damaged frame…...…….193

Figure 8.34: Temperature profiles of plinth beam…...…….194

Figure 8.35: Temperature profiles of roof beam…...…….....194

Figure 8.36: Temperature profiles of column…...…….194

Figure 8.37: Temperature profiles of slab…...…….…….195

Figure 8.38: Deformation shape of test frame in fire in OpenSees…….....….196

Figure 8.39: Deflection at the centre of slab of OpenSees model…...……...197

Figure 8.40: Horizontal displacement of column at roof level of OpenSees model……..197

Figure 8.41: Horizontal reaction in columns of OpenSees model….....….197

Figure 8.42: Vertical reaction in columns of OpenSees model…...……...198

List of Tables

Table 2.1: List of Milestone of finite element……...10

Table 2.2: List of techniques used for finite element method……....11

Table 2.3: Classification of finite element methods……..…..….13

Table 2.4: Summary of Cardington tests…….……..27

Table 4.1: Locations and weights of Lobatto quadrature……..…...84

Table 4.2 Element state determination algorithm……..…....90

Table 5.1: Properties of the beam model subjected to UDL and ΔT……..…....120

Table 5.2: Various boundary conditions of beam in the test……..…….....127

Table 5.3: Input parameters of the single beam model……..……..…….....127

Table 7.1: Input parameters of the plate model…….……....…….....159

Table 7.2: Input parameters of the plate model……...……...…….....163

Table 7.3: Input parameters for Drucker Prager material in OpenSees……...…164

Table 7.4: Input parameters for Drucker Prager material in ABAQUS……..…….164

Chapter 1

Introduction

1.1 Background to the project

Structural fire engineering is concerned with ensuring satisfactory performance of a building structure in the event of a fire.

The traditional method of ensuring that steel-framed buildings with composite slabs satisfy the regulatory requirements for fire resistance is to protect the exposed surfaces of all the columns and all of the supporting beams with an insulating material. The current method for specifying the required thickness of the insulating material is extremely conservative since it ignores the inherent fire resistance of the structure. The development of structural fire design codes e.g. BS5950: Part 8 and the Eurocodes provide a more solid scientific foundation for the provision of fire resistance. However, the guidance in these design codes is primarily based on considering isolated structural members in Standard Fire Tests. It is generally recognized that the whole-frame structural behaviour in fire cannot be represented by a test on an individual element, and furthermore, that the Standard Fire Test does not adequately represent a real local or compartment fire scenario.

Building codes worldwide are moving from prescriptive to performance-based approaches. Performance based codes establish fire safety objectives and leave the means for achieving those objectives to the designer. The performance based approach involves the assessment of three basic components comprising of: the likely fire behaviour; heat transfer to the structure; and the structural response. The overall complexity of the design depends on the assumptions and methods adopted to predict each of the three design components.

Following the observations from Cardington tests and other laboratory tests, numerous numerical models have been proposed in parallel with the development of special-purpose finite element program such as ADAPTIC, VULCAN and SAFIR. The results from the numerical analyses have been invaluable in understanding the behaviour of structures in fire and can be used to develop design guides as part of a performance based design procedure.

1.2 Aims of this research

The aim of this project was to extend the open source software framework OpenSees to deal with thermo-mechanical analysis of structures in fire. In the first phase of the work, OpenSees was extended to deal with framed structures in fire using 2D beam-column

elements. Later on, a geometrically nonlinear shell element was created (based on existing linear MITC4 shell element in OpenSees) using the total Lagrangian formulation for nonlinear thermo-mechanical analysis of plate structures. Finally, a number of benchmark cases and two of the Cardington tests were analysed to verify and validate the performance of the new developments implemented in OpenSees.

OpenSees is an open source object-oriented software framework developed at UC Berekeley and has so far been focussed on providing an advanced computational tool for analysing the non-linear response of structural frames subjected to seismic excitations. In view of its powerful nonlinear analysis capability, OpenSees was chosen to be extended to enable the modelling of earthquake damaged structural frames subjected to a subsequent fire. The development of this capability will involve fire load modelling, heat transfer analysis and structural thermo-mechanical analysis given the temperature distributions in structural members. The eventual aim is to enable full structural fire response to be simulated by directly reading member temperature distributions from the heat transfer analysis results. This aspect of the work is being dealt with by other members of the team and will not form part of this thesis, which focuses on the thermo-mechanical response of the structure to known temperature distributions (available from previous experimental or numerical analyses).

The existing beam-column element in OpenSees was first extended to model framed structures in fire. A thermal load class was developed to define the temperature distribution through the depth of a section and material properties at elevated temperature according to the Eurocodes were added to the existing uniaxial material classes (for steel and concrete) in OpenSees. A geometrically nonlinear shell element was developed based on the total Lagrangian formulation. The performance of the development work carried out in OpenSees was tested using a number of benchmark problems and two of the Cardington tests.

1.3 Outline of thesis chapters

Chapter 1

Introduction

A brief discussion of the background to the project and an overview of the research undertaken are presented.

Chapter 2

Overview of Geometrically Nonlinear Shell Elements and Computational Modelling of Steel Framed Composite Structures in Fire

A literature review of shell elements is presented including methods preventing the shear and membrane locking problems, drilling degrees of freedom and MITC shell elements. The development of experimental, numerical and analytical analysis of composite floor systems is also reviewed. The experimental review includes a brief description of Cardington tests.

Chapter 3

OpenSees Class Hierarchy and Workflow

The class hierarchy of OpenSees is presented including modelling classes, finite element classes, numerical analysis classes. The properties of existing uniaxial material classes Steel01, Concrete02, Pinching4 and multiaxial material class DruckerPrager are introduced which are extended to be temperature dependent in Chapter 5 and Chapter7, respectively, to conduct thermo-mechanical analyses in OpenSees.

Chapter 4

Nonlinear Corotational Beam-Column Element

The general corotational framework for beam elements is presented. Different beam theories (e.g. Euler-Bernoulli or Timoshenko beam) can be included in the framework by defining corresponding stiffness matrices in the base coordinate system. The displacement-based and flexibility-based methods for beam-column elements are presented including state determination at three levels (section state, element state and structural state).

Chapter 5

Thermo-mechanical Analysis of Framed Structures in OpenSees

The procedure of nonlinear structural analysis adopted in OpenSees is introduced followed by the procedure used in this work for nonlinear thermo-mechanical analysis. The unbalanced force is calculated from external loads, thermal forces as well as residual resisting forces due to the material degradation at elevated temperature. The existing materials Steel01, Concrete02 and Pinching4 are modified to include temperature dependent properties according to the Eurocodes. The performance of the extended codes in OpenSees is verified using two benchmark problems. The first involves one-half of a beam

subjected to a uniform temperature increase and the second is a beam with a range of restraint conditions (zero to infinite) subjected to a range of thermal gradients.

Chapter 6

Geometrically Nonlinear Total Lagrangian Shell Element

A geometrically nonlinear shell element has been developed using the total Lagrangian formulation. This new shell element is a flat four-node isoparametic element including the drilling degree of freedom formed by a combination of membrane element and Mindlin plate bending element. This work is further develops the linear MITC4 element provided in OpenSees which uses the mixed-interpolation of the transverse shear strain components to avoid shear locking problems.

Chapter 7

Thermo-mechanical Analysis of Shell Structures in OpenSees

The implementation of the thermo-mechanical analysis of shell structures in OpenSees is presented. A new thermal load class for shell elements is created to define the temperature distribution through the depth of shell section as well as in the horizontal plane of shell element. The existing elastic and elasto-plastic materials for shell elements were modified to include temperature dependent properties. The geometrically nonlinear shell element presented in Chapter 6 was extended to model shell structures in fire. Two benchmark problems based on cylindrical (or one-way) bending and two-way bending of rectangular plates are analysed to verify the performance of the new code developed in OpenSees.

Chapter 8

Validation Using Experimental Data

After testing the performance of the extended OpenSees using thermo-mechanical benchmark problems, further validation is carried out using experimental data. At first two steel frames subjected to a uniform temperature increase were analysed. Following this the Cardington Restrained Beam test and the British Steel Corner test were analysed to validate the OpenSees development for thermo-mechanical analysis of steel framed composite structures. Finally a fire experiment on an RC frame first subjected to simulated seismic damage conducting at IIT Roorkee in India was analysed to test the capability of the extended OpenSees in modelling RC structures subjected fires following an earthquake.

Chapter 9

Conclusions and Future Work

A general discussion is given, summarising the results and conclusions obtained from the project, together with suggestions for future work.

Chapter 2

Overview of Geometrically Nonlinear Shell Elements and Computational Modelling of Steel Framed Composite Structures in Fire

2.1 Introduction

Structural fire engineering is concerned with ensuring satisfactory performance of a building structure in the event of a fire. The traditional approach has been to follow prescriptive rules, but more advanced methods based on calculation have recently been developed. These range from simple approaches based on hand calculation or tabulated data, offering limited benefits, to sophisticated computer analysis which enables the full value of structural fire engineering to be realised. Such approaches to modelling structural behaviour are often complemented by consideration of more realistic fire scenarios than the standard fire. The performance based approach involves the assessment of three basic components comprising the likely fire behaviour, heat transfer to the structure and the structural response. The overall complexity of the design depends on the assumptions and methods adopted to predict each of the three design components.

The main work of this thesis focuses on developing a three-dimensional thermomechanical analysis capability of modelling steel-framed composite structures in OpenSees. The key work is to develop a geometrically nonlinear shell element based on an existing linear shell element in OpenSees. The existing shell element is a flat four-node isoparametric element including the drilling degree of freedom formed by a combination of membrane element and Mindlin plate bending element. MITC technique (i.e. Mixed Interpolation of Tensorial Components) (Dvorkin and Bathe 1984 and Bathe and Dvorkin 1986) was used to alleviate shear locking problem. The geometrical nonlinearity of the new shell element followed the Total Lagrangian procedure. Two Cardington tests were used to validate the performance of the developed thermomechanical analysis capabilities in OpenSees. To this end this chapter presents the literature review of research that is relevant to this thesis. The review includes three main topics:

1. Literature review of geometrically nonlinear shell finite elements including shear locking, membrane locking, drilling degrees of freedom (Section 2.2).

2. Literature review of research on the composite floor system in fire including experimental, numerical and mathematical analysis (Section 2.3).

3. Performance based design of structures in fire (Section 2.4)

2.2 Literature review of nonlinear shell elements

The wide application of plate/shell structures in engineering practice has caught the interest of many researchers. A large amount of research has been proposed over the five decades and the development of simple and efficient shell finite elements goes through three major approaches (Zienkiewicz and Taylor 2005): (1) the curved shell elements based on classical shell theory with curvilinear coordinates; (2) the degenerated shell elements derived from three-dimensional solid elements and (3) the flat shell elements obtained by the combination of the membrane and bending behaviour of plate elements. In general, it is difficult to identify which shell element is the most advantageous. The flat shell elements have been used extensively because of the simplicity in their formulation (directly formed by combination of membrane and bending plate element) as well as the effectiveness in performing computation.

In this section the review will focus on the development of the flat plate/shell elements. The development involves convergence requirements for finite element method, methods to alleviate shear and membrane locking problems as well as drilling degrees of freedom. Considering that the formulation of existing linear shell element ShellMITC4 in OpenSees was based on Reissner-Mindlin plate theory and MITC technique (Mixed Interpolation of Tensorial Components) was used to deal with the shear locking problem. Therefore the review in this section will starts from the continuity requirement of finite element method followed by methods to alleviate the shear locking problems.

Hrabok et al. (1984) presented a brief review of development on finite elements for thin plates followed by an extensive and detailed tabular listing of plate bending elements. A review of development of hybrid/mixed finite element method is in Pian (1995). Table 2.1 lists some milestone elements and Table 2.2 lists some techniques used for the plate and shell elements in the finite element method.

Element Name	Year	Researcher	Description
1st rectangular plate bending element	1960	Clough and Adini	12 dofs non-conforming
1st conforming rectangular PBE	1965	Bogner et al.	16 & 36 dofs

HCT	1965	Hsieh, Clough and Tocher	Triangular element with 3 sub-triangles within each element
BCIZ	1965	Bazeley et al.	Superimposing non-polynomial shape function
DKT	1968	Wempner	Impose the Kirchhoff constraints at discrete locations
QUAD4	1978	MacNeal	A simple 4-node quadrilateral element with special modifications
MITC	1984	Bath	Interpolate displacements and strains separately and connect these interpolations at selected points in element domain

Table 2.1: List of Milestone of finite element

Technique	Year introduced	Researchers	Description
Finite element method	1960	Clough	Numerical technique to approximate solution of equation by discretization of domain
Isoparamatric element	1961	Taig	Same shape function used to interpolate both coordinate and displacement
Hybrid method	1964	Pian	Hybrid stress method based on modified complementary potential energy principle
Area coordinate	1965	Bazeley et al.	Alternative to define triangular element
Completeness requirement	1965	Irons and Draper	Requirement of minimum orders of polynomial of shape function
Patch test	1965	Bazeley et al.	Procedure to ensure element fulfil the consistency requirement
Mixed method	1965	Herrmann	Based on a modified Reissner variational principle
Discrete Kirchhoff constraint	1968	Wempner	Impose the Kirchhoff constraints at discrete locations
Selective reduced integration	1969	Doherty et al.	Selected different integration scheme for shear and bending terms
Uniform reduced integration	1971	Zienkiewicz et al.	Same reduced integration scheme for the shear and bending terms
Amended shear factor	1973	Fried	Balancing the bending and shear energies via a shear stiffness multiplier

Free formulation	1975	Bergan and Hanseen	The element is derived directly from satisfying the patch test and the rigid body motion and constant strain requirements.
Loof and semi-loof element	1976	Irons	Nodal points are not located at the corners of the elements
Constraint index	1977	Hughes et al.	Predict locking in various FE mesh
Assumed natural strain (ANS)	1978	MacNeal	Independently assumed shear strain fields in element natural coordinate
Locking indicator	1978	Pugh Hinton	A measure of degree of singularity in a element stiffness matrix to check locking
Enhanced Assumed Strain (EAS)	1990	Simo and Rifai	An enhanced strain field based on the Hu-Washizu variational principle
Degenerated shell element	1990	Ahmad	Derived from three-dimensional solid elements

Table 2.2: List of techniques used for finite element method

2.2.1 Early conforming elements

At the beginning of the 1960s a number of elements were proposed by researchers such as Clough (1965), Adini (1960 1961), Melosh (1961) and Tocher (1962). By the mid 1960s the variational basis for the finite element method had become better understood, and coupled with this, came the realisation that completeness of polynomial used for shape functions and inter-element conformity were of importance to the convergence of the finite element. Let n be the highest order derivative of the field variable ϕ that appears in the functional Π. The completeness requires that within each element the assumed shape function for ϕ must contain a complete polynomial of degree n or higher (Cook et al. 2002). The conformity means that across element boundaries there must be continuity of ϕ and derivatives of ϕ through order (n-1) (Cook et al. 2002). In structural mechanics of in-plane problems (n=1), the physical meaning of completeness is that it is required that the finite element solution can represent the rigid body motion and constant strain conditions. Otherwise if nodal displacements representing rigid body motion or constant strain are assigned in the element, non-zero or non-constant strain will occur when the element size becomes infinite which makes the finite element solution not converge to the real solution. The physical meaning of conformity lies in that the displacement field should be continuous both within the element and across element boundaries which is called C^0 continuity and otherwise unexpected strain will be caused and added in the system strain energy resulting in divergence of finite element

solution. For Kirchhoff plate theory (n=2) the conformity requirement proved to be particularly problematic in that inter-element continuity is required for both the transverse displacement and the slope normal to the element boundary which is called C^1 continuity. Efforts had been made to achieve conformity including the sub-domain approach (HCT element) (Gallagher 1969; Clough and Tocher 1965), amended shape function (BCIZ element) (Bazeley et al. 1965), substitute shape function (Irons and Razzaque 1972) and using higher order elements (Argyris 1968; Bell 1969; Bogner et al. 1965; Irons 1969; and Visser 1969; Cowper et al. 1968)

The assumed displacement element formulation via the principle of minimum potential energy dominates the early development of plate bending element. Still other researchers sought elements based on alternative variational principles. One logical choice is the principle of minimum complementary energy. Initial work in this field was done by Veubeke (1965). Considerable clarification and simplification can be attributed to Southwell (1950), Morley (1967, 1968), Elias (1968) and Sander (1970).

2.2.2 Methods to bypass C1 continuity

Conforming plate elements were not only difficult to obtain, but with the exception of the higher order elements, they were found to be too stiff. There was considerable scepticism about the need to meet the C^1 continuity requirement (Melosh 1961) and researchers were looking for ways to bypass this requirement. Basically, there are two methods to solve these continuity problems：hybrid/mixed methods and Mindlin plate theory.

2.2.2.1 Hybrid/Mixed methods

A review of development of hybrid/mixed finite element method was in Pian (1995). One of the methods was the use of Lagrange multipliers to allow relaxation of the continuity requirement along inter-element boundaries thereby reducing the conformity requirement from C^1 to C^0. The first hybrid method developed was the stress hybrid method by Pian (1964, 1965, 1968) using a modified complementary potential energy principle. Other researchers such as Tong (1970), Kikuchi and Ando (1972), developed various displacement hybrid approaches based on modified forms of the principle of minimum potential energy.

Another alternative to the displacement method was presented by Herrmann (1965) and Pian and Tong (1969). It was a mixed method and was based on a modified Reissner variational principle. The generalized displacement method appeared to have been initiated by Jones (1964), Greene et al. (1969), Anderheggen (1970), and Harvey and Kelsey (1971). The generalized equilibrium method was introduced by Anderheggen (1969) and Sander (1970).

Hrabok et al. (1984) presented an overview of the different element categories based on their variational formulation as shown in Table 2.3.

Finite element method		Variational principle	Assumed functions inside the element	Along interelement boundaries	Unknowns in finial equations
Displacement		Minimum potential energy	Continuous displacements	Displacement compatibility	Nodal displacements
equilibrium		Minimum complementary energy	Continuous and equilibrium stresses	Equilibrium boundary tractions	(a) Generalized displacements (b) Stress parameters
Hybrid	Hybrid stress method	Modified complementary energy	Continuous and equilibrium stresses	Assumed compatible displacements	Nodal displacements
	Hybrid displacement method (1)	Modified potential energy	Continuous displacements	Assumed compatible displacements	Nodal displacements
	Hybrid displacement method (2)	Modified potential energy	Continuous displacements	Assumed equilibrating boundary tractions	Nodal displacements and boundary forces
Mixed	Reissner's principle	Reissner method as modified by Harrmann	Continuous stress and displacement functions	Combinations of boundary tractions and displacements	Combinations of displacements and tractions
	Generalized displacement method	Modified potential energy	Continuous displacements	Lagrangian multipliers (stresses)	Nodal displacements and Lagrangian multipliers
	Generalized equilibrium method	Modified complementary energy	Continuous and equilibrium stresses	Lagrangian multipliers (displacements)	Nodal displacements and Lagrangian multipliers

Table 2.3: Classification of finite element methods (Hrabok et al. 1984)

2.2.2.2 Mindlin plate element

In addition to the hybrid/mixed method, the Reissner-Mindlin plate element was introduced to bypass the C^1 continuity requirement where translations and rotations are interpolated independently. The detail of this theory can refer to references (Cook et al. 2002; Zienkiewicz and Taylor 2005).

2.2.3 Methods to alleviate shear locking

For the Mindlin plate element, the translations and rotations usually use the same interpolation functions which leads to the so called shear locking problems. The shear locking phenomenon happens when deflection of a plate/shell becomes zero as the thickness of its section becomes infinite small for bending-dominated problems. It is known that the zero deflection results from spurious stiffness provided by overestimating the strain energy induced by shear strain through the thickness of the section. Generally the overestimation of the shear strain energy is due to the different order of interpolation function taken for the slope dw/dx (w is the transverse displacement) and section rotation θ which leads to unexpected shear strain γ ($\gamma = dw/dx - \theta$) within the element and cross the element boundaries

Similar with shear locking problem, membrane locking is characterized by the occurrence of spurious membrane strain in originally curved beam and shell elements in a state of pure bending. Membrane locking results from the inability of an element to bend without stretching. Since any bending deformation of the element is then accompanied by stretching of the mid-surface, membrane energy is always generated in bending which increases the bending stiffness. Theoretically, the scheme for avoiding shear locking can be used for preventing the membrane locking considering both types of locking problems are due to the different orders of polynomials for interpolating horizontal and the transverse displacements. More work on the membrane locking problems of flat shell elements can be found in Cook (1994), Groenwold and Stander (1995), Choi and Lee (2003), and Cui et al. (2008).

The existing shell element in OpenSees uses MITC technique to alleviate shear locking problems and therefore this section will focus on review of methods to avoid shear locking. In general, the success or failure of an element formulation in avoiding shear locking is entirely related to the definition of the assumed shear strain and how they relate to the element displacement fields (Bucalem and Bathe 1997). Much effort has been aimed at

understanding and predicting and therefore eliminating locking, such as, selective/uniform reduced integration, Discrete Kirchhoff Theory (DKT), hybrid/mixed method, amended shear factor, unequal order interpolation of deflection and rotation, substitute shear strain field, assumed natural strain as well as enhanced assumed strain. The details of these methods will be presented in the following sections.

2.2.3.1 Understanding and predicting locking

Ziekiewicz (1976) and Pugh et al. (1978) introduced the concept of locking indicator as a measure of the degree of singularity in an element stiffness matrix. Based on the criterion mentioned by Zienkiewicz and Hinton (1976), a constraint index was developed by Hughes and his colleagues (Hughes el at. 1977, 1978; Malkus and Hughes 1978; Tsach 1981) to predict locking behaviour in various finite element meshes. These techniques proved useful in some situations, but were problem dependent, and sometimes led to overly pessimistic assessment of element behaviour (reviewed by Averill and Reddy 1990). A "Kirchhoff mode" criterion was introduced by Hughes and Tezduyar (1981) and Hughes and Taylor (1981) which served as a basis for constructing suitable shear strains. The interpolation function had to be chosen to allow the shear angles to obey the Kirchhoff constraints throughout the element domain or at certain key positions.

2.2.3.2 Selective/uniform reduced integration

The shear locking problem can be solved easily by reduced integration. Doherty et al. (1969) introduced the selective reduced integration scheme which "under integrated" the element transverse shear energy while exactly evaluated the bending energy. Zienkiewicz et al. (1971, 1976) and Pawsey and Clough (1971) suggested the uniform reduced integration by under-integrating all terms. These two integration methods were to dominate the plate bending research field from the mid 1970s to the present. The main contributions to this area have come from research groups associated with Hughes (Hughes et al. 1977, 1978b; Hughes and Cohen 1978a) and Hinton (Pugh el at. 1978 and Hinton and Bicanic 1979) and from Cook (1972), Stolarski and Belytschko (1983). Spilker and Munir (1980) extended the use of the reduced integration scheme to hybrid stress formulations. Reduced integration suffers from so-called "hourglass" or "zero energy" modes since it tends to be excessively flexible. Stabilization matrices are required to remove these spurious modes (Belytschko and Tsay 1983; Belytschko et al. 1983, 1984, 1989, 2000; Belytschko and Leviathan 1994). Over-stiff

solutions may still occur in problems with highly constrained boundaries. On the other hand, when such elements are used in problems with lightly constrained boundaries the solutions may fail due to the formation of mechanisms or be polluted by oscillations caused by near mechanisms (Hughes et al. 1978; Pugh et al. 1978; Cook and Zhao 1982; Park and Flaggs 1984; Belytschko et al. 1984). Equivalence has been established (Malkus and Hughes 1978) between mixed finite elements based on the modified Hellinger-Reissner functional and displacement elements developed using selective reduced integration.

2.2.3.3 DKT

One alternative to using selective or reduced integration for Mindlin plate elements is to impose, at the element level, the Kirchhoff constraints of normality at discrete locations such as the Gauss integration points. This idea was first introduced in 1968 by Wempner et al. and had been used for plates by researchers such as Stricklin (1969), Baldwin (1973), Fried (1973), Irons (1976), Batoz et al. (Batoz et al. 1980; Batoz 1982; Batoz and Tahar 1982) and Lyons (1977). A second alternative is to impose the Kirchhoff constraints in a weighted integral sense by using Lagrangian multipliers which is interpolated over the element tested by Hrabok (1981).

2.2.3.4 Hybrid/Mixed method

An alternative scheme for dealing with shear-locking problem is the hybrid/mixed formulation in which separate interpolations are used for the stresses and displacements (Lee and Pian 1978; Noor and Peters 1981)

2.2.3.5 Amended shear factor

One method is the use of shear factor to amend the shear strain. Fried et al. (Fried 1973, 1974; Fried and Yang 1973) proposed a remedy for the locking phenomenon by balancing the bending and shear energies via a shear stiffness multiplier of the form $C(h/L)^2$ where a positive constant C can be determined from numerical tests to give best convergence to the thin plate solution.

2.2.3.6 Unequal order interpolation of deflection and rotation

Other efforts to alleviate locking concentrated on special selection of the field variable interpolations and certain ad hoc schemes. MacNeal (1978) produced a four-node quadrilateral shell element called QUAD4 by an array of special modifications including reduced order integration for shear terms, enforcement of curvature compatibility, and the augmentation of transverse shear flexibility to account for a deficiency in the bending strain energy. Hughes and Tezduyar (1981) and Hughes and Taylor (1981) had extended the idea of MacNeal to produce four-node quadrilateral and three-node triangular elements, respectively. They suggested a "Kirchhoff mode" criteria which served as a basis for constructing suitable shear strains instead of the additional special techniques proposed by MacNeal. A special interpolation strategy, labelled interdependent variable interpolations, had been proposed by Tessler and his colleagues (Tessler and Dong 1981; Tessler 1981, 1982; Tessler and Hughes 1983) which the deflection interpolation polynomial were taken one order higher than the rotation polynomials.

2.2.3.7 Assumed natural strain (ANS)

The assumed natural strain (ANS) formulation has undergone rapid development. The key formulation step is the replacement, in the potential energy principle, of selected displacement-related strains by independently assumed strain fields in element natural coordinates. The transverse shear strain is interpolated from the displacement-dependent strains defined at the mid-side of element edges. This procedure is conceived as one of several competing methods to solve the element locking problems. Related approaches were MacNeal (1978), Hughes and Tezduyar (1981) and Hughes and Taylor (1981), Dvorkin and Bathe (1984), Bathe and Dvorkin (1985, 1986), Wempner (1982), Park (1986), Park and Stanley (1986), and Jang and Pinsky (1987). Militello and Felippa (1990) study the ANS formulation from a variational standpoint. The element developed by Dvorkin and Bathe (1984) was generalized to a new family of locking-free elements (Hinton and Huang 1986) using the corrected shear strain functions (Huang and Hinton 1984). In the paper by Hinton and Huang 1986 the selectively integrated Mindlin plate elements may be viewed as fully integrated elements with substitute shear strain fields.

As a possibility to overcome the problem of transverse shear locking in plates and shells Bletzinger et al. (2000) proposed the Discrete Shear Gap (DSG) method. The method has certain similarities to the existing concept ANS, but it has some unique features: first, it is

directly applicable to both triangles and quads, without any further considerations, like a particular choice of sampling points or the introduction of additional nodes or degrees of freedom. Second, it applies directly to elements of arbitrary polynomial order. It was then generalized to solve other locking problem (e.g. membrane locking) by rephrasing DSG as "Discrete Strain Gap" (Bischoff and Bletzinger 2001; Bischoff et al. 2003; Koschnick et al. 2005).

Based on the ANS formulation, the Mixed Interpolation of Tensorial Components (MITC) approach has been used successfully to propose shell elements. This MITC technique was used in the existing shell element in OpenSees to alleviate shear locking problems. The basic idea and also the key to success of the MITC technique are to interpolate displacements and strains separately and "connect" these interpolations at "tying points" (i.e. selected points in element domain). The key step of the mixed integration of the tensorial components approach is to define assumed strain component fields that are linked with the displacement variables and that lead to element formulations that are free from membrane and shear locking effects. MITC approach coined after the works of Dvorkin and Bathe (1984) and Bathe and Dvorkin (1986) and with roots in earlier publications (MacNeal 1978, 1982; Hughes and Tezduyar 1981; Park 1986; Park and Stanley 1986)

The technique was originally proposed for 4-node and 8-node quadrilateral shell elements (the MITC4 and MITC8 elements) by Dvorkin and Bathe (Dvorkin and Bathe 1984; Bathe and Dvorkin 1986) and was later extended to 9 and 16-node elements (the MITC9 and MITC16 elements) by Bucalem and Bathe (1993) and Bathe et al. (2003). The technique was also used for triangular 3, 6, 7 and 12-node shell elements (the MITC3, MITC6 MITC7 and MITC12 elements) (Bathe and Brezzi 1989b; Bucalem and Bathe 1997; Lee and Bathe 2004; Lee et al. 2007; Kim and Bathe 2009) and in particular regarding shell analyses shows further potential. The theoretical foundations of these MITC elements can be found in (Bathe and Brezzi 1985; Bathe and Brezzi 1987; Brezzi et al. 1989, 1991). The formulation and numerical performance of these MITC elements can be found in Dvorkin and Bathe 1984; Bathe and Dvorkin 1985, 1986; Bucalem and Bathe 1993, 1997; Bathe et al. 2000; Lee and Bathe 2004, 2010.

2.2.3.8 Enhanced assumed strain (EAS)

Another interesting scheme arising from mixed variational formulations is the Enhanced Assumed Strain (EAS) method first proposed by Simo and Rifai (1990) and further developed in the linear elastic range (Andelfinger and Ramm 1993; Bischoff and Ramm 1997; Eckstein and Basar 2000; Cardoso et al. 2006, 2007). The key point of this method lies in the use of a strain field composed of a compatible strain field and an enhanced strain field based on the Hu-Washizu variational principle to reduce shear locking.

The many formulations discussed so far are not all independent and a more comprehensive treatment of these topics and the equivalences between certain methods are discussed in the texts by Ziekiewicz (1977) and Gallagher (1975). Malkus and Hughes (1978) and Spilker (1980) have shown the equivalence of selective reduced integration and some mixed methods.

The reduced integration method is the most common method to alleviate shear locking problem but suffers zero energy mode which causes convergence difficulties. Hybrid or mixed method uses different variational principles but the expression is complicated. The assumed strain method is straightforward and is being used widespread. The MITC technique developed based on the ANS method was used in the existing shell element in OpenSees. In addition, the drilling degree of freedom was also used in the forming of this shell element which will be presented in the next section.

2.2.4 Drilling degrees of freedom

A difficulty arises if all the elements meeting at a node are co-planar. The difficulty is due to the assignment of the zero stiffness in the θ_z direction which leads to a singular structural stiffness matrix in the global coordinate system. A number of alternatives have been presented that avoid the presence of a singularity. A drilling degree of freedom is defined as an in-plane rotation about a normal to the plane of the element. The motivation for introducing the drilling rotation into a shell element is to avoid the problem of singularity in the stiffness matrix and also to improve the accuracy of numerical results. The main objectives behind this idea are as follows (Paknahad et al. 2007):

1. Improve the element performance while avoiding the use of midpoint degrees of freedom. Midpoint nodes have lower valency than corner nodes, demand extra effort in mesh definition and generation, and can cause modelling difficulties in nonlinear analysis and dynamics.

2. To solve the "normal rotation problem" of smooth shells analyzed with finite element programs that carry six degrees of freedom per node.

3. To simplify the modelling of connections between plates, shells and beams.

The early efforts to develop membrane elements with drilling degrees of freedom (Turner 1956; Abu-Gazaleh 1965; Fraeijs and Veubeke 1965; Argyris 1965; Felippa 1966; Bergan 1967; Carr 1967; Scordelis 1967; Tocher and Hartz 1967; Holand and Bergan 1968; MacLeod 1969; William 1969; Dungar and Severn 1969; Lyons 1970; Tinawi 1972; Yoshida 1974; Zienkiewicz 1977) focused on cubic displacement interpolation based on the principle of minimum potential energy or hybrid stress method. The above formulations are unsuccessful or partially successful due to the imposition of quite severe restrictions on the displacement in order to achieve compatibility. The readers can refer Allman 1984; Bergan and Felippa 1985; and MacNeal and Harter 1988 for a comprehensive review of these early papers. Although Robinson (1980) and Mohr (1982) had achieved a notably greater measure of success, it was not until the paper by Allman (1984) and Bergan and Felippa (1985) that the practical utility of corner rotations became firmly established for at least the case of three-node triangular membrane elements. Allman's triangular element combined the best features of the two elements described by Turner (1956), Fraeijs and Veubeke (1965) and Argyris (1965), namely: compatible quadratic displacements consist of two components of displacement and a vertex rotation located at the vertices only. The stiffness matrix was established by the principle of minimum potential energy and exhibited an unusual type of zero energy mode which can be easily suppressed by prescribing an arbitrary value for any one of the vertex rotation in the entire finite element model. Bergan and Felippa (1985) derived a triangular element using the so-called free formulation (Bergan and Hanssen 1975) which was coordinate invariant and passed the patch test for any geometry.

The key to their success appears to have been the use of quadratic displacement functions rather than the cubic functions employed in many earlier papers (MacNeal and Harter 1988). These papers have created a revival of interest in elements with drilling degrees of freedom as evidenced by numerous work by engineers (Carpenter et al. 1985; Taylor and Simo 1985; Bergan and Felippa 1986; Jetteur 1986, 1987; Cook 1986 1987; Jaamei 1987; MacNeal 1987; Allman 1988; Lee and Yoo 1988; MacNeal and Harter 1988; Robinson and Sheng 1988; Taylor 1988; Hughes et al. 1989; Hughes and Brezzi 1989; Ibrahimbegovic et al. 1990; Ibrahimbegovic and Wilson 1991; Hughes et al. 1995; Chinosi 1995; Chinosi et al. 1997). Hughes and Brezzi (1989) presented a rigorous framework based on independently interpolated rotation fields and a modified variational principle based on the Euler-Lagrange equations presented by Reissner (1965), but with improved stability properties in the context of the discrete approximations. Early finite element interpolations employing the formulation of Hughes and Brezzi were presented by Hughes et al. (1989), Ibrahimbegovic et al. (1990), as well as Ibrahimbegovic and Wilson (1991). Pimpinelli (2004) proposed a four-node quadrilateral membrane element with drilling degree of freedom based on the modified Hu-Washizu functional where the enhanced strain and the enhanced rotation fields are included.

These elements depend on a problem-dependent penalty parameter γ, which relates the in-plane translations to the in-plane rotation. The value of γ has been the topic of a number of studies (Hughes et al. 1989; Hughes and Brezzi 1989; Ibrahimbegovic et al. 1990; Hughes et al. 1995; Geyer and Groenwold 2002). The stability analysis in Hughes et al. (1995a) proposed an upper bound of $\gamma \leq G$ and all the elements tested showed excellent results up to this limiting range. For different conditions (e.g. orthotropy or elastodynamics), a greater sensitivity to the value of γ is to be expected and a smaller value of γ (i.e. value of γ/G between 1/10000 and 1) appears to give more accurate solutions (Liu et al. 2000; Pimpinelli 2004; Long et al. 2006). For dynamic problems, for example, Hughes et al. (1995b) proposed γ/G=0.1.

Some researchers focus on assumed stress hybrid/mixed formulations combined with drilling degrees of freedom (Yunus 1988; Yunus and Saigal 1989; Ibrahimbegovic 1990, 1993, 1994; Ibrahimbegovic and Frey 1992, 1994; Aminpour 1992; Sze et al. 1992; Sze and Ghali 1993; Rengarajan and Aminpour 1995; Geyer and Groenwold 2002, 2003; Groenwold et al. 2002, 2004)

Felippa and Militello (1992), Felippa and Alexander (1992), and Felippa (2003) studied the formulation of 3-node 9-dof membrane elements with drilling freedoms in the context of parameterized variational principles. They constructed an element of this type using the extended free formulation EFF in the context of the assumed natural deviatoric strain (ANDES) formulation.

For the existing shell element in OpenSees, a fictitious stiffness k_θ is assigned to the drilling degree of freedom θ_z and it can be incorporated into the total potential energy in a penalty manner with k_θ being the penalty parameter.

2.2.5 Geometrically nonlinear flat shell elements

It has been common to use flat elements to model shell behaviour by superposition of a membrane element and a plate bending element. It is simple to formulate but they exclude coupling of stretching and bending within the element such coupling is a major contributor towards load carrying mechanism in shells and other curved members.

Development of reliable and efficient techniques for nonlinear analysis of shell structures still remains one of the most challenging topics in finite element research today. Many strategies have been proposed for geometrically nonlinear analysis of shells (Bathe 1996; Zienkiewicz and Taylor 2005). The most commonly used procedures are the total and updated Lagrangian formulations which can deal with the large displacements (rotations) and large strain, however, for a special class of nonlinear problems, namely, large displacements (rotations) but small strain, the corotational framework is currently proposed by decomposing the rigid body motion from the pure deformational parts. (Bathe and Bolourchi 1980; Bathe 1996; Zienkiewicz and Taylor 2005; Oliver and Onate 2005, Felippa and Haugen 2005). The underlying mechanics of the Total Lagrangian formulation and Updated Lagrangian formulation is identical. The difference is that the measures of stress and strain in which derivatives and integrals are taken with respect to the initial configuration for Total Lagrangian formulation and last converged equilibrium configuration for Updated Lagrangian formulation. In the formulation of incremental theories for nonlinear problems, we need to select appropriately conjugate stress and strain measures. By ‘conjugate’ we mean that the stress and strain measures selected with respect to a certain reference configuration can consistently represent the energy measures of the body at its current configuration. Any expressions with stress and strain measures that are not appropriately

conjugate are by themselves lacking a valid physical basis. One sure way of selecting appropriately conjugate stress and strain measures is to carry out the derivation of the principle of virtual displacements for the theory in question. If the current deformed configuration is selected as the reference, it will be shown that the Cauchy stress tensor and Cauchy infinitesimal strain tensor form a conjugate pair. For the Lagrangian description, which is referred to the known configuration, we may select the second Piola-Kirchhoff stress tensor and Green strain tensor as the conjugate pair. For corotational framework, the engineering strain and stress can be used. The detail of corotational framework can refer to reference (Felippa and Haugen 2005).

In this thesis, the geometrically nonlinear beam element (2D and 3D) follows existing corotational framework in OpenSees and the newly created geometrically nonlinear shell element follows the Total Lagrangian formulation.

2.3 Composite slabs in fire

In the previous section, methods to alleviate shear locking problems were reviewed among which MITC technique was used for the existing shell element in OpenSees. This element was modified to include geometrical nonlinearity and thermomechanical analysis capability. This modified shell element was then used to model composite concrete floor in fire. In this section experimental, numerical and analytical studies on the composite concrete floor were reviewed. The review of experimental research focuses on the Cardington tests which provide a background for Chapter 8 where two of the Cardington tests (the Restrained Beam test and Corner test) were used to validate the performance of developed thermomechanical analysis capability in OpenSees.

Typical steel framed composite construction consists of a composite floor with a trapezoidal or re-entrant steel deck, topped by a concrete slab with mesh reinforcement supported on a grillage of steel beams which act compositely with the slab as shown in Figure 2.1 (Bailey 2003). The traditional method of ensuring that a steel-framed building with composite slabs satisfies the regulatory requirements for fire resistance is to protect all exposed surfaces of columns and supporting beams with an insulating material. Code based methods for specifying the required thickness of the insulating material is extremely conservative since they ignore the inherent fire resistance of the structure. The development of structural fire design codes, e.g. BS5950: Part 8 and the Eurocodes provide a more solid scientific

foundation for the provision of fire resistance. However, these design codes were based on isolated structural members in Standard Fire Tests. The Standard Fire Test uses a standard temperature time curve to represent an average compartment gas temperature which varies over the duration of the fire which does not incorporate any additional variables, apart from fire duration or the fires growth rate. This makes it unable to represent a real natural fire and the most severe fire conditions. Although there are disadvantages and limitations of assuming the standard temperature curve, the simplest and most common performance-based approaches have been developed based on the results and observation from standard fire tests. In addition it is generally recognized that the whole-frame structural behaviour in fire cannot be represented by a test on an individual element.

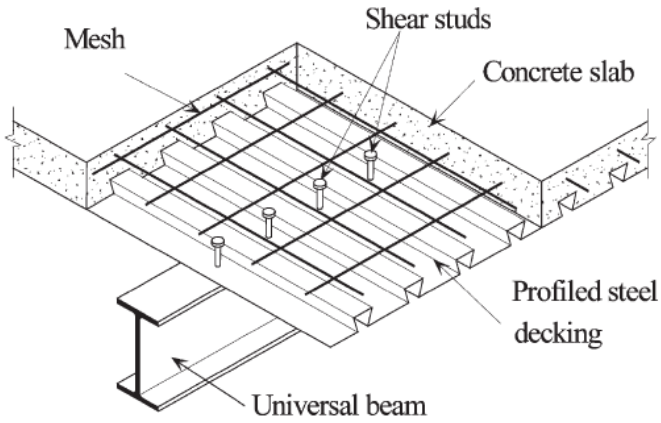


Figure 2.1: Typical composite floor (Bailey 2003)

Membrane action of concrete slabs is not a new topic and significant work on membrane action at both small and large displacement has previously been conducted (Wood 1961; Park 1964; Sawczuk 1965; Taylor 1965; Kemp 1967; Hayes 1968; Brotchie 1971). The membrane action is caused by the development of in-plane or membrane forces within the depth of concrete floor slabs. At small vertical deflections, compressive membrane forces will occur provided the slab perimeter is vertically supported and restrained horizontally. The development of compressive action significantly increases the load-carrying capacity of the slab compared with pure flexural behaviour. Practical application of membrane action of concrete slabs has been limited to the behaviour at small displacements due to serviceability restrictions on acceptable vertical deflections. Both theoretical and experimental research into membrane action of concrete slabs at large displacements is considered to have no application in the design of buildings. However, in an accidental design state, such as during a fire, large displacements in the structure are acceptable provided overall structural collapse

or breach of compartmentation is avoided. In large displacements, tensile membrane action can occur in the slab supported vertically and horizontally at the perimeter. It is also possible for tensile membrane action to occur in two-way spanning slabs that are vertically supported and horizontally unrestrained. In this case, the slab resists the load by tensile membrane action in the middle of the slab and by the formation of a "compressive ring" within the boundary of the slab.

A direct way to study the behaviour of structures in fire is to carry out fire tests on full-scale structures and small-scale structural members. The observations from the experimental study have led to progress in two main areas. On one hand, numerical modelling of the Cardington tests as well as typical composite concrete slabs by a number of authors (Izzuddin 1991; Bailey 1995; Huang et al. 1999a; Elghazouli and Izzuddin 2000; Franssen 2003) produced a detailed understanding of the structural behaviour of steel-framed composite floors in fire. Accompanied with the numerical modelling of structures in fire came the progressive development of the specialist programs such as ADAPTIC, VULCAN and SAFIR. The theory of forming plate/shell element and methods to alleviate shear locking problem mentioned in previous section had been applied comprehensively in these finite element programs to numerically analyse the behaviour of structures in fire. On the other hand, some progress has been made in extending the understanding developed from modelling into design guidance (Bailey and Toh 2007; Cameron 2003). The experimental study of Cardington tests will be described briefly in the next section to provide a background for the validation of developed OpenSees presented in Chapter 8 followed by a detailed review of numerical and mathematical studies.

2.3.1 Experimental research

It is only in relatively recent years since the Broadgate Phase 8 fire in London and the subsequent Cardington fire tests that researchers have fully investigated and understood the behaviour of whole composite steel-framed concrete structures in fire. The Cardington test results (Bravery 1993; Kirby 1997; O'Connor 1998; Bailey 1999; Huang et al. 2000c; Usmani 2000) have confirmed that steel members in real multi-story buildings have significantly greater fire resistance than isolated members in the standard fire test. It has been found that the concrete slab appears to play an important role in preventing structural collapse. Observations from these tests suggested that the improved performance of composite floor system is due to the ability of reinforced composite slabs to bridge over the

supporting fire-damaged steel beams and transfer load, by membrane action, to the undamaged parts of the structures (Bailey and Moore 2000).

The layout and results of the Cardington tests are presented in the following sections to provide background knowledge for the validation of developed OpenSees presented in Chapter 8. The eight-storey steel-framed test building was constructed by the Building Research Establishment at its Cardington Laboratory near Bedford. The building was designed and constructed to resemble a typical modern city centre office development. On plan, the building covered an area of 21m ×45m with an overall height of 33m. A plan of the building is shown in Figure 2.2 and 2.3.

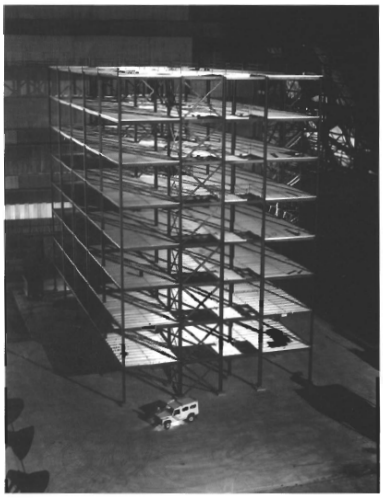


Figure 2.2: General view of Cardington test building (British Steel, 1999)

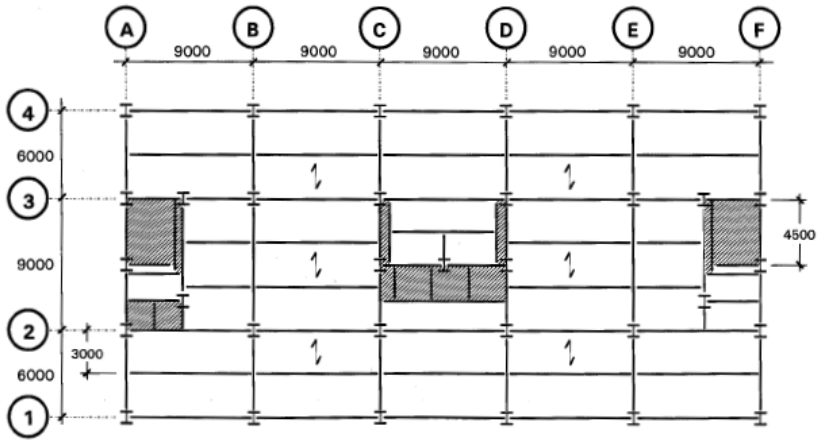


Figure 2.3: Floor layout of Cardington test (BS, 1999)

Two programmes of tests (BS/ECSC and BRE) took place between Jan. 1995 and Sept. 1996. In all, 6 major fire tests were carried out. These are summarised in Table 2.4 and their locations are shown in Figure 2.4. The details of each Cardington test are presented as follows.

Test date	Sponsor	Description	Location	Floor area (m^2)	Maximum vertical displacement (mm)	Maximum temperature (^oC)
Jan 95	British Steel	Restrained beam	7^{th} floor	24	232	887
Apr 95	British Steel	Plane frame	4^{th} floor	53	300	700
Oct 95	BRE	Corner	2^{nd} floor	54	269	903
Nov 95	British Steel	Corner	1^{st} floor	76	428	935
Apr 96	BRE	Large compartment	2^{nd} floor	340	557	691
Sep 96	British Steel	Office compartment	1^{st} floor	136	640	1150
Jan 03	European	Connection	4^{th} floor	80	919	1000

Table 2.4: Summary of Cardington tests

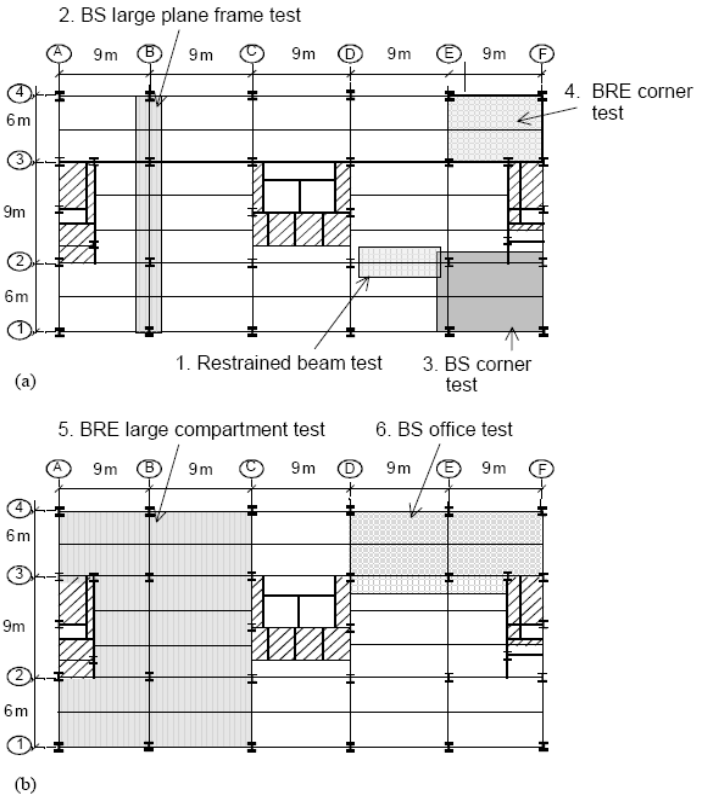


Figure 2.4: Locations of six Cardington fire tests: (a) test 1-4; (b) test 5and 6 (BRE, 2005)

2.3.1.1 BS restrained beam test

The restrained beam test was carried out on the 7th floor of the steel framed building at Cardington shown in Figure 2.5. A purpose built gas fired furnace, 8.0m long and 3.0m wide was designed and constructed to the underside of the composite floor (on the sixth floor) to heat a 305×165×40UB panning between two columns (254×254×89UC). The beam was heated over the middle 8.0m of its 9.0m length and kept the connections as near as possible at ambient temperature. The objective of the test was to provide validation for structural models on a single element with realistic boundary conditions.

The beam was heated between 3-10^{o}C per minute until temperature within the range of 800-900^{o}C was recorded through the profile of the section. The deformation shape of the restrained beam after fire is shown in Figure 2.6. Figure 2.7 illustrates the time-temperature-displacement relationship for the restrained beam test. At the beam's maximum temperature (887^{o}C in the lower flange) the mid-span deflection was 232mm. Even when the test was terminated, when the strength of the structural steel is less than 10% of its yield strength at

ambient temperature, the "runaway" displacement (instability) was not reached. Once the beam had cooled back to ambient temperature the mid-span deflection recovered to 113mm.

Figure 2.5: British Steel restrained beam test (BRE, 2005)

Figure 2.6: Post-test deformation of restrained beam test (BRE, 2005)

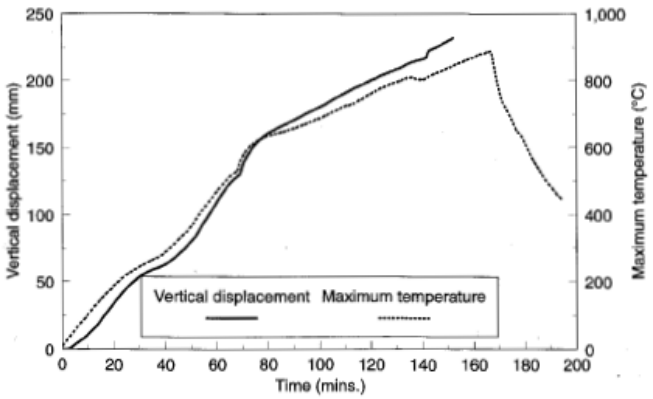


Figure 2.7: Maximum vertical displacement and temperature in the beam (BS, 1999)

2.3.1.2 BS plane frame test

The objective of the plane frame test shown in Figure 2.8 was to extend the model validation to a complete sub-frame consisting of a slice across the full width of the building incorporating two partially protected internal columns and two partially protected perimeter columns within the heated area. The test was to investigate the behaviour of the structure in the proximity of the connection and also the behaviour of the connections themselves.

The rate of vertical displacement of the central 9m steel beam increased rapidly between approximately 110 and 125 minutes. This was caused by the exposed areas of the internal columns squashing by approximately 180mm. The measured maximum displacement and temperature is shown in Figure 2.9.

Figure 2.8: Plane frame test (BRE, 2005)

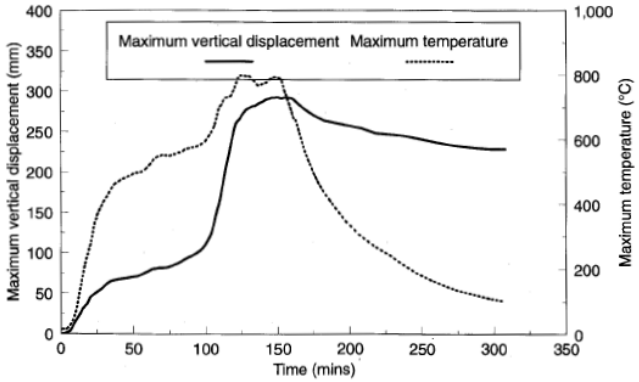


Figure 2.9: Maximum vertical displacement and temperature in the beam (BS, 1999)

2.3.1.3 BRE corner test

This test was carried out in a 9m×6m compartment on the second floor between gridlines E to F and 3 to 4 (see Figure 2.10 and 2.11). The maximum recorded value of 269mm for the slab displacement occurred in the centre of the compartment after 130 minutes (Figure 2.12). A permanent displacement of 160mm was recorded.

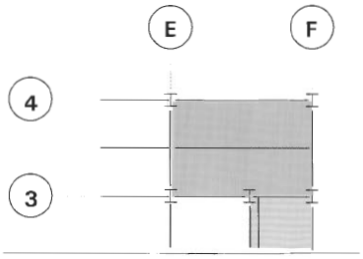


Figure 2.10: Schematic of BRE corner test (BS, 1999)

Figure 2.11: General arrangement of BRE corner test (BRE, 2005)

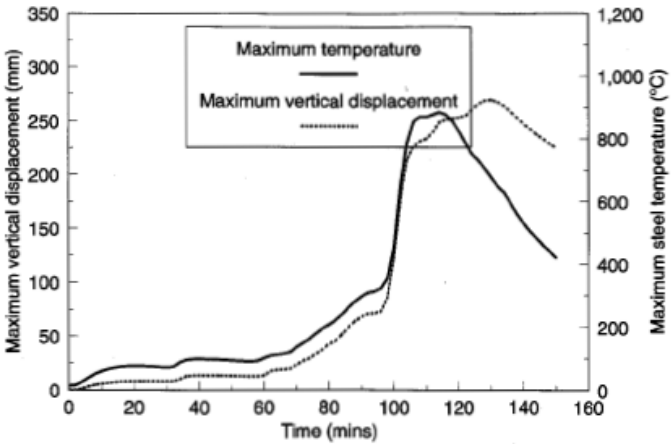


Figure 2.12: Maximum vertical displacement and temperature in the beam (BS, 1999)

2.3.1.4 BS corner test

A compartment 10m wide by 7.6m deep with a floor area of approximately 80m^2 was built on the first floor of the Cardington steel building in one corner shown in Figure 2.13-2.15. The objective of the test was to investigate the behaviour of a complete floor system and the role of bridging/membrane action of the floor in providing alternative load paths as the supporting steel frame reaches high temperatures.

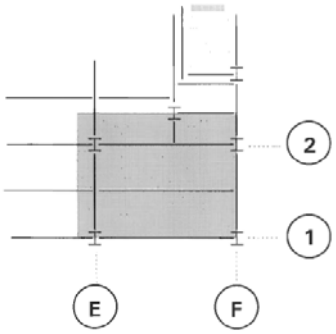


Figure 2.13: Schematic of BS corner test (BS, 1999)

The deformations of the frame after fire are shown in Figure 2.16 and 2.17. The maximum vertical displacement of 428mm occurred at the centre of the secondary beam spanning between gridlines E and F, located between gridlines 1 and 2. This recovered to a permanent displacement of 296mm after the cooling. The maximum recorded temperature of the secondary beam was 935^{o}C. (as shown in Figure 2.18)

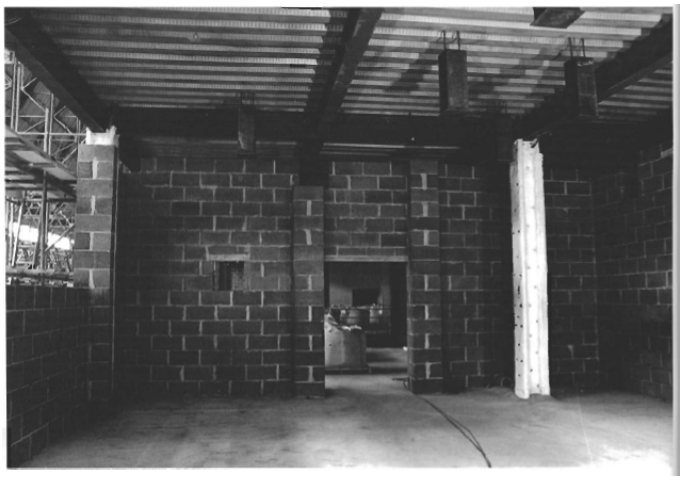


Figure 2.14: Internal view of the test compartment (BS, 1999)

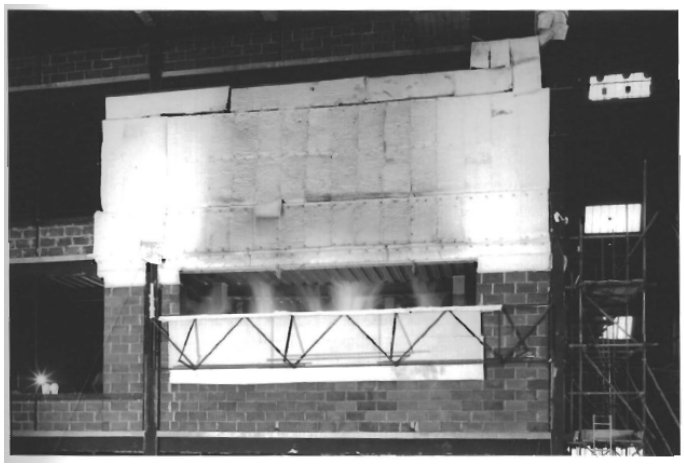


Figure 2.15: Front view of compartment during early fire development stage (BS, 1999)

Figure 2.16: Internal view of Post-test deformation of BS corner test (BS, 1999)

Figure 2.17: Front view of post-test deformation of BS corner test

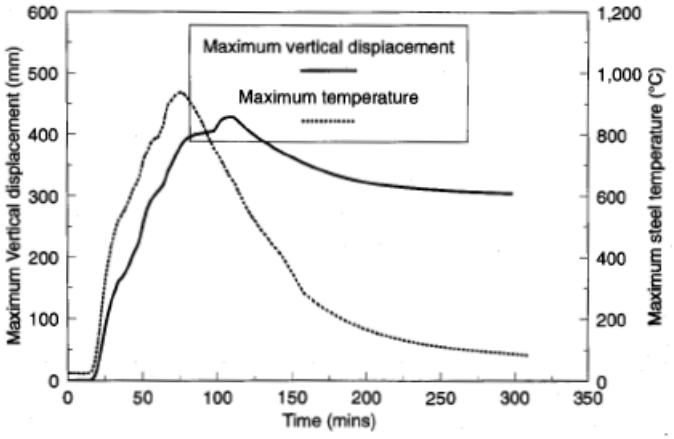


Figure 2.18: Maximum vertical displacement and temperature in the beam (BS, 1999)

2.3.1.5 BRE large compartment test

This test was carried out on the second floor and covered an area of approximately 340m^2 extending across the full width of the building between gridline A-C and gridline 1-4 (as shown in Figure 2.19). The time-temperature-deflection relationship is illustrated in Figure 2.20. The maximum slab displacement reached a value of 557mm which was recorded halfway between gridlines 2 and 3 and B and C. This recovered to a value of 481mm when the structure cooled.

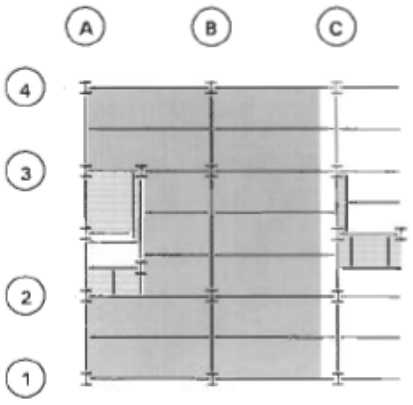


Figure 2.19: Schematic of BRE large compartment fire test (BS, 1999)

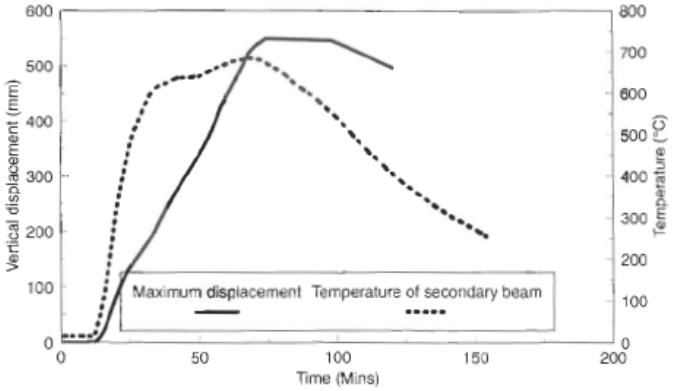


Figure 2.20: Maximum vertical displacement and temperature in the beam (BS, 1999)

2.3.1.6 BS office compartment test

This test considered of a compartment up to 18m wide and 10m deep on the first floor of the steel framed building. The aim of this test was to demonstrate that the type of structural behaviour observed in the earlier tests would also occur when the building was subjected to a more realistic fire scenario-office fire. Figure 2.21 showed the compartment fitted out with office furniture, computers and filing systems typical of a modern office. A maximum vertical displacement of 640mm was recorded after 62 minutes (Figure 2.22).

Figure 2.21: Internal view of the compartment (BS, 1999)

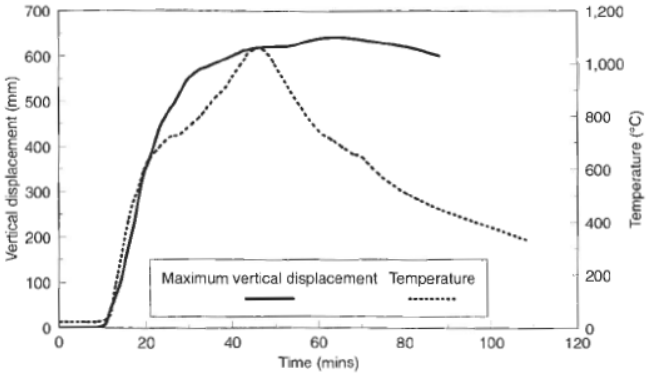


Figure 2.22: Maximum vertical displacement and temperature in the beam (BS, 1999)

Experimental observations from the Cardington tests provide significant conclusions on the structural behaviour of composite steel frames, briefly listed as follows:

1. The composite floor slab played an important role in enhancing the load-carrying capacity of the overall structure. The contribution of the composite slab to the fire resistance of the structure arose from the tensile membrane action in the middle and compressive membrane action near the perimeter of the slabs.

2. Tensile membrane action allows load from the weakened fire affected part of the structure to be transferred to the cooler and stronger surrounding areas via the steel reinforcement in the slabs acting as a net or membrane.

3. Restrained thermal expansion and thermal bowing of structural members dominate the structural behaviour in fire. Material degradation and loading only become important near impeding "runaway" failure which was not reached in the tests.

4. Local buckling typically occurred in the lower flange of the heated steel beams in the proximity of the connections at an early stage of the fire (between 100^{o}C and 200^{o}C). Therefore, conservatively, pinned connections should be assumed in fire design.

5. The whole structure showed no signs of collapse even with the unprotected steel beam temperatures in excess of 1000^{o}C.

6. The vertical residual displacement of the heated beams and composite floor slabs were large and in some cases exceed 500mm.

In addition to the Cardington tests, many other full-scale (Lin et al. 1989; Hamerlinck et al. 1990; Both and Haar 1994) and small-scale fire tests (Bailey and Toh 2007; Foster 2006) were conducted on reinforced concrete slabs. The tests were aimed to investigate the membrane action in the slab at large displacements under fire conditions as well as the failure mechanism of the concrete slab considering cracking and crushing of the concrete and rupture of the reinforcement. Furthermore, the results from the tests can be provided for the validation of the numerical and analytical modelling of the behaviour of composite slabs in fire.

2.3.2 Numerical simulation

Following the observations from Cardington tests and other laboratory tests, numerous numerical models were developed in parallel with development of special-purpose finite element program such as ADAPTIC (Izzuddin 1991), VULCAN (Bailey 1995) and SAFIR (Franssen et al. 2000; Franssen 2003) developed by researchers at Imperial College, the University of Sheffield and the University of Liege (Belgium), respectively. VULCAN, a specialist finite element code, has been progressively developed (Najjar 1994; Bailey 1995; Bailey et al. 1996; Najjar and Burgess 1996; Huang et al. 1999a). The historical development of numerical simulation of composite slabs experienced the following issues (details will be presented later).

1. Shell elements were used to model the composite slab in place of a grillage of beam elements. Single element was employed to model both the upper solid part of the slab and the lower ribbed part which replaced the separated modelling of these two parts connected by some kind of rigid link or spring link.

2. The material model for the concrete was developed from the temperature independent elastic material to temperature dependent material considering the orthotropy of composite slab, cracking, crushing and spalling of the concrete, and other thermal properties at elevated temperature such as creep, shrinkage.

One of the first models of the Cardington test frame (Wang et al. 1995) was a 2D model that produced useful conclusions in the critical role of columns. A further theoretical study by Wang (1996) indicated the importance of tensile membrane action in maintaining the robustness of composite slabs. Rose et al. (1997, 1998) published one of the first 3D models of the Cardington tests which showed a good agreement between predicted and test deflections for the restrained beam test, corner test and plane frame test. The Reissner-Mindlin plate theory was employed by Bailey (1995) to model the behaviour of composite building frames in fire. An isotropic elastic material model was assumed for concrete which was temperature independent. Based on Bailey's work Huang et al. (1999a) proposed a nonlinear layered finite element procedure for predicting the structural response of reinforced concrete slabs subjected to fire. The complex features of structural behaviour in fire conditions, such as thermal expansion, cracking or crushing, and temperature-dependent material properties were considered excluding the spalling of concrete. Huang et al. (2000a) extended the layered procedure (Huang et al. 1999a) to include the orthotropic properties of composite slabs by introducing an effective stiffness approach (Polak 1996) (see Figure 2.23). The basic idea was to use the nominal thickness of the composite slab as the thickness of the slab element, and to use an effective stiffness factor to modify the material stiffness matrices of plain concrete in order to consider the orthotropic properties of the ribbed slab.

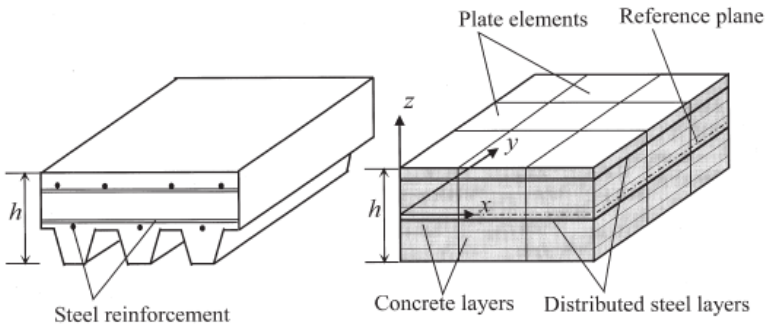


Figure 2.23: Model of composite slabs with uniform thickness (Huang et al. 2000a)

A three-dimensional, nonlinear finite element procedure for modelling composite steel-framed buildings in fire was presented by Huang et al. (2000b). The proposed model shown in Figure 2.24 was an assembly of beam-column, spring, and layered flat shell elements. The spring element of zero length (Huang et al. 1999b) was used to model the connection of steel beam and reinforced concrete slab. In this study, the reference plane was assumed to coincide with the mid-surface of the concrete slab element and remained fixed throughout the analysis. The reinforcing steel mesh was modelled by an equivalent smeared steel layer with stiffness only in the direction of the reinforcement.

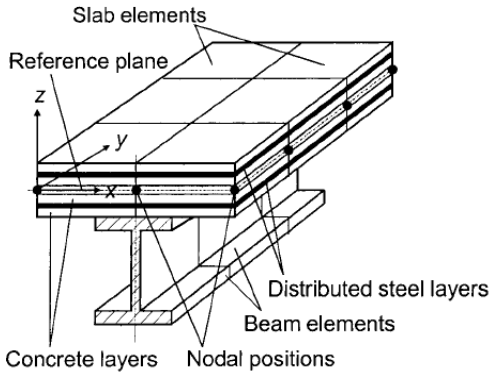
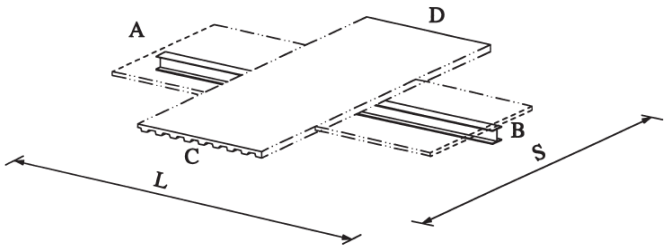
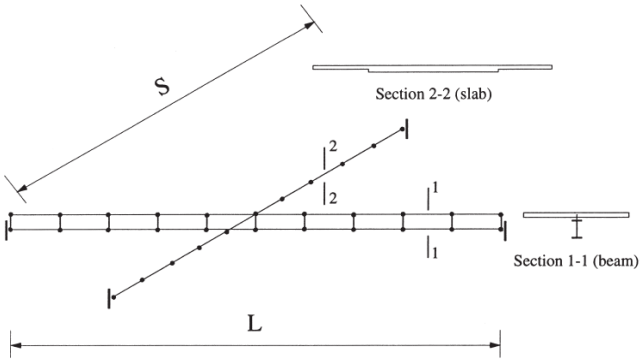


Figure 2.24: Composite slabs consisting of beam and slab elements (Huang et al 2000b)

Elghazouli and Izzuddin (2000) studied the behaviour of an idealised composite beam-slab systems under fire conditions using the program ADAPTIC. The system consisted of three members including a longitudinal steel beam, the solid part of the slab parallel to and above the steel beam as well as the ribbed slab in the transverse direction shown in Figure 2.25. All the members were modelled by beam-column elements and connected by rigid links. The transverse ribbed slab was idealised into one T-shape section. Kinematic bilinear and multi-linear stress-strain relationships were employed for steel and concrete respectively. The temperature-dependent properties of the material were defined by a reduction factor following a trilinear curve over the temperature domain. Nonlinear and non-monotonic temperature distribution can be applied across the section and along the length of a modelled member. It was indicated that thermal expansion may have beneficial and detrimental consequences on the performance, depending on the particular structural configuration and adopted failure criteria. In restrained systems, bucking in the beam due to thermal expansion can have adverse effects on the mechanical strains in the slab reinforcement and beneficial effects of thermal expansion lay in introducing the pre-compression and hence reducing the tensile forces in the beam connections to the surrounding structure.



(a) Composite beam-slab system



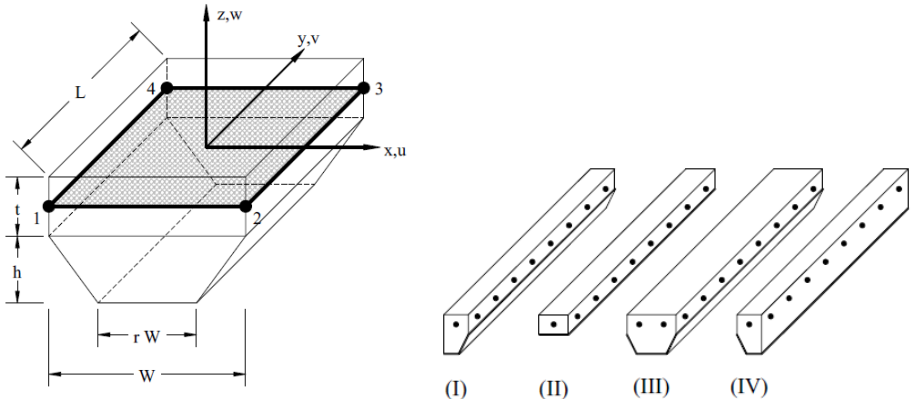
(b) Outline of finite element model

Figure 2.25: Layout of Composite beam-slab system (Elghazouli and Izzuddin 2000)

After the analysis of the simplified single beam-slab system, numerical simulation of two Cardington tests, restrained beam and corner test, were undertaken using ADAPTIC by Elghazouli el at. (2000) and Elghazouli and Izzuddin (2001). A grillage representation of the composite floor was used in which all slab and beam components were represented by the same cubic elasto-plastic beam-column elements (Elghazouli and Izzuddin 2000). The results showed that, for both the tests, thermal expansion caused an early buckling of the floor system due to compressive action considering full restraint of the beam from the surrounding cool structure which led to a rapid increase in deformation. As the temperature increased, the response became dominated by tensile membrane action within the composite slab which enhanced the load-carrying capacity.

Izzuddin et al. (2004) pointed out that previous studies had focused on solving the difficulty arising from modelling the geometric orthotropy of composite slabs by 2D shell elements and classified them into two main approaches. The first approach employed geometric simplification, where uniform thickness shell elements are used with an effective stiffness approach to approximate the influence of the geometric orthotropy (Huang et al. 2000a). The second approach employed dimensional simplification, where a grillage 1D element was

used to approximate the 2D bending and membrane response (Elghazouli el at. 2000; Elghazouli and Izzuddin 2001). In contrast to the previous grillage beam-column elements, Izzuddin et al. (2004) introduced a new flat shell element for ribbed composite slabs accounting for geometric and material nonlinearities as shown in Figure 2.26. The proposed element can deal with the geometric orthotropy of composite floor by means of incorporating two additional displacement fields corresponding to stretching and shear modes in the rib region. The slab element was formulated in a co-rotational framework (Izzuddin 2002). A new robust material model for concrete was employed which considered the tensile cracking and compressive nonlinearity as well as the effects of elevated temperature. The extensive verification of the proposed nonlinear analysis method was provided in the companion paper (Elghazouli and Izzuddin 2004).



(a) Geometric configuration (b) Variants of composite shell element

Figure 2.26: Layout of composite slabs (Izzuddin et al. 2004)

Macorini et al. (2006) presented a new finite element model for composite beam which allows a more realistic evaluation of the effects due to the planar shear strains in the slab. One-dimensional beam elements were used to model the steel beam and two-dimensional shell elements (Izzuddin et al. 2004) for the concrete slab, where the two types of element were interconnected by means of rigid links and special-purpose spring elements (Izzuddin 2003). The concrete was modelled taking into account creep, shrinkage and the nonlinear cracking behaviour. Figure 2.27 shows the finite element model of the composite slab.

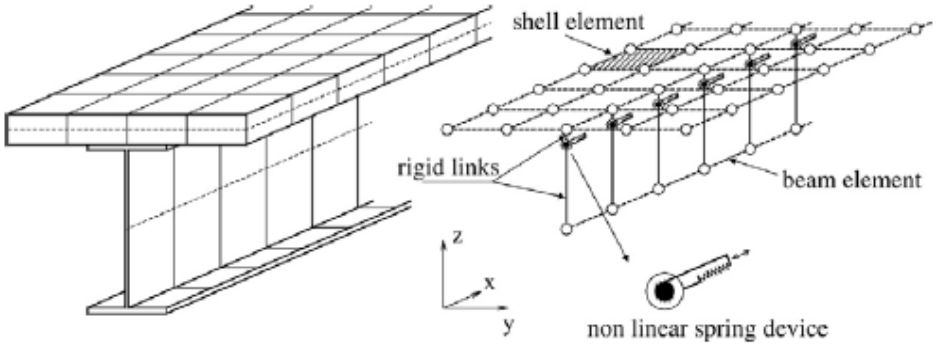


Figure 2.27: Finite element representation of the composite beam (Macorini et al. 2006)

The effects of geometric nonlinearity were ignored in the early analyses of concrete slabs subjected to fire. For accurate determination of large displacement and membrane action of concrete slabs in fire, the layered procedures previously developed (Huang et al. 1999; Huang et al. 2000b) were extended to model reinforced concrete slabs in fire including geometric nonlinearity based on a total Lagrangian approach (Huang et al. 2003). A quadrilateral 9-node higher-order isoparametric element developed by Bathe (1996) was used to model the concrete slab in place of the previous 4-node geometrically linear element (Huang et al. 2000a). The influence of different thermal expansions, tensile membrane action and various temperature distributions across the thickness of slabs were investigated. In this study 13 layers were used to represent adequately the temperature distribution through the slab section. This advanced geometrically nonlinear model was later used to model composite concrete floors by Huang et al. (2004).

Yu et al. (2008) presented a new model for orthotropic slabs in fire considering the non-uniform temperature distribution in the horizontal plane of the slab element shown in Figure 2.28. The slab element was modelled by assembly of a 9-node solid slab element which represented the upper solid part of slab and 3-node beam element which represented the lower ribbed part. It was assumed that the reference axis of the beam element coincided with the mid-surface of the slab element. In this model, different from the individual rib model (Lim 2003; Lim et al. 2004), the beam element was used to represent a group of ribs of the composite slab and hence the equivalent width of its cross-section is determined according to the dimensions of the solid slab element and the cross-section of the ribbed profile. Different temperatures can be ascribed to the adjacent Gauss integration points in the horizontal plane of the slab to consider the non-uniform temperature distribution due to the fact that the presence of the ribs made the temperature higher in the thinner portion than in the thicker ribbed portion.

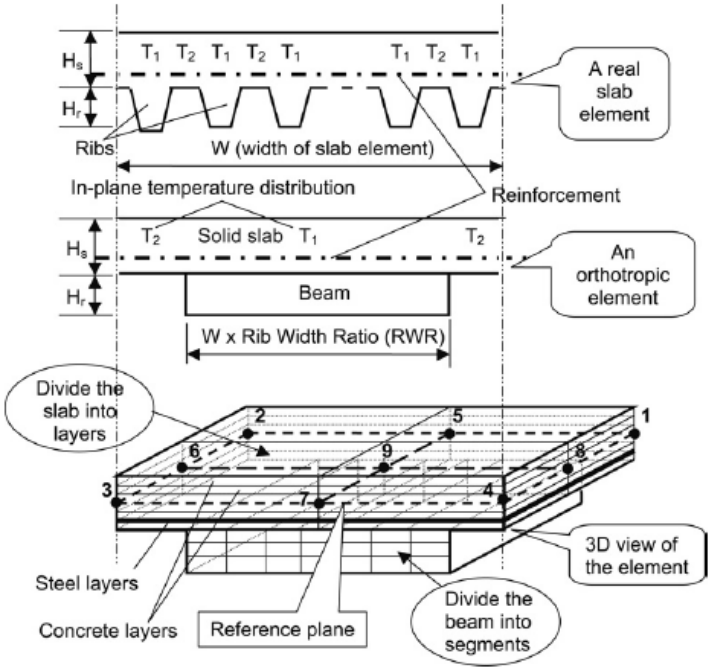


Figure 2.28: An orthotropic slab element model (Yu et al. 2008)

Recently, Huang (2010) extended the previous layered procedure (Huang et al. 2003) to take into account the effects of concrete spalling on both thermal and structural behaviour of concrete slabs in fire. The spalling behaviour was modelled by making certain concrete layers void (with zero mechanical strength, stiffness and thermal properties).

SAFIR is a special purpose finite element program developed at the University of Liege, Belgium, for analysing the behaviour of structures in fire. SAFIR possesses a variety of finite elements such as beam, truss, solid and shell elements for performing nonlinear 2D and 3D analysis of steel, concrete and composite structures in fire. Recently a four-node shell element (Talamona and Franssen 2000) was developed to model structures in fire which supersedes an earlier triangular shell element. The properties of the shell element were based on the Discrete Kirchhoff Quadrilateral theory (DKQ). There were four Gauss integration points on the surface of the shell element and different number of integration points ranging from 2 to 10 through the thickness of the section can be defined by the user.

The research team in the University of Edinburgh used commercial package ABAQUS to model the Cardington tests and obtain an extensive and in-depth understanding of the structural behaviour of the composite steel-framed buildings under fire conditions (Sanad et al. 2000; Gillie et al. 2001, 2002; Lamont et al. 2004). A relatively simple numerical model

of the first Cardington test using ABAQUS (restrained beam test) was presented in (Sanad et al. 2000a). The composite slab, in the transverse direction, was modelled by a grillage of beam elements representing the ribbed slab. In the longitudinal direction parallel to the secondary beam, a composite beam element was used to model the heated secondary beam. The temperature dependent material properties were defined according to Eurocodes. Different heating regimes including slab mean temperature and thermal gradient were applied on the proposed model to further investigate the structural behaviour in fire (Sanad et al. 2000b, c). In the later paper, the concrete slab was modelled using a specially developed FEAST program (Finite Element Analysis of Shells at High Temperature) developed by Gillie (Gillie 2000; Gillie et al. 2001a). FEAST can model slabs of arbitrary geometry and nonlinear material by a depth integration technique. One limit is that the uniaxial material behaviour was assumed for the concrete. Using ABAQUS and FEAST program, two Cardington tests including restrained beam test and corner test were modelled (Gillie et al. 2001b, 2002) and the results indicated that the response of the structure in fire was dominated by the effects of thermal expansion and thermal bowing rather that the effect of material degradation and loading. Lamont et al. (Lamont et al. 2004; Lamont and Usmani 2003) modelled the structural behaviour of a small generic composite floor frame in two different single floor compartment fires. One was named “short-hot” fire with high temperature but a relatively short post-flashover duration and the other “long-cool” fire represented lower maximum temperatures but a long post-flashover duration. The results showed contrasting structural behaviour of the two fire scenarios due to the different temperature distribution in the concrete slab and steel beam. In contrast to the traditional thinking about fire severity in terms of area under the temperature-time curve and the equivalent time concept, it was concluded that the most detrimental fire in terms of the structure response seemed to be the short-hot fire.

The results from the aforementioned numerical analyses have been invaluable in extending the knowledge of how composite slabs and the supporting steel beam grillage behave in fire. The researchers, however, have found difficulties in modelling the cracking behaviour of the concrete slab together with an accurate prediction of the fracture of the reinforcement. The cracking of the concrete will accelerate the heat transfer in the concrete and always leads to high temperature in the reinforcement which causes its fracture due to the degraded properties at high temperature. The fracture of the reinforcement will dependent on the location and width of the cracking of concrete. In this thesis these effects are not considered

which are expected to be the future work. Difficulty has also been found in simplifying the models such that practical design procedures can be produced.

2.3.3 Analytical methods

The numerical analysis is considered to be robust and powerful but complicated and case dependent which is not welcome for the engineers who need relatively simple design method (analytical methods in this section) with sufficient accuracy. A number of researchers have proposed analytical methods for calculating the tensile membrane capacity of a composite floor slab in fire. Bailey et al. have developed a design method (Bailey and Moore 2000; Bailey 2001, 2003, 2004 Newman et al. 2006) based on a series of small-scale test on the reinforced concrete slabs (Bailey and Toh 2007). Bailey and Moore (2000) provided an accurate estimate of the load-carrying capacity by dividing the floor slab of a building into a number of slab panels. Each slab panel was designed in fire using membrane action by considering the strength at elevated temperature. The method assumed a yield line type failure mechanism and the maximum allowable displacement is based on an estimate of the strains in the reinforcement together with the thermal curvature of the slab. One limitation of this method is the use of isotropic reinforcement which is acceptable for square slabs but is uneconomical for rectangular slabs. Another limitation lies in that conservative assumption of no horizontal restraint from the surrounding structure. This original method was extended to allow orthotropic reinforcement to be used (Bailey 2001, 2003). The previous design method was extended (Bailey 2004) to incorporate the membrane action of the slab and beam system action compositely and allowed variation in the deflection forms of the slab and changing yield line patterns as the slab and beam system is heated in a fire. Bailey and Toh (2007) presented the latest development of the simple design method used to predict the membrane action of unrestrained composite floors under fire conditions. The development included a more accurate estimate of the in-plane stress distribution and a limit on the load-capacity due to the crushing of the concrete in the corners of the slab.

Cameron and Usmani (2005) discarded both the assumption of yield line theory and no horizontal restraint in the Bailey's method to develop a more realistic design method. The method considered the temperature distribution through the slab and the resultant thermal strains were decomposed in terms of thermal expansion and thermal curvature (Usmani and Cameron 2004). The load capacity was then calculated based on an assumed failure mechanism and the remaining strength of the steel reinforcement using an energy method.

The failure mechanism referred to a limiting value for the mechanical strain based on the ductility of the reinforcing bars.

Li et al. (2007) proposed a new analytical method based on the force and moment equilibrium requirement on slabs in fire. The slab in fire was divided into five parts including four rigid plates near the edges and an elliptic paraboloid in the centre.

The observations from Cardington tests provide a better understanding to behaviour of structures in fire which stimulate a large amount of numerical simulation followed by simplified design method. So far the numerical models of composite slabs in fire can be classified into two groups. One is to use grillage consisting of beam and flat plate/shell elements to model the concrete ribs and concrete flat slab respectively and interconnects them with link elements. The other way is to use a single plate/shell element and orthotropy of the composite slab is considered in the stiffness matrix. Compared with this single shell element, the grillage method is straightforward and easy to combine different advanced beam and shell elements to model the composite slab flexibly. Therefore this grillage method is accepted in this thesis to model the composite slabs in the Cardington tests to validate the performance of the developed OpenSees presented in Chapter 8.

2.4 Performance based design

Building codes worldwide are moving from prescriptive to performance-based approaches. Performance based codes establish fire safety objectives and leave the means for achieving those objectives to the designer. One of the main advantages of performance based design is that the most recent models and fire research can be used by practising engineers inevitably leading to innovative and cost effective design. Performance based design has been documented in the literature extensively over the past 10 years.

The performance-based codes in UK include a general guidance applying to most building work (Approved Document B and other government guidance to support legislative requirements), a high-level guidance on fire safety engineered solutions for larger and more complex buildings (BS 7974) as well as advanced guidance (BS 9999) essentially introducing a middle route between ADB (Approved Document Part B Fire Safety) and BS 7974. BS 9999 is a new standard which uses risk profiles instead of prescribed methods as used in ADB.

The Eurocodes provide a collection of the most recent methodologies for design. The design of concrete, steel and composite structures in fire are defined in Eurocode 2, Eurocode 3 and Eurocode 4 respectively. Each Eurocode is supplemented by a National Application Document (NAD) appropriate to the country. All Eurocodes are presented in a limit state format where partial safety factors are used to modify loads and material strengths. EC3 and EC4 are very similar to BS 5950 Part 8 although some of the terminology differs. EC3 and EC4 Parts 1.2 and BS 5950 Part 8 are only concerned with calculating the fire resistance of steel or composite sections. Three levels of calculation are described in EC3 and EC4, i.e. tabular method, simple calculation and advanced calculation. Tabular methods are used for direct design based on parameters such as loading, geometry and reinforcement. They relate to most common designs. Simple calculations are based on principles such as plastic analysis taking into account reduction in material strength with temperature. These are more accurate than tabular methods. Advanced calculation methods relate to computer analyses and are not used in general design.

The performance-based design idea prompts the development of numerical analysis of structures in fire. The framework OpenSees was chosen by the research team in the University of Edinburgh to develop a three-dimensional thermomechanical analysis capability of structures. This 3D model in OpenSees includes a shell element representing the flat concrete flat and 3D beam/column elements representing the concrete rib and beam/columns. These components are interconnected by rigid link element. A new geometrically nonlinear shell element was created by modifying an existing linear MITC element with drilling degrees in OpenSees. The theories of the beam and shell element are presented in Chapter 4 and Chapter 6 respectively with its application in OpenSees presented in Chapter 5 and Chapter 7 respectively. Before introducing the theory and application of elements in OpenSees, the existing class hierarchy of OpenSees framework is presented in Chapter 3 which is of help to better understand the modification made in OpenSees. Two Cardington tests (restrained beam test and BS corner test) were modelled in Chapter 8 to validate the performance of the developed OpenSees.

Chapter 3

OpenSees Class Hierarchy and Workflow

3.1 Introduction

From this chapter, the modifications of OpenSees to deal with thermomechanical analysis of structures and its validation will be presented in the following chapters. The modifications in OpenSees involve element, material and load classes as well as interfaces between them which are based on its existing class hierarchy. These classes can be classified into four groups such as Modelling classes, Finite Element Model classes, Analysis classes and Numerical classes. The main modifications of OpenSees framework are based on the existing Finite Element Model classes including new element and load classes. A brief introduction of these four classes groups will be presented in this chapters and the detail can reference to reference (McKenna 1997). In addition, the existing material classes in OpenSees do not include temperature effects which will be added according to Eurocodes presented in Chapter 5. The stress-strain relation of the existing classes of uniaxial and multi-dimensional steel and concrete material will be presented in this chapter to provide background knowledge for the inclusion of temperature dependent properties in Chapter 5.

3.2 Introduction to OpenSees

OpenSees is an open source object oriented software framework developed at University of California, Berkeley and supported by PEER and Nees (OpenSees website). OpenSees has so far been focussed on providing an advanced computational tool for analysing the non-linear response of structural frames subjected to seismic excitations. Given that OpenSees is open source (available for free download at opensees.berkeley.edu) and has been available for best part of this decade it has spawned a rapidly growing community of users as well as developers who have added to it's capabilities over this period. For instance it has significant geotechnical modelling capabilities developed by this community so that the seismic response analyses can include full soil structure interaction if required. It also has a structural reliability and sensitivity analysis capability offering many reliability calculation tools. Furthermore it has an HPC or parallel version for solving large problems on high-performance computing hardware. In addition to it's availability as an analysis tool OpenSees is also the software platform of choice for the US Nees network that enables earthquake engineers to organise and share data, participate in remote experiments and perform hybrid simulations. It therefore represents the largest community of this kind in structural engineering and has the potential to bring together the best structural engineering

computational modelling capabilities under one platform accessible to all facilitating new collaborations across geographical boundaries to solve ever more challenging problems.

There are three types of problems encountered in engineering (structural) analysis: steady-state (or quasi-static) problems, transient (dynamic) problems, and eigenvalue (stability) problems. Each of these can be further classified as being either linear or nonlinear. For linear problems, the basic steps in a finite element analysis are:

1. Discretisation of the domain into nodes and elements

2. Formulation of the element matrices and vectors

3. Formulation of the system of equations

4. Incorporation of the boundary conditions

5. Solution of the system of equations for the nodal degrees-of-freedom (DOF)

6. Computation of response within each element given values for the DOFs

The differences between the three types of problems are in the equilibrium equations that are formed. For each type of problem, there are many ways to solve these equations. When dealing with nonlinear problems, the system of equations is nonlinear, and iteration regime is typically employed, e.g. Newton-Raphson method to achieve a solution.

3.3 Class Hierarchy

The classes defined in OpenSees can be grouped into four broad categories (McKenna 1997):

1. Modeling classes: classes used to create the finite element model for a given problem

2. Finite Element Model classes: classes used to describe the finite element model and to store the results of an analysis on this model

3. Analysis classes: classes used to form and solve the governing equations

4. Numerical classes: classes used to handle the numerical operations in the solution procedure.

These classes have accesses between each other for solving problems stated in Section 3.2. Step one of the analysis is performed by the analyst through building components (node, element, constraint, etc) by Modelling classes which are then added to domain, a container class of Finite Element Model classes. Step two is performed by the element classes. The Analysis class is used to implement steps three through six. In this work, particular attention is paid to the incremental solution technique used to solve quasi-static problems. The details of the four class types are presented in the following sections.

3.3.1 Modeling class

A TclModelBuilder object can be used as a Modelling class to read the command script, build the model components and add them to the domain. The components of the structural model include node, material, section, element, loading, constraint and coordinate transformation, etc. One TclModelBuilder object is associated with one domain object which is the repository for these components.

3.3.2 Finite Element Model Classes

The main class abstractions used to describe the finite element model are: Node, Element, Constraint, Load and Domain. The domain class is a container class to store the other components. The classes and their relationship are illustrated in Figure 3.1, which shows the class diagram using Rumbaugh notation (Rumbaugh et al. 1991)

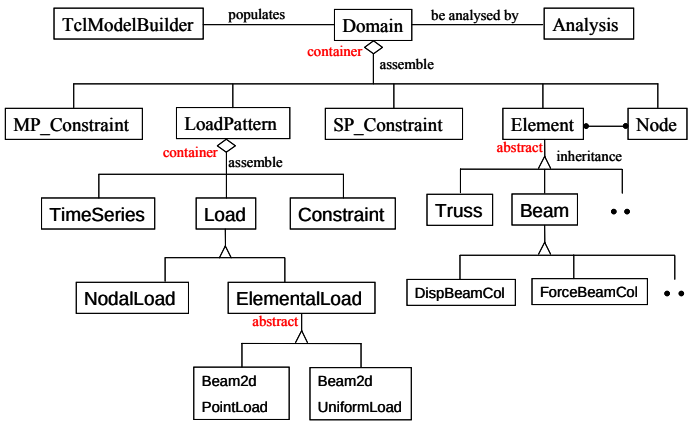


Figure 3.1: Hierarchy diagram of finite element model class

3.3.3 Analysis Class

Figure 3.2 shows the hierarchy of Analysis classes. An analysis object is a container of AnalysisModel, SolutionAlgorithm, Integrator, ConstraintHandler, DOF_Numberer, ConvergenceTest, SystemOfEqn and Solver classes. The details of each class above will be presented in the following sections.

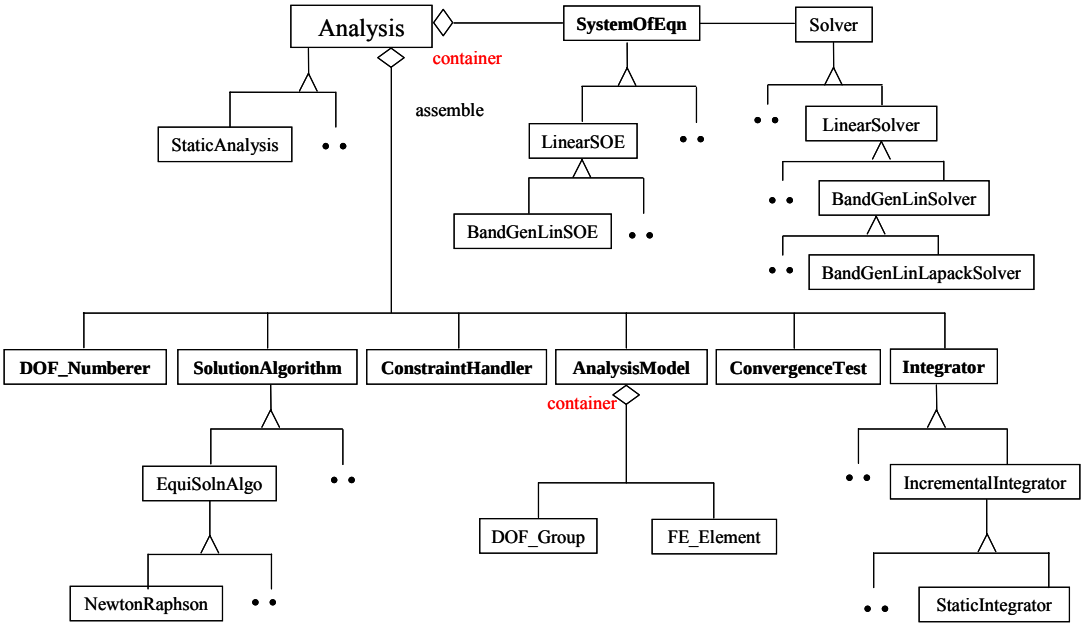


Figure 3.2: Class diagram of Analysis framework

The Analysis class is an abstract base class, defining a pure virtual method. The function "domainChanged()" can be invoked to inform the analysis procedure that the domain has changed. Its subclasses define the implementation of function "analyze()" which is invoked by the analyst to perform an analysis on the domain. Figure 3.3 shows an example of a StaticAnalysis class.

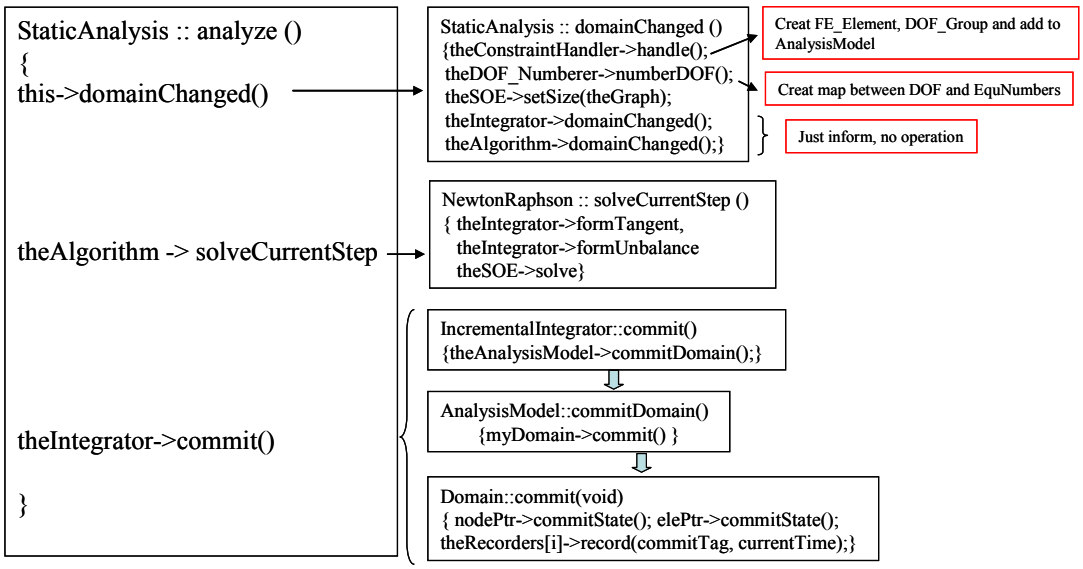


Figure 3.3: Work flow of Analysis class

3.3.3.1 AnalysisModel

The AnalysisModel object is a container class holding and providing access to the FE_Element and DOF_Group objects which are created by ConstraintHandler object. The access of AnalysisModel is shown in Figure 3.4.

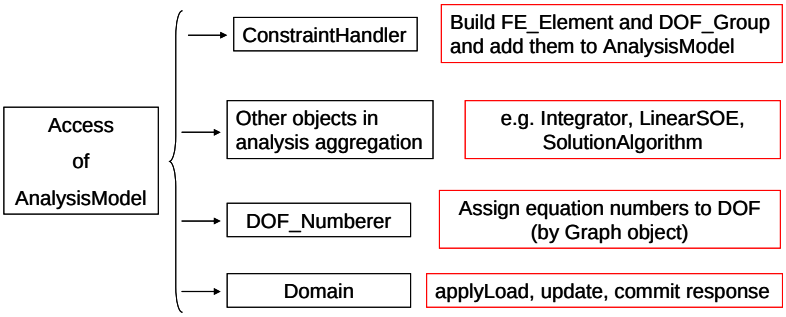


Figure 3.4: Access of AnalysisModel object to other objects

DOF_Group object: one object represents one DOF at a node or new DOF introduced into the analysis to enforce the constraints. It keeps track of mapping between DOF and equation numbers.

FE_Element object: one object associated with one Element in Domain used to define the contribution of EleTangent and EleResidual to the system of structure.

The DOF_Group and FE_Element are important to the design because they remove from the Node and Element objects the need to worry about the mapping between DOF and equation numbers.

3.3.3.2 ConstraintHandler

The ConstraintHandler object is responsible for creating the DOF_Group and FE_Element objects and adding them to the AnalysisModel by the member function "handle()".

3.3.3.3 Integrator

The Integrator class, an abstract base class, is responsible for iterating over the FE_Element and DOF_Group to form structural tangent matrix and residual force vector used to form the equilibrium equations and updating the response at the DOF_Group objects. Figure 3.5 shows the hierarchy of Integrator class.

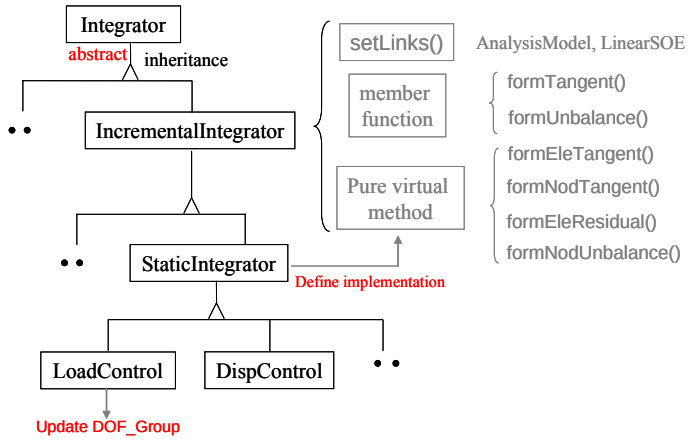


Figure 3.5: Class diagram of Integrator

3.3.3.4 SolutionAlgorithm

The SolutionAlgorithm object is used to control the solution steps in the analysis. It specifies the sequence of operations that are invoked on the different objects in the analysis aggregation. It is an abstract base class with functions "addRecorder()" and "record()" and its subclass defines the implementation of functions "setLinks()" and "solveCurrentStep()". The framework of EquiSolnAlgo class, a subclass of SolutionAlgorithm, is shown in Figure 3.6.

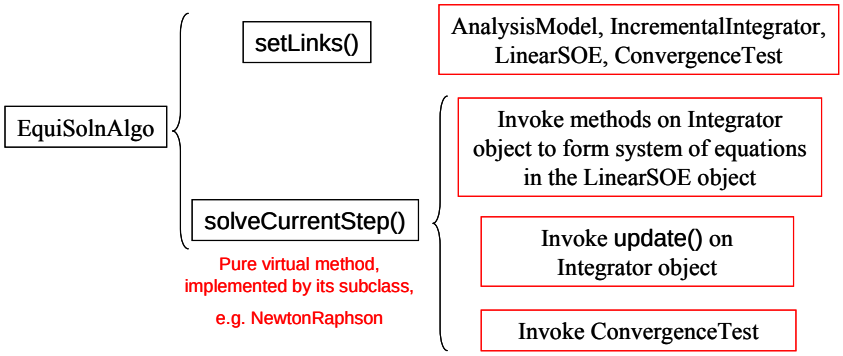


Figure 3.6: Methods in EquiSolnAlgo object

3.3.3.5 DOF_Numberer

The DOF_Numberer object goes through each individual DOF_Group object and assigns equation numbers to DOF, and then makes each FE_Element object to determine its mapping.

3.3.3.6 ConvergenceTest

It is an abstract base class to test the convergence of the algorithm given the tolerance set by analyst.

3.3.4 Numerical Classes

The finite element method requires numerical computations performed by the Numerical classes including Matrix and Vector classes, Tensor classes, and Linear System of Equation classes. There are two classes provided in the Linear System of Equation class: SystemOfEqn and Solver. The SystemOfEqn is responsible for storing the equations. The Solver is used to solve the equations by corresponding numerical methods. Figure 3.7 shows the general solution procedure for an individual loading step in OpenSees.

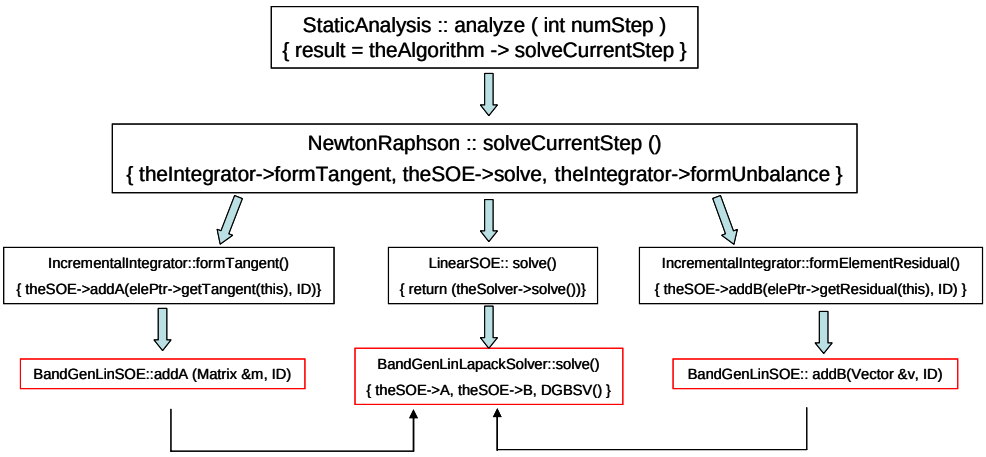


Figure 3.7: Procedure of the solution of the system of equations

In the following sections the material classes of steel and concrete in OpenSees are presented. New temperature dependent material classes are created in OpenSees by modifying these existing material classes according to Eurocodes. The modification will be presented in Chapter 5 and Chapter 7.

3.4 Steel material

The material properties at elevated temperature are of great importance for behaviour of structures in fire. The existing material classes in OpenSees do not include the temperature effects and modifications are made to include the temperature dependent properties which will be presented in Chapter 5. In this section the stress-strain relation of the existing steel material is presented and concrete material in the next section. There are two main uniaxial steel materials at ambient temperature in OpenSees: <Steel01> and <Steel02> in OpenSees (Mazzoni et al. 2007). Steel01 is a bilinear steel material with kinematic hardening and optional isotropic hardening described by a non-linear evolution equation. Steel02 is a Giuffre-Menegotto-Pinto steel material with isotropic strain hardening (Filippou 1983). A new temperature dependent material class Steel01Thermal was developed by modifying the existing Steel01 material class (presented in Chapter 5) and was used to model the steel material for the 2D and 3D fibred beam section. Figure 3.8 shows the stress-strain relationship for Steel01 material defined by three parameters including yield stress f_y, initial elasticity modulus $E0$ and strain hardening slope Esh.

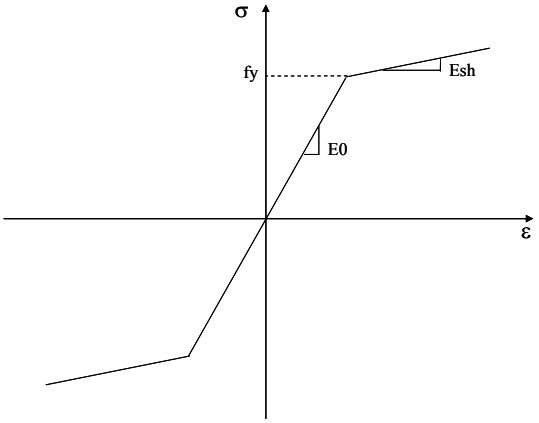


Figure 3.8: Stress-strain relationship of Steel01 in OpenSees

The script command to define Steel01 material in OpenSees can be presented as:

```
uniaxialMaterial Steel01 $Tag $fy $E0 $rat <$a1 $a2 $a3 $a4>
```

where rat is the ratio between *E*sh and *E*0 and a1-a4 is the optional isotropic hardening parameters.

3.5 Concrete material

Similar to the steel material presented in Section 3.4, in this section uniaxial concrete material classes Concrete02 and Pinching4 are introduced to model the undamaged and damaged concrete in the beam/column elements (2D and 3D) respectively. Multi-dimensional material class DruckerPrager is introduced to model the concrete material in the slab. Three corresponding temperature dependent material classes Concrete02Thermal, Pinching4Thermal and DruckerPragerThermal were developed by modifying these three existing material classes in OpenSees which will be presented in Chapter 5.

3.5.1 Uniaxial concrete material

3.5.1.1 Undamaged concrete model

There are three uniaxial concrete material classes in OpenSees: Concrete01, Concrete02 and Concrete03 (Mazzoni et al. 2007). Concrete01 is defined based on zero tensile strength and degraded linear unloading/reloading stiffness. Concrete02 considers the tensile

strength and linear tension softening (Yassin 1994). Nonlinear tension softening is defined in Concrete03.

Due to the similarity in the stress-strain relationship between the three concrete materials, in this section, the Concrete02 is taken for example to highlight the key properties of the concrete material in OpenSees. Seven parameters are needed to define Concrete02 with stress-strain relationship shown in Figure 3.9.

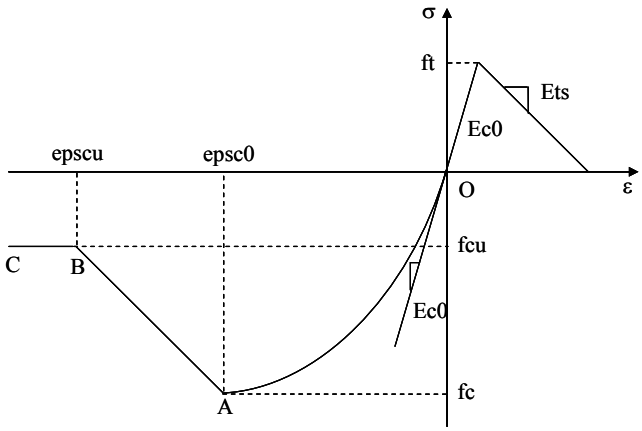


Figure 3.9: Stress-strain relationship of Concrete02 in OpenSees

where

fc —— compressive strength of concrete
epsc0 —— strain at maximum strength of concrete
fcu —— ultimate stress (crushing strength)
epscu —— strain at ultimate stress
rat —— ratio between unloading slope at epscu and initial slope
ft —— tensile strength
Ets —— tension softening stiffness

The formula for the compressive curve can be expressed as follows (Kent and Park 1997):

For curve OA:

$$\sigma^{OA} = fc\left[2\left(\frac{\varepsilon}{\varepsilon_0}\right)-\left(\frac{\varepsilon}{\varepsilon_0}\right)^2\right] \tag{3.1}$$

$$E^{OA} = \frac{d\sigma}{d\varepsilon} = Ec0\left(1 - \frac{\varepsilon}{\varepsilon_0}\right) \tag{3.2}$$

For straight line AB:

$$\sigma^{AB} = fc + \frac{fcu - fc}{\varepsilon_u - \varepsilon_0}\left(\varepsilon - \varepsilon_0\right) \tag{3.3}$$

$$E^{AB} = \frac{fcu - fc}{\varepsilon_u - \varepsilon_0} \tag{3.4}$$

For horizontal line BC

$$\sigma^{BC} = fcu \tag{3.5}$$

$$E^{BC} = 0 \tag{3.6}$$

The script command to define Concrete02 can be presented as:

```
uniaxialMaterial Concrete02 $Tag $fc $epsc0  $fcu  $epscu  $rat  $ft  $Ets
```

3.5.1.2 Damaged concrete

The Pinching4 material in OpenSees is used to construct a uniaxial material that represents a "pinched" load-deformation response considering degradation under cyclic loading. The material is defined through a response envelope, an upload-reload path, and three hysteric damage rules that control evolution of these paths. The hysteretic damage is simulated through deterioration in unloading and reloading stiffness as well as strength degradation. Figure 3.10 shows the load-deformation history of this material. Four points are used to define the multi-linear path of the positive and negative response envelope, respectively. Six parameters are required to define the two trilinear unload-reload paths and twelve parameters to define the hysteric damage rules shown in Figure 3.11-3.13 (Lowes 2004). The theory and implementation details of this Pinching4 material in OpenSees can be found in Lowes 2004. The script command to define Pinching4 material in OpenSees can be presented as:

```
uniaxialMaterial Pinching4 $Tag $ePf1 $ePd1 $ePf2 $ePd2 $ePf3 $ePd3 $ePf4 $ePd4
        <$eNf1 $eNd1 $eNf2 $eNd2 $eNf3 $eNd3 $eNf4 $eNd4 > $rDispP $rForceP
        $uForceP <$rDispN $rForceN $uForceN > $gK1 $Gk2 $gK3 $gK4 $gKLim
        $gDd1 $gDd2 $gDd3 $gDd4 $gDLim $gFd1 $gFd2 $gFd3 $gFd4 $gFLim
        $gE $ dmgType
```

where

ePf1-ePf4 ——force values on the positive response envelop
ePd1-ePd4 ——deformation values on the positive response envelop
eNf1-eNf4 ——force values on the negative response envelop
eNd1-eNd4 ——deformation values on the negative response envelop

rDispP ——The ratio of the deformation at which reloading occurs to the maximum historic deformation demand
rForceP ——The ratio of the force at which reloading occurs to force corresponding to the maximum historic deformation demand
uForceP ——The ratio of strength developed upon unloading from negative load to the maximum strength developed under monotonic loading
rDispN ——The ratio of the deformation at which reloading occurs to the minimum historic deformation demand
rForceN ——The ratio of the force at which reloading occurs to force corresponding to the minimum historic deformation demand
uForceN ——The ratio of strength developed upon unloading from negative load to the minimum strength developed under monotonic loading
gK1-gK4, gKLim ——The values to control cyclic unloading stiffness degradation
gD1-gD4, gDLim ——The values to control cyclic reloading stiffness degradation
gF1-gF4, gFLim ——The values to control cyclic strength degradation
gE ——The value to define maximum energy dissipation under cyclic loading
dmgType ——The string to indicate type of damage (option:"cycle"; "energy")

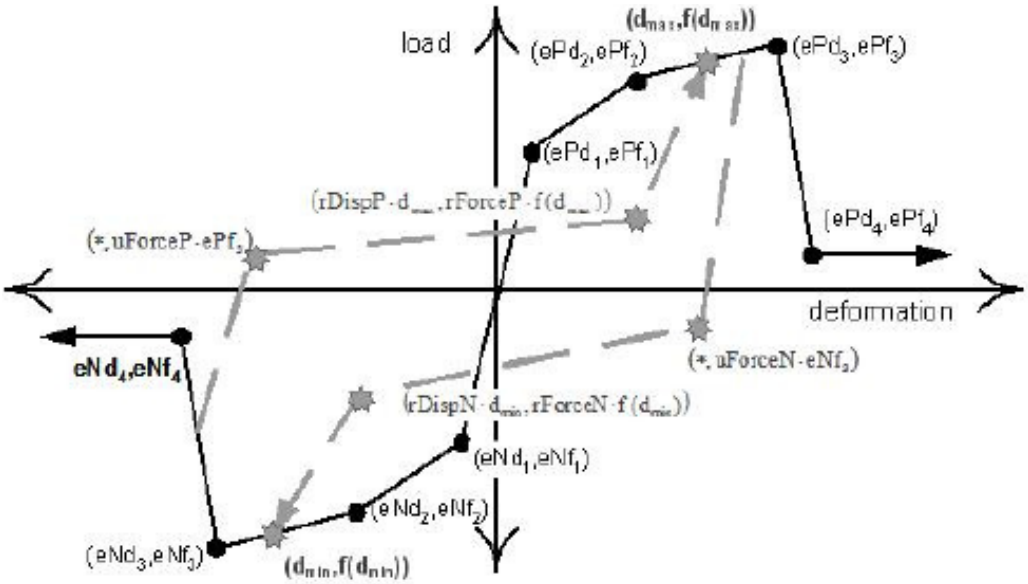


Figure 3.10: Definition of Pinching4 uniaxial material model (Mazzoni et al. 2007)

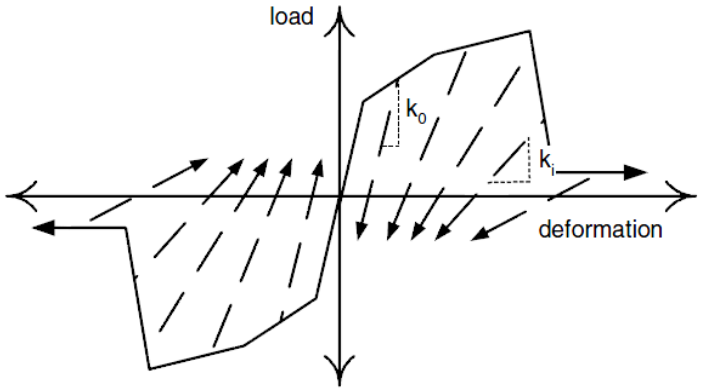


Figure 3.11: Unloading stiffness degradation (Lowes, 2004)

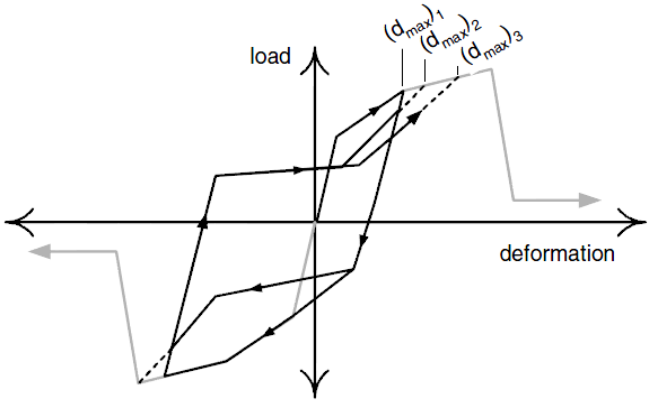


Figure 3.12: Reloading stiffness degradation (Lowes, 2004)

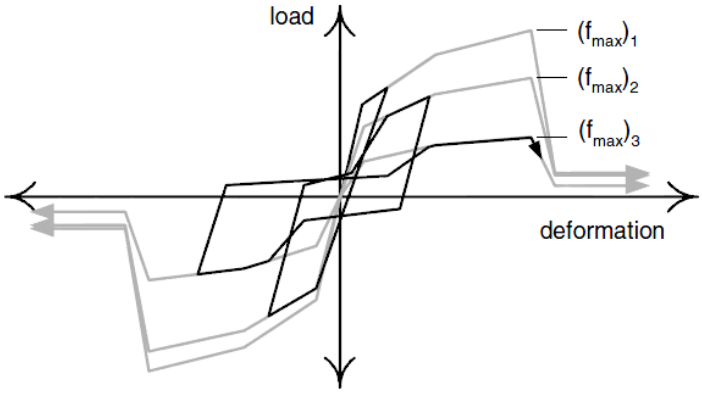


Figure 3.13: Strength degradation (Lowes, 2004)

3.5.2 Drucker-Prager concrete material

The multi-dimensional material class DruckerPrager is used to model concrete in the slab consisting of a Drucker-Prager yield criterion and tension cut-off model (Drucker and Prager 1952; Chen and Saleeb 1994).

The yield condition for the Drucker-Prager model can be expressed as

$$\sqrt{2J_2} = \sqrt{\frac{2}{3}}\sigma_Y - \rho I_1 \tag{3.7}$$

Where ρ and σ_Y are material constants. J_2 is the second invariant of the deviatoric stress tensor and I_1 is the first invariant of the stress tensor defined by the following relationships:

$$J_2 = \frac{1}{6}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right] \tag{3.8}$$

$$I_1 = \sigma_1 + \sigma_2 + \sigma_3 \tag{3.9}$$

Where σ_i is the three-dimensional principal stress.

The constants ρ and σ_Y can be determined by substituting the uniaxial compressive and tensile yield stress *f*c and *f*t into Equation 3.7 as

$$\rho = \frac{\sqrt{6}}{3}\frac{f_c - f_t}{f_c + f_t} \tag{3.10}$$

$$\sigma_Y = 2\frac{f_c f_t}{f_c + f_t} \tag{3.11}$$

The script command to define DruckerPrager material in OpenSees can be presented as:

```
nDMaterial DruckerPrager $Tag $k $G  $sigmaY  $rho $rhoBar  $Kinf  $Ko $delta1
$delta2 $H $theta $density
```

where

k —— bulk modulus, $k=E/3(1-2\nu)$, *E* is elastic modulus and ν is the Poisson's ratio
G —— shear modulus, $G=E/2(1+\nu)$
sigmaY —— yield stress
rho —— frictional strength parameter
rhoBar —— controls evolution of plastic volume change
Kinf —— nonlinear isotropic strain hardening parameter
Ko —— nonlinear isotropic strain hardening parameter
delta1 —— nonlinear isotropic strain hardening parameter
delta2 —— tension softening parameter
H —— linear strain hardening parameter
theta —— controls relative proportions of isotropic and kinematic hardening
density —— mass density of the material

The detailed theory and examples can be found from the OpenSees Wiki site.

In order to determine the values of parameters *k* and *G*, the initial elasticity modulus of the concrete should be calculated first. In this case the compressive curve OA defined in Concrete02 material class was accepted and the corresponding *E* can be calculated from the derivative of σ^{OA} with respect to ε as

$$E = \frac{d\sigma^{OA}}{d\varepsilon} = 2\frac{fc}{\varepsilon_0} \tag{3.12}$$

The extension of OpenSees focuses on the Finite Element Model classes group. This involves the modification of existing material classes to include temperature dependent properties, modification of existing section and element classes to include temperature related messages and methods. In the next chapter the theory of the existing two-dimensional beam element in OpenSees will be presented which its geometrical nonlinearity is based on the corotational framework. The modification of these corresponding classes in OpenSees (material, section and element) will be presented in Chapter 5.

Chapter 4

Nonlinear Corotational Beam-column Element

4.1 Introduction

In the previous chapter the class hierarchy of existing OpenSees framework was presented. A two-dimensional thermomechanical analysis model was developed in OpenSees based on the existing OpenSees framework. The development involves modifications on the beam element, section and material classes which will be presented in Chapter 5. For the existing two-dimensional beam/column element in OpenSees, there are three kinds of coordinate transformation scheme: linear, corotational and P-Delta transformation. The linear and P-Delta transformation perform a linear geometric transformation of beam stiffness and resisting force from the basic system to the global coordinate system but the latter considers the second-order P-Delta effects. The corotational transformation performs an exact or nonlinear geometric transformation. In this thesis, the existing corotational transformation was used to deal with thermomechanical analysis in OpenSees. To help understand the modifications in OpenSees, the existing corotational framework for the two-dimensional beam/column element in OpenSees was presented in this chapter. A corotational formulation seeks to separate rigid body motions from strain producing deformations at the local element level. This is accomplished by attaching a local element reference frame (or coordinate system), which rotates and translates with the beam element. A local elemental stiffness matrix is first calculated referring to this local reference frame and the global stiffness matrix is then formed by corotational coordinate transformation scheme. The local stiffness matrix of this beam element used in OpenSees is based on the geometrically linear Euler-Bernoulli beam theory. The corresponding local stiffness matrices of nonlinear Euler-Bernouli and Timoshenko beam theory Linear and nonlinear are also presented. Two kinds of beam elements based on stiffness method and force method are available in OpenSees and the corresponding theories are also presented in this chapter.

4.2 General corotational beam element

In this section the corotational framework is presented based on Euler and Timoshenko beam. An individual beam element subjected to external loadings potentially does three things: it rotates, translates and deforms. The rotation and translation are rigid body motions, which may be removed from the motion of the beam. If this is done, all that remains are the strain causing deformations of the beam element. The strain causing local deformations are related to the forces induced in the beam element. The framework of a corotational beam element is

presented first followed by different local stiffness matrices defined for Euler and Timoshenko beams.

4.2.1 Co-rotational framework

In this section the relations between the local and global expressions of the resisting force vector and tangent stiffness matrix are presented. This presentation is mainly taken from reference (Battini 2002).

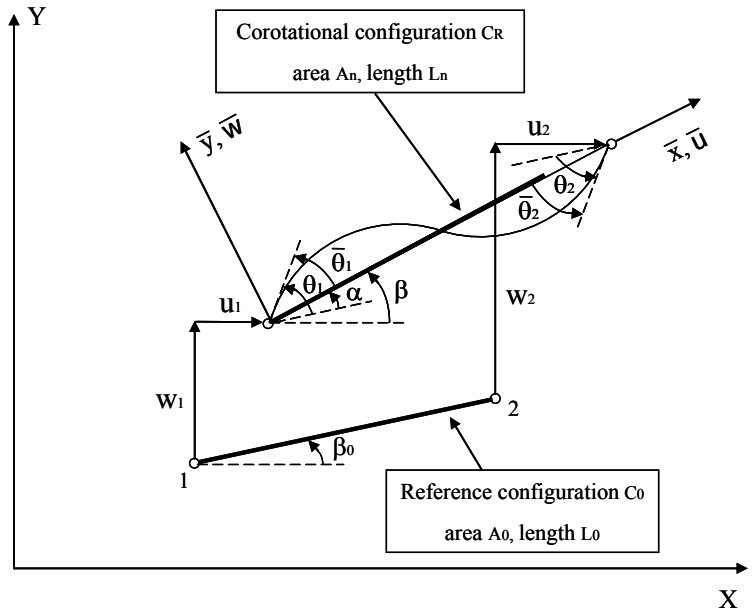


Figure 4.1: Reference and corotational configuration of a typical beam element

4.2.1.1 Beam kinematics

The notations used in this section are defined in Figure 4.1. The coordinates for the nodes 1 and 2 of the beam element in the global coordinate system (X, Y) are (X_1, Y_1) and (X_2, Y_2). The vector of global displacement q_g is defined by

$$q_g = \begin{bmatrix} u_1 & w_1 & \theta_1 & u_2 & w_2 & \theta_2 \end{bmatrix}^T \tag{4.1}$$

The vector of basic element displacement is defined by

$$q_l = \begin{bmatrix} \bar{u} & \bar{\theta}_1 & \bar{\theta}_2 \end{bmatrix}^T \tag{4.2}$$

The components of q_l can be computed according to

$$\bar{u} = l_n - l_0 \tag{4.3a}$$

$$\bar{\theta}_1 = \theta_1 - \alpha \tag{4.3b}$$

$$\bar{\theta}_2 = \theta_2 - \alpha \tag{4.3c}$$

Where l_0 and l_n is the initial and current lengths of the element, α denotes the rigid rotation angle. The definition of these parameters can be expressed as

$$l_n = \sqrt{x_{21}^2 + y_{21}^2} = \sqrt{\left(X_{21} + u_{21}\right)^2 + \left(Y_{21} + w_{21}\right)^2} \tag{4.4}$$

$$l_0 = \sqrt{X_{21}^2 + Y_{21}^2} \tag{4.5}$$

$$c = \cos\beta = \left(X_{21} + u_{21}\right)/l_n \tag{4.6a}$$

$$s = \sin\beta = \left(Y_{21} + w_{21}\right)/l_n \tag{4.6b}$$

$$\alpha = \beta - \beta_0 \tag{4.6c}$$

Where $X_{21} = X_2 - X_1$; $Y_{21} = Y_2 - Y_1$; $u_{21} = u_2 - u_1$; $w_{21} = w_2 - w_1$

4.2.1.2 Virtual displacement

The virtual local displacements can be obtained through differentiation of Equation 4.3 as

$$\delta\bar{u} = \delta l_n = c\delta u_{21} + s\delta w_{21} = \begin{bmatrix} -c & -s & 0 & c & s & 0 \end{bmatrix} \delta \mathrm{q_g} \tag{4.7a}$$

$$\delta\bar{\theta}_1 = \delta\theta_1 - \delta\alpha = \delta\theta_1 - \delta\beta \tag{4.7b}$$

$$\delta\bar{\theta}_2 = \delta\theta_2 - \delta\alpha = \delta\theta_2 - \delta\beta \tag{4.7c}$$

Where $\delta\beta$ can be calculated by differentiation of Equation 4.6b as

$$\delta\beta = \frac{\delta w_{21} l_n - \left(Y_{21} + w_{21}\right)\delta l_n}{c l_n^2} = \frac{\delta w_{21} l_n - \left(Y_{21} + w_{21}\right)\left(c\delta u_{21} + s\delta w_{21}\right)}{c l_n^2} \tag{4.8}$$

Substituting Equation 4.6b into Equation 4.8, $\delta\beta$ becomes

$$\delta\beta = \frac{-s\delta u_{21} + c\delta w_{21} l_n}{l_n} \tag{4.9}$$

which, after simplifications gives

$$\delta\beta = \frac{1}{l_n}\begin{bmatrix} s & -c & 0 & -s & c & 0 \end{bmatrix}\delta \mathrm{q_g} \tag{4.10}$$

Therefore, substituting $\delta\beta$ into Equation 4.3 gives

$$\delta\bar{u} = \begin{bmatrix} -c & -s & 0 & c & s & 0 \end{bmatrix}\delta \mathrm{q_g} \tag{4.11a}$$

$$\delta\bar{\theta}_1 = \begin{bmatrix} -s/l_n & c/l_n & 1 & s/l_n & -c/l_n & 0 \end{bmatrix}\delta \mathrm{q_g} \tag{4.11b}$$

$$\delta\bar{\theta}_2 = \begin{bmatrix} -s/l_n & c/l_n & 0 & s/l_n & -c/l_n & 1 \end{bmatrix}\delta \mathrm{q_g} \tag{4.11c}$$

The transformation matrix B, defined by

$$\delta \mathrm{q_l} = \mathrm{B}\delta \mathrm{q_g} \tag{4.12}$$

is given by

$$\mathrm{B} = \begin{bmatrix} -c & -s & 0 & c & s & 0 \\ -s/l_n & c/l_n & 1 & s/l_n & -c/l_n & 0 \\ -s/l_n & c/l_n & 0 & s/l_n & -c/l_n & 1 \end{bmatrix} \tag{4.13}$$

4.2.1.3 Internal force

The relationship between the local and global internal force vector p_l, p_g is obtained by equating the virtual work in both the local and global systems

$$V = \delta \mathrm{q}_g^T \mathrm{p}_g = \delta \mathrm{q}_l^T \mathrm{p}_l = \delta \mathrm{q}_g^T \mathrm{B}^T \mathrm{p}_l \tag{4.14}$$

Considering any arbitrary $\delta \mathrm{q}_g$ applied in Equation 4.14 the global internal force p_g is given by

$$\mathrm{p}_g = \mathrm{B}^T \mathrm{p}_l \tag{4.15}$$

In which the local internal force $\mathrm{p}_l = [N \;\; M_1 \;\; M_2]^{\mathrm{T}}$ depends on the element definition and will be calculated in the following sections.

4.2.1.4 Tangent stiffness matrix

The global tangent stiffness matrix $\mathrm{K_g}$ can be defined by

$$\delta \mathrm{p}_g = \mathrm{K_g} \delta \mathrm{q}_g \tag{4.16}$$

which is obtained by differentiation of Equation 4.15

$$\delta \mathrm{p}_g = \mathrm{B}^T \delta \mathrm{p}_l + \delta \mathrm{B}^T \mathrm{p}_l = \mathrm{B}^T \delta \mathrm{p}_l + \delta \begin{bmatrix} -c & -s/l_n & -s/l_n \\ -s & c/l_n & c/l_n \\ 0 & 1 & 0 \\ c & s/l_n & s/l_n \\ s & -c/l_n & -c/l_n \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N \\ M_1 \\ M_2 \end{bmatrix} \tag{4.17}$$

The first term of Equation 4.17 can be computed as

$$\mathrm{B}^T \delta p_l = \mathrm{B}^T K_l \mathrm{B} \delta p_g \tag{4.18}$$

where K_l is the local tangent stiffness matrix, which depends on the element definition (Euler or Timoshenko beam) and will be presented in Section 4.2.2-4.2.4.

By defining these notations b_{i} = ith column vector of matrix B^{T}

$$\mathrm{b}_1 = \begin{bmatrix} -c & -s & 0 & c & s & 0 \end{bmatrix}^T \tag{4.19a}$$

$$\mathrm{b}_2 = \begin{bmatrix} -\dfrac{s}{l_n} & \dfrac{c}{l_n} & 1 & \dfrac{s}{l_n} & -\dfrac{c}{l_n} & 0 \end{bmatrix}^T \tag{4.19b}$$

$$\mathrm{b}_3 = \left[-\frac{s}{l_n} \quad \frac{c}{l_n} \quad 0 \quad \frac{s}{l_n} \quad -\frac{c}{l_n} \quad 1 \right]^T \tag{4.19c}$$

Equation 4.17 becomes

$$\delta \mathrm{p}_g = \mathrm{B}^T K_l \mathrm{B} \delta p_g + N \delta b_1 + M_1 \delta b_2 + M_2 \delta b_3 \tag{4.20}$$

Equation 4.19 can be transformed into

$$\mathrm{b}_1 = \mathrm{r} \tag{4.21a}$$

$$\mathrm{b}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T - \frac{\mathrm{z}}{l_n} \tag{4.21b}$$

$$\mathrm{b}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T - \frac{\mathrm{z}}{l_n} \tag{4.21c}$$

Where r and z is expressed as

$$\mathrm{r} = \begin{bmatrix} -c & -s & 0 & c & s & 0 \end{bmatrix}^T \tag{4.22a}$$

$$\mathrm{z} = \begin{bmatrix} s & -c & 0 & -s & c & 0 \end{bmatrix}^T \tag{4.22b}$$

of which differentiation forms as

$$\delta \mathrm{r} = \mathrm{z} \delta \beta = \frac{\mathrm{z}\mathrm{z}^T}{l_n} \delta \mathrm{q}_\mathrm{g} \tag{4.23a}$$

$$\delta \mathrm{z} = -\mathrm{r} \delta \beta = -\frac{\mathrm{r}\mathrm{z}^T}{l_n} \delta \mathrm{q}_\mathrm{g} \tag{4.23b}$$

By substituting of Equation 4.23 into Equation 4.21 the differentiation of Equation 4.21 can be presented as

$$\delta \mathrm{b}_1 = \delta \mathrm{r} = \frac{\mathrm{z}^T \mathrm{z}^T}{l_n} \delta \mathrm{q}_\mathrm{g} \tag{4.24a}$$

$$\delta \mathrm{b}_2 = \delta \mathrm{b}_3 = -\frac{\delta \mathrm{z}}{l_n} + \frac{\mathrm{z} \delta l_n}{l_n^2} = \frac{1}{l_n^2} \left(\mathrm{r}\mathrm{z}^T + \mathrm{z}\mathrm{r}^T \right) \delta \mathrm{q}_\mathrm{g} \tag{4.24b}$$

Substituting of Equation 4.24 into Equation 4.20 gives

$$\delta \mathrm{p}_g = \left[\mathrm{B}^T K_l \mathrm{B} + \frac{N}{l_n} \mathrm{z}^T \mathrm{z}^T + \frac{M_1 + M_2}{l_n^2} \left(\mathrm{rz}^T + \mathrm{zr}^T \right) \right] \delta p_g \tag{4.25}$$

From Equation 4.25, the global tangent stiffness matrix is shown as

$$\mathrm{K}_\mathrm{g} = \mathrm{K}_g^M + \mathrm{K}_g^G \tag{4.26}$$

where K_g^M is called material matrix and K_g^G for geometric matrix

$$K_g^M = \mathrm{B}^T K_t^l \mathrm{B} \tag{4.27a}$$

$$K_g^G = \frac{N}{l_n} \mathrm{z}^T \mathrm{z}^T + \frac{M_1 + M_2}{l_n^2} \left(\mathrm{rz}^T + \mathrm{zr}^T \right) \tag{4.27b}$$

Equations 4.15 and 4.27 give the relations between the local internal forces $\mathbf{p}_l$ and tangent stiffness matrix $\mathbf{K}_l$ and the global $\mathbf{p}_g$ and $\mathbf{K}_g$. These relations are independent on the choosing of different beam theories (Euler or Timoshenko beam). Although the linear Euler-Bernoulli beam theory is used in the existing beam element, other theories such as nonlinear Euler-Bernoulli and Timoshenko beam theory (linear and nonlinear) will also be presented in the following sections which are considered to be implemented in OpenSees in the future.

4.2.2 Local linear Euler-Bernoulli beam

4.2.2.1 Strain-displacement relationship

The Euler-Bernoulli beam element is based on the classical linear beam theory, using a linear interpolation for the axial displacement u and a Hermite cubic for the vertical displacement w.

$$u = \xi \overline{u} \tag{4.28a}$$

$$w = \left(\xi - 2\xi^2 + \xi^3 \right) l_0 \overline{\theta}_1 + \left(-\xi^2 + \xi^3 \right) l_0 \overline{\theta}_2 = N_2 \overline{\theta}_1 + N_4 \overline{\theta}_2 \tag{4.28b}$$

where ξ is the natural coordinate and $\xi = x/l_0$.

The curvature κ and strain ε are defined by

$$\kappa = \frac{\partial^2 w}{\partial x^2} = \frac{1}{l_0^2}\frac{\partial^2 w}{\partial \xi^2} = \frac{1}{l_0}\left(-4+6\xi\right)\bar{\theta}_1 + \frac{1}{l_0}\left(-2+6\xi\right)\bar{\theta}_2 \tag{4.29}$$

$$\varepsilon = \frac{\partial u}{\partial x} - y\kappa = \frac{\bar{u}}{l_0} + \frac{y}{l_0}\left(4-6\xi\right)\bar{\theta}_1 + \frac{y}{l_0}\left(2-6\xi\right)\bar{\theta}_2 \tag{4.30}$$

4.2.2.2 Local internal force

For the nonlinear stress-strain relation, it is not possible to derive analytical expressions for the internal forces and the local tangent stiffness matrix and a numerical integration method has to be adopted. For the elastic case, the analytical formula can be obtained as follows.

For the principle of virtual work, the internal force can be calculated by

$$V = \int_v \sigma\delta\varepsilon dv = N\delta\bar{u} + M_1\delta\bar{\theta}_1 + M_2\bar{\theta}_2 \tag{4.31}$$

which by introducing Equation 4.30 gives

$$V = \int_v \sigma\delta\varepsilon dV = \int_v \sigma\left(\frac{\delta\bar{u}}{l_0} + \frac{y}{l_0}\left(4-6\xi\right)\delta\bar{\theta}_1 + \frac{y}{l_0}\left(2-6\xi\right)\delta\bar{\theta}_2\right)dv \tag{4.31}$$

The local internal forces are given by

$$N = \int_v \frac{\sigma}{l_0} dv \tag{4.32a}$$

$$M_1 = \int_v \sigma\frac{y}{l_0}\left(4-6\xi\right)dv \tag{4.32b}$$

$$M_2 = \int_v \sigma\frac{y}{l_0}\left(2-6\xi\right)dv \tag{4.32c}$$

4.2.2.3 Local tangent stiffness matrix

Differentiation of stress σ gives

$$\delta\sigma = E\left(\frac{\delta\bar{u}}{l_0} + \frac{y}{l_0}\left(4-6\xi\right)\delta\bar{\theta}_1 + \frac{y}{l_0}\left(2-6\xi\right)\delta\bar{\theta}_2\right) \tag{4.33}$$

The local tangent stiffness matrix K_l is obtained by differentiation of the Equation 4.32

$$\delta N = \int_v \delta\sigma \frac{1}{l_0} dv \tag{4.34a}$$

$$\delta M_1 = \int_v \delta\sigma \frac{y}{l_0}\left(4-6\xi\right) dv \tag{4.34b}$$

$$\delta M_2 = \int_v \delta\sigma \frac{y}{l_0}\left(2-6\xi\right) dv \tag{4.34c}$$

which from Equation 4.33 gives

$$\begin{aligned}
K_l^{11} &= \frac{\partial N}{\partial \bar{u}} = \frac{1}{l_0^2}\int_v E dv \\
K_l^{22} &= \frac{\partial M_1}{\partial \bar{\theta}_1} = \frac{1}{l_0^2}\int_v E y^2 \left(4-6\xi\right)^2 dv \\
K_l^{33} &= \frac{\partial M_2}{\partial \bar{\theta}_2} = \frac{1}{l_0^2}\int_v E y^2 \left(2-6\xi\right)^2 dv \\
K_l^{12} &= K_{21}^l = \frac{\partial N}{\partial \bar{\theta}_1} = \frac{\partial M_1}{\partial \bar{u}} = \frac{1}{l_0}\int_v E y \left(4-6\xi\right) dv \\
K_l^{13} &= K_{31}^l = \frac{\partial N}{\partial \bar{\theta}_2} = \frac{\partial M_2}{\partial \bar{u}} = \frac{1}{l_0}\int_v E y \left(2-6\xi\right) dv \\
K_l^{23} &= K_{32}^l = \frac{\partial M_1}{\partial \bar{\theta}_2} = \frac{\partial M_2}{\partial \bar{\theta}} = \frac{1}{l_0}\int_v E y^2 \left(4-6\xi\right)\left(2-6\xi\right) dv
\end{aligned} \tag{4.35}$$

The closed form solution of local stiffness matrix can be shown as

$$K_l = \begin{bmatrix} \frac{EA}{l_0} & 0 & 0 \\ 0 & \frac{4EI}{l_0} & \frac{2EI}{l_0} \\ 0 & \frac{2EI}{l_0} & \frac{4EI}{l_0} \end{bmatrix} \tag{4.36}$$

4.2.3 Local nonlinear Euler-Bernoulli beam

4.2.3.1 Strain-displacement relationship

The Green strain is used to measure the strain ε_G of the neutral axis of the beam

$$\varepsilon_G = \frac{l_n^2 - l_0^2}{2l_0^2} \tag{4.37}$$

Substituting of $l_n = l_0 + \overline{u}$, $u = \left(x/l_0\right)\overline{u}$ into Equation 4.37 gives

$$\varepsilon_G = \frac{\overline{u}}{l_0} + \frac{1}{2}\left(\frac{\overline{u}}{l_0}\right)^2 \tag{4.38}$$

With the same displacement shape function with Equation 4.28 the curvature κ and strain ε are defined by

$$\kappa = \frac{\partial^2 w}{\partial x^2} = \frac{1}{l_0^2}\frac{\partial^2 w}{\partial \xi^2} = \frac{1}{l_0}\left(-4+6\xi\right)\overline{\theta}_1 + \frac{1}{l_0}\left(-2+6\xi\right)\overline{\theta}_2 \tag{4.39a}$$

$$\varepsilon = \varepsilon_G - y\kappa = \frac{\overline{u}}{l_0} + \frac{1}{2}\left(\frac{\overline{u}}{l_0}\right)^2 + \frac{y}{l_0}\left(4-6\xi\right)\overline{\theta}_1 + \frac{y}{l_0}\left(2-6\xi\right)\overline{\theta}_2 \tag{4.39b}$$

4.2.3.2 Local internal force

For the principle of virtual work, the internal force can be calculated by

$$V = \int_v \sigma\delta\varepsilon dv = N\delta\overline{u} + M_1\delta\overline{\theta}_1 + M_2\overline{\theta}_2 \tag{4.40}$$

which by introducing Equation 4.39b gives

$$V = \int_v \sigma\delta\varepsilon dV = \int_v \sigma\left[\frac{1}{l_0}\left(1+\frac{\overline{u}}{l_0}\right)\delta\overline{u} + \frac{y}{l_0}\left(4-6\xi\right)\delta\overline{\theta}_1 + \frac{y}{l_0}\left(2-6\xi\right)\delta\overline{\theta}_2\right]dv \tag{4.41}$$

The local internal forces are given by

$$N = \int_v \frac{\sigma}{l_0}\left(1+\frac{\overline{u}}{l_0}\right)dv \tag{4.42a}$$

$$M_1 = \int_v \sigma \frac{y}{l_0}\left(4-6\xi\right)dv \tag{4.42b}$$

$$M_2 = \int_v \sigma \frac{y}{l_0}\left(2-6\xi\right)dv \tag{4.42c}$$

4.2.3.3 Local tangent stiffness matrix

Differentiation of stress σ gives

$$\delta\sigma = E\left(\frac{1}{l_0}\left(1+\frac{\overline{u}}{l_0}\right)\delta\overline{u} + \frac{y}{l_0}\left(4-6\xi\right)\delta\overline{\theta}_1 + \frac{y}{l_0}\left(2-6\xi\right)\delta\overline{\theta}_2\right) \tag{4.43}$$

The local tangent stiffness matrix K_l is obtained by differentiation of the Equation 4.42

$$\delta N = \int_v \delta\sigma \frac{1}{l_0}\left(1+\frac{\overline{u}}{l_0}\right)dv + \int_v \frac{\sigma}{l_0^2}\delta\overline{u}dv \tag{4.44a}$$

$$\delta M_1 = \int_v \delta\sigma \frac{y}{l_0}\left(4-6\xi\right)dv \tag{4.44b}$$

$$\delta M_2 = \int_v \delta\sigma \frac{y}{l_0}\left(2-6\xi\right)dv \tag{4.44c}$$

which from Equation 4.43 gives

$$
\begin{aligned}
K_{11}^{l} &= \frac{\partial N}{\partial \bar{u}} = \frac{1}{l_0^2}\int_v \frac{E}{l_0^2}\left(1+\frac{\bar{u}}{l_0}\right)^2 dv + \int_v \frac{\sigma}{l_0^2} dv \\
K_{22}^{l} &= \frac{\partial M_1}{\partial \bar{\theta}_1} = \frac{1}{l_0^2}\int_v Ey^2\left(4-6\xi\right)^2 dv \\
K_{33}^{l} &= \frac{\partial M_2}{\partial \bar{\theta}_2} = \frac{1}{l_0^2}\int_v Ey^2\left(2-6\xi\right)^2 dv \\
K_{12}^{l} &= K_{21}^{l} = \frac{\partial N}{\partial \bar{\theta}_1} = \frac{\partial M_1}{\partial \bar{u}} = \frac{1}{l_0}\int_v Ey\left(4-6\xi\right) dv \\
K_{13}^{l} &= K_{31}^{l} = \frac{\partial N}{\partial \bar{\theta}_2} = \frac{\partial M_2}{\partial \bar{u}} = \frac{1}{l_0}\int_v Ey\left(2-6\xi\right) dv \\
K_{23}^{l} &= K_{32}^{l} = \frac{\partial M_1}{\partial \bar{\theta}_2} = \frac{\partial M_2}{\partial \bar{\theta}} = \frac{1}{l_0}\int_v Ey^2\left(4-6\xi\right)\left(2-6\xi\right) dv
\end{aligned}
\tag{4.45}
$$

The closed form solution of local stiffness matrix can be shown as

$$
K_l = \begin{bmatrix}
\dfrac{EA}{l_0}\left(1+\dfrac{\bar{u}}{l_0}\right)^2 + \dfrac{N_0}{l_0} & 0 & 0 \\
0 & \dfrac{4EI}{l_0} & \dfrac{2EI}{l_0} \\
0 & \dfrac{2EI}{l_0} & \dfrac{4EI}{l_0}
\end{bmatrix}
\tag{4.46}
$$

where $N_0 = \int_v \frac{\sigma}{l_0} dv$

Compared with Equation 4.36, the stiffness matrix depends on the horizontal displacement and there is an additional term (N_0/l_0) in Equation 4.46 which accounts for the P-δ effect in this geometrically nonlinear beam element. It is noted that this P-δ effect is considered in this local coordinate system and global P-δ effect can be considered in the corotational transformation of elemental stiffness matrix. Similar deductions with this Euler beam can be conducted to obtain the local stiffness matrices for Timoshenko beam which are presented in the next section.

4.2.4 Local Timoshenko beam

Timoshenko beam assumes that the rotation is independent of vertical displacement, using a linear interpolation for both the axial displacement u and rotation θ as

$$u = \xi \overline{u} \tag{4.47a}$$

$$w = 0 \tag{4.47b}$$

$$\theta = \left(1 - \xi\right)\overline{\theta}_1 + \xi \overline{\theta}_2 \tag{4.47c}$$

The curvature κ, shear strain γ and axial strain ε are defined by

$$\kappa = \frac{\partial \theta}{\partial x} \tag{4.48a}$$

$$\gamma = \frac{\partial w}{\partial x} - \theta \tag{4.48b}$$

$$\varepsilon = \frac{\partial u}{\partial x} - y\kappa \tag{4.48c}$$

The local internal forces taking into account of the shear deformation can be expressed as

$$V = \int_v \left(\sigma \delta \varepsilon + \tau \delta \gamma\right) dv = N \delta \overline{u} + M_1 \delta \overline{\theta}_1 + M_2 \overline{\theta}_2 \tag{4.49}$$

Apart from the difference above, the deduction of internal force and tangent stiffness matrix for linear and nonlinear Timoshenko beam follow the same procedure with Euler-Bernoulli beam mentioned in Section 4.2.2 and 4.2.3.

The closed form solution of local stiffness matrix for Timoshenko beam can be shown as

For linear Timoshenko beam:

$$K_l = \begin{bmatrix} \dfrac{EA}{l_0} & 0 & 0 \\ 0 & \dfrac{EI}{l_0} + \dfrac{GAl_0}{3} & -\dfrac{EI}{l_0} + \dfrac{GAl_0}{6} \\ 0 & -\dfrac{EI}{l_0} + \dfrac{GAl_0}{6} & \dfrac{EI}{l_0} + \dfrac{GAl_0}{3} \end{bmatrix} \tag{4.50}$$

For nonlinear Timoshenko beam:

$$K^l = \begin{bmatrix} \frac{EA}{l_0}\left(1+\frac{\bar{u}}{l_0}\right)^2 + \frac{N_0}{l_0} & 0 & 0 \\ 0 & \frac{EI}{l_0}+\frac{GAl_0}{3} & -\frac{EI}{l_0}+\frac{GAl_0}{6} \\ 0 & -\frac{EI}{l_0}+\frac{GAl_0}{6} & \frac{EI}{l_0}+\frac{GAl_0}{3} \end{bmatrix} \tag{4.51}$$

The same difference can be seen in the local matrices above of linear and nonlinear Timoshe nko beam as that for Euler beam.

The corotational framework defines that how the local stiffness matrix and resisting force transformed from local reference configuration to global reference configuration. Given what variable field (displacement or force) is assumed unknown, there are two methods in the finite element analysis: displacement method and force method which displacements are taken assumed field for the former method and force for the latter method. The formulations of these two methods in OpenSees will be presented in the following sections which provide background knowledge for the thermomechanical development of OpenSees presented in the next chapter.

4.3 Stiffness method theory for OpenSees beam elements

The theoretical foundation of dispBeamColumn2d class in OpenSees is the linear Euler-Bernouli beam based on the displacement or stiffness method which is based on appropriate interpolation functions for the transverse and axial displacement of the element. The beam theory of stiffness method and force method presented in this section and Section 4.4 is mainly from reference (Taucer et al. 1991; Spacone et al. 1992, 1996). Cubic Hermitian polynomials are used to approximate the transverse deformations along the element and linear Lagrange shape functions for axial displacement. The elements proposed are limited to small displacements and deformations and assume that plane sections remain plane. The formulation of fibre elements can be divided into three levels: (a) Section state determination, the determination of section stiffness matrix (or flexibility matrix) and section resisting forces that are required in the calculation of the element stiffness matrix (or flexibility matrix) and element resisting forces, respectively. (b) Element state determination, which involves the determination of element stiffness matrix and resisting forces. (c) Structure state

determination. The elemental stiffness matrix and resisting force vector of each element can be assembled into structural correspondences. Then the equilibrium equations can be solved to obtain the new displacement for the current load step.

4.3.1 Section state determination

The section of beam element is subdivided into longitudinal fibres shown in Figure 4.2. The fibre is defined by its location in the local coordinate system and its area A. The constitutive relation of the section is obtained by integration of the responses of the fibres. The details of the section state determination process are presented as follows.

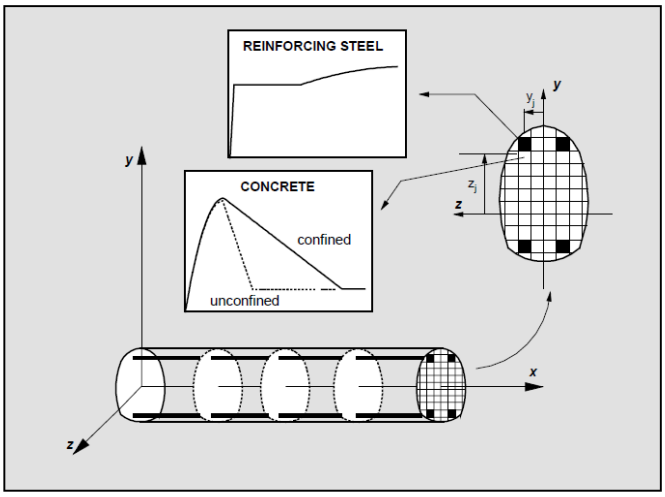


Figure 4.2: Beam section subdivided into fibres (Taucer et al. 1991)

4.3.1.1 Strain of each fibre

The strain $\varepsilon(x,y)$ of each fibre of location x, y can be calculated as

$$\varepsilon(x,y) = \begin{bmatrix} 1 & -y \end{bmatrix} \begin{bmatrix} \varepsilon_0 \\ \varphi \end{bmatrix} = I(y)\,d(x) \tag{4.52}$$

Where ε_0 is the strain at the centroid of the section; φ is the section curvature; $I(y)$ is the geometric vector; $d(x)$ is the section deformation vector.

4.3.1.2 Section stiffness matrix

According to the virtual force principle, the section stiffness matrix $k(x)$ can be determined as

$$k(x) = \int_{A(x)} I^T(y) E(x,y) I(y)\, dA \tag{4.53}$$

where $E(x,y)$ is the tangent material modulus.

By the integration of each fibre through the thickness of the section, the formula above can be transformed into

$$k(x) = \begin{bmatrix} \sum_i E_i A_i & -\sum_i E_i A_i y_i \\ -\sum_i E_i A_i y_i & \sum_i E_i A_i y_i^2 \end{bmatrix} \tag{4.54}$$

where E_i, A_i and y_i is the elasticity modulus, cross-section area and location of each fibre, respectively.

4.3.1.3 Section resisting force

The section resisting force can also be determined by the virtual force principle as

$$D_R(x) = \int_A I(y)^T \sigma(x,y)\, dA \tag{4.55}$$

Where $\sigma(x,y)$ is the stress of an individual fibre.

By the integration of each fibre through the thickness of the section, Equation 4.55 becomes

$$D_R(x) = \begin{bmatrix} \sum \sigma_i A_i \\ -\sum \sigma_i A_i y_i \end{bmatrix} \tag{4.56}$$

4.3.2 Element state determination

4.3.2.1 Displacement interpolation function

The transverse displacement is interpolated by Hermite cubic polynomial defined as

$$\omega(\xi) = N_1\omega_1 + N_2\theta_1 + N_3\omega_2 + N_4\theta_2 \tag{4.57}$$

where $N_1 = 1-3\xi^2+2\xi^3$; $N_2 = \left(\xi - 2\xi^2 + \xi^3\right)L$; $N_3 = 3\xi^2 - 2\xi^3$; $N_4 = \left(-\xi^2 + \xi^3\right)L$; $\xi = x/L\left(0 \le \xi \le 1\right)$

The linear interpolation function is used for the axial displacement

$$u(\xi) = N_5u_1 + N_6u_2 \tag{4.58}$$

where $N_5 = 1-\xi$; $N_6 = \xi$

4.3.2.2 Strain-displacement relation

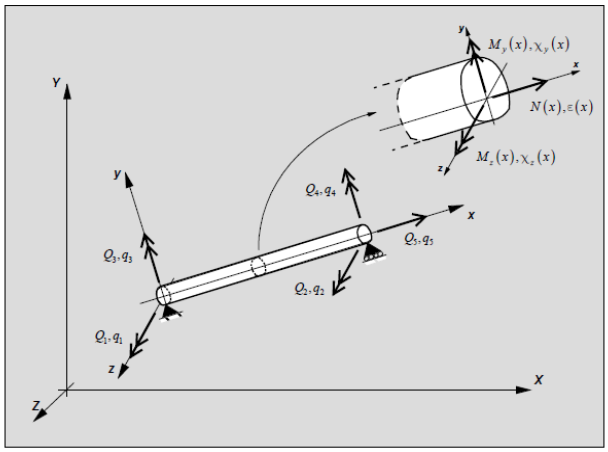


Figure 4.3: Forces and displacements defined in the local coordinate system (Taucer et al. 1991)

There are five DOFs in a 3D element model (q_1-q_5) and only three DOFs for 2D problem (q_1, q_2 and q_5). For a 2D beam element the displacement at any point along the element can be expressed in terms of the element displacement.

$$\begin{bmatrix} u(x) \\ \omega(x) \end{bmatrix} = \mathbf{Nq} = \begin{bmatrix} N_6 & 0 & 0 \\ 0 & N_2 & N_4 \end{bmatrix} \begin{bmatrix} q_5 \\ q_1 \\ q_2 \end{bmatrix} \tag{4.59}$$

where N is the displacement interpolation function.

Substituting of shape function of Equation 4.57 and 4.58 gives

$$\begin{bmatrix} u(x) \\ \omega(x) \end{bmatrix} = \begin{bmatrix} \xi & 0 & 0 \\ 0 & \left(\xi - 2\xi^2 + \xi^3\right)L & \left(-\xi^2 + \xi^3\right)L \end{bmatrix} \begin{bmatrix} u \\ \theta_1 \\ \theta_2 \end{bmatrix} \tag{4.60}$$

The strain of the neutral axis of the section is defined as

$$d(x) = \begin{bmatrix} \varepsilon \\ \varphi \end{bmatrix} = \begin{bmatrix} \dfrac{\partial u}{\partial \xi} \\ \dfrac{\partial^2 \omega}{\partial x^2} \end{bmatrix} \tag{4.61}$$

where ε is the translational strain and φ is the curvature

Substitution of Equation 4.60 into Equation 4.61 gives the strain-displacement relation as

$$d(x) = \begin{bmatrix} \dfrac{1}{L} & 0 & 0 \\ 0 & \dfrac{6\xi - 4}{L} & \dfrac{6\xi - 2}{L} \end{bmatrix} \begin{bmatrix} u \\ \theta_1 \\ \theta_2 \end{bmatrix} = \mathrm{B}(x)\mathrm{q} \tag{4.62}$$

Where B(x) is the strain-displacement transformation matrix.

4.3.2.3 Element stiffness matrix

The element stiffness matrix K_b^e in the basic reference system can be expressed as

$$K_b^e = \int_0^L B^T(x)\,k(x)\,B(x)\,dx \tag{4.63}$$

The Gauss-Lobatto quadrature is used to solve the integration. It is a kind of Gaussian quadrature in which the endpoints of the interval [-1, 1] are included in a total of n integration points to yield an exact result for polynomials of degree (2n-1) or less.

The locations and weights of integration points for three-point and five-point Lobatto quadrature scheme are listed in Table 4.1 for the interval [0, 1].

Number of points	Location *x*i	Weight *w*i
3	0	0.17
	0.5	0.67
	1	0.17
5	0	0.05
	0.17	0.27
	0.5	0.35
	0.83	0.27
	1	0.05

Table 4.1: Locations and weights of Lobatto quadrature

The element stiffness matrix can be derived using Lobatto quadrature as

$$K_b^e = L \cdot \sum_i \left[B^T(x_i)\,k(x_i)\,B(x_i)\,w_i \right] \tag{4.64}$$

4.3.2.4 Element resisting force

The elemental resisting force vector is defined as

$$Q_b^e = \int_0^L B^T(x)\,D_R(x)\,dx \tag{4.65}$$

which can be obtained by integration of section resisting force vector.

$$Q_b^e = L \cdot \sum_i \left[B^T(x_i)\,D_R(x_i)\,w_i \right] \tag{4.66}$$

Figure 4.4 shows the flow chart of element state determination for stiffness method.

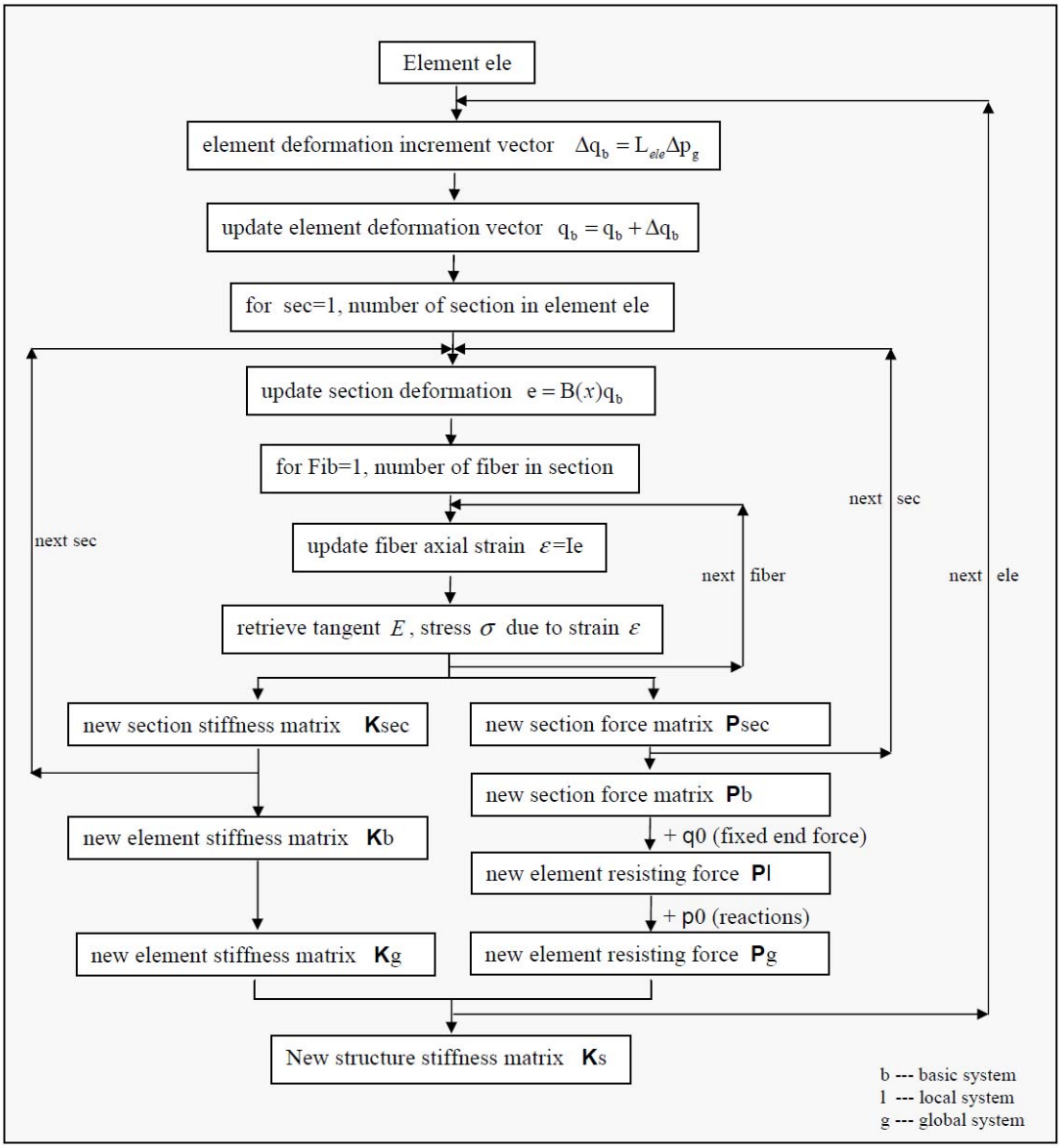


Figure 4.4: Flow chart of element state determination for stiffness method

4.3.3 Structure state determination

The elemental stiffness matrix and resisting force formed in Section 4.3.2 should be assembled into structural stiffness matrix and resisting force by using coordinate transformation matrix. There are three kinds of coordinate system such as basic system, local system and global system. The basic system is the local system without the rigid motion of the element.

Before determination of the section state the displacement in global system should be transformed into displacements in basic system using Equation 4.12. After determination of element state the elemental stiffness matrix and resisting force should be transformed from basic system to global system using Equation 4.15 and 4.17.

For geometrically linear beam, the equilibrium equation is established on the initial configuration and β_0 should be used (shown in Figure 4.1) to conduct the transformation. On the other hand, the equilibrium equation is established on the deformed configuration for geometrically nonlinear beam and β should be used (shown in Figure 4.1) for the coordinate transformation.

Figure 4.5 shows the flow chart of structural state determination of stiffness method.

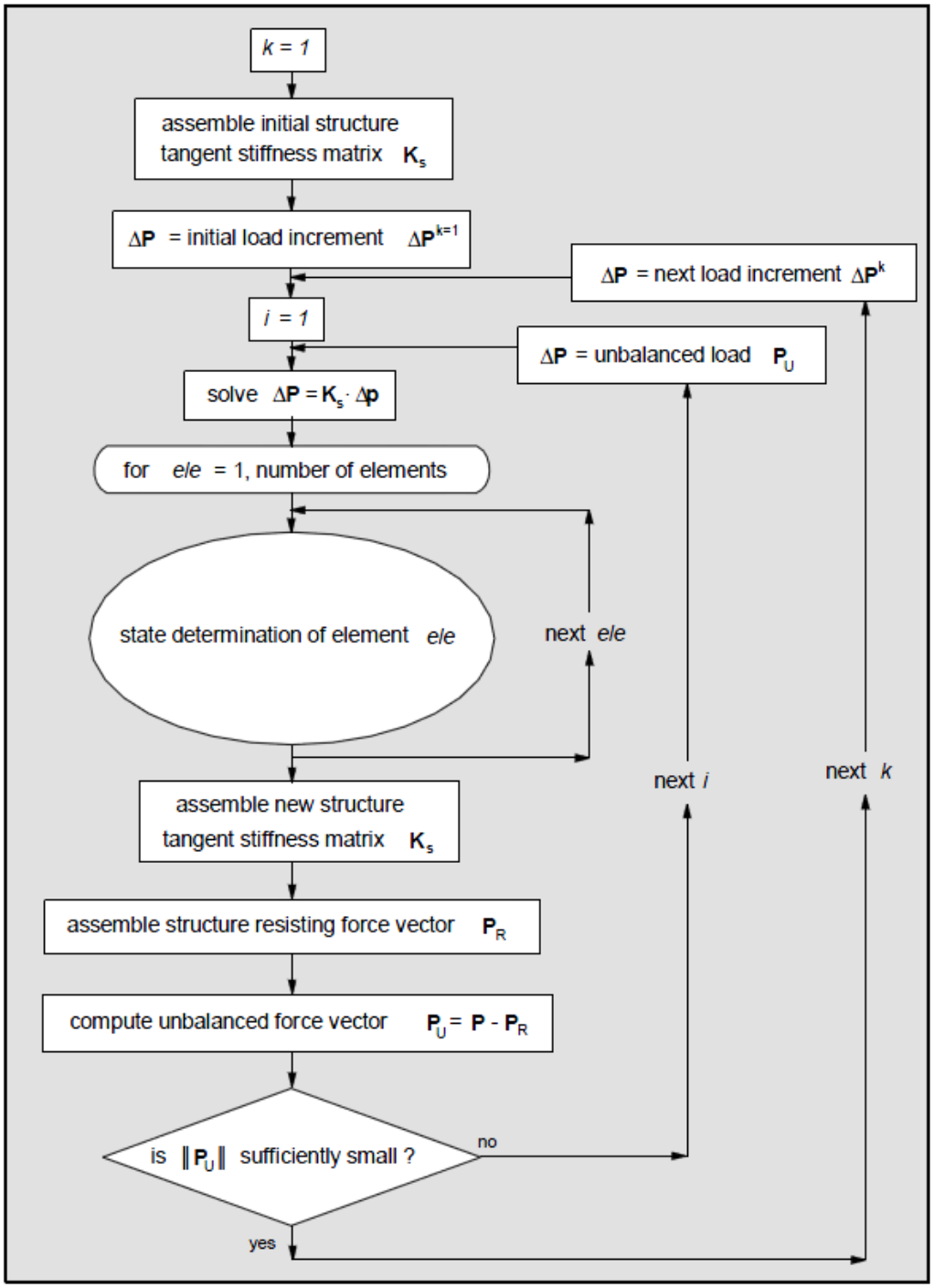


Figure 4.5: Flow chart of structural state determination (Taucer et al. 1991)

4.4 Force method theory for beam element

Different from the displacement method, the force method or flexibility-based method is based on the force interpolation function within the element. It is common to assume that the bending moment distribution inside the element is linear and that the axial force distribution is constant. This means the element equilibrium is satisfied in a strict sense. The critical issue in flexibility-based element is the implementation in an existing finite element program, which are typically based on the direct stiffness method. In general, the force method is used in the element state determination stage and stiffness method for the structural state level.

4.4.1 Section state determination

The section stiffness matrix will be transformed into section flexibility matrix as

$$f_s(x) = \left[k(x)\right]^{-1} \tag{4.67}$$

In the force method the section force field is interpolated by the elemental force described as

$$\mathrm{D}(x) = \mathrm{b}(x)\mathrm{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{x}{L}-1 & \frac{x}{L} \end{bmatrix} \begin{bmatrix} P \\ M_1 \\ M_2 \end{bmatrix} \tag{4.68}$$

It is important to point out that the equilibrium equation 4.68 is exact in a strict sense when no elemental loadings are present. This is a major advantage of the force method over the stiffness method. The force interpolation matrix b(x) is exact irrespective of the element material behaviour, while the deformation interpolation matrix is only exact in the linear elastic case.

4.4.2 Element state determination

The element residual deformation can be obtained by the integration of section residual deformation as

$$\mathrm{s} = \int_0^L \mathrm{b}^T(x) \cdot \mathrm{r}(x)\, dx \tag{4.69}$$

A comparison of the element state determination algorithm for the force and stiffness method is provided in Table 4.2.

Step	Flexibility-based algorithm	Stiffness-based Algorithm
1. Displacement increment	$\Delta q^j = -s^{j-1}$	
2. element force increment	$\Delta \mathrm{Q}^j = \left[F^{j-1} \right]^{-1} \cdot \Delta q^j$	$q^i = q^{i-1} + \Delta q^i$
3. section force increment	$\Delta \mathrm{D}^j = \mathrm{b}(x) \cdot \Delta \mathrm{Q}^j$	
4. update section force	$D^j(x) = D^{j-1}(x) + \Delta \mathrm{D}^j(x)$	
5. section deformation increment	$\Delta \mathrm{d}^j(x) = \mathrm{f}^{j-1}(x) \Delta \mathrm{D}^j(x)$	$\mathrm{d}^i(x) = B(x) q^i$
6. update section deformation	$\mathrm{d}^j(x) = \mathrm{d}^{j-1}(x) + \Delta \mathrm{d}^j(x)$	
7. update section flexibility and section force	$f^j = \left[kData^j(x) \right]^{-1}$ $D_R^j = sData^j(x)$	$k^i(x) = kData^i(x)$ $D^i(x) = sData^i(x)$
8. unbalance section force	$D_U^j(x) = D^j(x) - D_R^j(x)$	$Q^i = \int_0^L B^T(x) D^i(x) dx$
9. residual section deformation	$\mathrm{r}^j(x) = \mathrm{f}^j(x) \cdot D_U^j(x)$	$P_U^i = P_E^k - Q^i$
10. residual element displacement	$\mathrm{s}^j = \int_0^L \mathrm{b}^T(x) \cdot \mathrm{r}^j(x) dx$	$K^i = \int_0^L B^T(x) k^i(x) B(x)$
11. update element flexibility matrix	$F^j = \int_0^L b^T(x) f^j(x) b(x) dx$	$K^i \cdot \Delta q^{i+1} = P_U^i$
12. determine convergence	dw ?< tol	Δq^{i+1} ?< tol
13. if no, go to next j+1 iteration	$\Delta q^{j+1} = -s^j$	go to next i+1 N-R iteration
14. if yes, finish this i iteration		go to next k+1 load step

Table 4.2 Element state determination algorithm

The section and element state determination procedures in Table 4.2 are illustrated in Figure 4.6.

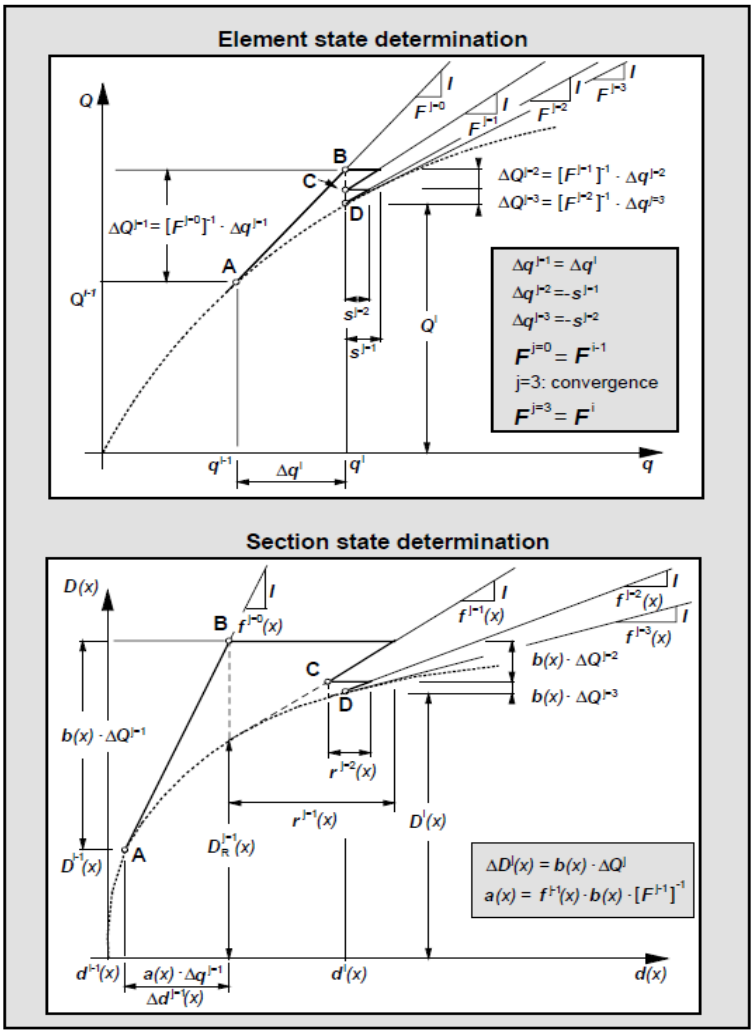


Figure 4.6: Element and section state determination for flexibility-based element (Taucer et al. 1991)

4.4.3 Structural state determination

The structural state determination is the same as the stiffness method as shown in Figure 4.5.

The force method mentioned above need two iteration regimes and has some convergence problems. Therefore the stiffness method was used to develop OpenSees to enable thermomechanical analysis capability. The development involves modification of material

classes Steel01 and Concrete02, section classes FiberSection2d and beam element class DispBeamColumn2d which is based on the stiffness method theory following coratational transformation formulation. The existing material classes Steel01 and Concrete02 do not include the effects of temperature and the modifications of these material classes focus on adding temperature dependent material properties based on the Eurocodes. The details of these modifications will be shown in the next chapter.

Chapter 5

Thermomechanical Analysis of Framed Structures in OpenSees

5.1 Introduction

Based the mechanical theory of material and beam element classes presented in Chapter 3 and Chapter 4, the development of thermomechanical analysis of framed structures in OpenSees will be presented in this chapter which involves modifications of these material, section and beam element classes to include temperature related messages and methods. New thermal load classes have been created in OpenSees to define the temperature distribution in the structural members and existing element, section, material classes have been modified to include temperature dependency. The key thermal effects on an individual structural member under various boundary conditions were studied followed by two benchmark cases to test the performance of the new developments implemented in OpenSees.

5.2 Existing mechanical analysis in OpenSees

The framework of existing mechanical analysis in OpenSees is shown in Figure 5.1. At the beginning of each load step, the external loads are applied to the element of which nodal loads are directly applied and elemental load are transferred to equivalent fixed end force by the function "addLoad()" defined in DispBeamColumn2d element. Then the iterations are conducted by the function "solveCurrentStep()" following the predefined iterative procedures such as Newton-Raphson, Modified Newton-Raphson, etc. to derive convergence once the predefined error tolerance is satisfied.

Figure 5.2 shows the methods in the function "solveCurrentStep()" applied for Newton-Raphson method. The analysis procedure is according to the stiffness method mentioned in Section 4.3. The corresponding invoking of functions in DispBeamColumn2d element is shown in Figures 5.3-5.5. At the first iteration of each load step, the out of balance force ΔF is updated given the displacements from the last committed load step and then a do-while loop is used to obtain converged results. Once the equilibrium equation is solved the displacements are updated at the element level and the deformation of the specified section at the integration point can be calculated based on the updated displacement. Then the deformation of each fibre in the section calculated based on the section deformation is sent to the material class from which the stress and elasticity modulus are sent back to calculate the section force. The elemental resisting force and stiffness matrix can then be updated by functions "getResistingForce()" and "getTangentStiff()" based on the updated section force and section stiffness.

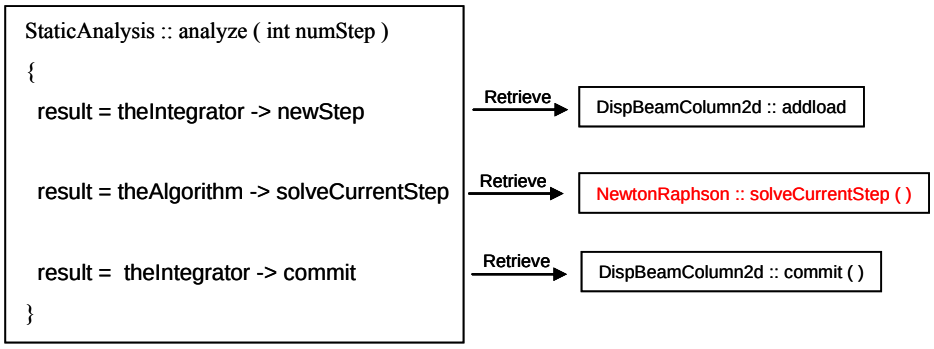


Figure 5.1: Static analysis program in OpenSees

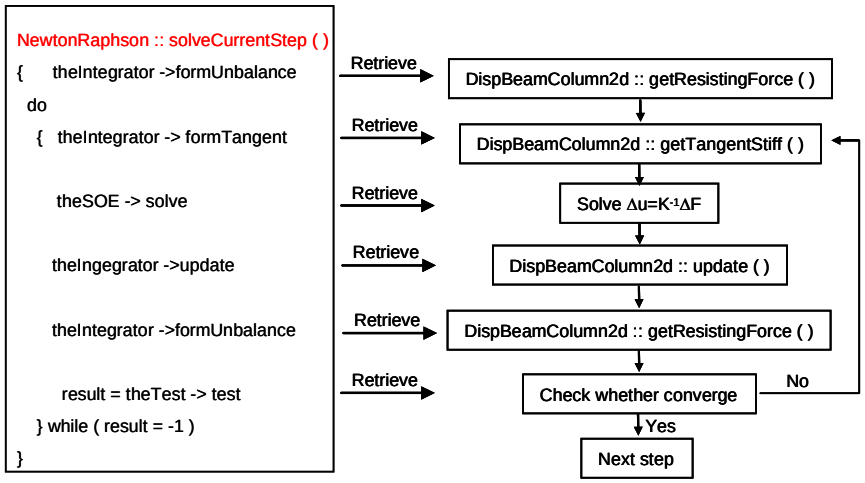


Figure 5.2 Newton-Raphson method in OpenSees

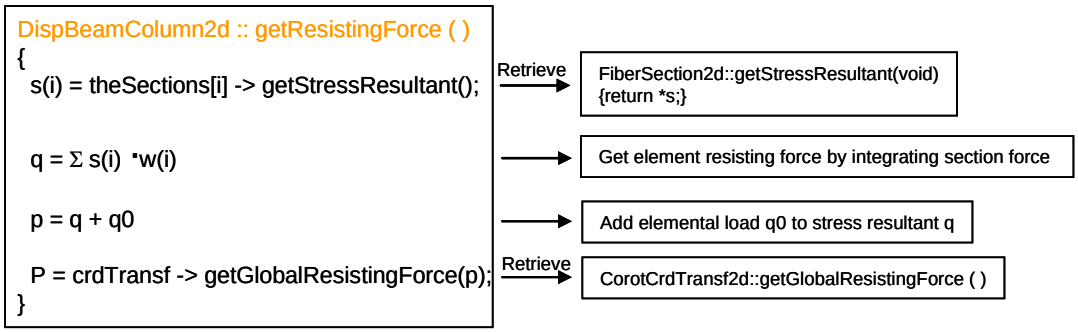


Figure 5.3: Function "getResistingForce()" in DispBeamColumn2d class

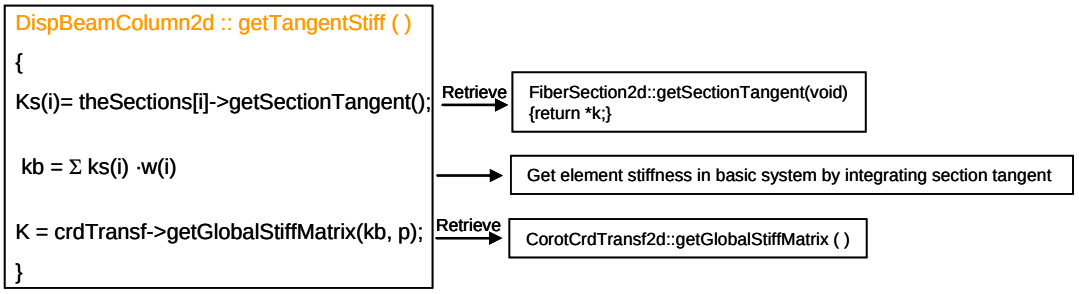


Figure 5.4: Function "getTangentStiff()" in DispBeamColumn2d class

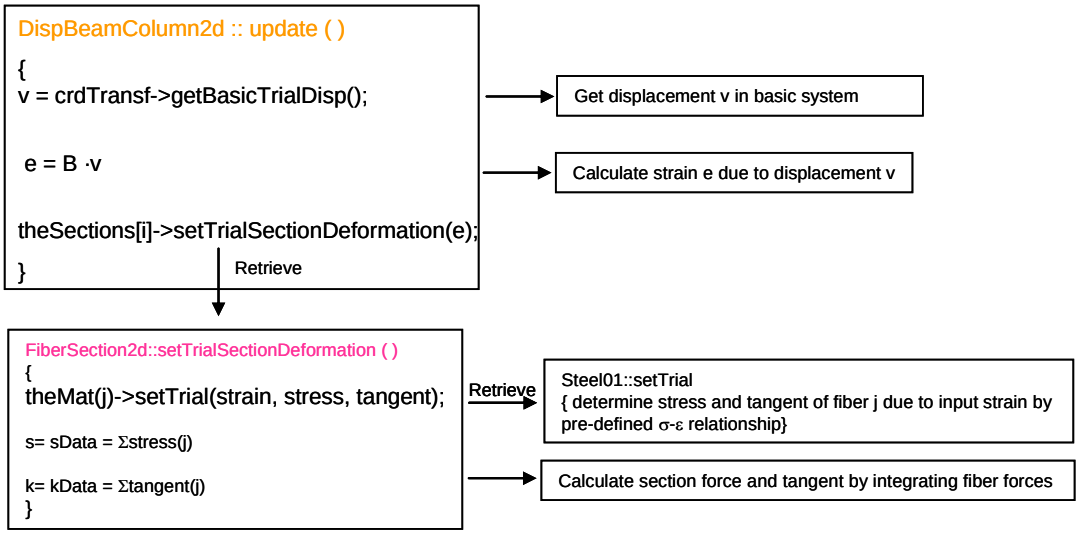


Figure 5.5: Function "update()" in DispBeamColumn2d class

Based on the existing mechanical analysis framework in OpenSees mentioned above, modifications were made to develop thermomechanical analysis capability involving creating new thermal load class, modifying existing material classes to include temperature dependent properties as well as section and element classes. These modifications follows the thermomechanical theory （Usmani 2005）which will be presented first in the next section.

5.3 Extended thermomechanical analysis of framed structures in OpenSees

Based on the existing mechanical analysis procedure, the OpenSees framework had been developed to deal with thermoemchanical analysis of framed structures. The corresponding theory is presented as follows. In an incremental-iterative nonlinear analysis, three phases can be identified: Predictor, corrector and convergence check. The predictor needs to predict an initial out of balance force and calculate the displacement increment due to this

unbalanced force given the stiffness matrix at the previous step. For thermomechanical analysis, in addition to the general external load increment, the unbalance force should include the equivalent fixed end force due to thermal load and material softening. The corrector is concerned with the recovery of element force increment from the displacement increment obtained in the predictor phase. The total strain is updated for the new geometry of the structure and the stress state can be determined by subtracting thermal strain from the total strain. The resisting force can be obtained by integrating the resisting stress along the section and used to calculate the out of balance force for this iteration. Equilibrium of the structure is checked at the end of each iteration to ensure that convergence is achieved in the new deformed configuration.

5.3.1 Predictor

The unbalance force resulting from thermal load and material softening should be calculated in the predictor phase. The thermal load can be considered as elemental load derived from the temperature distribution along the section. In the finite element analysis, the elemental load should be transformed into equivalent nodal load. Figure 5.6 shows a general fibre section, which is subdivided into longitudinal fibres, with the geometric properties and temperature conditions, as defined by a uniform temperature increment, ΔT_r, and a through-depth thermal gradient, $(T_{,z})_r$, for a given fibre, r. Thermal gradient has not been implemented in OpenSees, only mean temperature is used for simplicity, however this can conceivably be implemented in future to model very steep thermal gradients with fewer fibres (Usmani 2005). If the beam that the section belongs to is fully restrained, each fibre will have a force and moment associated with it. Integrating the forces in each fibre gives section force $F_{sec} = \left[\bar{F} \quad \bar{\mathrm{M}} \right]$ defined as

$$\bar{F} = \sum_r E_r A_r \alpha_r \Delta T_r \tag{5.1}$$

$$\bar{M} = \sum_r F_r \left(z_r - \bar{z} \right) \tag{5.2}$$

Where $\bar{z}$ is the centroid of the section given by (Usmani 2005).

$$\bar{z} = \frac{\sum_r A_r E_r \times z_r}{\sum_r A_r E_r} \tag{5.3}$$

The integration of section force along the element will gives thermally induced elemental load F_{th} (Taucer and Filippou 1991)

$$F_{th} = \int_0^l B^T(x)\, F_{\text{sec}}(x)\, dx \tag{5.4}$$

Where $B(x)$ is the strain-displacement transformation matrix (Taucer and Filippou 1991)

$$B(x) = \begin{bmatrix} \frac{1}{L} & 0 & 0 \\ 0 & \frac{6x-4L}{L^2} & \frac{6x-2L}{L^2} \end{bmatrix} \tag{5.5}$$

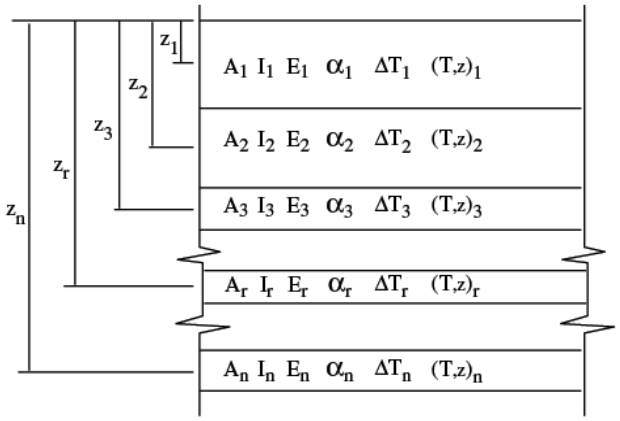


Figure 5.6: A general section divided into n fibres (Usmani 2005)

Another source of unbalanced force is the Material softening or material degradation, which means the reduction of resisting capability of the material due to the increment of the temperature. The reduction is mainly due to the degradation of the elasticity modulus and yield stress of the material at elevated temperature. The imbalance between the applied external load and declined resisting force leads to further deformation of the structure. Therefore, at the beginning of each thermal load step, the temperature-dependent material properties should be updated given current temperature and then the resisting force should be calculated again given the converged deformation at last step using the updated material properties.

The out of balance force F_{u} at the beginning of each load step is determined by

$$F_u = F_{ex} + F_{th} - F'_{re} \tag{5.6}$$

Where

F_{ex} = external load including nodal load and elemental load

F_{th} =thermal load;

F'_{re} = updated resisting force due to material softening

The initial displacement increment can then be determined by the updated out of balance force using the stiffness matrix at previous converged step.

5.3.2 Corrector

Once the initial displacement increment is obtained, iterations are needed to determine the converged displacements for the nonlinear problem. In this case, when forming the out of balance force, there is no need to consider thermal load, i.e.

$$F_u = F_{ex} - F'_{re} \tag{5.7}$$

Also, the stress state depends only on the mechanical strain

$$\varepsilon_{mechanical} = \varepsilon_{total} - \varepsilon_{thermal} \tag{5.8}$$

With these two modifications, the corrector phase of thermo-mechanical analysis can followed the general procedure of mechanical analysis of structures (Spacone and Filippou 1992). Figure 5.7 shows the flow chart of element state determination of thermo-mechanical analysis mentioned above. There are two key steps and one is to update the material properties at elevated temperature at the start of each load step. The other is to update mechanical strain by subtracting thermal strain from total strain. The implementations of these thermomechanical theory in OpenSees will be presented in the next section.

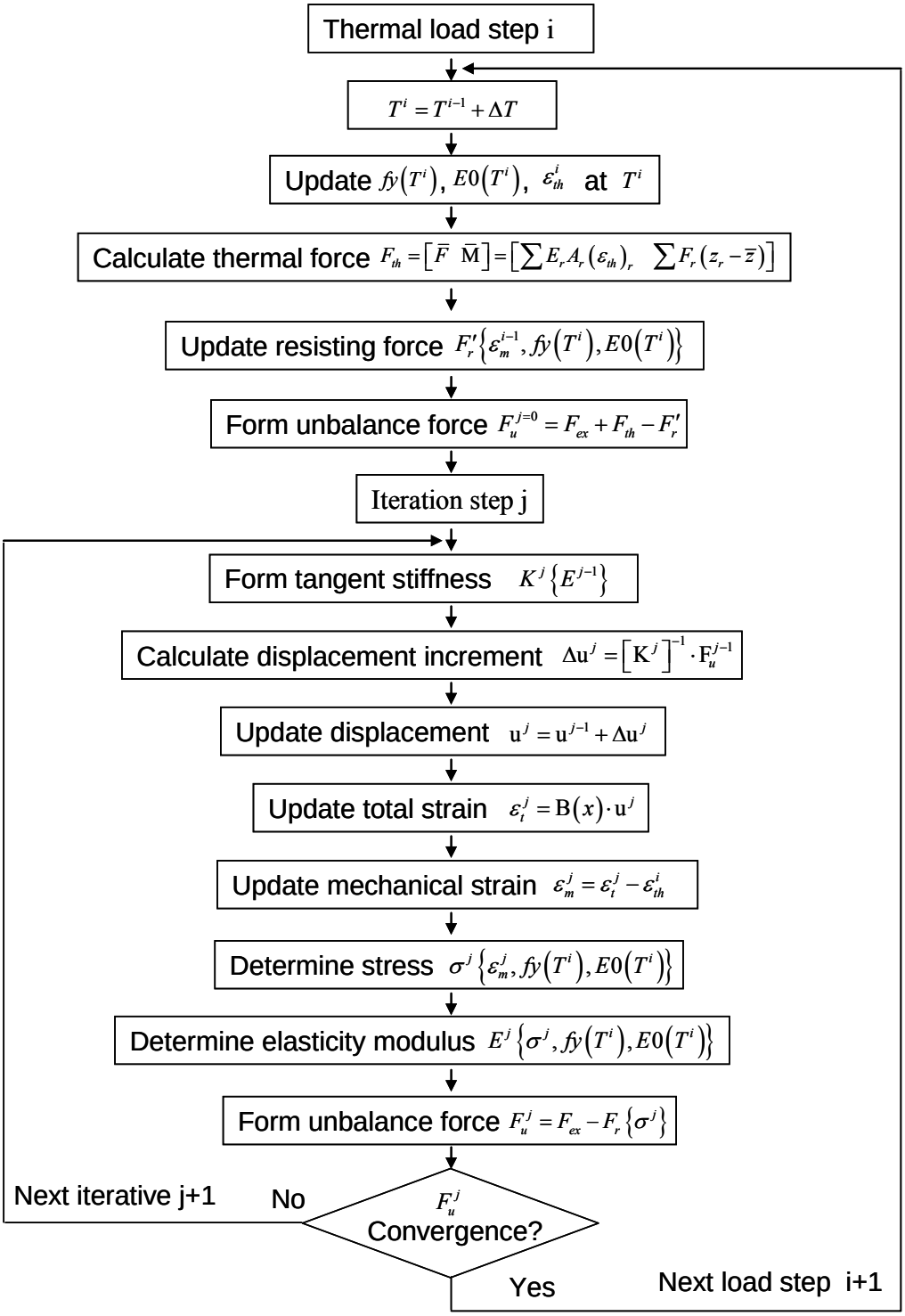


Figure 5.7: Flow chart of thermal-mechanical analysis in OpenSees

5.4 Implementation of thermo-mechanical analysis in OpenSees

In order to apply the aforementioned solution algorithm in OpenSees, a new thermal load class was created to store the temperature distribution in the structure and the existing material classes were modified to include temperature dependent properties. New functions and interfaces were added into the existing element and section classes to calculate thermally induced unbalanced force.

5.4.1 Thermal load class

The thermal analysis and structural analysis is uncoupled in OpenSees so far which means that temperature distribution along the element should be provided as input before the mechanical analysis. Parallel work is progressing on automatically generating time varying structural temperature data from a heat transfer analysis within OpenSees however direct inputs will always be required such as for modelling of experiments. Therefore new thermal load class <Beam2dThermalAction> was created to store the temperature distribution through the depth of the section defined by coordinate and corresponding temperature. The temperature of each fibre in the section will be determined by the interpolation of the temperature at the nearest coordinate point according to its location. The class diagram of this thermal load class is shown in Figure 5.8. The Beam2dThermalAction is considered as an elemental load ranked with concentrated load and uniformly distributed load.

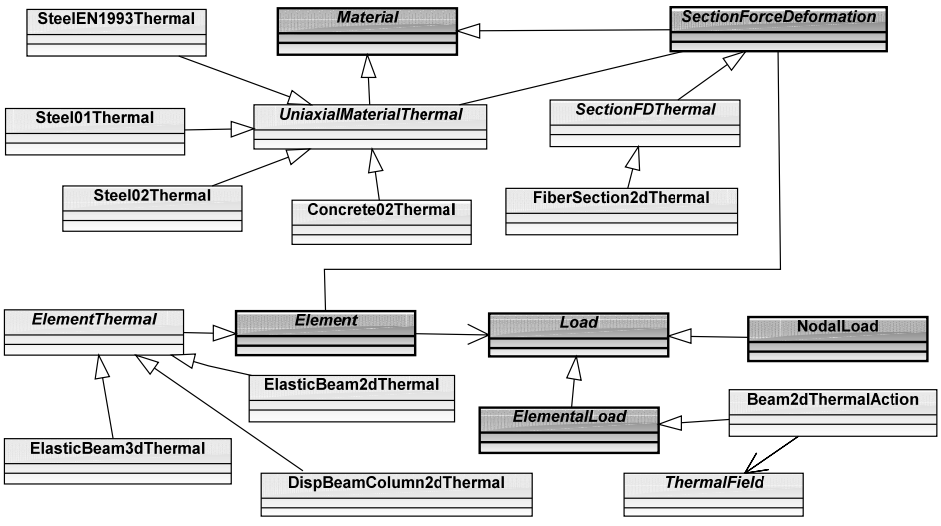


Figure 5.8: Class hierarchy of the thermal load class

The class Beam2dThermalAction, whose functions are as shown in Figure 5.9, has the following features:

1. The beam2dThermalAction is considered as a subclass of elemental load ranked with concentrated load and uniformly distributed load.

2. The constructors are defined based on 9, 5, 2 or 0 temperature points and two parameters, i.e. temperature and location, are assigned to each temperature point.

3. The constructor of the two temperature points can be used to define the linear thermal gradient and uniform temperature distribution through the depth of the section by assigning different and same temperatures at the top and bottom of the section, respectively.

```
class Beam2dThermalAction : public ElementalLoad
{
  public:
  //Constructors based on 9, 5, 2 or 0 temperature points;each
    point has two parameters, i.e. $temperature(ti) $location(locYi)
  Beam2dThermalAction(int tag,
            double t1, double locY1, double t2, double locY2,
            double t3, double locY3, double t4, double locY4,
            double t5, double locY5, double t6, double locY6,
            double t7, double locY7, double t8, double locY8,
            double t9, double locY9, int theElementTag);
  Beam2dThermalAction(int tag,
            double t1, double locY1, double t2, double locY2,
       double t3, double locY3, double t4, double locY4,
            double t5, double locY5, int theElementTag);
  Beam2dThermalAction(int tag,
   double t1,double locY1,double t2,double locY2,int theElementTag);
  Beam2dThermalAction();
  ~Beam2dThermalAction();
  const Vector &getData(int &type, double loadFactor);
  protected:
  private:
            double T1;double LocY1;double T2;double LocY2;
            double T3;double LocY3;double T4;double LocY4;
            double T5;double LocY5;double T6;double LocY6;
            double T7;double LocY7;double T8;double LocY8;
```

```
            double T9;double LocY9;
static Vector data;
};
```

Figure 5.9: Functions for Beam2DThermalAction class

5.4.2 Modified material classes

There are many types of material models available in OpenSees for steel and concrete, defining their mechanical constitutive relation, however, some of these are needed to be modified to include temperature dependent properties. At this stage temperature dependence will only be added to the unixial material models as this data is not reliably available for the multiaxial cases. The uniaxial properties at elevated temperature will be primarily based on Eurocode stipulations.

Two temperature dependent material classes <Steel01Thermal> and <Concrete02Thermal> are created by modifying existing uniaxial material class <Steel01> and <Concrete02> in OpenSees. The hierarchy of these two new material classes is shown in Figure 5.10.

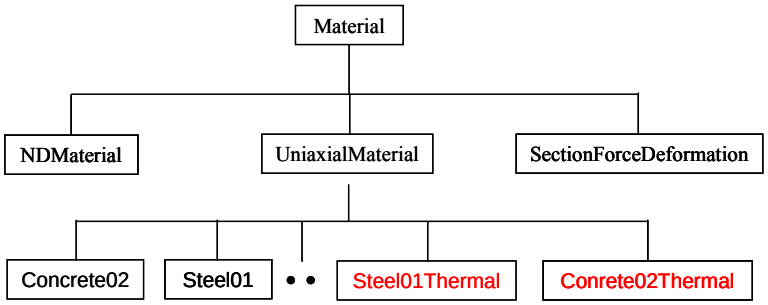


Figure 5.10: Modified material class in OpenSees

5.4.2.1 Steel material at elevated temperature

A temperature dependent steel material class <Steel01Thermal> was created based on existing steel material class <Steel01>, which has a bilinear stress-strain relationship. The yield stress and modulus of elasticity at elevated temperature were defined according to Eurocode 3. Eurocode 3 Part 1.2 defined the behaviour of steel at elevated temperature by reduction factors in a tabulated form. The mechanical behaviour of carbon steel at elevated temperature is shown in Figure 5.11. The temperature dependent mechanical properties of

the steel are determined by the effective yield strength $f_{y,T}$, proportional limit stress $f_{p,T}$ and modulus of elasticity E_T. The values of these parameters at elevated temperature are defined based on reduced factor shown in Figure 5.12.

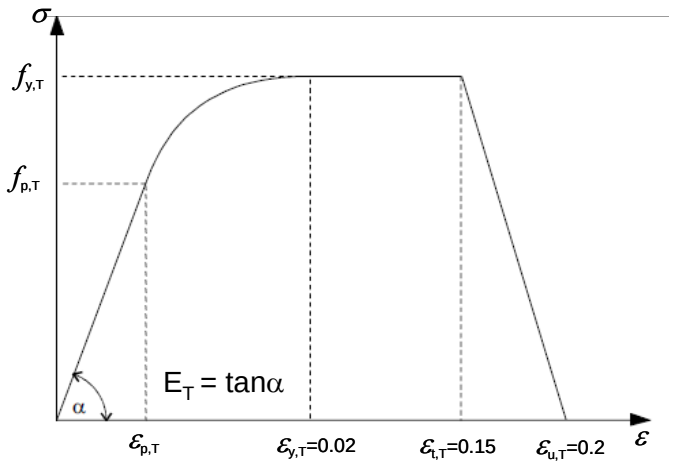


Figure 5.11: Stress-strain relationship for carbon steel at elevated temperature

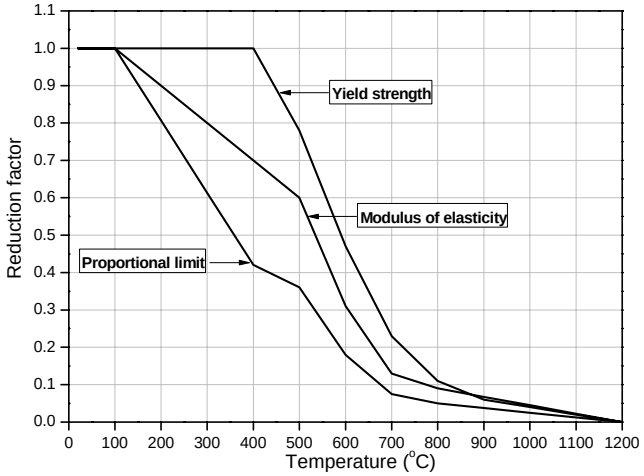


Figure 5.12: Reduction factors for mechanical properties of carbon steel at elevated temperature

The thermal elongation strain of steel ε_{sth} can be determined according to different temperature range as follows:

For $20^oC \leq T < 750^oC$

$$\varepsilon_{sth} = -2.416\times10^{-4} + 1.2\times10^{-5}T + 0.4\times10^{-8}T^2 \qquad (5.9a)$$

For 750°C ≤ T ≤ 860°C

$$\varepsilon_{sth} = 1.1\times10^{-2} \tag{5.9b}$$

For 860°C < T ≤ 1200°C

$$\varepsilon_{sth} = -6.2\times10^{-3} + 2\times10^{-5}T \tag{5.9c}$$

where T is the steel temperature.

The variation of the thermal elongation strain with temperature in Equation 5.9 can be illustrated in Figure 5.13.

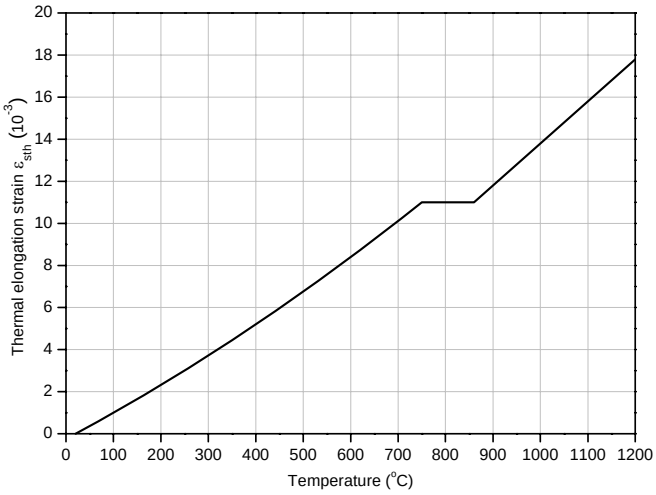


Figure 5.13: Variation of the thermal elongation strain of steel at elevated temperature

Figure 5.14 shows the functions defined in the Steel01Thermal class. Compared with the existing Steel01 class two new functions are defined in Steel01Thermal: "getThermalElongation()" and "setThermalTangentAndElongation()". The details of these two functions are shown in Figure 5.15 and 5.16. The complete version of Figure 5.16 implemented in OpenSees is presented in Appendix A.1. The yield stress and modulus of elasticity at certain temperature are interpolated from those at the adjacent bound temperatures defined in Eurocode 3 mentioned above.

```
class Steel01Thermal : public UniaxialMaterial
{
 public:
   Steel01Thermal(int tag, double fy, double E0, double b,
   double a1 = STEEL_01_DEFAULT_A1, double a2 = STEEL_01_DEFAULT_A2,
   double a3 = STEEL_01_DEFAULT_A3, double a4 =STEEL_01_DEFAULT_A4);
   Steel01Thermal();
   ~Steel01Thermal();
   const char *getClassType(void) const {return "Steel01Thermal";};
   int setTrialStrain(double strain, double strainRate = 0.0);
   int setTrial (strain,&stress, &tangent,strainRate = 0.0);
   double getStrain(void);
   double getStress(void);
   double getTangent(void);
   double getInitialTangent(void) {return E0;};

   double getThermalElongation(void);
   double setThermalTangentAndElongation(double &, double&,double&);
 protected:
 private:
    double Temp;  // total temperature increment
      double ThermalElongation; // eps(theata) = α * ΔT
    double fyT; // temperature-dependent yield stress
    double E0T; // temperature-dependent elasticity modulus
};
```

Figure 5.14: Functions for Steel01Thermal

```
double Steel01Thermal::getThermalElongation(void)
{
  return ThermalElongation;
}
```

Figure 5.15: Implementation of the function "getThermalElongation()"

```
double Steel01Thermal::setThermalTangentAndElongation(double &TempT,
double&ET, double&Elong)
{
// EN 1993 pt 1-2-1. Carbon steel at elevated temperatures
// first update yield stress and elasticity modulus
  if (TempT <= 100) {
       fy = fyT;
       E0 = E0T;
  }
  else if (TempT <= 200) {
      fy = fyT;
      E0 = E0T*(1 - (TempT - 100)*0.1/100);
  }
. . .
// then calculate thermal elongation
  else if (TempT <= 750) {
     ThermalElongation = -2.416e-4+1.2e-5*TempT+0.4e-8 *TempT*TempT;
  }
. . .

  ET = E0;
  Elong = ThermalElongation;
  return 0;
}
```

Figure 5.16: Implementation of the function "setThermalTangentAndElongation()"

5.4.2.2 Concrete material at elevated temperature

A temperature dependent concrete material class <Concrete02Thermal> (shown in Figure 5.10) was created based on existing concrete material class <Concrete02>, which was reviewed in Section 3.5.1. The properties of concrete and reinforcing material at elevated temperature are defined according to Eurocode 2. Eurocode 2 Part 1.2 includes a detailed algebraic representation of concrete behaviour (normal weight concrete with siliceous and calcareous aggregates) at elevated temperature. The compressive stress-strain relationship of concrete material is shown in Figure 5.17 defined by three temperature dependent parameters: the compressive strength $f_{c,T}$, the strain $\varepsilon_{c1,T}$ corresponding to $f_{c,T}$, and ultimate strain $\varepsilon_{cu,T}$. Figure 5.18 shows the variation of these parameters against temperature and Figure 5.19 shows the stress-strain relationship of concrete in compression at elevated temperature.

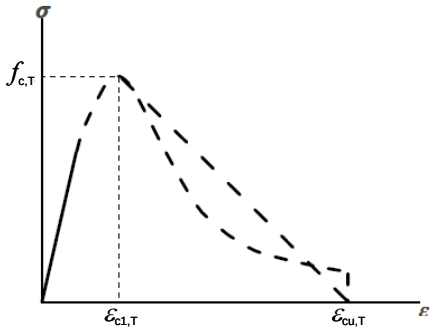
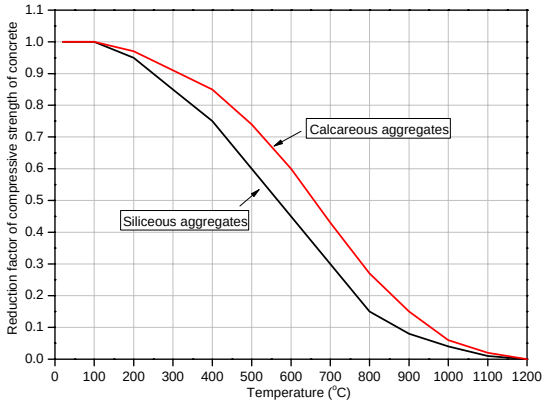
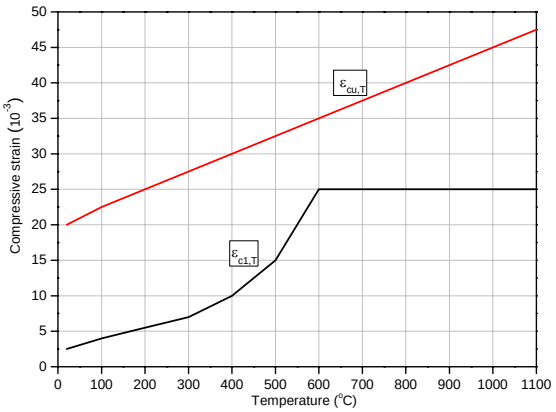


Figure 5.17: Stress-strain relationship of concrete under compression



(a) Compressive strength against temperature



(b) Compressive strain against temperature

Figure 5.18: Mechanical properties of concrete under compression at elevated temperature

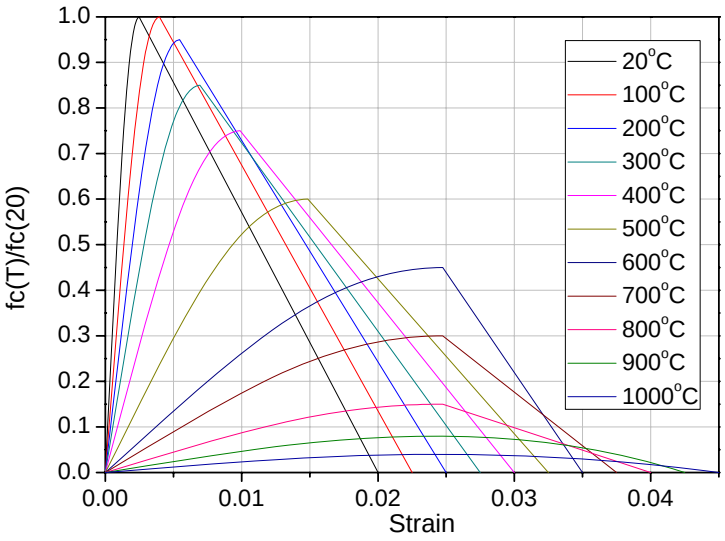


Figure 5.19: Stress-strain relation of concrete under compression at elevated temperature

Conservatively, the tensile strength of concrete should normally be ignored. The reduction of tensile strength $f_{ct,T}$ of concrete at elevated temperature is defined by the reduction factor $k_{ct}=f_{ct,T}/f_{ct}$ expressed as

For $20^oC \le T \le 100^oC$

$$k_{ct} = 1 \tag{5.10a}$$

For $100^oC < T \le 600^oC$

$$k_{ct} = 1 - \frac{T-100}{500} \tag{5.10b}$$

The thermal elongation strain of concrete ε_{cth} with siliceous and calcareous aggregates can be determined as follows:

Siliceous aggregates:

For $20^oC \le T \le 700^oC$:

$$\varepsilon_{cth} = -1.8\times10^{-4} + 9\times10^{-6}T + 2.3\times10^{-11}T^3 \quad (5.11a)$$

For $700^oC < T \leq 1200^oC$:

$$\varepsilon_{cth} = 1.4\times10^{-2} \quad (5.11b)$$

Calcareous aggregates:

For $20^oC \leq T \leq 805^oC$:

$$\varepsilon_{cth} = -1.2\times10^{-4} + 6\times10^{-6}T + 1.4\times10^{-11}T^3 \quad (5.12a)$$

For $805^oC < T \leq 1200^oC$:

$$\varepsilon_{cth} = 1.2\times10^{-2} \quad (5.12b)$$

The variation of the thermal elongation strain of concrete against temperature in Equation 5.11 and 5.12 is illustrated in Figure 5.20.

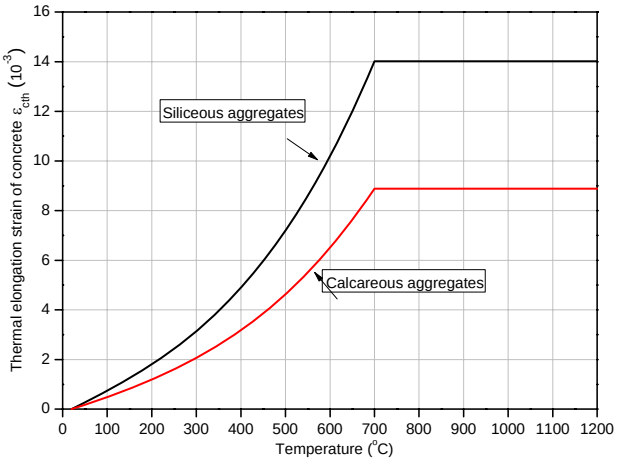


Figure 5.20: Variation of the thermal elongation strain of concrete at elevated temperature

The added functions in the Concrete02Thermal are similar with those in Steel01Thermal and the details are show in Appendix A.2.

5.4.2.3 Pinching material at elevated temperature

In section 3.5.1 a uniaxial material Pinching4 in OpenSees was introduced to model the "pinched" load-deformation response considering degradation under cyclic loading. New material classes Pinching4ThermalSteel and Pinching4ThermalConcrete were created based on this existing Pinching4 material for steel and concrete material respectively. The temperature dependent properties are according to Eurocodes (Eurocode 3 for steel and Eurocode 2 for concrete). The functions defined in these two new uniaxial material classes are shown in Appendix A.3 and A.4.

5.4.3 Section class

The existing section class <FiberSection2d> was modified to < FiberSection2dThermal> by adding a new function "getTemperatureStress()" to calculate the thermally induced section force. Figure 5.21 shows the main functions defined in the FiberSection2dThermal class. The script code of the added function "getTemperatureStress()" is shown in Figure 5.22 and its complete version is presented in Appendix B. The corresponding procedure can be explained as follows:

1. First receive the temperature distribution vector "dataMixed" from the element class.

2. Determine the temperature of each fibre by interpolation between the adjacent temperature points and retrieve the function "setThermalTangentAndElongation()" in the corresponding material class. On one hand, the material of each fibre was updated according to the temprature of the fibre. On the other hand, the value of elasticity modulus and thermal elongation at elevated temperature are sent back which can be used to calculated the thermal force. The mechanical strain of each fibre is calculated to derive the section resisting force by subtracting thermal elongation from the total strain.

3. Calculate the centroid of the fiber section using Equation 5.5 given the corresponding modulus of elasticity, area and location of each fiber.

4. The thermal force at the section is then calculated by integration through the depth of the section using Equation 5.3 and 5.4.

```
class FiberSection2dThermal : public SectionForceDeformation
{
  public:
    FiberSection2dThermal();
    FiberSection2dThermal(int tag, int numFibers, Fiber **fibers);
    FiberSection2dThermal(int tag, int numFibers, UniaxialMaterial
**mats,SectionIntegration &si);
   ~FiberSection2dThermal();
      int   setTrialSectionDeformation(const Vector &deforms);
    const Vector &getSectionDeformation(void);
      const Vector &getStressResultant(void);
      const Matrix &getSectionTangent(void);
     const Matrix &getInitialTangent(void);
const Vector &getTemperatureStress(double dataMixed[18]); // get
Ft=EA*Elongation//
. . .
private:
      int numFibers;      // number of fibers in the section
  UniaxialMaterial **theMaterials; // array of pointers to materials
    double   *matData;      // data for the materials [yloc and area]
    double   kData[4];     // data for ks matrix
    double   sData[2];    // data for s vector
  double   sTData[2];  // to store section thermal force
  Vector *sT;  // section thermal forces
  double  *TemperatureTangent; // E at elevated temperature
  double *LocElong;//thermal strain in getTemperatureStress(double *)
};
```

Figure 5.21: Functions for FiberSection2dThermal class

```
const Vector&
FiberSection2dThermal::getTemperatureStress(double *dataMixed)
{
    for (int i = 0; i < numFibers; i++) {
      UniaxialMaterial *theMat = theMaterials[i];
      //caculate the fiber tempe, T=T1-(Y-Y1)*(T1-T2)/(Y1-Y2)
    if (  fiberLocs[i] <= dataMixed[1])
    {opserr error}
    else if (fiberLocs[i] <= dataMixed [3])
    {
FiberTemperature = dataMixed [0] - (dataMixed [1] - fiberLocs[i]) *
(dataMixed [0] - dataMixed [2])/(dataMixed [1] - dataMixed [3]);
    }
. . .
    else if ( fiberLocs[i] <= dataMixed [17] ){ . . .}

theMat->setThermalTangentAndElongation(FiberTemperature, tangent,
elongation);
  ThermalTangent[i]=tangent;
  ThermalElongation[i]=elongation;
  }
//calculate centroid of section yBarT for composite section,i.e.
yBar is related to tangent E;  ybarT = ∑E_i A_i y_i / ∑E_i A_i
  double SigmaEAy = 0;
  double SigmaEA = 0;
  for (int i = 0; i < numFibers; i++) {
          SigmaEAy += ThermalTangent[i]*fiberArea[i]*fiberLocs[i];
          SigmaEA += ThermalTangent[i]*fiberArea[i];
  }
  yBarT = SigmaEAy/SigmaEA;
 // calculate section resisting force due to thermal load
  double FiberForce;
  for (int i = 0; i < numFibers; i++) {
FiberForce = ThermalTangent[i]*fiberArea[i]*ThermalElongation[i];
      sTData[0] += FiberForce;
      sTData[1] -= FiberForce*(fiberLocs[i] - yBarT);
  }
  return *sT;
}
```

Figure 5.22: Implementation of the function “getTemperatureStress()”

5.4.4 Element class

The existing stiffness-based beam element class <DispBeamColumn2d> was modified to <DispBeamColumn2dThermal>. No new member functions were created but the existing function "addLoad()" and "getResistingForce()" were modified to include thermal force and material softening. Figure 5.23 shows the main functions defined in the DispBeamColumn2dThermal class.

The implementation of the function "addLoad()" in the DispBeamColumn2dThermal class shown in Figure 5.24 can be explained as follows:

1. The load pattern class passes the temperature data to the function "addLoad()".

2. The temperature messages are then passed over to the material class by calling the function "getTemperatureStress()" in the section class which in turn calls the function "setThermalTangentAndElongation()" in the material class where the properties of the material are then updated according to this input temperature data.

3. The element thermal force can be calculated in the function "addLoad()" by integrating the section thermal force derived by the sum of fibre force given the updated stress-strain relationship of the material.

Similar with 2D beam element DispBeamColumn2dThermal, 3D beam element DispBeamColumn3dThermal was created by modifying the existing DispBeamColumn3d element in OpenSees.

```
class DispBeamColumn2dThermal : public Element
{
  public:
    DispBeamColumn2dThermal(int tag, int nd1, int nd2,
            int numSections, SectionForceDeformation **s,
            BeamIntegration &bi, CrdTransf2d &coordTransf,
            double rho = 0.0);
    DispBeamColumn2dThermal();
    ~DispBeamColumn2dThermal();
    void setDomain(Domain *theDomain);
    int addLoad(ElementalLoad *theLoad, double loadFactor);
```

```
    const Vector &getResistingForce(void);
    const Matrix &getTangentStiff(void);
private:
int numSections;
   SectionForceDeformation **theSections; // the ND material objects
   CrdTransf2d *crdTransf;// coordinate tranformation object
   static Matrix K;// Element stiffness
   static Vector P;// Element resisting force vector

    Vector Q;      // Applied nodal loads
    Vector q;      // Basic force
    double q0[3];  // Fixed end forces in basic system
    double p0[3];  // Reactions in basic system
double *dataMix; //store temperature and coordinate
double q0Temperature[3];  //Fixed end forces of tempe of current
step in basic system
double q0TemperatureP[3];  // Fixed end forces of tempe of last step
in basic system
int counterTemperature; //used to remove thermal force in the second
interation
};
```

Figure 5.23: Functions for DispBeamColumn2dThermal

```
int
DispBeamColumn2dTemperature::addLoad(ElementalLoad *theLoad, double
loadFactor)
{
  const Vector &data = theLoad->getData(type, loadFactor);
  if (type == LOAD_TAG_Beam2dUniformLoad) {. . .}
  else if (type == LOAD_TAG_Beam2dPointLoad) {. . .}
  else if (type == LOAD_TAG_Beam2dThermalAction) {
    //dataMix = data;
  for (int i = 0; i < 17; i+2)
    {
      dataMix[i] = data(i)*loadFactor;// store temperatures
    }
     for (int j = 1; j < numSections; j+2)
    {
       dataMix[j] = data(j);// store location of temperature points
    }
  counterTemperature = 0;
  q0Temperature[0] = 0.0;
  q0Temperature[1] = 0.0;
```

```
  q0Temperature[2] = 0.0;
  // Loop over the integration points
  for (int i = 0; i < numSections; i++) {
    // Get section stress resultant
    const Vector &s = theSections[i]->getTemperatureStress(dataMix);
    double si;
    for (int j = 0; j < order; j++) {
      si = s(j)*wt[i];
      switch(code(j)) {
      case SECTION_RESPONSE_P:
    q0Temperature[0] += si; break;
      case SECTION_RESPONSE_MZ:
q0Temperature[1] += (xi6-4.0)*si;
q0Temperature[2] += (xi6-2.0)*si; break;
      default:
    break;
      }}};
```

Figure 5.24: Implementation of function "addLoad()"

The complete versions of function "addLoad()" and "getResistingForce()" are presented in Appendix C.

In a word, the temperature distribution in the structure is input by thermal load class and the material properties are updated according to the corresponding temperature of each fiber. Once the equilibrium equations are solved, the displacements of the structure is upated and then the mechanical strain is updated by subtracting thermal elongation strain from the derived total strain. The robustness and stability of the developed OpenSees framework should be tested by comparing with analytical solution and other software (e.g. ABAQUS). Therefore in the next section mathematic formulas calculating the deflection of a single beam subjected to thermal gradient will be derived followed by benchmark cases which the OpenSees results are compared with analytical solution and ABAQUS results.

5.5 Benchmark testing of developed codes in OpenSees

The key features of structures in fire are presented first followed by the analytical solutions of the deflection and reaction force of an individual beam subjected to different boundary and thermal load conditions. These analytical solutions are derived to judge the performance of the developed OpenSees when temperature independent linear elastic material is used. In

section 5.5.2 and 5.5.3, the developed OpenSees is used to analyse the behaviour of a single beam under fire conditions. These cases include a fully restrained beam partly subjected to a uniform temperature increase, a single beam with finite translational and rotational end restraint subjected to uniform distributed load and a thermal gradient through the section depth. In these cases a temperature dependent elastic material is used and the OpenSees results are compared with analytical solution as well as ABAQUS.

5.5.1 Key features of thermomechanical response for a structural member

The most fundamental relationship that governs the behaviour of structures in fire is (Usmani et al. 2001):

$$\varepsilon_{total} = \varepsilon_{mechanical} + \varepsilon_{thermal} \tag{5.13}$$

with

$$\sigma = f\left(\varepsilon_{mechanical}\right), \quad \delta = f\left(\varepsilon_{total}\right) \tag{5.14}$$

The total strains $\varepsilon_{\text{total}}$ govern the deformed shape of the structure through kinematic or compatibility considerations. The stress state in the structure σ (elastic or plastic) depends only on the mechanical strains $\varepsilon_{\text{mechanical}}$.

In structural members made of high conductivity materials (e.g. steel), the temperature gradient between the two surfaces (surface exposed to fire and the opposite side) is not obvious during the fire and the uniform temperature increment in the member will dominate its structural response. For a translationally restrained beam subjected to uniform temperature increment, two kinds of responses can be expected: yielding for a stocky beam and buckling for a slender beam. The yielding or buckling response of the beam is because of the high compressive force restored in the beam due to the restrained thermal expansion (Usmani 2005).

In contrast for structural members made of low conductivity materials (e.g. concrete), the surfaces exposed to fire will be at a much higher temperature than the surfaces on the opposite side. This causes the exposed surfaces to expand more than the unexposed surfaces

which lead to curvature in the member. This effect is called thermal bowing and is one of the main reasons for the deformation of concrete slabs and masonry walls in fire. Another important reason for thermal bowing in composite members is the large difference between the temperatures of the steel beam and concrete slabs.

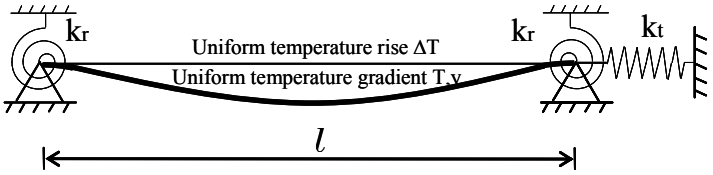


Figure 5.25: Schematic of a beam model with finite end restraints

Figure 5.25 shows a general beam model with finite rotational and translational restraints at both ends to simulate the boundary condition in real structures. The beam is subjected to thermal gradient through the depth of the section. The deflection of the beam will depend on the values of the stiffness of the end restraint. The equilibrium equation of the beam can be found to be

$$\frac{d^2y}{dx^2} = \phi + \frac{Py}{EI} - M_r \tag{5.15}$$

Where $P = \dfrac{EA\left(\alpha\Delta T - \varepsilon_\phi\right)}{\left(1 + \dfrac{EA}{k_t l}\right)}$; $M_r = \dfrac{EI\alpha T_{,y}}{\left(1 + \dfrac{2EI}{k_r l}\right)}$; ε_ϕ is the contraction strain due to the thermal bowing effect (Usmani 2005).

Many variations of mean temperature and gradient exist in structural elements under different heating regimes thus many displacement and force patterns will also exist. Figure 5.26 shows the various displacement configurations which could occur in an axially restrained pinned-end beam. When there is only a mean temperature applied there are compressive forces in the beam and the deflection response is that of the pre-buckling, post-buckling shape. As a thermal gradient is incorporated the deflected shape becomes smoother and the magnitude of the deflections increase. Compressions are also absorbed resulting in a possible zero stress case. Where large thermal gradients exist the beam is in tension.

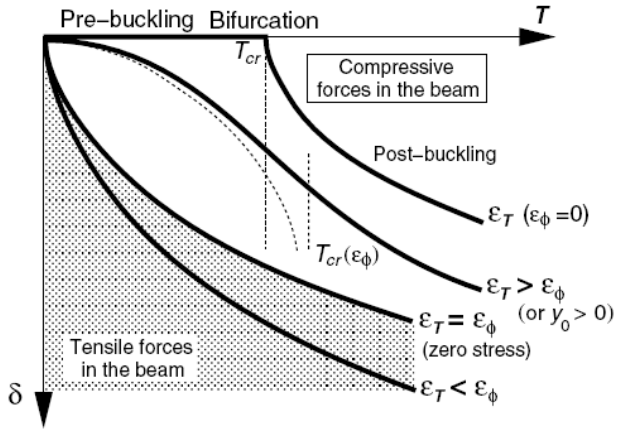


Figure 5.26: Temperature deflection responses for combination of ε_T and ε_ϕ (Usmani et al. 2001)

The closed form solution of structural response of a single beam element subjected to external load and thermal load is presented as follows.

5.5.1.1 Combined UDL and uniform temperature rise

First consider a beam pinned supported at both ends subjected to a uniformly distributed load q (UDL) and uniform temperature rise ΔT (as shown in Figure5.27). The restrained thermal expansion will cause compressive reaction force P in the supports.

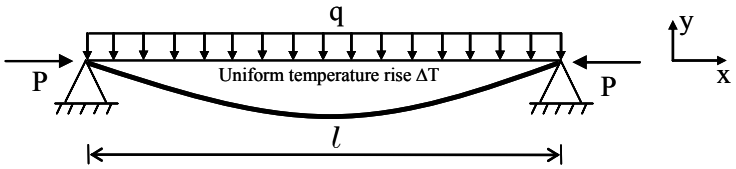


Figure 5.27: Pinned-end beam subjected to UDL and ΔT

The equilibrium equation of the beam is

$$\frac{d^2y}{dx^2} = -M_q(x) - \frac{Py}{EI} \tag{5.16}$$

where $M_q(x)$ represents the bending moment of cross section of the beam at coordinate x defined as

$$M_q(x) = \frac{ql}{2}x - \frac{q}{2}x^2 \tag{5.17}$$

The solution of Equation 5.16 is

$$y(x)=\frac{q}{EIk^4}\left[\frac{1-\cos kl}{\sin kl}\sin kx+\cos kx+\frac{k^2}{2}\left(x^2-lx\right)-1\right] \tag{5.18}$$

Where $k=\sqrt{P/EI}$.

The maximum deflection at the mid span of the beam can be derived from Equation 5.18 by letting $x=l/2$ as

$$y_{mid}=\frac{q}{EIk^4}\left[\frac{1-\cos kl}{\sin kl}\sin\frac{kl}{2}+\cos\frac{kl}{2}-\frac{k^2l^2}{8}-1\right] \tag{5.19}$$

To solve Equation 5.19, the axial force P should be determined first. However P is also a function of the deflection w as it results from the combined effects of restrained thermal expansion and extension developed in the beam due to the large deflection.

The extension of the beam produced by the deflection $y(x)$ is equal to the difference between the length of the arc l_{arc} along the deflection curve and the chord length l. If the sin curve of deformation shape is assumed, the extension strain $\varepsilon_e = (l_{arc} - l)/ l$ can be calculated as (Usmani et al. 2001).

$$\varepsilon_e=\sqrt{\frac{1}{2}\left(\frac{\pi y_{mid}}{l}\right)^2+1}-1 \tag{5.20}$$

The axial force in the support can then be written as

$$P=EA\left[\varepsilon_T-\varepsilon_e\right] \tag{5.21a}$$

or

$$P=EA\left[\alpha\Delta T-\sqrt{\frac{1}{2}\left(\frac{\pi y_{mid}}{l}\right)^2+1}+1\right] \tag{5.21b}$$

The uniform temperature rise ranges from 0-1000 °C and a constant modulus of elasticity E is assumed. The analytical results were derived by iteratively solving Equation 5.19 and 5.21. The properties of the beam mode are listed in Table 5. 1. Figures 5.28 and 5.29 show the comparison of mid-span deflection and axial reaction force in the beam subjected to UDL and ΔT between the analytical and OpenSees results. Good agreement for the mid-span deflection is achieved between analytical and OpenSees results. However obvious difference can be seen for the horizontal reaction force in the support in Figure 5.29. The difference may result from the assumed sine curve of deformation shape compared with a circular arc of the real deformation shape. This difference is considered to be obvious when deflection of the beam becomes large at high temperature. This difference is also seen in Figure 5.32 in Section 5.5.1.2.

L (m)	E (N/m^2)	α (/°C)	A (m^2)	Height of section (m)	UDL (kN/m)
6m	2×10^{11}	12e-6	0.02	0.2	100

Table 5.1: Properties of the beam model subjected to UDL and ΔT

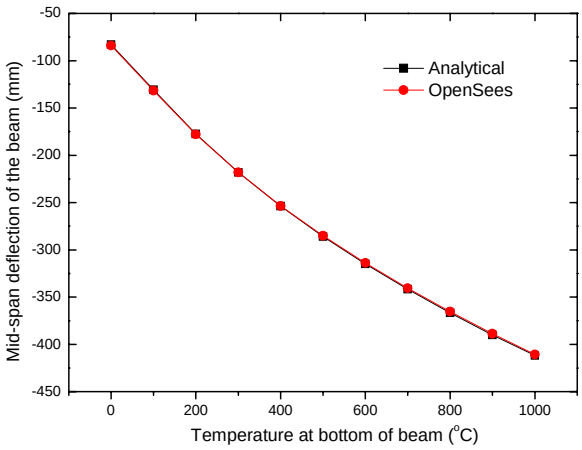


Figure 5.28: Mid-span deflection of pinned-end beam subjected to UDL and Δ*T*

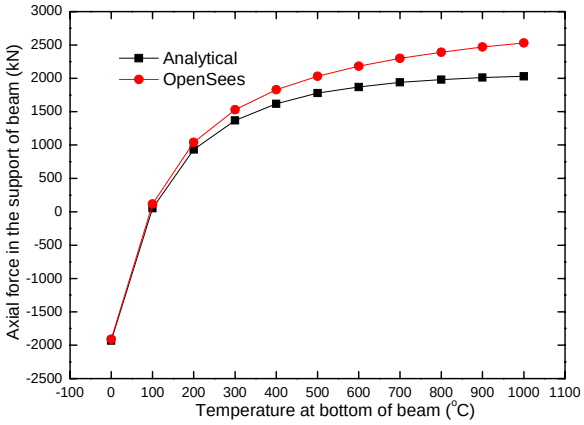


Figure 5.29: Axial restraining force of pinned-end beam subjected to UDL and ΔT

5.5.1.2 Combined UDL and pure thermal gradient

Figure 5.30 shows a pinned-end beam subjected to UDL and pure thermal gradient $T_{,y}$. The "pure" means there is no resultant uniform temperature rise in the beam and the thermal gradient only causes bending of the beam with no horizontal reaction force in the support. There is tension in the support. The contraction effect due to the pure thermal gradient causes tensile reaction force P in the supports.

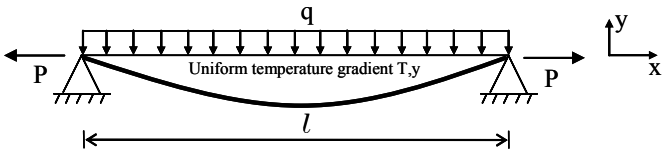


Figure 5.30: Pinned-end beam subjected to UDL and $T_{,y}$

The equilibrium equation of the beam can expressed as

$$\frac{d^2y}{dx^2} = -\phi - M_q(x) + \frac{Py}{EI} \tag{5.22}$$

The solution of the equation is

$$y(x) = \frac{q - EI\phi k^2}{EIk^4}\left[\frac{1-\cosh kl}{\sinh kl}\sinh kx + \cosh kx + \frac{qk^2}{2\left(q - EI\phi k^2\right)}\left(lx - x^2\right) - 1\right] \tag{5.23}$$

The maximum deflection at the mid span of the beam is

$$y_{mid} = \frac{q - EI\phi k^2}{EIk^4}\left[\frac{1-\cosh kl}{\sinh kl}\sinh\frac{kl}{2} + \cosh\frac{kl}{2} + \frac{qk^2l^2}{8\left(q - EI\phi k^2\right)} - 1\right] \tag{5.24}$$

The axial reaction force in the support results from the combined effect of thermal contraction and extension developed in the beam due to the large deflection. Therefore the axial force P in the support can then be written as

$$P = EA\left[\varepsilon_\phi + \varepsilon_e\right] \tag{5.25a}$$

or

$$P = EA\left[\sqrt{\frac{1}{2}\left(\frac{\pi y_{mid}}{l}\right)^2 + 1} - \frac{\sin\frac{l\phi}{2}}{\frac{l\phi}{2}}\right] \tag{5.25b}$$

The same beam model as listed in Table 5.1 was used and the temperature at top of the beam was assumed to be 0^{o}C and it varied linearly over the depth of the beam to temperatures at the bottom of 100^{o}C to 1000^{o}C. In OpenSees the pure thermal gradient is realized by closing the interfaces account for the thermal expansion due to uniform temperature rise. Figure 5.31 and 5.32 shows the comparison of mid-span deflection and axial reaction force in the beam subjected to UDL and $T_{,y}$ between the analytical and OpenSees results. Reasonable agreement is achieved between OpenSees and analytical results.

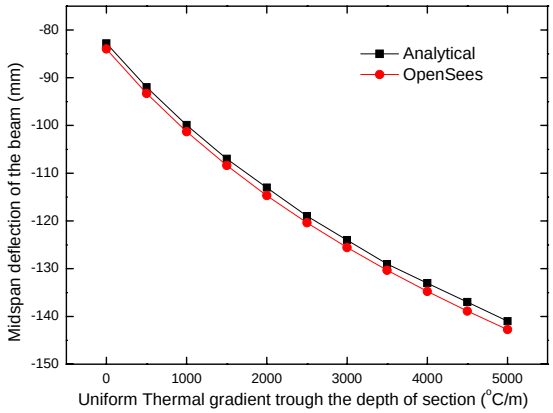


Figure 5.31: Mid-span deflection of pinned-end beam subjected to UDL and $T_{,y}$

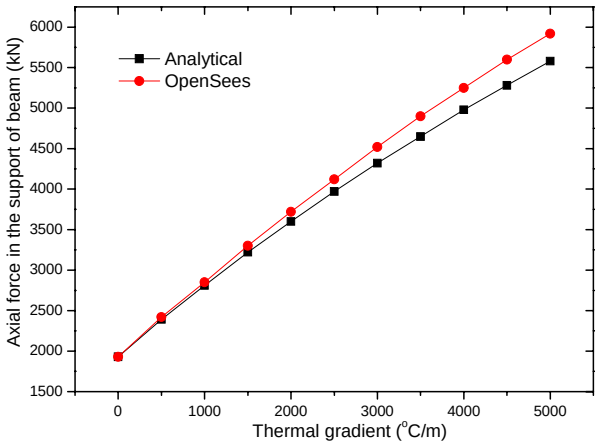


Figure 5.32: Axial restraining force of pinned-end beam subjected to UDL and $T_{,y}$

5.5.1.3 Combined UDL and thermal gradient with uniform temperature rise

Figure 5.33 shows a pinned-end beam subjected to UDL, ΔT and pure thermal gradient $T_{,y}$. In this case, the thermal expansion is assumed to dominate the structural behaviour and compressive reaction force is caused due to the dominant thermal expansion effect.

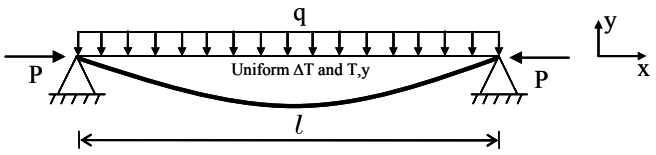


Figure 5.33: Pinned-end beam subjected to UDL, ΔT and $T_{,y}$

The equilibrium equation of the beam in Figure 5.33 can be expressed as

$$\frac{d^2y}{dx^2} = -\phi - M_q(x) - \frac{Py}{EI} \tag{5.26}$$

The solution of the equation is

$$y(x) = \frac{q + EI\phi k^2}{EIk^4}\left[\frac{1-\cos kl}{\sin kl}\sin kx + \cos kx + \frac{qk^2}{2\left(q + EI\phi k^2\right)}\left(x^2 - lx\right) - 1\right] \tag{5.27}$$

The maximum deflection at the mid span of the beam is

$$y_{mid} = \frac{q + EI\phi k^2}{EIk^4}\left[\frac{1-\cos kl}{\sin kl}\sin\frac{kl}{2} + \cos\frac{kl}{2} - \frac{qk^2l^2}{8\left(q + EI\phi k^2\right)} - 1\right] \tag{5.28}$$

The axial reaction force in the support results from the combined effect of thermal expansion, thermal contraction and extension developed in the beam due to the large deflection. Therefore the axial force P in the support can then be written as

$$P = EA\left[\varepsilon_T - \varepsilon_\phi - \varepsilon_e\right] \tag{5.29a}$$

or

$$P = EA\left[\alpha\Delta T - \sqrt{\frac{1}{2}\left(\frac{\pi y_{mid}}{l}\right)^2 + 1} + \frac{\sin\frac{l\phi}{2}}{\frac{l\phi}{2}}\right] \tag{5.29b}$$

Figure 5.34 shows the comparison of mid-span deflection in the beam subjected to UDL, ΔT and $T_{,y}$ between the analytical and OpenSees results. The OpenSees results agrees well with analytical solution. The same beam model as listed in Table 5.1 was used and the temperature at top of the beam was assumed to be 0^{o}C and it varied linearly over the depth of the beam to temperatures at the bottom of 100^{o}C to 1000^{o}C.

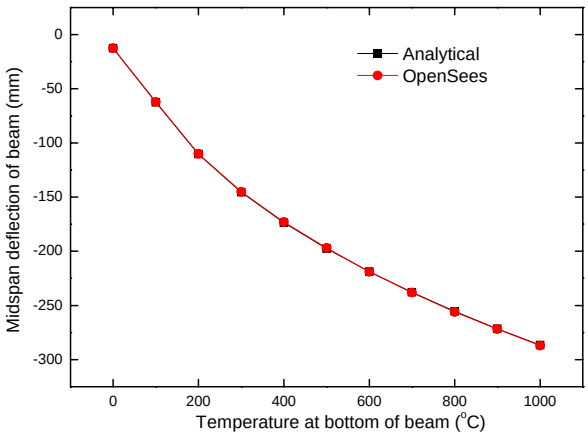


Figure 5.34 Mid-span deflection of pinned-end beam subjected to UDL, ΔT and $T_{,y}$

5.5.2 Restrained beam under thermal expansion

Figure 5.35 shows a 2m beam, only the left half of which is subjected to a uniform temperature increment from 0°C to 800°C. The right half of the beam keeps ambient temperature and acts as a translational spring to restrain the displacement of the left part. Two elements are used in the model. Temperature dependent elastic material was assumed and the properties are taken from Eurocode 3 shown in Figure 5.12. The initial modulus of elasticity at 0°C is 200GPa and a constant coefficient of thermal expansion α=12×10^{-6}/°C is assumed.

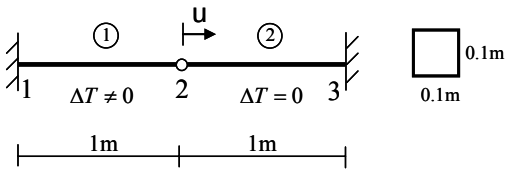


Figure 5.35: Schematic of restrained beam subjected to uniform temperature rises

The horizontal displacement of mid point 2 can be calculated analytically as (Usmani 2005)

$$u = \frac{E(T)A\alpha\Delta T}{\frac{E_0 A}{l} + \frac{E(T)A}{l}} \tag{5.30}$$

Where E_0 and $E(T)$ are the elasticity modulus at ambient and elevated temperature, respectively.

The horizontal displacement of node 2 against temperature is shown in Figure 5.36. The OpenSees result agrees well with analytical solution from Equation 5.30. The command scripts of OpenSees model is presented in Appendix D. Node 2 displaces towards the right driven by thermal expansion until 500 ^{o}C and then begins to move back as the decreasing of modulus of elasticity in the left element is unable to resist the stored strain energy and elastic rebound of the unheated right element.

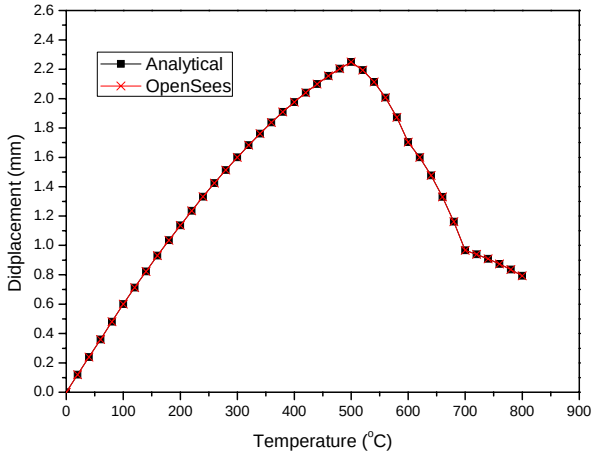


Figure 5.36: Horizontal displacement of midpoint of the beam against temperature

5.5.3 Single beam with finite boundary conditions

Figure 5.37 shows a 2D single beam with finite end restraints which is subjected to a uniformly distributed load (UDL) and linear thermal gradient along the section height. The finite end restraints are represented by translational and rotational springs with constant stiffnesses K_t and K_r, respectively. Different boundary conditions can be achieved by setting values of K_t and K_r. Table 5.2 shows six most practical boundary conditions of beam elements in steel frames. OpenSees was used to analyses all six cases and the results were compared with ABAQUS. The command scripts of OpenSees model is presented in Appendix E.

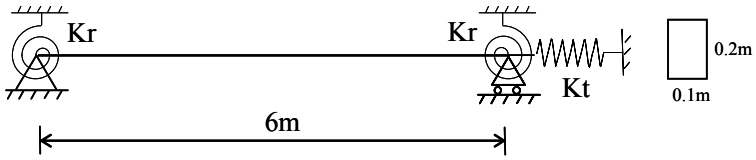


Figure 5.37: Schematic of a single beam with translational and rotational spring at the end

	1	2	3	4	5	6
						
K_t	0	>0	∞	∞	∞	>0
K_r	0	0	0	>0	∞	>0

Table 5.2: Various boundary conditions of beam in the test

The temperature at top of the beam was assumed to be 0°C and it varied linearly over the depth of the beam to temperatures of 100°C to 1000°C. The same material properties as in Section 5.5.2 are used, the details are listed in Table 5.3.

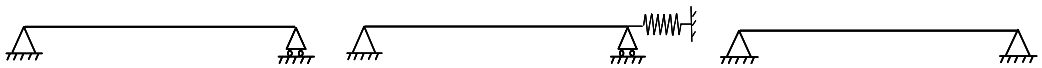
Length (m)	Area (m^2)	E (0°C) (N/m^2)	UDL (N/m)	Thermal gradient (°C/m)	α (/°C)	K_t (N/m)	K_r (N.m/rad)
6	0.02	2e11	1000	500-5000	12×10^{-6}	6.7e8	3e6

Table 5.3: Input parameters of the single beam model

Three cases were studied as follows involving beams with translational restraint alone, with rotational restraint alone and with both translational and rotational restraints. Nonlinear analysis was carried out using OpenSees (using corotational transformation to conduct the geometrically nonlinear analysis) and then compared with ABAQUS results.

5.5.3.1 Case1 Beams with translational end restraint

The influence of various translational end restraints, as shown in Figure 5.38, is presented here. Selected displacements of the beams are shown in Figure 5.39-5.41 where the results from OpenSees agree well with those of ABAQUS.



(a) (b) (c)

Figure 5.38: Schematic of beams with different translational end restraint: (a) free end; (b) spring end; (c) pinned end

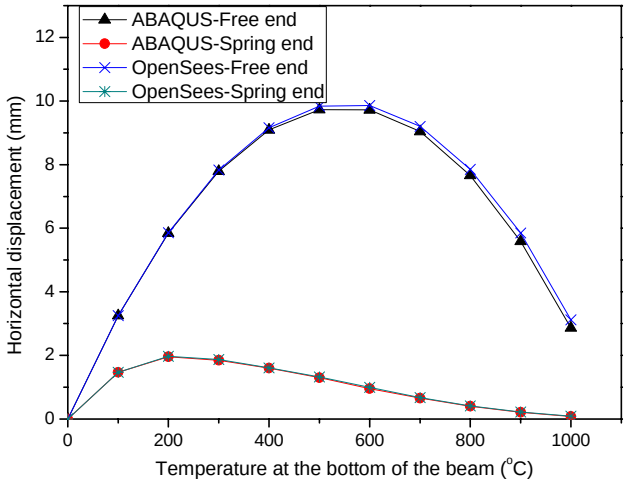


Figure 5.39: Horizontal displacement of beams with different translational end restraints

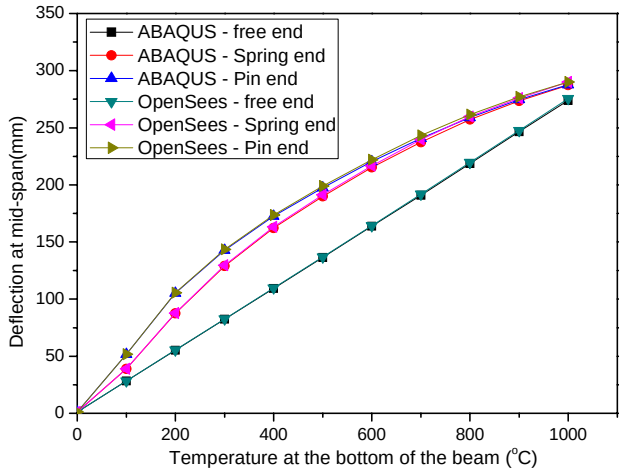


Figure 5.40: Vertical mid-span deflection of beams with different translational end restraints

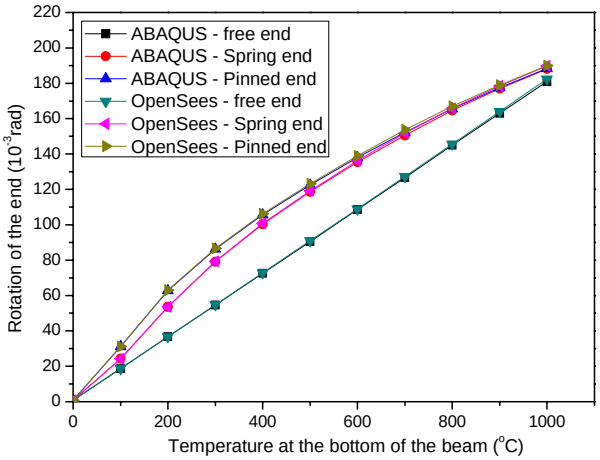


Figure 5.41: End rotation of beams with different translational end restraints

The horizontal displacement of the spring end follows a similar pattern to the example in Section 5.5.2. At the beginning of heating, the beam expands as temperature increases which is driven by the thermal elongation effect considering the increasing uniform temperature of the beam and moves back because of increasing thermal gradient which "pulls" the movable end back as well as losing resisting capability due to the material degradation at elevated temperature. The beam with spring end moves back earlier than the beam with free end. This is because the spring pushes the moveable end back when the beam with weakened material at elevated temperature can not resist the force stored in the spring. It is obvious that the translational restraint dominates the structural behaviour. The vertical deflection at mid-span of beam arises from a combination of the UDL, which will produce additional deflection given degrading material, the thermal gradient and the additional $P-\delta$ moment along the beam caused by the end restraints. Figure 5.42 shows relative deflections at the mid-span for different combination of causes, illustrating that thermal gradient provides the greatest contribution to deflection.

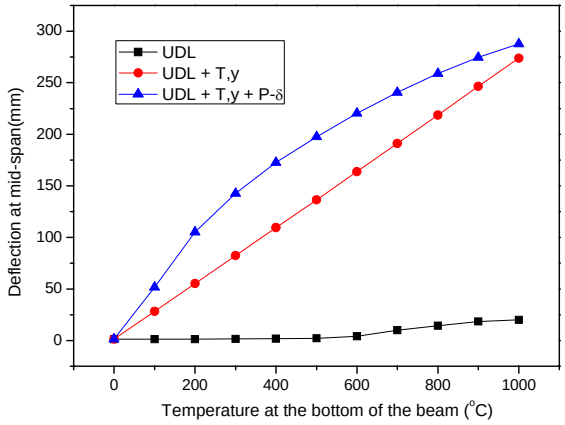


Figure 5.42: Mid-span vertical deflection contributions from different causes

The axial force in the beams with end-restraints is plotted against temperature in Figure 5.43. Produced by restrained thermal expansion, the axial force increases until approximately 200 ^{o}C before declining. At this stage there is very little change in material properties the thermal bowing induced curvature cancels some of the restrained thermal expansion and indirectly leads to the reduction in axial forces.

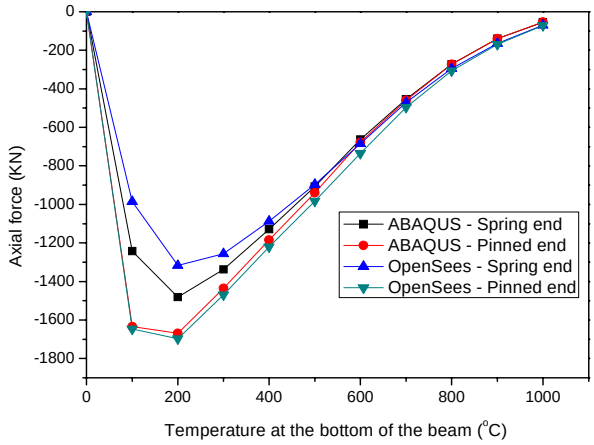


Figure 5.43: Axial force in beams with different translational end restraints

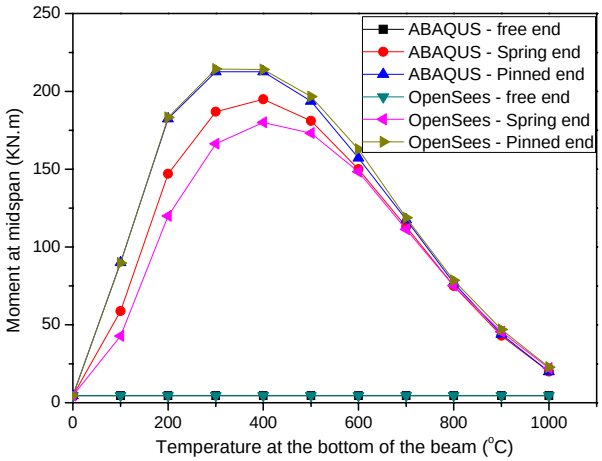


Figure 5.44: Moment at mid-span of beams with different translational end restraints

The mid-span moment in all the beams is shown in Figure 5.44. The restrained beams develop increasing $P-\delta$ moment until 400°C, after which it declines because of the reducing axial forces even though the deflection keeps increasing. At 1000°C, the axial force drops down to nearly zero and the mid-span moment in restrained beam declines to nearly the same value as the UDL only induced moment in the free-end beam. It is therefore obvious that the translational restraint dominates the structural behaviour over midrange temperatures (between 200-400°C) resulting in high axial forces and mid-span moment in end-restrained beams.

5.5.3.2 Case2 Beams with different rotational end restraint

Models of beams with different rotational end restraint are shown in Figure 5.45.

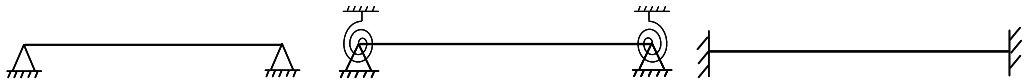
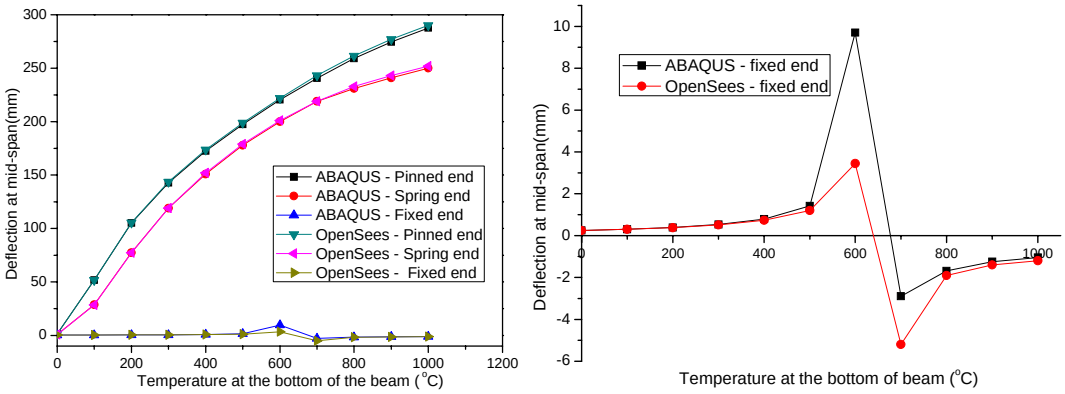


Figure 5.45: Schematic of beams with different rotational end restraint: (a) pinned end; (b) spring end; (c) fixed end

Figure 5.46 shows that the deflection at mid-span of finite restrained beam was much larger than that in fixed-end beam. This is because, for the fully fix ended beam subjected to thermal gradient $T_{,z}$, an equal and opposite curvature induced by the support moments cancels out the thermal curvature and, therefore, the fixed ended beam remains 'straight' with a constant moment $M=EI\alpha T_{,z}$ along its length. However unstable behaviour of the beam can be seen between temperatures of 500°C and 800°C of Figure 5.46(b). At 600°C the beam bends downward to its peak deflection (because of thermal bowing) and then snaps through to the opposite direction. This "thermal snap through" is driven by the additional hogging moment that occurs in the beam resulting from the centre of stiffness of the beam section moving upwards (due to the greater material degradation in the bottom) as the temperature increases, which creates an "eccentricity" for the axial forces and produces a moment opposite to the once caused by thermal bowing. Both ABAQUS and OpenSees results reproduce this effect, however as this represents an abrupt and unstable transition stage, the exact magnitudes of the displacements (which are of the order of a few mm compared to the beam depth of 200mm) are not a measure of program accuracy.

The end rotation for the spring-end beam in Figure 5.47 again shows an increase in rotation followed by reduction due to material softening.



(a) (b)

Figure 5.46: Vertical deflection of beams with different rotational end restraints: (a) vertical deflection at mid-span; (b) detail of fixed-end beam in (a)

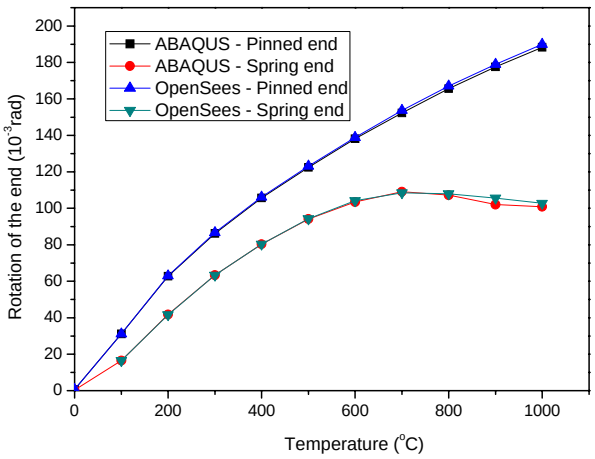


Figure 5.47. End rotation of beams with different rotational end restraint

Figure 5.48 shows that higher axial forces result from higher rotational end restraints as expected. Figures 5.49 and 5.50 show mid-span and end moments in beams with rotational restraint stiffnesses ranging from pinned end to fixed end.

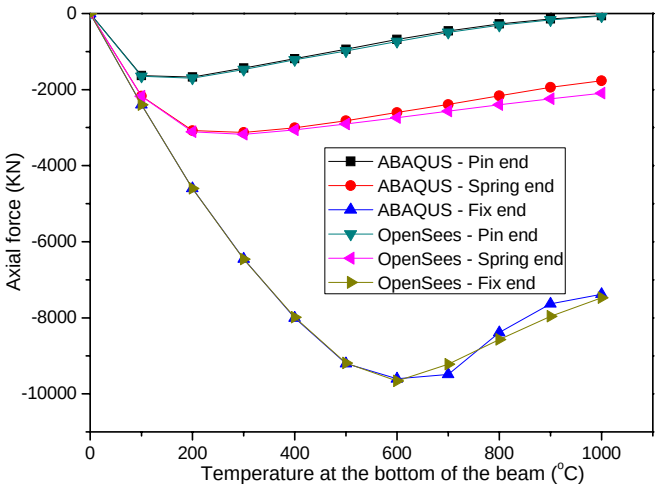


Figure 5.48: Axial force of beams with different rotational end restraint

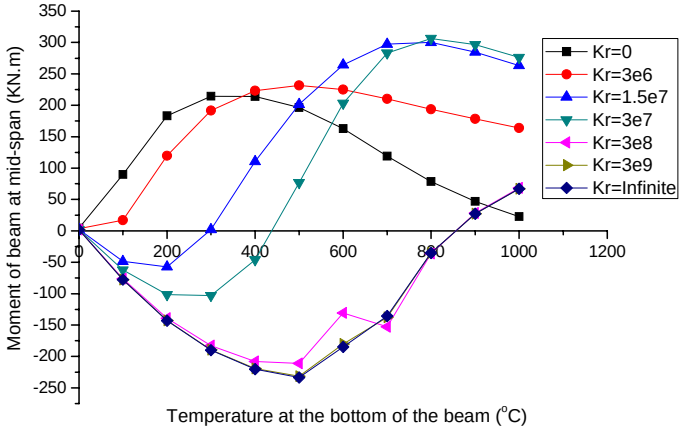


Figure 5.49: Moment at mid-span of beams with various end rotational stiffness

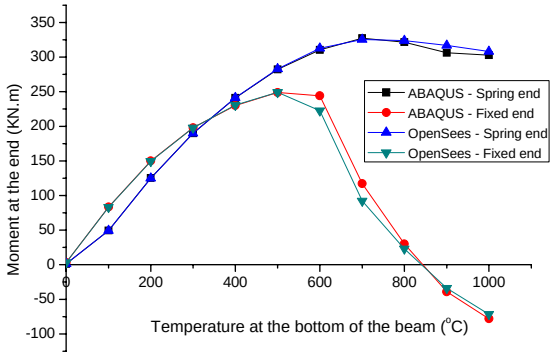


Figure 5.50: Reaction Moment at the left end of beam for fixed and spring end restraints

5.5.3.3 Case3 Beams with translational and rotational end restraint

Figure 5.51 shows a single beam with both translational and rotational end restraints represented by constant stiffness K_t and K_r, respectively. The beam is subjected to UDL and linear thermal gradient along the section height. The responses of the beam are shown in Figure 5.52-5.54. Figure 5.52 shows the horizontal displacement of right end of the beam. The movable end does not move back as much as that shown in Figure 5.39 because the rotational restraint reduces the deflection of the beam which causes the pulling back effect on the beam movable end. The mid-span deflection of the beam in Figure 5.53 is smaller than that of the beam with spring end in Figure 5.40 due to the restraint of rotational spring at the two ends. The reduction in the deflection of the beam makes the horizontal displacement of the right end of the beam not move back too much after the peak compared with that in Figure 5.39.

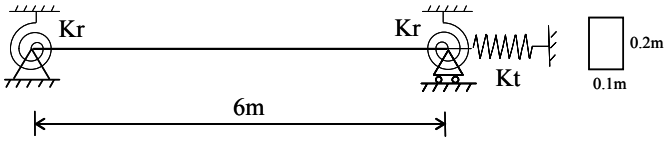


Figure 5.51: Schematic of a single beam with translational and rotational spring at the end

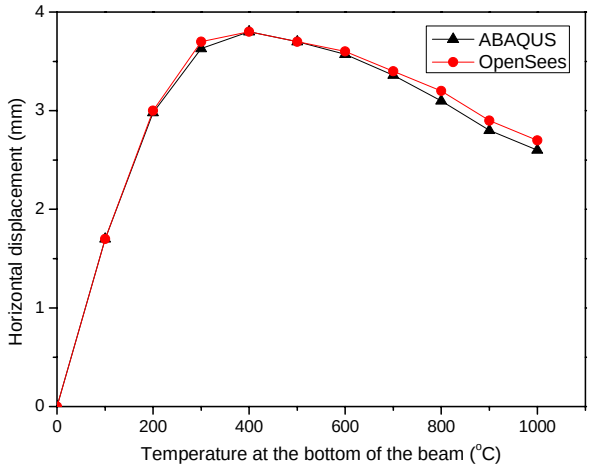


Figure 5.52: Horizontal displacement of beams with translational and rotational restraints

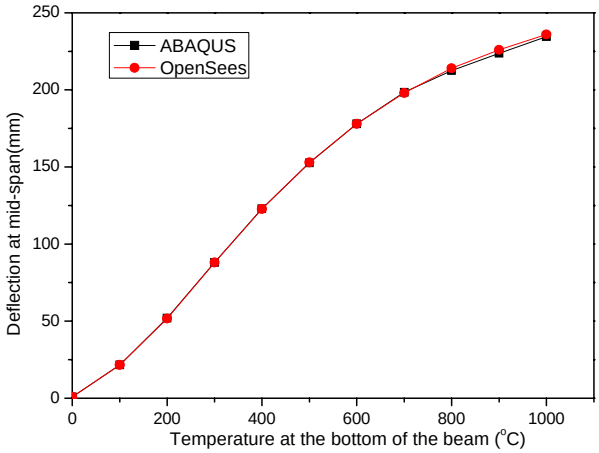


Figure 5.53: Vertical mid-span deflection of beams with translational and rotational restraints

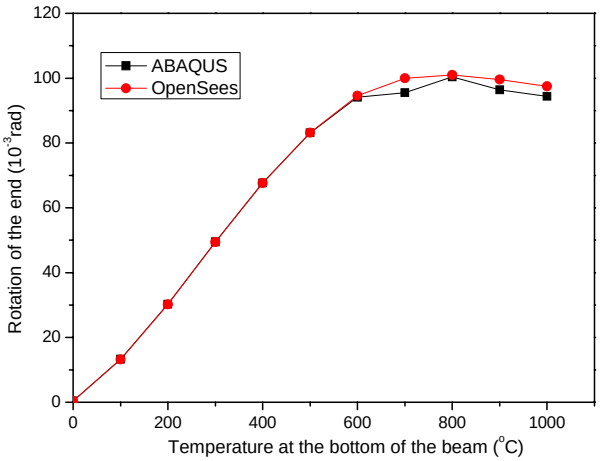


Figure 5.54: End rotation of beams with translational and rotational restraints

The development of OpenSees to deal with thermomechancial analysis of structures involves creating a new thermal load class and modifying existing material, section and beam element classes. The performance of the developed two-dimensional thermomechanical model in OpenSees has been verified by benchmark tests comparing with analytical solution and ABAQUS results. Further validation (both 2D and 3D model) will be presented in Chapter 8 by comparing OpenSees predictions with experimental measurements. In addition to the 2D model in OpenSees, a 3D thermomechanical analysis model is developed in OpenSees which involves creating new geometrically nonlinear shell element and new 3D beam/column elements modified based on the existing correspondences. The 3D beam/column elements in this model follow similar modification process with that mentioned in this chapter. Therefore in the next two chapters the development of a geometrically nonlinear shell element in

OpenSees will be present with its theory presented in Chapter 6 and applications in OpenSees shown in Chapter 7.

Chapter 6

Nonlinear Total Lagrangian Shell Element

6.1 Introduction

In order to develop a three-dimensional theromechanical model in OpenSees, a geometrically nonlinear shell element had been developed based on the existing linear shell element in OpenSees. This chapter presents the formulation of this new shell element with geometrical nonlinearity following the Total Lagrangian formulation. Its application in OpenSees will be presented in the next chapter.

6.2 Geometrically nonlinear shell element following Total Lagrangian Formulation

New geometrically nonlinear shell element ShellMITC4GNThermal is created by modifying the exiting linear shell element ShellMITC4 in OpenSees. This new shell element is a flat four-node isoparametric element including the drilling degree of freedom formed by a combination of membrane element and Mindlin plate bending element. These modifications followed the Total Lagrangian procedure with a simplified Green strain. The shell element is subdivided into several layers shown in Figure 6.1. However the layers are not uniformly distributed and their locations are pre-defined through the thickness of the element according to Lobatto integration scheme which includes points at the top and bottom of the element. The elemental resisting force and stiffness matrix are calculated by integrating the section located at 2×2 Gauss integration points. The section force and stiffness matrix were integrated from the pre-defined Lobatto integration points located through the thickness of the element.

6.2.1 Kinematics

Consider first a four-node isoparametric element of geometric 2×2 (for which the local x, y coordinates could coincide the natural coordinate ξ, η) shown in Figure 6.1, Bilinear shape functions are used to interpolate both coordinates and displacements of a generic point within the element from nodal coordinates and displacements respectively. The element has six degrees of freedom per node $\mathrm{u} = \left\{ u \quad v \quad w \quad \theta_x \quad \theta_y \quad \theta_z \right\}^T$ (three translational displacements u, v, w and three rotations θ_x, θ_y, θ_z as shown in Figure 6.1(b)).

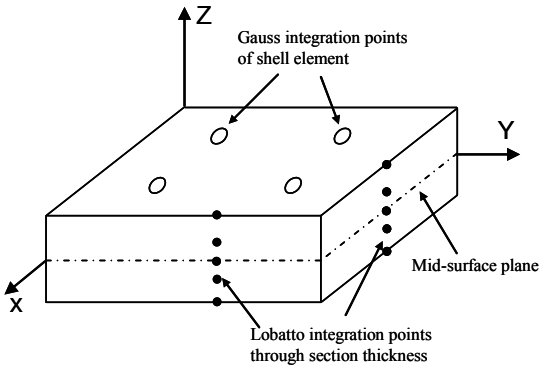
$$x = \sum_{i=1}^{4} N_i x_i \text{ , and } y = \sum_{i=1}^{4} N_i y_i \tag{6.1a}$$

$$u = \sum_{i=1}^{4} N_i u_i \text{ ; } v = \sum_{i=1}^{4} N_i v_i \text{ and } w = \sum_{i=1}^{4} N_i w_i \text{ ;} \tag{6.1b}$$

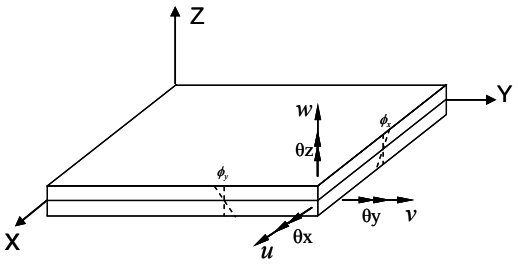
$$\theta_x = \sum_{i=1}^{4} N_i \theta_{xi} \text{ ; } \theta_y = \sum_{i=1}^{4} N_i \theta_{yi} \text{ and } \theta_z = \sum_{i=1}^{4} N_i \theta_{zi} \tag{6.1c}$$

where N_i is the bilinear shape function of the rectangular four-node element defined as

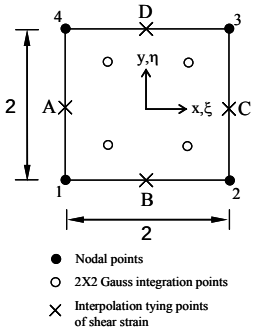
$$N_1 = \frac{(1-\xi)(1-\eta)}{4}; N_2 = \frac{(1+\xi)(1-\eta)}{4}; N_3 = \frac{(1+\xi)(1+\eta)}{4}; N_4 = \frac{(1-\xi)(1+\eta)}{4} \tag{6.2}$$



(a) Integration scheme of shell element



(b) Nodal degrees of freedom



(c) Geometry of 2x2 element

Figure 6.1 Geometry of a four-node shell element in the *x*, *y* plane

Based on the Reissner-Mindlin plate theory, the displacement components u', v', w' of a generic point in the element with coordinate *x*, *y*, *z* can be expressed by their corresponding mid-surface displacements *u*, *v*, *w* and rotations ϕ_x, ϕ_y as

$$
\begin{aligned}
u'(x,y,z) &= u(x,y) - z\phi_x(x,y);\\
v'(x,y,z) &= v(x,y) - z\phi_y(x,y);\\
w'(x,y,z) &= w(x,y).
\end{aligned}
\tag{6.3}
$$

Where ϕ is the rotation of section normal to the mid-surface of the shell.

It is convenient to use the physical rotation ϕ of a mid-surface normal to express the strain-displacement relation, which can be later transformed into θ by

$$
\begin{Bmatrix} \phi_x \\ \phi_y \end{Bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{Bmatrix} \theta_x \\ \theta_y \end{Bmatrix} \tag{6.4}
$$

A simplified Green strain can be expressed as

$$
\varepsilon = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = \begin{Bmatrix} u'_{,x} + \frac{1}{2}\left(w'_{,x}\right)^2 \\ v'_{,y} + \frac{1}{2}\left(w'_{,y}\right)^2 \\ u'_{,y} + v'_{,x} + w'_{,x}w'_{,y} \\ u'_{,z} + w'_{,x} \\ v'_{,z} + w'_{,y} \end{Bmatrix} \tag{6.5}
$$

where comma represents the derivative with respect to x or y.

By substitution of Equation 6.3, the membrane part in Equation 6.5 can be expressed as

$$
\varepsilon^M = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} u_{,x} + \frac{1}{2}\left(w_{,x}\right)^2 - z\partial\phi_x/\partial x \\ v_{,y} + \frac{1}{2}\left(\overline{w}_{,y}\right)^2 - z\partial\phi_y/\partial y \\ u_{,y} + v_{,x} + w_{,x}w_{,y} - z\left(\partial\phi_x/\partial y + \partial\phi_y/\partial x\right) \end{Bmatrix} \tag{6.6}
$$

Substituting of the interpolation function of displacements, the membrane strain-displacement relationship can now be derived as

$$
\mathrm{d}\varepsilon^M = \sum_{i=1}^{4} \mathbf{B}_i^M \, \mathrm{d}\mathbf{u}_i \tag{6.7}
$$

in which $\mathbf{B}_i^M$ is the strain-displacement matrix in forms of

$$\mathbf{B}_i^M = \begin{bmatrix} N_{i,x} & 0 & N_{i,x}N_{j,x}w_j & 0 & zN_{i,x} & 0 \\ 0 & N_{i,y} & N_{i,y}N_{j,y}w_j & -zN_{i,y} & 0 & 0 \\ N_{i,y} & N_{i,x} & \left(N_{i,x}N_{j,y}+N_{i,y}N_{j,x}\right)w_j & -zN_{i,x} & zN_{i,y} & 0 \end{bmatrix} \tag{6.8}$$

The MITC technique was used to form the shear strain to avoid the shear locking problems. The key formulation step is the replacement, in the potential energy principle, of selected displacement-related strains by independently assumed strain fields in element natural coordinates. The transverse shear strain was interpolated from the displacement-dependent strains defined at the mid-side of element edges as shown in Figure 6.1(c) as

$$\gamma = \begin{Bmatrix} \gamma_{\xi z} \\ \gamma_{\eta z} \end{Bmatrix} = \frac{1}{2}\begin{Bmatrix} (1-\eta)\gamma_{\xi z}^B + (1+\eta)\gamma_{\xi z}^D \\ (1-\xi)\gamma_{\eta z}^A + (1+\xi)\gamma_{\eta z}^C \end{Bmatrix} \tag{6.9}$$

where $\gamma_{\eta z}^A$, $\gamma_{\eta z}^C$ $\gamma_{\xi z}^B$, $\gamma_{\xi z}^D$ are the physical shear strains at points A, B, C, and D as shown in Figure 6.1(c) evaluated by the general shear strain defined in Equation 6.5.

For the element shown in Figure 6.1, the MITC shear strain can be written as

$$\gamma_{\xi z} = \frac{1}{2}(1-\eta)\left(\frac{w_2 - w_1}{2} - \frac{\phi_{x1}+\phi_{x2}}{2}\right) + \frac{1}{2}(1+\eta)\left(\frac{w_3 - w_4}{2} - \frac{\phi_{x3}+\phi_{x4}}{2}\right) \tag{6.10a}$$

$$\gamma_{\eta z} = \frac{1}{2}(1-\xi)\left(\frac{w_1 - w_4}{2} - \frac{\phi_{y1}+\phi_{y4}}{2}\right) + \frac{1}{2}(1+\xi)\left(\frac{w_3 - w_2}{2} - \frac{\phi_{y2}+\phi_{y3}}{2}\right) \tag{6.10b}$$

Based on Equation 6.9 the shear strain-displacement matrix can be defined as

$$\mathrm{d}\gamma = \sum_{i=1}^{4} \mathbf{B}_i^S \mathrm{d}u_i \tag{6.11}$$

A combined strain-displacement matrix **B** can now be derived by assembly of membrane and shear matrix of Equation 6.8 and 6.11 as

$$\mathrm{d}\varepsilon = \begin{Bmatrix} \mathrm{d}\varepsilon^M \\ \mathrm{d}\gamma \end{Bmatrix} = \sum_{i=1}^{4} \begin{bmatrix} \mathbf{B}_i^M \\ \mathbf{B}_i^S \end{bmatrix} \mathrm{d}u_i = \mathrm{B}\mathrm{d}u \tag{6.12}$$

The derivatives of the shape function with respect to x and y in strain-displacement matrix **B** are not available directly and they can be transformed from those with respect to ξ and η by Jocobian matrix [*J*] defined as

$$\begin{bmatrix} N_{i,x} \\ N_{i,y} \end{bmatrix} = [J]^{-1} \begin{bmatrix} N_{i,\xi} \\ N_{i,\eta} \end{bmatrix} \tag{6.13}$$

where

$$[J] = \begin{bmatrix} x_{,\xi} & y_{,\xi} \\ x_{,\eta} & y_{,\eta} \end{bmatrix} = \begin{bmatrix} \sum N_{i,\xi} x_i & \sum N_{i,\xi} y_i \\ \sum N_{i,\eta} x_i & \sum N_{i,\eta} y_i \end{bmatrix} \tag{6.14}$$

A fictitious stiffness k_θ is assigned to the drilling degree of freedom θ_z and it can be incorporated into the total potential energy in a penalty manner with k_θ being the penalty parameter.

$$\Pi_{drill} = \frac{1}{2} \int_V k_\theta \left(\omega - \theta_z \right)^2 \mathrm{d}V \tag{6.15}$$

where ω is the physical in-plane rotation of shell defined as

$$\omega = \frac{1}{2} \left(v_{,x} - u_{,y} \right) \tag{6.16}$$

and the corresponding strain-displacement matrix can be written as

$$\mathrm{d}\left(\omega - \theta_z \right) = \mathrm{B}^{dr} \mathrm{d}u \tag{6.17}$$

in which

$$\mathrm{B}^{dr} = \left[-\frac{1}{2} N_{i,y} \quad \frac{1}{2} N_{i,x} \quad 0 \quad 0 \quad 0 \quad -N_i \right] \tag{6.18}$$

So far the strain-displacement matrix is defined at local coordinate system and it can be transformed into global coordinate by transformation matrix T before the calculation of stiffness matrix. The transformation matrix T is defined as

$$\mathrm{u}=\mathrm{T}\bar{\mathrm{u}} \tag{6.19}$$

where u and $\bar{\mathrm{u}}$ are the displacements in the local and global coordinate system, respectively.

The strain-displacement matrix of Equation 6.12 and 6.17 can be written in the global system as

$$\bar{\mathrm{B}} = \mathrm{BT}\,;\ \bar{\mathrm{B}}^{dr} = \mathrm{B}^{dr}\mathrm{T} \tag{6.20}$$

where T is defined as

$$\mathrm{T} = \begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda \end{bmatrix};\ \Lambda = \begin{bmatrix} g_1(0) & g_1(1) & g_1(2) \\ g_2(0) & g_2(1) & g_2(2) \\ g_3(0) & g_3(1) & g_3(2) \end{bmatrix} \tag{6.21}$$

where $\{g_1, g_2, g_3\}$ is the right-hand orthonormal basic vectors for the shell element.

6.2.2 Stress-strain relation

For the isotropic elastic material, the stress-strain relation can be written as

$$\sigma = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{yz} \end{Bmatrix} = [\mathrm{D}](\varepsilon - \varepsilon_T) \tag{6.22}$$

where [D] is the elasticity matrix; ε_T is the thermal strain due to the temperature rise.

6.2.3 Element stiffness matrix

The total potential energy including the drilling rotation term can be written as

$$\Pi = \frac{1}{2}\int_V \varepsilon^T \sigma dV + \frac{1}{2}\int_V k_\theta \left(\omega - \theta_z\right)^2 dV - \int_V u^T f dV \tag{6.23}$$

The first variation of Equation 6.23 yields the governing equilibrium equation as

$$\delta\Pi = \int_V \delta\varepsilon^T \sigma dV + \int_V \left(\delta\omega - \delta\theta_z\right) k_\theta \left(\omega - \theta_z\right) dV - \int_V \delta u^T f dV = 0 \tag{6.24}$$

By substituting the strain-displacement relationship of Equation 6.12, Equation 6.24 becomes

$$\int_V [B]\sigma dV + \left(\int_V \left[B^{dr}\right]^T k_\theta \left[B^{dr}\right] dV\right) u = \{F\} \tag{6.25}$$

Where $\{F\} = \int_V f dV$ is the equivalent elemental load.

Solutions of Equation 6.25 have to be approached iteratively for the geometrically nonlinear analysis and taking variation with respect to du we obtain

$$\int_V d\left([B]^T \sigma\right) dV + \left(\int_V \left[B^{dr}\right]^T k_\theta \left[B^{dr}\right] dV\right) du = d\{F\} \tag{6.26}$$

The first term of Equation 6.26 can be expressed as

$$\int_V d\left([B]^T \sigma\right) dV = \int_V d[B]^T \sigma dV + \int_V [B]^T d\sigma dV \tag{6.27}$$

and together with the variation of stress in the form of

$$d\{\sigma\} = [D]d\{\varepsilon\} = [D][B]d\{u\} \tag{6.28}$$

Hence Equation 6.26 becomes

$$\int_V \mathrm{d}[\mathrm{B}]^T \sigma \mathrm{d}V + \left(\int_V [\mathrm{B}]^T [\mathrm{D}][\mathrm{B}]\mathrm{d}V\right)\mathrm{d}u + \left(\int_V \left[\mathrm{B}^{dr}\right]^T k_\theta \left[\mathrm{B}^{dr}\right]\mathrm{d}V\right)\mathrm{du} = \mathrm{d}\{\mathrm{F}\} \quad (6.29a)$$

Or

$$[\mathrm{K_T}]\mathrm{du} = \mathrm{d}\{\mathrm{F}\} \quad (6.29b)$$

where $[K_T] = [K]^M + [K_\sigma] + [K]^D$ is the tangent stiffness matrix; $[K]^M = \int_V [B]^T [D][B]\mathrm{d}V$ is the so-called material matrix which is the function of initial geometry and displacement; $[K]^D = \int_V \left[B^{dr}\right]^T k_\theta \left[B^{dr}\right]\mathrm{d}V$ is the additional "drilling stiffness" matrix. $[K_\sigma]$ is known as geometric matrix defined as

$$\int_V \mathrm{d}[\mathrm{B}]^T \sigma \mathrm{d}V = [\mathrm{K}_\sigma]\mathrm{du} \quad (6.30)$$

The geometrical nonlinearity of this shell element is based on the Total Lagrangian formulation. The shear locking problem is overcome by MITC technique. The drilling degree of freedom is considered when forming the elemental stiffness matrix. The applications of this new shell element in OpenSees will be presented in the following chapter.

Chapter 7

Thermo-mechanical Analysis of Shell Structures using OpenSees

7.1 Introduction

The development of a two-dimensional thermomechanical model in OpenSees had been presented in Chapter 5. Based on the theory of a geometrically nonlinear shell element presented in Chapter 6, its application as well as modification of existing framework in OpenSees to enable a three-dimensional thermomechanical analysis capability will be presented in this chapter. This involves creating new geometrically nonlinear shell element ShellMITC4GNLThermal by modifying the existing linear shell element ShellMITC4 in OpenSees. A new thermal field class ShellThermalAction was created to define the temperature distribution in the shell element. The existing elastic and elasto-plastic three-dimensional material classes were modified to include temperature-dependent properties according to Eurocodes. Two benchmark examples, one-way bending of a plate subjected to a uniform temperature rise and two-way bending of a rectangular plate subjected to a linear thermal gradient, were used to test the performance of the development work in OpenSees.

7.2 Extended thermomechanical analysis of plate structures in OpenSees

New thermal load class ShellThermalAction was created to define the temperature distribution in the shell element. New geometrically nonlinear shell element ShellMITC4GNThermal was created by modifying the existing linear shell element ShellMITC4 in OpenSees. New multiaxial material class DruckerPragerThermal was created to model the concrete in the slab.

7.2.1 Thermal load class

A new thermal load class ShellThermalAction was created to define the temperature distribution in the shell element. Figure 7.1 shows the class diagram of the shell thermal load class. Figure 7.2 shows the scheme of temperature distribution in the shell element. The temperature distribution through the shell section is defined by temperature and corresponding location. At each Gauss point in the plane of the shell element, there are seven predefined integration points of which the temperature can be interpolated from the adjacent temperature points. In addition different sets of temperature can be defined at the Gauss integration points to consider the variation of temperature in the horizontal plane of the shell element. In this stage uniform temperature distribution is assumed in the shell plane and the

constructor and function defined for class ShellThermalAction was similar with class Beam2dThermalAction and the details are shown in Appendix

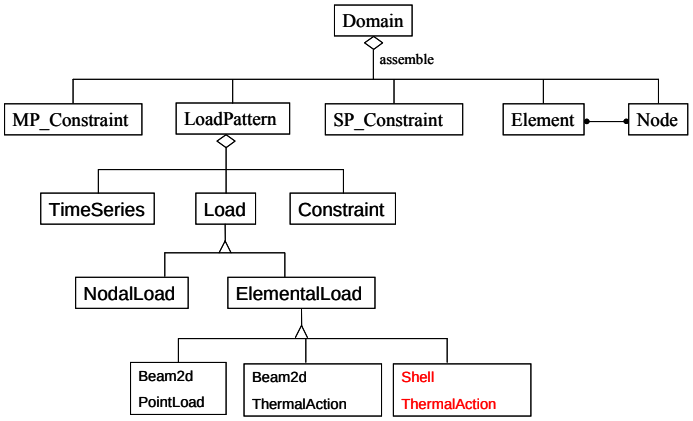


Figure 7.1. Hierarchy for thermal load class in OpenSees

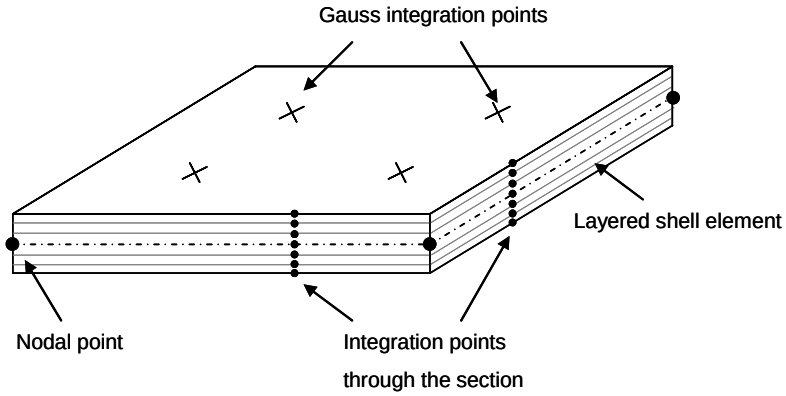


Figure 7.2 Temperature distribution of the shell element in OpenSees

7.2.2 3D material class

3D temperature dependent material classes ElasticIsotropic3DThermal and DruckerPragerThermal were created to model the elastic material and concrete material for the shell element.

7.2.2.1 Elastic material

A new temperature dependent material class ElasticIsotropic3DThermal was created by modifying the existing 3D elastic isotropic material class ElasticIsotropic3D. The modifications are similar to the Steel01Thermal class mentioned in Section 5.4.2. The modulus of elasticity at elevated temperature is referred to Eurocode.

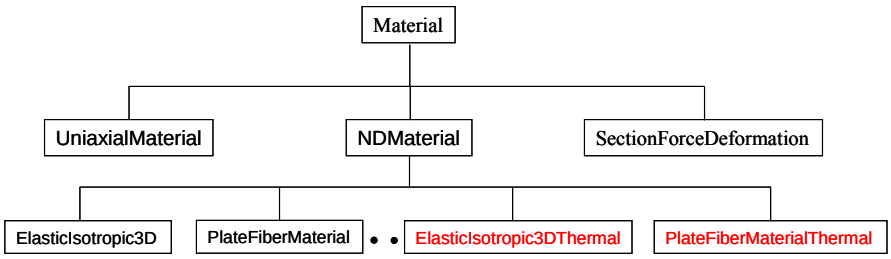


Figure 7.3 Hierarchy for 3D temperature dependent material class in OpenSees

7.2.2.2 Drucker Prager concrete material

A new 3D concrete material class DruckerPragerThermal was created based on the exising DruckerPrager material class in OpenSees. The input parameters for the existing DruckerPrager material class were presented in Section 3.5.2. Figure 7.4 shows the functions implemented in the DruckerPragerThermal class and the temperature dependent properties of concrete are according to Eurocode 2 as shown in Figure 7.5. The complete expression of Figure 7.5 is shown in Appendix F.

```
class DruckerPragerThermal : public NDMaterial
{
  public:
    // Full Constructor
    DruckerPragerThermal(int tag, int classTag, double bulk, double
    shear, double s_y, double r, double r_bar, double Kinfinity,
    double Kinit, double d1, double d2, double H, double t, double
    massDen, double atm = 101.0);


  DruckerPragerThermal();

  ~DruckerPragerThermal();

  NDMaterial *getCopy(const char *type);

  int commitState(void);
  int revertToLastCommit(void);
  int revertToStart(void);

  NDMaterial *getCopy(void);
  const char *getType(void) const;
  int getOrder(void) const;

  double setThermalTangentAndElongation(double &TempT, double &,
                                        double &);

  int updateParameter(int responseID, Information &eleInformation);
```

```
  double getRho(void) {return massDen;};

protected:

  //material parameters
  double mKref;           // reference Bulk Modulus
  double mGref;           // reference Shear Modulus
  double mPatm;    // reference stress first invariant (pressure)
  double mK;          // bulk modulus
  double mG;          // shear modulus
  double msigma_y;    // yield strength
  double mrho;        // volumetric term
  double mrho_bar;    // nonassociative flow term
  double mKinf;           // nonlinear isotropic hardening term
  double mKo;         // nonlinear isotropic hardening term
  double mdelta1; //exponential hardening term for drucker prager
                   surface
  double mdelta2;//exponential hardening term for tension cutoff
                   surface
  double mHard;           // hardening constant
  double mtheta;            // hardening constant
  double mTo;                 // initial tension cutoff strength

  double mKref0;          // record initial Bulk Modulus
  double mGref0;          // record initial Modulus
  double msigma_y0;       // record initial yield strength
  double ThermalElongation;

  //functions
  void initialize();  // initializes variables
  int updateElasticParam(void); //updated Elastic Parameters based
                                     on mean stress

  //plasticity integration routine
  void plastic_integrator(void);

};
```

Figure 7.4: Functions for DruckerPragerThermal

```
double
DruckerPragerThermal::setThermalTangentAndElongation(double &TempT,
double&ET, double&Elong)
{
  // TempT is actural temperature; ambient temperature is 20;
  // ET is temperature dependent elasticity modulus.
  // Elong is the thermal elongation at elevated temperature
  // The temperature dependent properties are based on concrete with
      Calcareous aggregates EN 1992 part 1-2-1, Page20

    double feT;//elastic compressive strength
    double ee0; // elastic strain corresponding to feT
    double fcT;// compressive strength at elevated temperature
      double ecT;//strain at fcT;
```

```
      fc0 = 5.5*msigma_y0;// compressive strength at ambient
                                        temperature
    ec0 = 0.0025; //strain at fcT at ambient temperature

   if (TempT <= 20) {

       ET = 2*fc0/ec0;

  }
    else if (TempT <= 100) {

    mKref = mKref0/(0.0025 + (0.004-0.0025)*(TempT - 20)/80)*ec0;
    mGref = mGref0/(0.0025 + (0.004-0.0025)*(TempT - 20)/80)*ec0;;
    mK = mKref;
    mG = mGref;
    msigma_y = msigma_y0;
    fcT = fc0;
    ecT = (0.0025 + (0.004-0.0025)*(TempT - 20)/80);
    ET = 2*fcT/ecT;
    ee0 = 1/3*fcT/ET;
      mHard = 2/3*fcT/(ecT-ee0)/2.8;

 }
. . .

  else if (TempT <= 1000) {

      mKref = mKref0*(0.08 - (TempT - 900)*0.04/100)/10;
      mGref = mGref0*(0.08 - (TempT - 900)*0.04/100)/10;
      mK = mKref;
      mG = mGref;
      msigma_y = msigma_y0*(0.08 - (TempT - 900)*0.04/100);
      fcT = fc0*(0.08 - (TempT - 900)*0.04/100);
      ecT = 0.025;
      ET = 2*fcT/ecT;
      ee0 = 1/3*fcT/ET;
       mHard = 2/3*fcT/(ecT-ee0)/2.8;
  }

// Calculate thermal elongation
  if (TempT <= 0) {
      ThermalElongation = 0.0;
  }
  else if (TempT <= 700) {
      ThermalElongation = -1.8e-4 + 9e-6*TempT + 2.3e-
                              11*TempT*TempT*TempT;

  }
  else if (TempT <= 1200) {
      ThermalElongation = 14e-3;
  }
  else {
      opserr << "the temperature is invalid\n";
  }

Elong = ThermalElongation;
this->initialize();
this->updateElasticParam( );
```

```
this->plastic_integrator( );
  return 0;
}
```

Figure 7.5: Implementation of the function "setThermalTangentAndElongation()"

7.2.3 Section class

The existing section class <MembranePlateFiberSection> was modified to <MembranePlateFiberSectionThermal> to include temperature data. Figure 7.6 shows the main functions defined in this class including a similar function "getTemperatureStress()" with FiberSection2dThermal class.

```
class MembranePlateFiberSectionThermal : public
SectionForceDeformation{

 public :

    //full constructor
    MembranePlateFiberSectionThermal(int tag,double thickness,
                                NDMaterial &Afiber) ;

    virtual ~MembranePlateFiberSectionThermal( ) ;

    //get the strain and integrate plasticity equations
    int setTrialSectionDeformation( const Vector
                  &strain_from_element ) ;

    const Vector &getTemperatureStress(double dataMixed[18]);

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

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

    //send back the tangent
    const Matrix& getSectionTangent( ) ;

private :

    double h ; //plate thickness

    NDMaterial *theFibers[5] ;  //pointers to five materials (fibers)

    double sTData[2]; //Data for section resisting force due to
                           thermal load
    Vector  *sT;  //  Pointer to sTData
```

```
    double  ThermalElongation[5]; // store thermal elongation of
                                  five fibres
} ;
```

Figure 7.6: Functions for MembranePlateFiberSectionThermal class

7.2.4 Shell element class

A geometrically nonlinear shell element class <ShellMITC4GNThermal> was created based on the existing linear shell element <ShellMITC4>. The modifications follow the procedure of the Total Lagrangian formulation mentioned in Chapter 6. Figure 7.7 shows the main functions defined in the ShellMITC4GNThermal class. The implementation of the function "addLoad()" is similar with that of DispBeamColumn2dThermal class shown in Figure 5.17

```
class ShellMITC4GNThermal : public Element {

 public:

  ShellMITC4GNThermal(int tag, int node1,int node2,int node3,int
                   node4, SectionForceDeformation &theMaterial);

  virtual ~ShellMITC4GNThermal( ) ;

  void setDomain(Domain *theDomain) ;

  int update(void);

    //return stiffness matrix
    const Matrix &getTangentStiff( );
    const Matrix &getInitialStiff();
    const Matrix &getMass();

    // methods for applying loads
    void zeroLoad(void);
    int addLoad(ElementalLoad *theLoad, double loadFactor);

    //get residual
    const Vector &getResistingForce( );

 private :

    Node *nodePointers[4] ;      //pointers to four nodes

    double Ktt ; //drilling stiffness
```

```
    SectionForceDeformation *materialPointers[4]; //pointers to four
                                                    materials
    //shell basis vectors
    double g1[3] ;
    double g2[3] ;
    double g3[3] ;

    void computeBasis( ); //compute local coordinates and basis

    void formResidAndTangent(int tang_flag);//form residual and
                                                    tangent
    //compute Jacobian matrix and inverse at point {L1,L2}
    void computeJacobian(double L1, double L2, const double x[2][4],
                         Matrix &JJ, Matrix &JJinv);
    //compute Bdrill matrix
    double* computeBdrill(int node,const double shp[3][4]);

    //assemble a B matrix
    const Matrix& assembleB(const Matrix &Bmembrane, const Matrix
&Bbend, const Matrix &Bshear);

    //compute Bmembrane matrix
    const Matrix& computeBmembrane(int node,const double shp[3][4]);

    //compute Bbend matrix
    const Matrix& computeBbend(int node, const double shp[3][4]);

    //compute standard Bshear matrix
    const Matrix& computeBshear(int node, const double shp[3][4]);

    //compute Bbar shear matrix
    const Matrix& computeBbarShear(int node, double L1, double L2,
                    const Matrix& Jinv);
    //compute the gamma's
    void computeGamma(const double xl[2][4],const Matrix &J);

    //shape function routine for four node quads
      void shape2d(double ss, double tt, const double x[2][4],
                         double shp[3][4], double &xsj);

   double *dataMix; routine //store temperature distribution

      int counterTemperature;//Recorded parameter used to determine
                         the application of thermally induced
                         force at first iteration and
                         cancellation for the following
                         iterations

      double residThermal[24];//store thermally induced unbanlanced
                                                    load
} ;
```

Figure 7.7: Functions for DispBeamColumn2dThermal

7.3 Benchmark testing of developed codes in OpenSees

A series of benchmark problems were analyzed to test the extended codes in OpenSees and comparisons have been made with analytical solutions. These include the bending of a cylindrical plate subjected to uniform temperature rise, a rectangular plate subjected to UDL and thermal gradient as well as a concrete plate subjected to different loads to check the performance of the developed DruckerPragerThermal material.

7.3.1 One-way bending of plate

In this case the plate is assumed to have infinite extent in the y direction with loading and support conditions independent of y. The two infinite edges along y direction are restrained in lateral translation (in the x direction) but free to rotate about y axis. They are also free to translate along the y direction. The plate is subjected to a uniformly distributed load (UDL) and uniform temperature rises. An equivalent strip can be cut out from the plate, as shown in Figure 7.8. The axial force P is produced by the combined effects of restrained thermal expansion and large deflections. Denoting by q the intensity of the UDL, the equilibrium equation governing the bending of the plate can be written as

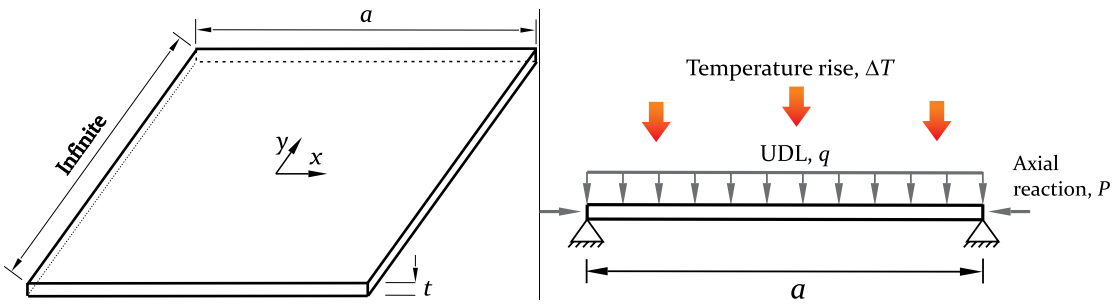


Figure 7.8: Schematic view of cylindrical bending of a rectangular plate

$$\frac{d^2w}{dx^2}+\frac{Pw}{D}=-\frac{qax}{2D}+\frac{qx^2}{2D} \tag{7.1}$$

Where $D = Et^3 \big/ 12\left(1-\nu^2\right)$ is the bending stiffness of the plate; E is the elasticity modulus; t is the thickness of the plate; ν is Poisson ratio.

Equation 7.1 is similar with Equation 5.16 for the beam case except the different expressions of bending stiffness and similar solutions can be deduced. Introducing the notation

$$k = \sqrt{\frac{P}{D}} \tag{7.2}$$

the general solution of Equation 7.1 can be written in the following form:

$$w = C_1 \sin kx + C_2 \cos kx + \frac{q}{2Dk^2}x^2 - \frac{qa}{2Dk^2}x - \frac{q}{Dk^4} \tag{7.3}$$

The constants of C_1 and C_2 can be determined from the boundary conditions and the solution of deflection w becomes

$$w = \frac{q}{Dk^4}\left[\frac{1-\cos ka}{\sin ka}\sin kx + \cos kx - 1\right] + \frac{q}{2Dk^2}x^2 - \frac{qa}{2Dk^2}x \tag{7.4}$$

The mid-span deflection can be derived from Equation 7.4 as

$$w_{mid} = \frac{q}{Dk^4}\left[\frac{1-\cos ka}{\sin ka}\sin\left(\frac{ka}{2}\right) + \cos\left(\frac{ka}{2}\right) - 1\right] - \frac{qa^2}{8Dk^2} \tag{7.5}$$

To solve Equation 7.5, the axial force P should be determined first. However P is also a function of the deflection w as it results from the combined effects of restrained thermal expansion and extension developed in the strip due to the large deflection.

The extension of the strip produced by the deflection w is equal to the difference between the length of the arc a_{arc} along the deflection curve and the chord length a. If the sin curve of deformation shape is assumed, the mid-span deflection can be calculated as (Usmani el at. 2001)

$$w_{mid} = \frac{2a}{\pi}\sqrt{\varepsilon_w + \frac{\varepsilon_w{}^2}{2}} \tag{7.6}$$

where $\varepsilon_w = \frac{a_{arc} - a}{a}$ is the tensile strain and can be expressed in turn as

$$\varepsilon_w = \sqrt{\frac{1}{2}\left(\frac{\pi w_{mid}}{l}\right)^2 + 1} - 1 \tag{7.7}$$

Introducing the thermal strain ε_T

$$\varepsilon_T = \alpha \Delta T \tag{7.8}$$

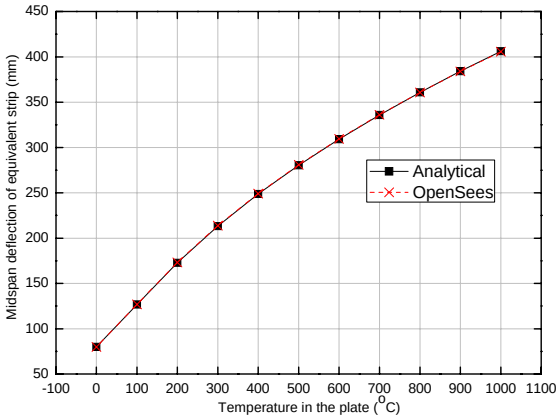
The reaction in the support can be written as

$$P = EA\left[\varepsilon_T - \varepsilon_w\right] \tag{7.9a}$$

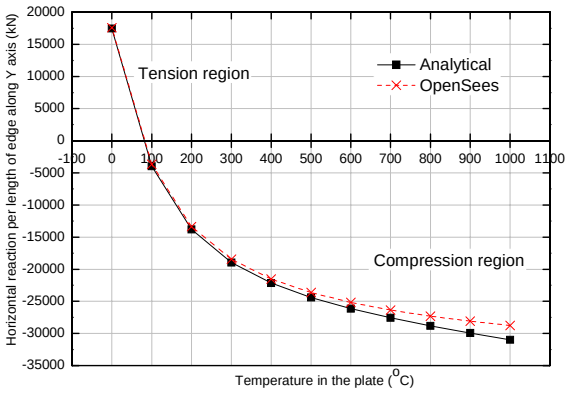
or

$$P = EA\left[\alpha \Delta T - \sqrt{\frac{1}{2}\left(\frac{\pi w_{mid}}{l}\right)^2 + 1} + 1\right] \tag{7.9b}$$

Figure 7.9 shows the comparison of analytical and OpenSees results. The command scripts in OpenSees is presented in Appendix G.1. As can be seen the OpenSees results were in good agreements with the analytical results. The plate geometric parametres and material properties are listed in Table 7.1. The infinite length edge was considered to be 30m long in OpenSees and the constant *E* was used. A mesh of 16×16 was used for the OpenSees model. The analytical results were derived by iterative solving of Equation 7.5 and 7.9. The horizontal reactions from OpenSees is not uniformly distributed along the edge as shown in Figure 7.10 and the average values are taken to be compared with the analytical solutions. It was also found, from Figure 7.9 (b) and 7.10, that the increments of the reaction slowed down as the temperature increased which may be driven by the increasing centenary tension due to the extension of the strip under large deflection.



(a) Mid-span deflection



(b) Axial reaction

Figure 7.9: Results of bending of uniformly heated cylindrical plate

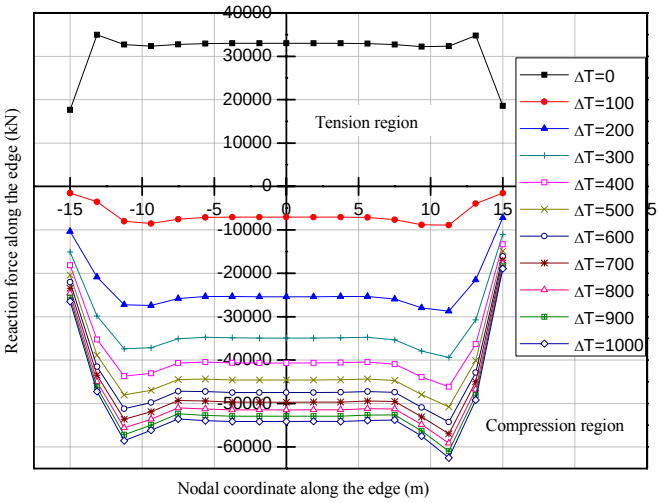


Figure 7.10: Horizontal reaction distribution along the edge in the y direction

Side length a (m)	E (N/m^2)	α (/^{o}C)	ν	Thickness of plate t (m)	UDL (kN/m^2)
6	2×10^{11}	12×10^{-6}	0.3	0.2	1000

Table 7.1: Input parameters of the plate model

If the thermal gradient is considered, more generally, Equation 7.1 can be modified as

$$\frac{d^2w}{dx^2}+\frac{Pw}{D}=-\frac{qax}{2D}+\frac{qx^2}{2D}-\phi_T \tag{7.10}$$

where $\phi_T = \partial T_{,z}$ is the curvature due to thermal gradient $T_{,z}$.

The mid-span deflection can be calculated followed the similar process from Equation 7.10 as

$$w_{mid}=\frac{q+D\phi_T}{Dk^4}\left[\frac{1-\cos kl}{\sin kl}\sin\left(\frac{kl}{2}\right)+\cos\left(\frac{kl}{2}\right)-1\right]-\frac{ql^2}{8Dk^2} \tag{7.11}$$

7.3.2 Rectangular plate

Now we consider a simply supported rectangular plate subjected to a UDL and a uniform thermal gradient through the thickness. For linear analysis the equation of maximum deflection for plates under UDL is given as (Ugural 1999)

$$w=\frac{16q}{D\pi^6}\sum_m^\infty\sum_n^\infty\frac{\sin(\alpha_m x)\sin(\gamma_n y)}{mn\left[(m/a)^2+(n/b)^2\right]^2} \tag{7.12}$$

The analytical solution for the deflection of the plate subjected to a uniform thermal gradient is given by Ugural (1999)

$$w = \frac{16M^T}{D\pi^4}\sum_m^\infty\sum_n^\infty \frac{\sin(\alpha_m x)\sin(\gamma_n y)}{mn\left[(m/a)^2+(n/b)^2\right]} \tag{7.13}$$

where $\alpha_m = m\pi/a$ and $\gamma_n = n\pi/b$, $M^T = \alpha T_{,z} D(1+\nu)$ is the equivalent thermal load, $\Delta T = 0.5 + T_{,z}$ z where $T_{,z}$ is the thermal gradient per unit depth. The expression of deflection converges very rapidly. The total deflection at the centre of the plate subjected to UDL and thermal gradient is the sum of Equations 7.12 and 7.13

$$w = \frac{16q}{D\pi^6}\sum_m^\infty\sum_n^\infty \frac{\sin(m\pi/2)\sin(n\pi/2)}{mn\left[(m/a)^2+(n/b)^2\right]^2} + \frac{16M^T}{D\pi^4}\sum_m^\infty\sum_n^\infty \frac{\sin(m\pi/2)\sin(n\pi/2)}{mn\left[(m/a)^2+(n/b)^2\right]} \tag{7.14}$$

For nonlinear analysis the Airy stress function $F(x,y)$ is introduced in order to satisfy the nonlinear governing equations

$$\sigma_x = F_{,yy}, \qquad \sigma_y = F_{,xx}, \qquad \tau_{xy} = -F_{,xy} \tag{7.15}$$

The strain components are obtained in terms of stress function

$$\begin{aligned} \varepsilon_x &= \frac{1}{E}(F_{,yy} - \nu F_{,xx}) + \alpha\Delta T \\ \varepsilon_y &= \frac{1}{E}(F_{,xx} - \nu F_{,yy}) + \alpha\Delta T \\ \gamma_{xy} &= -\frac{2(1+\nu)}{E}F_{,xy} \end{aligned} \tag{7.16}$$

The compatibility equation can be expressed by

$$\varepsilon_{x,yy} + \varepsilon_{y,xx} - \gamma_{xy,xy} = w_{,xy}^2 - w_{,xx}w_{,yy} \tag{7.17}$$

Substitution from Equation 7.16 into the compatibility Equation 7.17 leads to

$$F_{,xxxx} + 2F_{,xxyy} + F_{,yyyy} + E\alpha((\Delta T)_{,xx} + (\Delta T)_{,yy}) - E(w_{,xy}^2 - w_{,xx}w_{,yy}) = 0 \tag{7.18}$$

By integration over the thickness of the plate, the following equation is derived

$$t(F_{,xxxx} + 2F_{,xxyy} + F_{,yyyy}) + (N^T_{,xx} + N^T_{,yy}) - Et(w^2_{,xy} - w_{,xx}w_{,yy}) = 0 \tag{7.19}$$

where $N^T = E\alpha t/2$. The governing equation of plates under UDL is given as

$$M_{xx,xx} + 2M_{xy,xy} + M_{yy,yy} + N_{xx}w_{,xx} + 2N_{xy}w_{,xy} + N_{yy}w_{,yy} + q(x,y) = 0 \tag{7.20}$$

where the force and stress resultants are defined by

$$\begin{aligned}
(N_{xx}, N_{yy}, N_{xy}) &= \int_{-t/2}^{t/2} (\sigma_{xx}, \sigma_{yy}, \tau_{xy}) dz = t(F_{,yy}, F_{,xx}, -F_{,xy}) \\
M_{xx} &= \int_{-t/2}^{t/2} \sigma_{xx} z dz = -D(w_{,xx} + \nu w_{,yy}) - M^T \\
M_{yy} &= \int_{-t/2}^{t/2} \sigma_{yy} z dz = -D(w_{,yy} + \nu w_{,xx}) - M^T \\
M_{xy} &= \int_{-t/2}^{t/2} \tau_{xy} z dz = -(1-\nu) D w_{,xy}
\end{aligned} \tag{7.21}$$

Substitution of the above force and moment resultants into Equation 7.20 gives

$$D(w_{,xxxx} + 2w_{,xxyy} + w_{,yyyy}) - t(F_{,yy}w_{,xx} - 2F_{,xy}w_{,xy} + F_{,xx}w_{,yy}) + M^T_{,xx} + M^T_{,yy} - q(x,y) = 0 \tag{7.22}$$

The following functions satisfy two partial differential Equations 7.19 and 7.22 for a plate with simply supported boundary conditions which has zero external edge load in its plane

$$\begin{aligned}
w(x,y) &= \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} w_{mn} \sin(\alpha_m x)\ \sin(\gamma_n y) \\
F(x,y) &= \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} F_{mn} \sin(\alpha_m x)\ \sin(\gamma_n y) \\
q(x,y) &= \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} q_{mn} \sin(\alpha_m x)\ \sin(\gamma_n y)
\end{aligned} \tag{7.23}$$

and double Fourier series are assumed for thermal loading resultants

$$\begin{aligned}
N^T(x,y) &= \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} N^T_{mn} \sin(\alpha_m x)\ \sin(\gamma_n y) \\
M^T(x,y) &= \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} M^T_{mn} \sin(\alpha_m x)\ \sin(\gamma_n y)
\end{aligned} \tag{7.24}$$

where

$$(N_{mn}^T, M_{mn}^T, q_{mn}) = \frac{4(-1+(-1)^m)(-1+(-1)^n)}{mn\pi^2}(N^T, M^T, q) \tag{7.25}$$

Substituting Equations 7.23-7.25 into governing Equations 7.19 and 7.22 leads to

$$\begin{aligned}&(\alpha_m^2+\gamma_n^2)^2 tF_{mn}\sin(\alpha_m x)\sin(\gamma_n y) - (\alpha_m^2+\gamma_n^2)N_{mn}^T\sin(\alpha_m x)\sin(\gamma_n y)\\ &\quad - Et(\alpha_m\alpha_r\gamma_n\gamma_s w_{mn}w_{rs}\cos(\alpha_m x)\cos(\gamma_n y)\cos(\alpha_r x)\cos(\gamma_s y)\\ &\quad - \alpha_m^2\gamma_s^2 w_{mn}w_{rs}\sin(\alpha_m x)\sin(\gamma_n y)\sin(\alpha_r x)\sin(\gamma_s y)) = 0\end{aligned} \tag{7.26a}$$

$$\begin{aligned}&D(\alpha_m^2+\gamma_n^2)^2 w_{mn}\sin(\alpha_m x)\sin(\gamma_n y)\\ &-t(\alpha_m^2\gamma_s^2\sin(\alpha_m x)\sin(\gamma_n y)\sin(\alpha_r x)\sin(\gamma_s y) - 2\alpha_m\alpha_r\gamma_n\gamma_s\cos(\alpha_m x)\cos(\gamma_n y)\cos(\alpha_r x)\cos(\gamma_s y))\\ &\quad + (\alpha_r^2\gamma_n^2\sin(\alpha_m x)\sin(\gamma_n y)\sin(\alpha_r x)\sin(\gamma_s y))w_{mn}F_{rs}\\ &-(\alpha_m^2+\gamma_n^2)M_{mn}^T\sin(\alpha_m x)\sin(\gamma_n y) - q_{mn}\sin(\alpha_m x)\sin(\gamma_n y) = 0\end{aligned} \tag{7.26b}$$

Using the expansion theorem we obtain

$$(\alpha_m^2+\gamma_n^2)^2 tF_{mn} - (\alpha_m^2+\gamma_n^2)N_{mn}^T - 4Et(\alpha_m\alpha_r\gamma_n\gamma_s w_{mn}w_{rs}\xi_{mnrs} - \alpha_m^2\gamma_s^2 w_{mn}w_{rs}\eta_{mnrs}) = 0 \tag{7.27a}$$

$$D(\alpha_m^2+\gamma_n^2)^2 w_{mn} - 4tF_{rs}(\alpha_m^2\gamma_s^2\eta_{mnrs} - 2\alpha_m\alpha_r\gamma_n\gamma_s F_{rs}\xi_{mnrs} + \alpha_r^2\gamma_n^2\eta_{mnrs})w_{mn} - (\alpha_m^2+\gamma_n^2)M_{mn}^T - q_{mn} = 0 \tag{7.27b}$$

where

$$\begin{aligned}\xi_{mnrs} &= \frac{(-m+2m(-1)^{2m+r} - m(-1)^r)(-n+2n(-1)^{2n+s} - n(-1)^s)}{\pi^2(4m^2-r^2)(4n^2-s^2)}\\ \eta_{mnrs} &= \frac{(-2m^2+2m^2(-1)^r - (-1)^r r^2 + r^2(-1)^{2m+r})(-2n^2+2n^2(-1)^s - (-1)^s s^2 + s^2(-1)^{2n+s})}{rs\pi^2(4m^2-r^2)(4n^2-s^2)}\end{aligned} \tag{7.28}$$

In this case, a 6m×6m square plate with thickness 0.1 m with the same properties as listed in Table 1. The temperature at top of the plate was assumed to be 0^oC and it varied linearly over the depth of the plate to temperatures of 100^oC to 1000^oC at bottom. The comparison of analytical and OpenSees results of the deflection at the centre of plate are shown in Figure 7.11. A mesh of 16×16 was used for this OpenSees model. The command scripts in OpenSees are presented in Appendix G.2. The analytical solutions were derived from the first nine terms of the series in Equation 7.23. Notice that due to the complexity of the

nonlinear analysis, it was not possible to obtain the deflection of the plate for very high temperature gradients, but there is a good agreement between the analytical and OpenSees results. Large displacement of the plate generates membrane tensions during the application of the UDL. This reduces the deflection when seen in comparison with the linear solution.

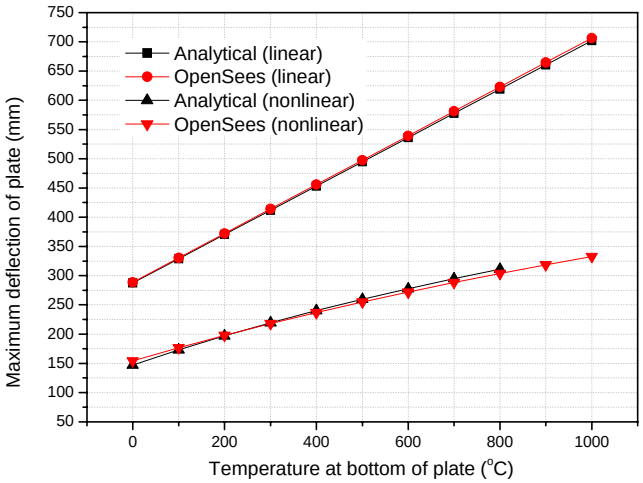


Figure 7.11: Small and large deflections of rectangular plate

7.3.3 Plate example to test DruckerPragerThermal

For a real case the two-dimensional material DruckerPrager in OpenSees should be used to model the concrete material of the slab. Therefore, before the modelling of Cardington tests using OpenSees presented in the next chapter, the performance of the developed two-dimensional concrete material DruckerPragerThermal at ambient and elevated temperature is verified in this section by comparing results from OpenSees and ABAQUS. A 6×6m square concrete plate was modelled in OpenSees with properties shown in Table 7.2 to test the performance. Four cases were studied including the concrete plated subjected to edge compression, uniformly distributed load (UDL), UDL and uniform temperature rise and thermal gradient alone. The command scripts in OpenSees is presented in Appendix G.3.

Side length L (m)	Compressive strength fc (MPa)	Tensile strength ft (MPa)	α (/°C)	ν	Thickness of plate (m)
6	48	4.8	12×10^{-6}	0.2	0.1

Table 7.2 Input parameters of the plate model

Substituting the properties in Table 7.2 into Equation 3.10, 3.11, 3.12, the input parameters for DruckerPragerThermal material in OpenSees can be calculated as shown in Table 7.3. The strain corresponding to the compressive strength is assumed to be 0.0025 in this case according to Eurocode 2. The corresponding input parameters for Drucker-Prager material in ABAQUS are shown in Table 7.4.

k (Pa)	G (Pa)	σ_Y (Pa)	ρ	H	theta	Kinf=Ko	delta1=delta2
2.13×10^{10}	1.6×10^{10}	7.44×10^{6}	0.437	5.5×10^{9}	1	0	1

Table 7.3 Input parameters for Drucker Prager material in OpenSees

Angle of friction	FlowStress ratio	Dilation angle	Young's modulus	Poisson's ratio
55	0.778	30	3.84×10^{10}	0.2

Yield stress	**Plastic strain**
1.6×107	**0**
4.8×107	**0.00125**

Table 7.4 Input parameters for Drucker Prager material in ABAQUS

7.3.3.1 Edge compression

One side of the plate was subjected to compressive UDL= 48MPa (chosen to be equal to the compressive strength of concrete) as shown in Figure 7.12. This case was carried out to check the one-dimensional performance of the Drucker-Prager material. The force-displacement curve is shown in Figure 7.13. Three models are compared. One is OpenSees model of three-dimensional plate. The second is ABAQUS model of three-dimensional plate. The third model is the stress-strain relation of concrete material Concrete02. In this case only one-dimensional compressive load is applied on the edge of the plate and so the force-displacement relation will exactly follows the one-dimensional stress-strain relation of Drucker-Prager material. Compared with Concrete02 shown in Section 3.5.1, the Drucker-Prager material in OpenSees and ABAQUS is to use bilinear constitutive relationship to model the realistic curved compressive stress-strain relationship of concrete.

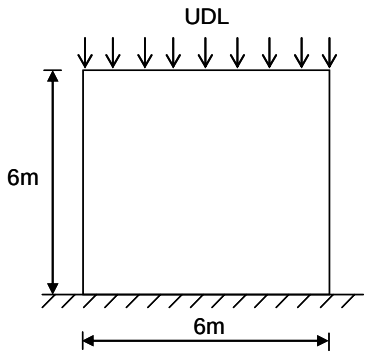


Figure 7.12: Plate subjected to edge compression

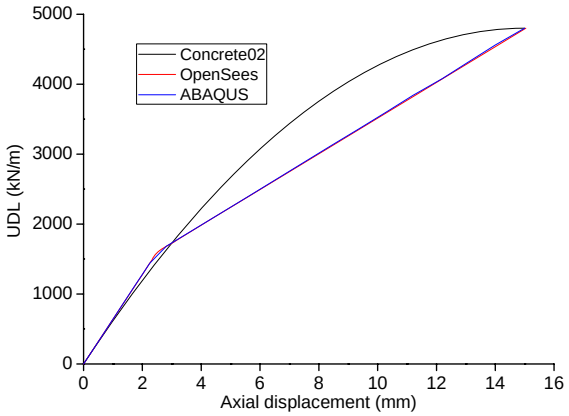


Figure 7.13: Vertical displacement of plate subjected to compressive UDL

7.3.3.2 Plane UDL

The plate was subjected to a UDL=12kN/m^2 and the four edges of the plate were translationally restrained but rotational free. Figure 7.14 shows the deflection at the centre of the plate.

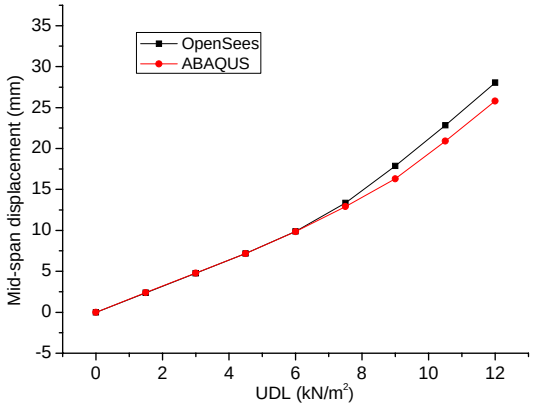


Figure 7.14: Deflection at centre of plate subjected to UDL

7.3.3.3 UDL and uniform heating

The plate was subjected to a UDL=5.5kN/m^2 and uniform temperature rise. Geometrical linear shell element was used to model the plate and the four edges of the plate were translational restrained but rotational free. Figure 7.15 shows the deflection at the centre of the plate.

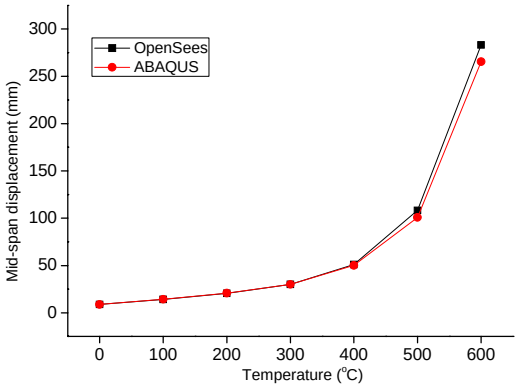


Figure 7.15: Deflection at the centre of plate subjected to UDL and uniform temperature rise

7.3.3.4 Thermal gradient alone

The plate was subjected to thermal gradient alone and the four edges of the plate were translational restrained but rotational free. The temperature at top of the plate was assumed to be 0°C and it varied linearly over the depth of the plate to temperatures of 100°C to 600°C at bottom. Figure 7.16 shows the deflection at the centre of the plate.

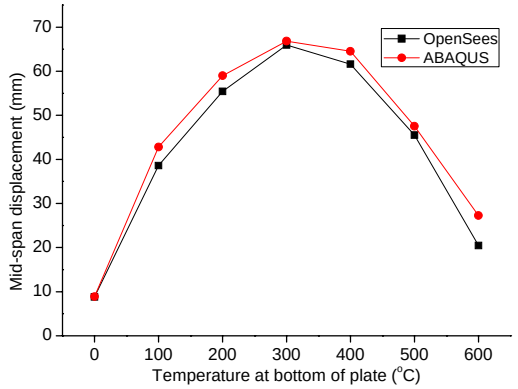


Figure 7.16: Deflection at the centre of plate subjected to thermal gradient alone

In this section the performance of the DruckerPragerThermal material were verified and the results of these four geometrically linear cases from OpenSees agreed well with

ABAQUS. The performance of the geometrically nonlinear shell element ShellMITC4GNThermal was also verified in Section 7.3. In the next chapter two Cardington restrained tests and a reinforced concrete frame test was modelled in OpenSees using this Drucker-Prager material and nonlinear shell element. In addition, two steel frame tests were used to validate the performance of two-dimensional thermomechanical ananlysis capability in OpenSees.

Chapter 8

Validation using Experimental Data

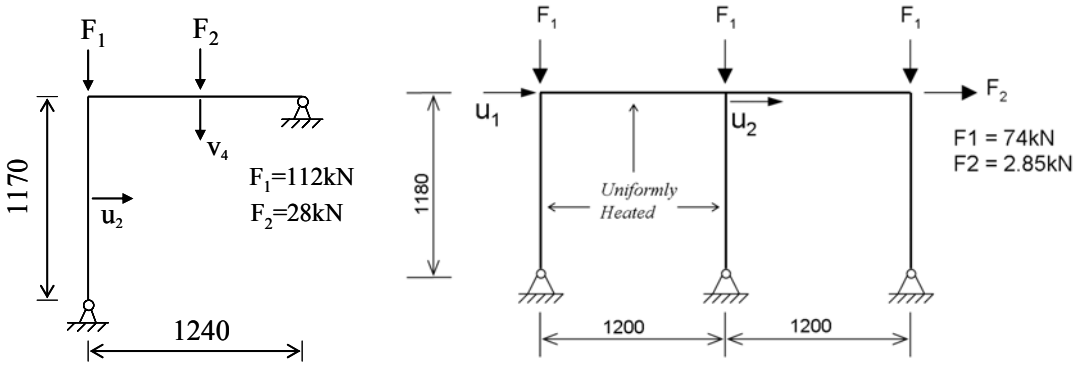
8.1 Introduction

The formation of two-dimensional and three-dimensional thermomechanical ananlysis capability in OpenSees had been presented in Chapter 5 and Chapter 7 respectively. Their performances had been verified by comparing with analytical solutions and ABAQUS results. In this chapter further validations of the extended OpenSees framework were carried out to model experiments. These involves two plane steel frames at elevated temperatures, the Cardington Restrained Beam Test and the Cardington British Steel Corner Test and a full-scale reinforced concrete (RC) frame subjected to simulated seismic damage followed by a fire. For the plane frame tests, DispBeamColumn2dThermal and Steel01Thermal were used to model the two-dimensional beam/column elements and steel materials at elevated temperature respectively. The geometrically nonlinear shell element ShellMITC4GNThermal was used to model the concrete slab and 3D beam/column element DispBeamColumn3dThermal was used to model beams and columns in the frame. The DruckerPragerThermal material in OpenSees was used to the model concrete in the composite floor in the Cardington tests and in the RC frame test. The Pinching4Thermal material in OpenSees was used to model the beams-column joints in the RC frame to consider the cyclic degradation of strength and stiffness under cyclic load. The OpenSees results shows reasonable agreement with the experimental results except the RC frame test which no experimental measurements of structural responses at elevated temperature were recorded due to its malfunction subjected to high temperature.

8.2 Steel frame test

A series of tests on plane steel frames at elevated temperatures were performed in Germany (Rubert and Schaumann 1986). A schematic diagram of two steel frames EHR3 and ZSR1 are shown in Figure 8.1. The braced two-bar frame (EHR3) was subjected to a uniform temperature rise and only one bay of the two-portal frames (ZSR1) was uniformly heated. All structural elements were made of IPE80 I-shaped steel. The temperature dependent stress-strain relationship of Steel 37 used for the experiments is shown in Figure 8.2. The yield stresses and modulus of elasticity are 382MPa and 210000MPa at ambient temperature for EHR3 and 355MPa and 210000MPa for ZSR1, respectively. Steel01 material class was used to model the properties of the Steel 37 material and nonlinear static analysis was conducted in OpenSees. Eight elements were meshed for each beam/column member. This mesh is considered to be fine enough in this case because the same results were obtained if

the mesh was refined by applying sixteen elements. Comparing with modelling from other researchers (Wang et al. 1995; Huang et al. 2000c), the OpenSees predicted deflections show reasonable agreement with the test results as well as those from theses numerical analyses as shown in Figure 8.3. For this steel frame example, a bilinear material was used to model the realistic steel material in the experiment and reasonable qualitative agreement was achieved. The author consider this to be adequate validation considering that actual test conditions (such as restraint, temperature distributions and material behaviour at elevated temperature etc.) in large scale thermal testing (as this was) can not really be fully or accurately represented in models.



(a) Frame EHR3 (b) Frame ZSR1

Figure 8.1: Schematic of the tested steel frames (mm)

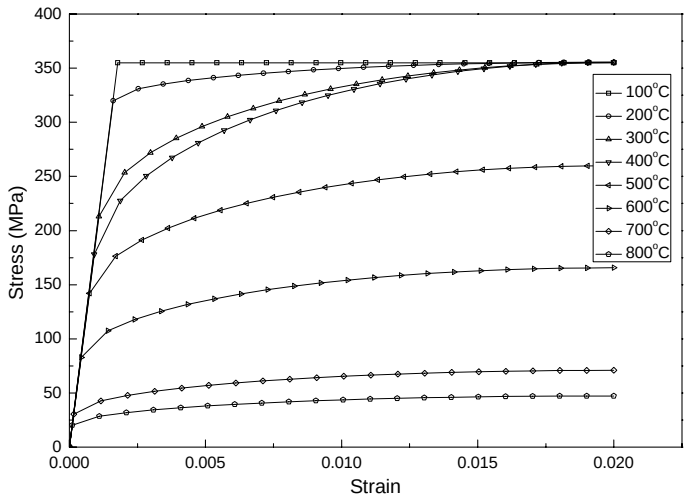
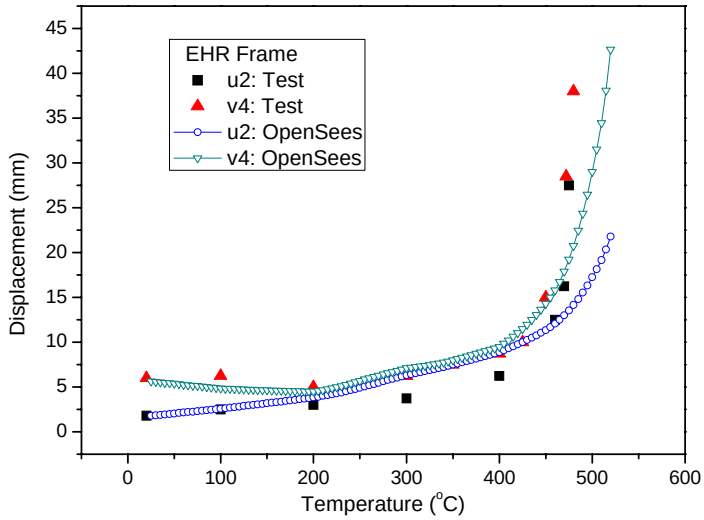
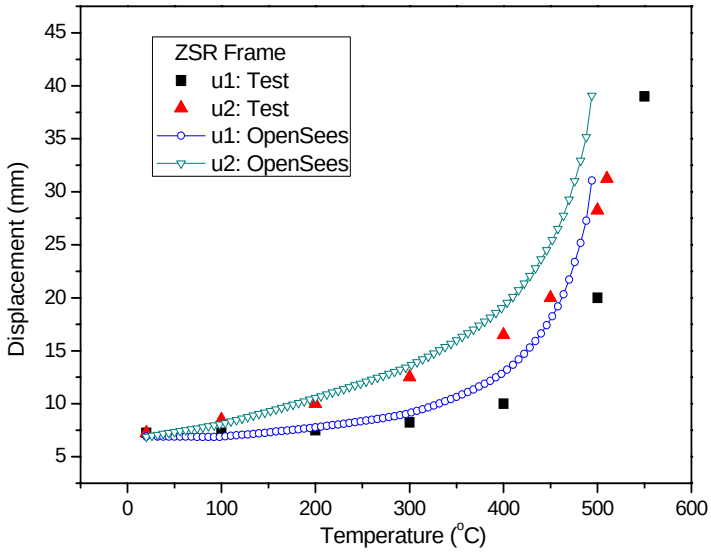


Figure 8.2: Temperature dependent stress-strain curves of steel 37 used in test



(a) Frame EHR3



(b) Frame ZSR1

Figure 8.3: Comparison between predicted and test deflection results

8.3 Cardington restrained beam test

The details of Cardington restrained beam test was introduced in Section 2.3.1. Figure 8.4 shows the locations of the test and the dimensions of the composite slab are shown in Figure 8.5. This test had been numerically modelled by many researchers (Rose et al. 1998; Huang et al. 2000c; Elghazouli et al. 2000; Gillie et al. 2001b). The results show that thermal expansion dominates the response of highly redundant structures under fire condition and that local yielding and large deflections can be beneficial in reducing damage to the

complete structure. However few references show the modelling of horizontal displacement of column at floor level which is modelled in this section using OpenSees. The geometric arrangement, materials, temperature distribution and results comparisons are presented as follows.

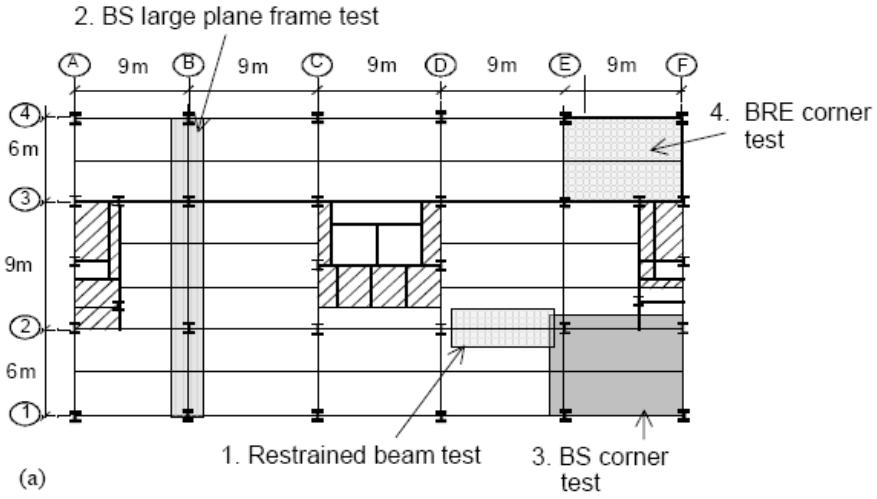
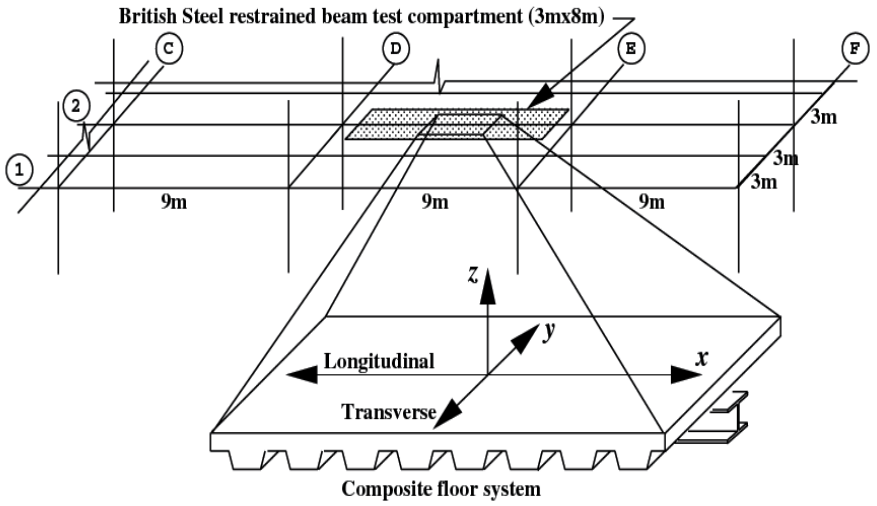
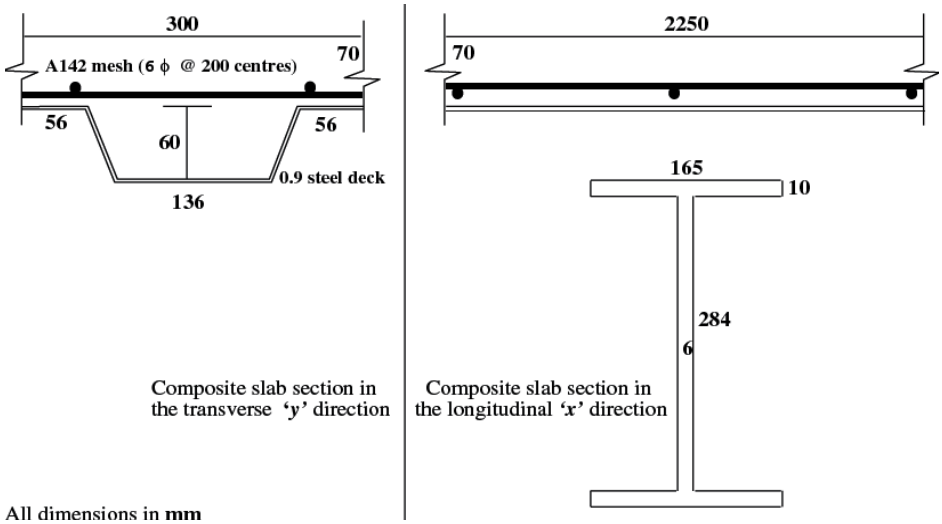


Figure 8.4: Location of the restrained beam test (BRE, 2005)



(a) Layout of the Cardington restrained beam test



(b) Composite cross sections in x and y directions

Figure 8.5 layout and dimensions of the restrained beam (Usmani 2005)

8.3.1 Geometric model

The 3D model of Cardington restrained beam test was set up in OpenSees shown in Figure 8.6. Exploiting symmetry, only half the compartment was modelled and the effect of the surrounding floor was also represented by symmetry boundary conditions. The model consisted of flat slab, concrete ribs and a primary beam in the transverse direction, a column in the middle and the heated (restrained) beam in the longitudinal direction as well as the other two parallel secondary beams. The decked slab was modelled separately by the flat reinforced concrete slab and concrete ribs. The shell element ShellMITC4GNThermal was used to model the flat slab and 3D beam element DispBeamColumn3DThermal was used for modelling the concrete ribs, column and beams. The slab, ribs, column and beams were connected by multi-point constraints (equalDOF). The column was modelled from one floor below the floor on which the test took place to one floor above it. The bottom end of the column was fully fixed whilst at the top only vertical deflection were permitted. The mesh of this restrained beam model in OpenSees is shown in Figure 8.7. A 30×18 elements mesh was used in the *x* and *y* direction of the slab respectively and 16 elements for the column. A mesh convergence study was not carried out which is planned to be checked in the future.

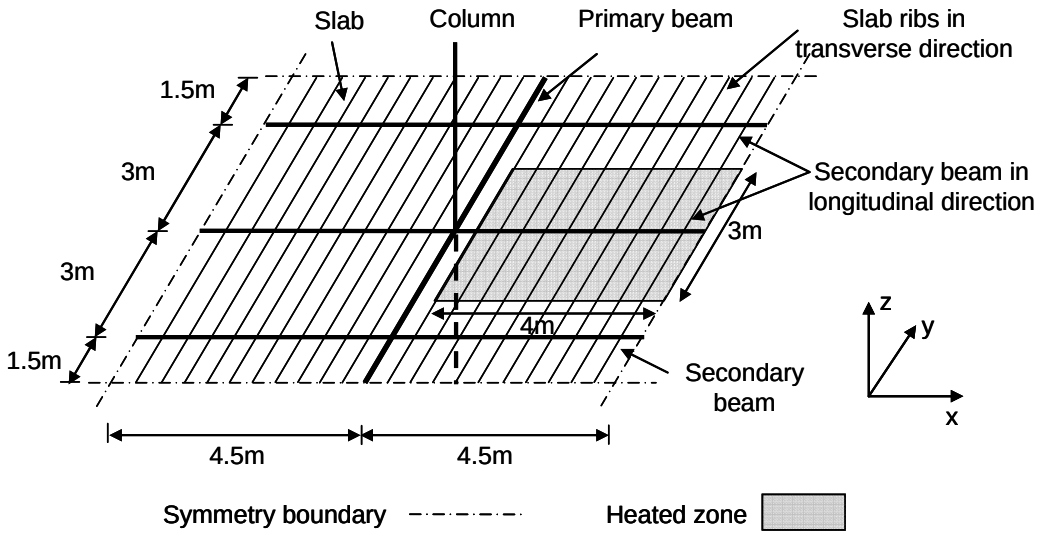


Figure 8.6 Geometric model of the restrained beam test

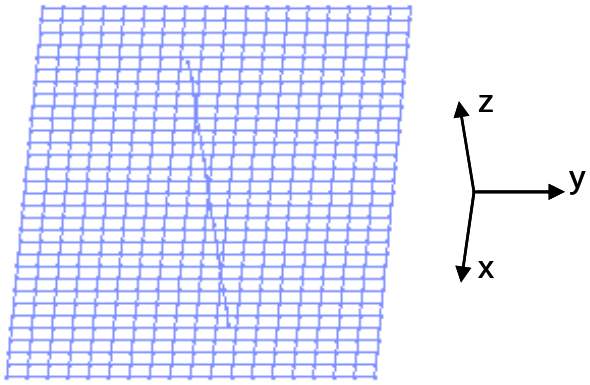


Figure 8.7 Mesh of Cardington restrained beam model in OpenSees

8.3.2 Material properties

The compressive strength of concrete is 48MPa and the yield stress of steel is 280MPa. The material class DruckerPragerThermal was used to model the concrete in slab and Concrete02 for the concrete in the beam/column members. Steel01 is used to model the steel I beam and reinforcements in the beam/column. The reinforcement in the slab was modelled by a smeared layer using elastic material mentioned in Section 7.2.2.1. The corresponding input parameters for these three material classes are shown in Appendix H. The material properties at elevated temperature are according to Eurocode 2 and Eurocode 3.

8.3.3 Results

A uniformly distributed load of 5.48 kN/m^2 (Gillie et al. 2001b) was applied over the entire floor slab and the temperature profile was shown in Figure 8.8. The loading was applied in two stages. First the static load was applied while the structure was unheated. The structure was then heated according to the recorded test temperature-time curves while keeping the static load constant. Nonlinear explicit dynamic analysis was carried out in OpenSees. Newmark method with γ=0.5 and β=0.25 and Newton-Raphson solution algorithm were applied. Compared with the static analysis, this dynamic analysis was easier to converge and derived the same results with static analysis. In addition, the solution algorithm can be automatically transformed to Modified Newton algorithm or Broyden algorithm (Mazzoni et al. 2007) when convergence is difficult to achieve. This explicit dynamic analysis procedure was applied to all the cases (except Section 8.2) in this Chapter. Figure 8.9 shows the deformed shape of the 3D model in OpenSees after 120 minutes. Figure 8.10 shows the mid-span deflection of the restrained beam. Comparing with many other references (Rose et al. 1998; Huang et al. 2000c; Elghazouli et al. 2000; Gillie et al. 2001b), OpenSees result shows reasonable agreement with experimental data as well as these numerical modelling. A relatively large difference between OpenSees results and experimental data may be due to that the DruckerPragerThermal material can not model the compressive descending branch of concrete material in the slab which is not the case in other's work mentioned above. Another reason may be that the temperature distributions applied in the OpenSees modelling were taken as an average value of different locations of thermal couples in the restrained beam and slab. The differences between these average temperature distributions and real measurements may lead to the differences seen in Figure 8.10. In addition to the mid-span deflection of the slab, however few researchers focused on the horizontal displacement of the column at floor level (Sanad et al. 2000a). This comparison is considered to be significant because it is strongly related to the overall expansion of the composite beam, which is partly restrained by the relatively cooler surrounding structure. After an initial increment due to the thermal expansion of restrained beam, a plateau of the column horizontal displacement is observed in the test at moderate temperatures which corresponds to the restrained beam reaching its axial resistance followed by a later increase in displacement associated with thermal expansion of the concrete slab which becomes considerably hotter. The reference (Sanad et al. 2000a) successfully models this trend but underestimates the displacement. In this thesis, good qualitative and quantitative agreements of the horizontal displacement of column are achieved as follows.

Figure 8.11 shows the horizontal displacement of the column. Figure 8.11(a) shows the results that the "equalDOF" was used to model the connection between slab, rib and beam/column. This command is used to constraint the degree of freedom of slave nodes to be the same as those at the master nodes. The agreement is poor and suggests that the "equalDOF" is unable to capture the correct structural behaviour in the test. If the rigid link element was used instead of "equalDOF" connection type, the OpenSees results shows good agreement with test data as shown in Figure 8.11(b). The OpenSees manual does not give clear explanation about the theory of the command "equalDOF" as well as rigid link element. It seems that the "equalDOF" can not consider the coupled horizontal movement of the two components (top concrete slab and bottom steel beam) which means the thermal expansion of the steel beam is not restrained by the concrete slab at the beginning of heating and in turn the thermal expansion of the concrete slab can not drive the movement of the steel beam at high temperature. Therefore the OpenSees results shown in Figure 8.11(a) just represent the horizontal displacement of the column due to the thermal expansion of the restrained beam and not plateau can be modelled because the effect of concrete slab is not considered. In contrast, the rigid link element can capture the coupled effect of the steel beam and concrete slab. Further work should be done to check the corresponding codes of these two commands to give a clear explanation. It is interesting to see that there is a plateau in the horizontal displacement of column as shown in Figure 8.11(b). As temperature rises, the horizontal displacement of the column increases until about 250 ^{o}C and keeps almost unchanged until 500 ^{o}C and starts to increase again. The initial increase is primarily due to the thermal expansion of the steel beam. As the reducing yield stress of steel at elevated temperature, the compressive force in the steel beam induced by restrained expansion by surrounding members begins to decrease and meanwhile significant uniform temperature develops in the concrete slab resulting in a large thermal expansion which produces increasing compressive force in the composite beam. This increasing compressive force due to restrained expansion of slab makes up the decreasing compressive force in the steel beam and maintains the resultant force in the composite beam in a steady condition. The second part of the increase in the column horizontal displacement is due to the thermal expansion of the concrete slab considering a rapidly increasing uniform temperature developed after 80 minutes of the fire duration

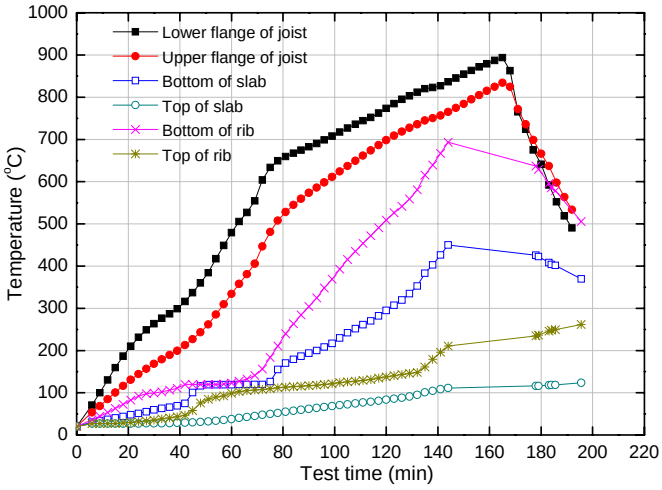
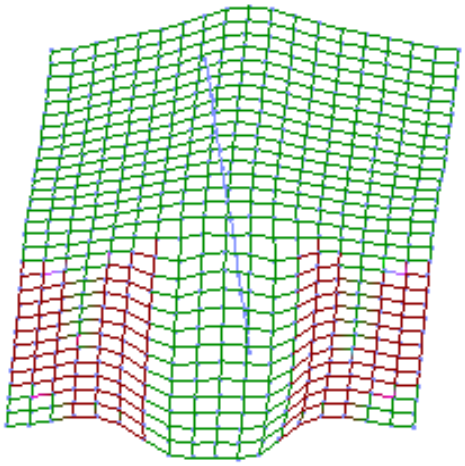
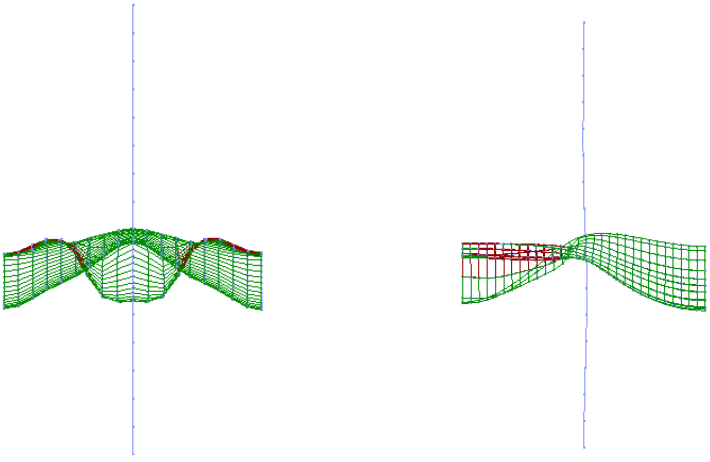


Figure 8.8: Temperature distribution in the composite slab of restrained beam test



(a) 3D



(b) ZY plane **(c) ZX plane**

Figure 8.9: Deformed shape of Cardington restrained beam model in OpenSees (×100)

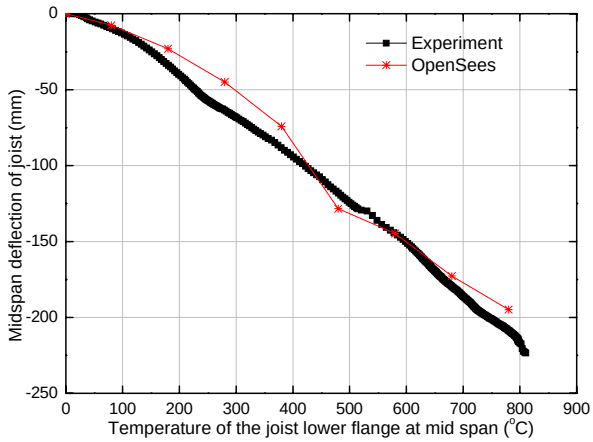
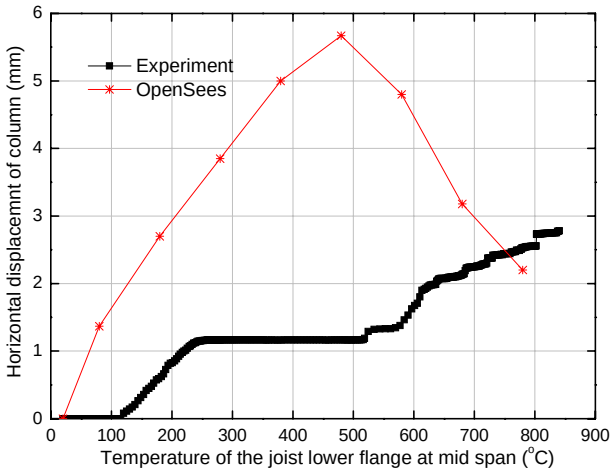
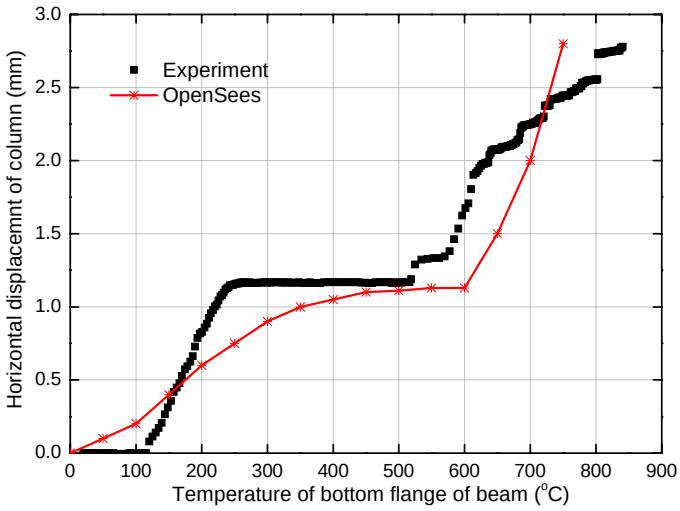


Figure 8.10: Mid-span deflection of the restrained beam against temperature



(a)



(b)

Figure 8.11: Horizontal displacement of the column against temperature: (a) equalDOF; (b) rigid link.

8.4 Cardington corner test

The British Steel corner test was introduced in Section 2.3.1. This corner test was also modelled by many researchers (Huang et al. 2000c; Gillie et al. 2002; Elghazouli et al. 2000). These analyses found that a significant load carrying mechanism is tensile action in the reinforcement in the concrete floor slab at extreme temperatures. However few researchers focused on the modelling of horizontal displacement of the internal and edge column which was modelled in this section using OpenSees. The whole corner compartment was modelled in OpenSees and the effect of the surrounding floor was also represented by symmetry boundary conditions (as shown in Figure 8.13). The same material and element classes were used to model the slab, ribs, beams and columns as for the restrained beam test model described earlier. The mesh of this corner test model in OpenSees is shown in Figure 8.14. A 45×15 element mesh was used in the x and y direction of the slab respectively and 16 elements for the column. A mesh convergence test was not carried out and will be checked in the future.

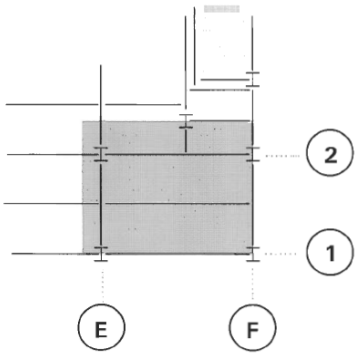


Figure 8.12: Schematic of BS corner test (BS, 1999)

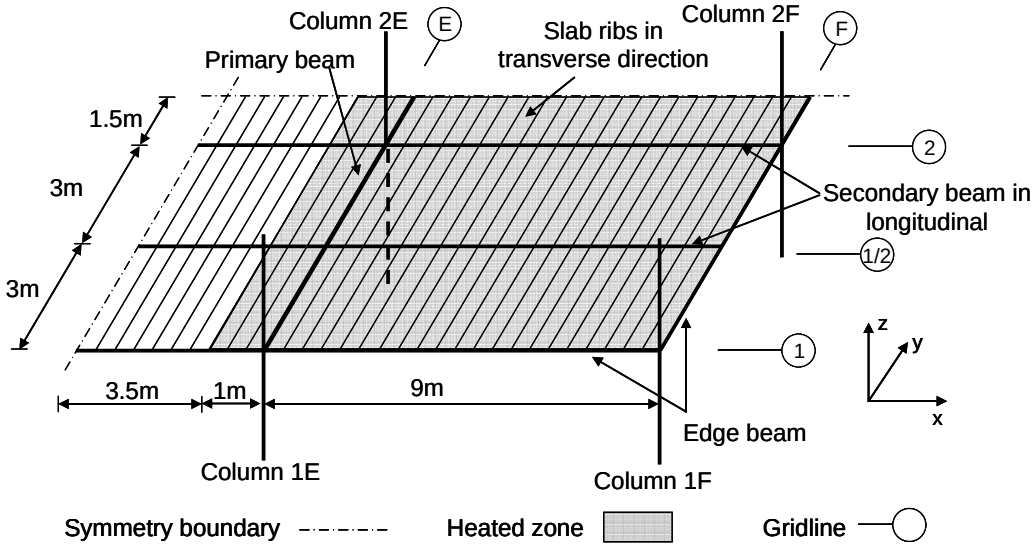


Figure 8.13: Geometric model of the restrained beam test

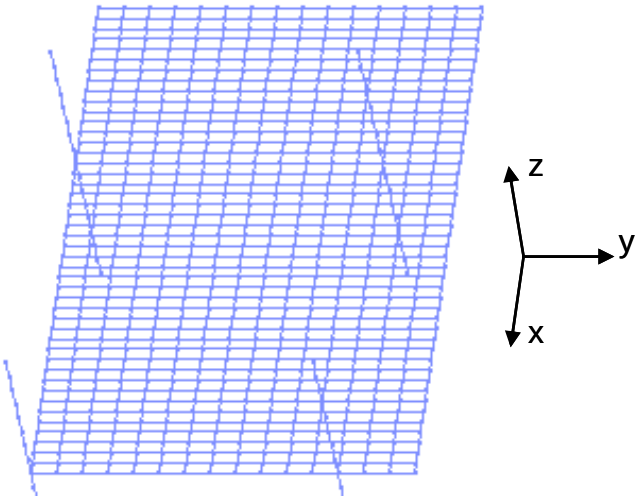
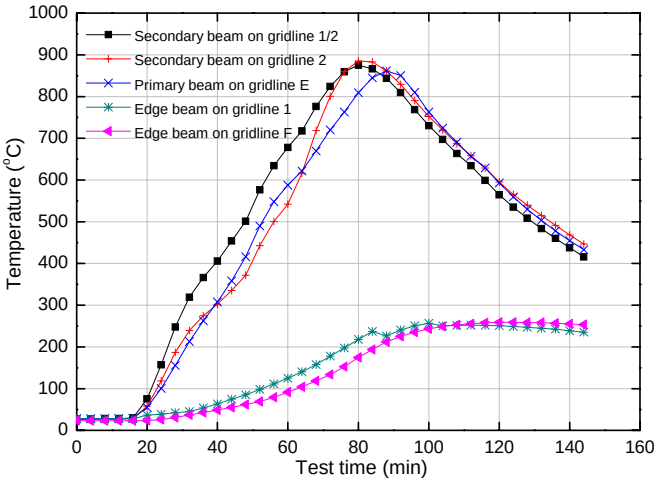


Figure 8.14: Mesh of Cardington corner test model in OpenSees

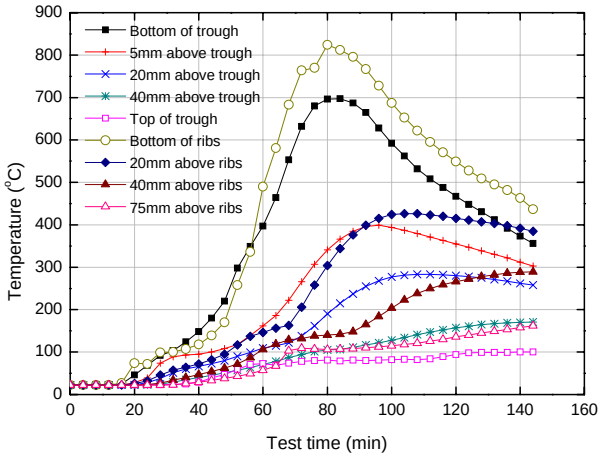
The floor slab was subjected to a uniformly distributed load of 5.81 kN/m^2 (Gillie et al. 2002) and the temperature distributions in the structural components were shown in Figure 8.15. Figure 8.16 shows the deformed shape of the 3D model in OpenSees after 80 minutes. Figure 8.17 shows the mid-span deflection of the beam. Comparing with other forms of analysis (Huang et al. 2000c; Gillie et al. 2002; Elghazouli et al. 2000), reasonable agreement was achieved between OpenSees results and experimental data. Similar reasons can be explained with restrained beam test for the difference as shown in Figure 8.17. Similar with the restrained beam test, few studies had focused on the horizontal displacement of the internal and edge columns in this test. Therefore Figure 8.18 shows comparisons of OpenSees results

of the horizontal displacement of the four columns and corresponding experimental measurements. The horizontal displacement of column 1E in the y direction is larger than that of column 1F and this is because the temperature in the beam on gridline E is higher than that of beam on gridline F. Similarly, the horizontal displacement of column 2F in the x direction is larger than that of column 1F because of higher temperature in the beam on gridline 2. The OpenSees results agree well with the experiment results.

This sort of comparison is usually very difficult to make because of the difficulties of representing the complex realities of a real test structure, particularly with a relatively simple model such as this. It can therefore be concluded with reasonable confidence that OpenSees has been validated for steel framed composite structures subjected to fire.

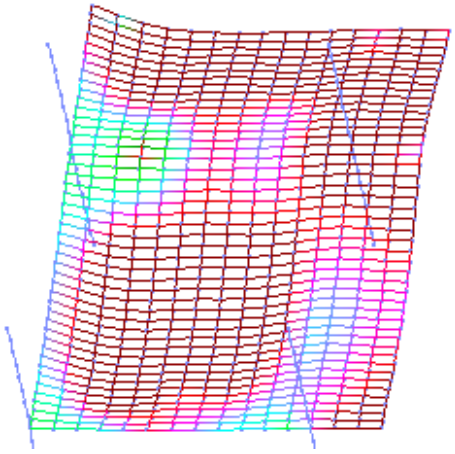


(a) Temperature distribution in the beam

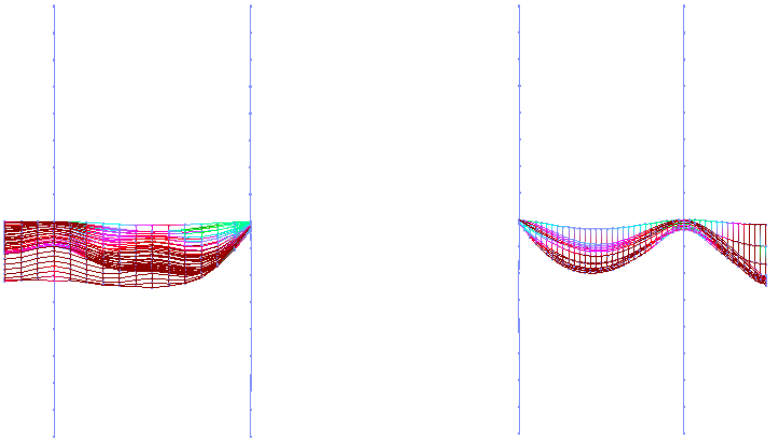


(b) Temperature distribution in the slab

Figure 8.15: Temperature distribution of the corner test



(a) 3D



(b) ZY plane **(c) ZX plane**

Figure 8.16: Deformation shape of Cardington restrained beam model in OpenSees (×100)

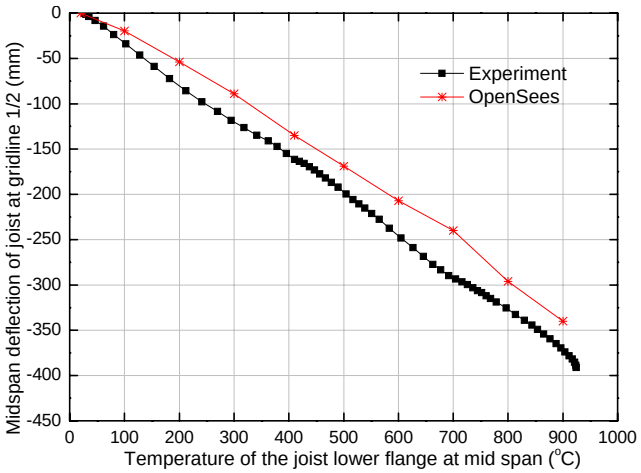


Figure 8.17: Mid-span deflection of the secondary beam against temperature

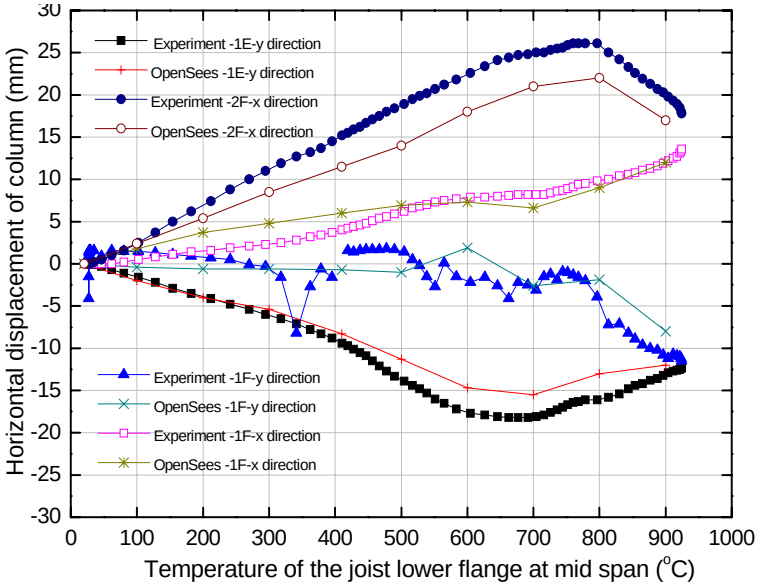


Figure 8.18: Horizontal displacement of the columns against temperature

8.5 Roorkee RC frame subjected to fire following simulated seismic damage

As part of a collaborative international research project involving IIT Roorkee in India and University of Edinburgh in UK, a series of tests on a reinforced concrete (RC) frame were planned. These tests began in February 2011 and the first test frame was subjected to cyclic loading to simulate earthquake damage at first and then it was subjected to a one-hour compartment fire fed by a uniform flow of kerosene into a $1m^2$ tray in the centre of the compartment floor providing a relatively uniform post-flashover temperature in the compartment. Numerical simulations were conducted to predict the behaviour of the RC frames under seismic loading and subsequent fire using OpenSees. Unfortunately, no test data of the structural responses in fire was recorded because of the malfunction of displacement measurement due to high temperature. Therefore there are no comparisons of OpenSees results to experimental results. The observation from the tests and OpenSees predictions are expected to generate a useful understanding of the behaviour of earthquake damaged RC structures in fire.

8.5.1 Layout of the test RC frame

The test frame is assumed to be a single storey sub-assemblage of a four storey building as shown in Figure 8.19. Figure 8.20 shows the layout of the tested RC frame. The frame

consists of four 300×300mm columns supporting four 230×230mm roof beams and a 4 metre square 120mm thick slab. Four 230×230mm plinth beams centred 3 metres below the mid-surface of the slab connect all the columns. The details of dimension and reinforcement of column, beam and slab are shown in Figures 8.21-8.23.

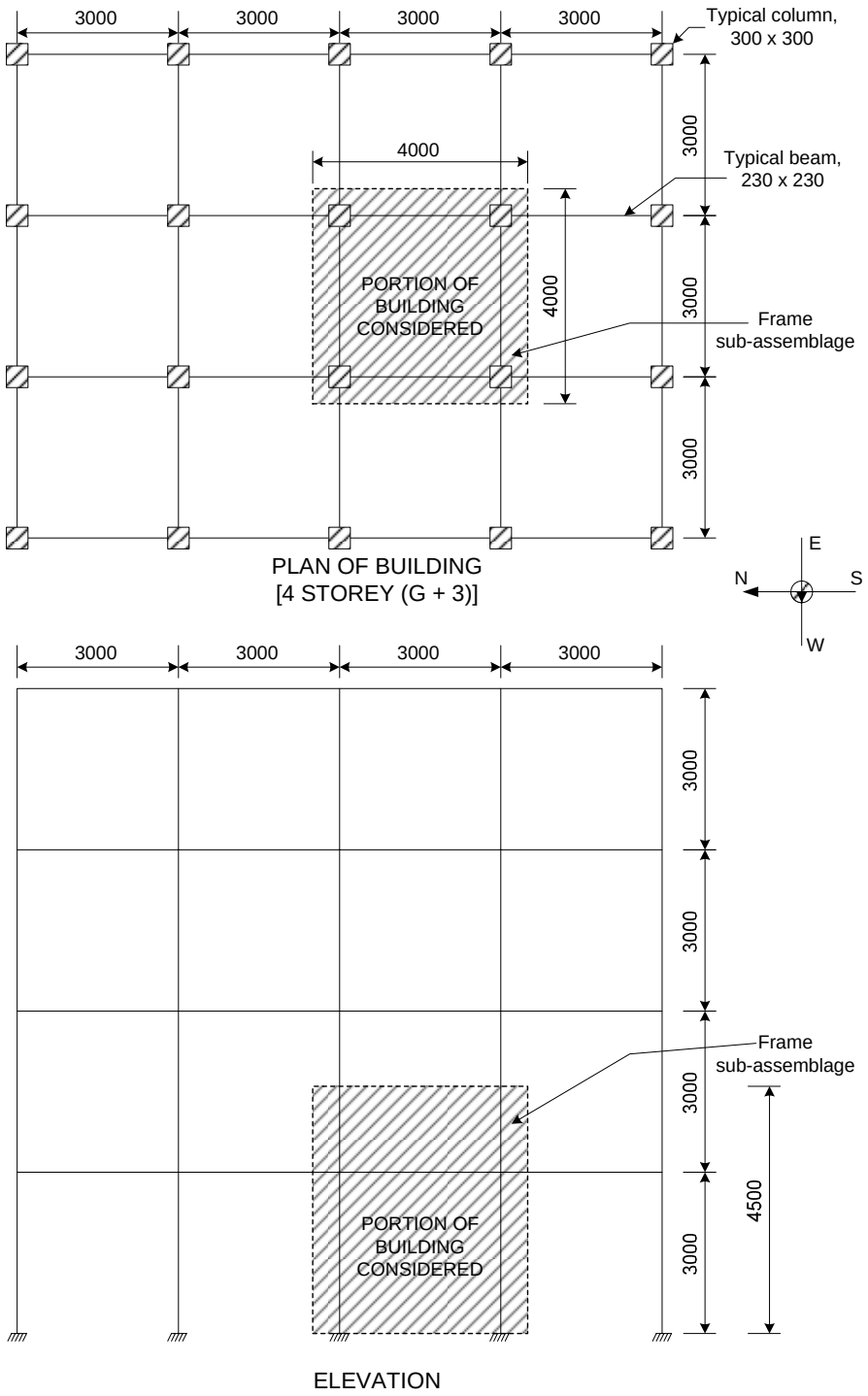
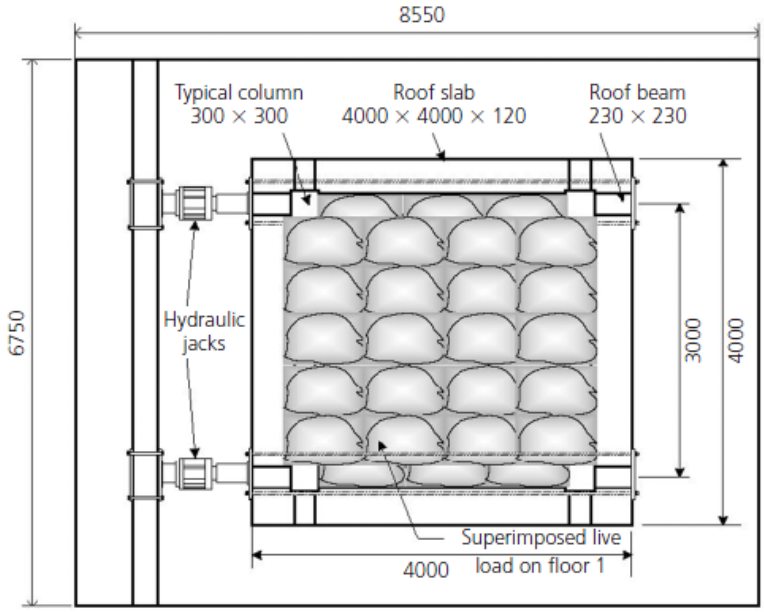
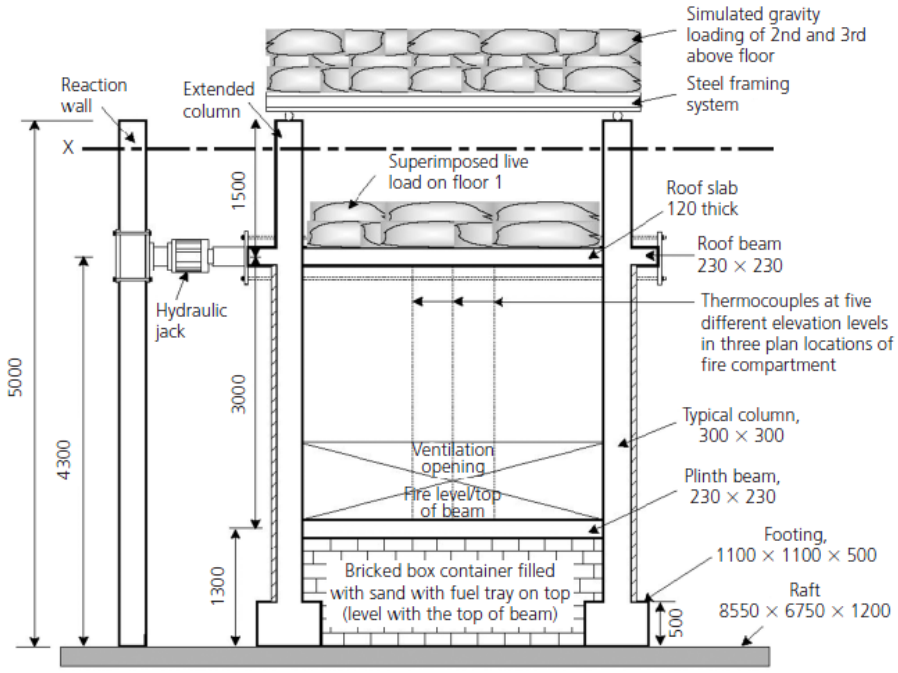


Figure 8.19: Building configuration and the frame sub-assemblage.



(a) Plan of the test setup (at X-X)



(b) elevation of the test setup

Figure 8.20: Schematic of the test RC frame (all dimensions in mm)

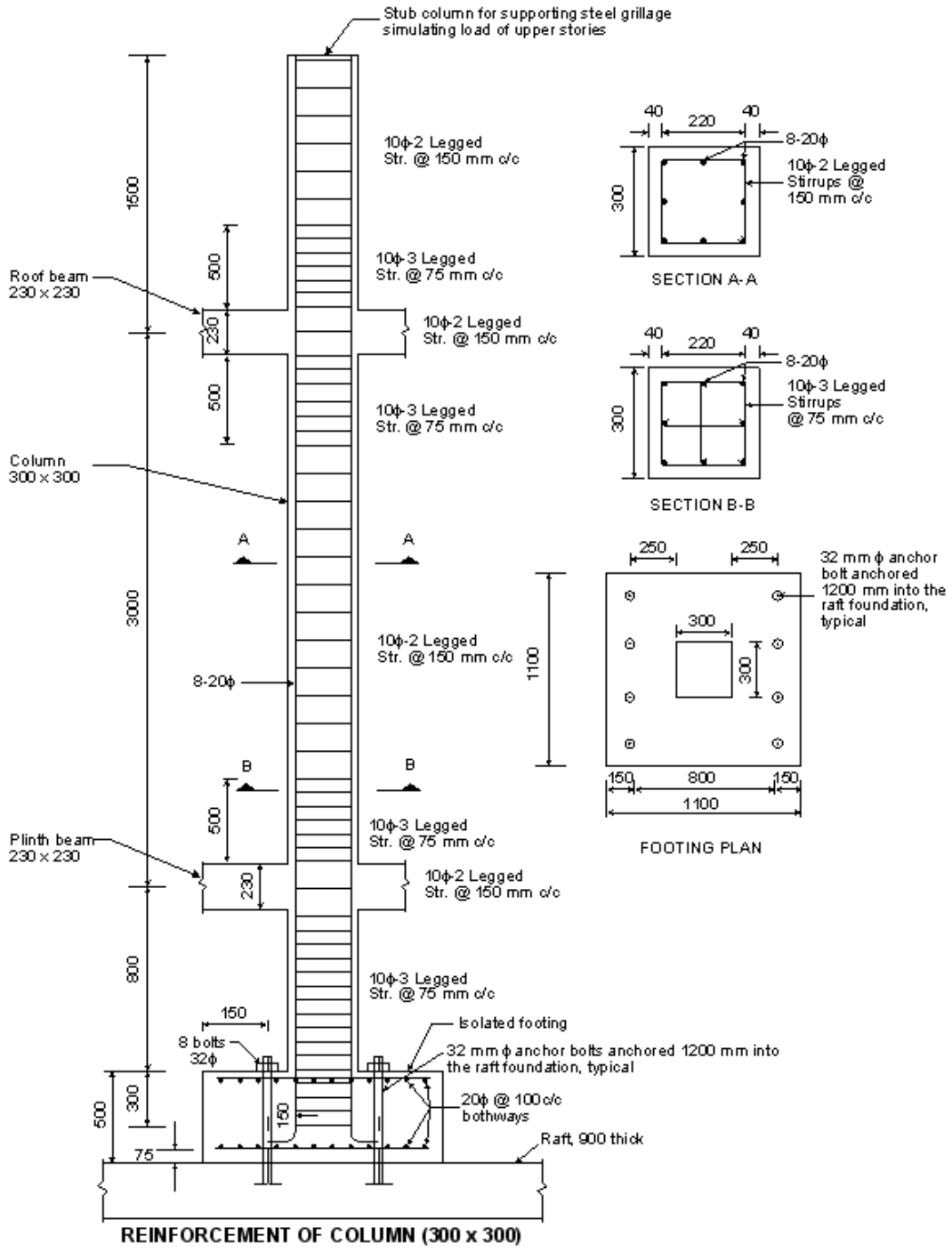
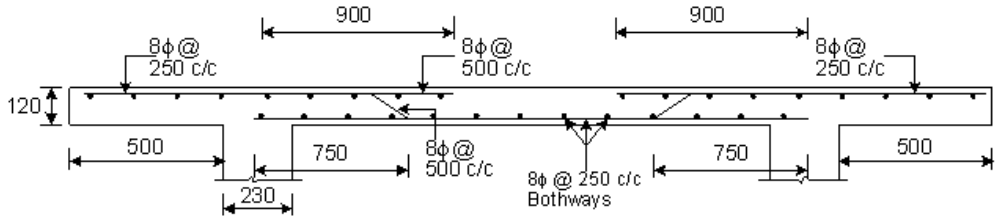
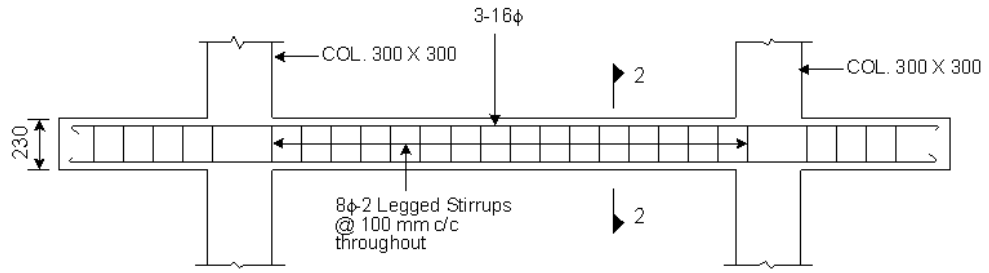


Figure 8.21: Details of a typical column.

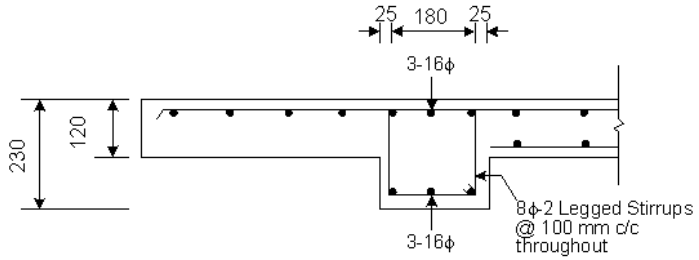


SECTION THROUGH SLAB

Figure 8.22: Detailing of slab reinforcement



L-SECTION OF BEAM (230 X 230)



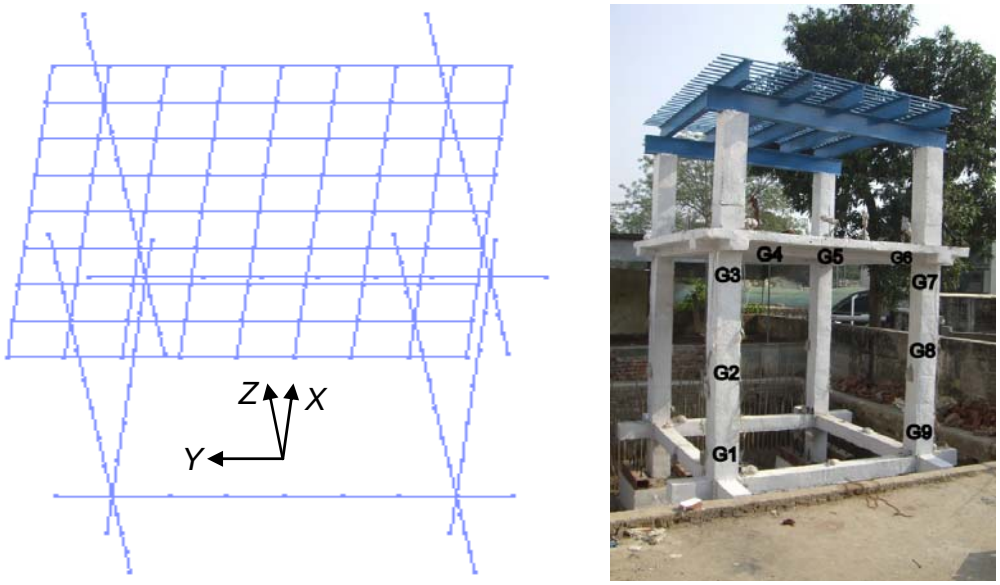
SECTION 2-2

Figure 8.23: Details of beam reinforcement

8.5.2 OpenSees model

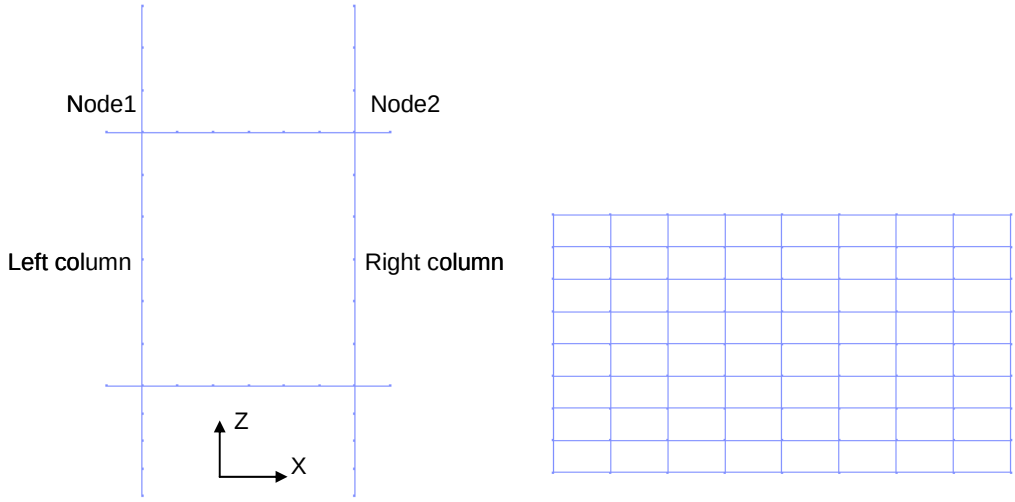
A 3D model was set up in OpenSees to model the test RC frame shown in Figure 8.24. The geometrically nonlinear shell element ShellMITC4GNThermal was used to model the reinforced concrete slab and 3D beam/column element DispBeamColumn3dThermal was used to model the columns, roof beams and plinth beams. Figure 8.25 shows the mesh of the 3D model in OpenSees. 14 and 8 elements were used for columns and beams respectively and an 8×8 mesh was used for the slab. A mesh convergence study was not carried out and it was planned to be checked in the future. The reinforcements in the column, beam and slab were modelled by a combined smeared reinforcement layer defined in the section of shell and beam element. An equivalent uniformly distributed load of 2.3kN/m^2 was applied on the

slab and a concentrated force of 80kN was applied on the top of each column. The concrete in the slab was modelled by DruckerPragerThermal material and Concrete02 and Steel01 were used to model the concrete and reinforcement in the columns and beams, respectively. Pinching4Thermal material was used to model the beam and column element around the beam-column joint. The compressive and tensile strength of concrete at ambient is 34MPa and 3.4MPa, respectively. The initial elasticity modulus of concrete is 27.2GPa. The yield stress of reinforcing steel is 450MPa. The input data for these materials are listed in Appendix I.



(a) OpenSees **(b) Test**

Figure 8.24: 3D model of the test frame



(a) **(b)**

Figure 8.25: Mesh of the beams, columns and slab: (a) beams and columns in ZX (ZY) plane; (b) slab in XY plane

8.5.3 Results

The desired nominal cyclic loading history of the test frame is shown in Figure 8.26 and the hysteretic curve of the test frame is shown in Figure 8.27. The cyclic load was applied to the frame in the X direction (Figure 8.24). The maximum displacement of 95mm in the "push" cycle and 85mm in the "pull" cycle was recorded corresponding to base shears of 316 and 267kN, respectively. Figure 8.28 shows the hysteretic curve of OpenSees model. The comparison of peak load-displacement relationship from test and OpenSees is shown in Figure 8.29. The deformed shapes of the frame model in OpenSees are shown in Figure 8.30 (a magnification factor of 20 was used). Figure 8.31 shows the vertical reaction forces of the columns in the ZX plane and their peak values at the peak displacement of the roof during cyclic load are shown in Figure 8.32. During the cyclic loading, the vertical reaction force of the columns increases with increasing peak displacement. Vertical reaction in the left column changes from compression to tension when the frame is pushed in the positive X direction and vice versa for the right column moving in the negative X direction. When the frame moves to the peak displacement of x=90mm, compressive reaction of 216kN and tensile reaction of 38kN occurred in the right and left columns, respectively. Similarly, when for the opposite peak displacement of x=-90mm, compression of 214kN and tension of 35kN exists in the left and right columns, respectively.

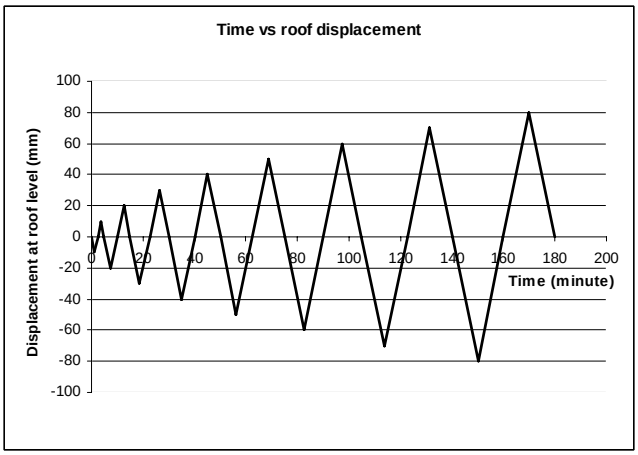


Figure 8.26: Proposed loading history of the test frame

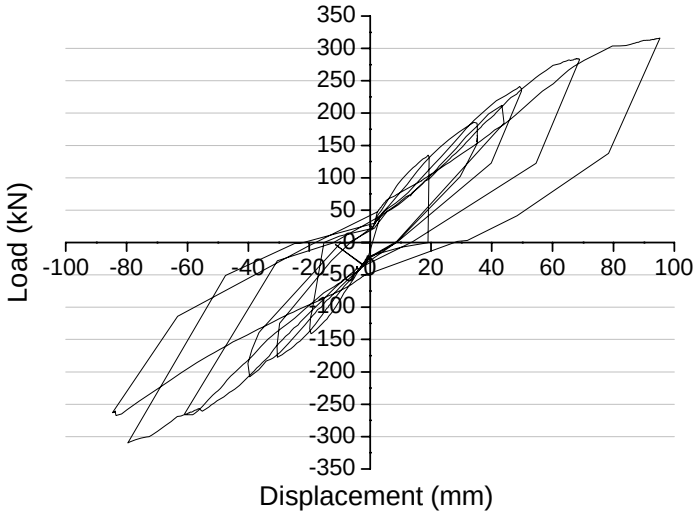


Figure 8.27: Hysteretic curve of the test results

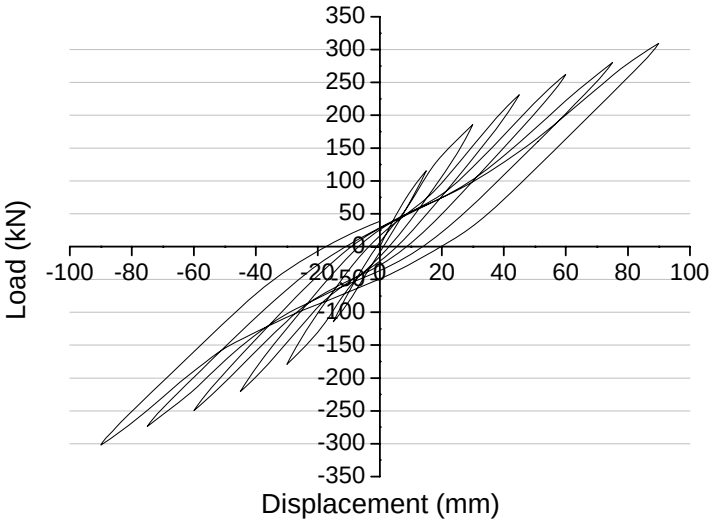


Figure 8.28: Hysteretic curve of OpenSees results

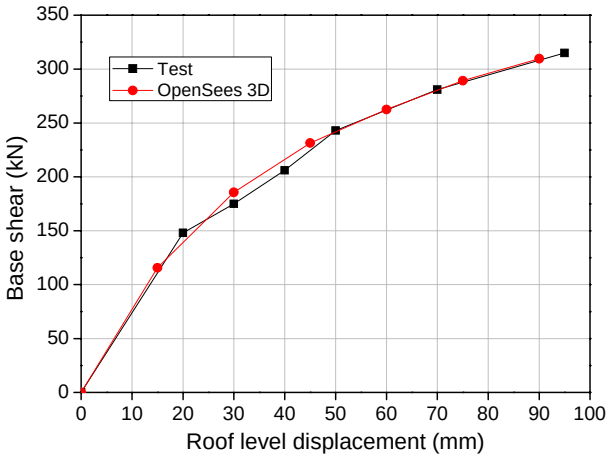
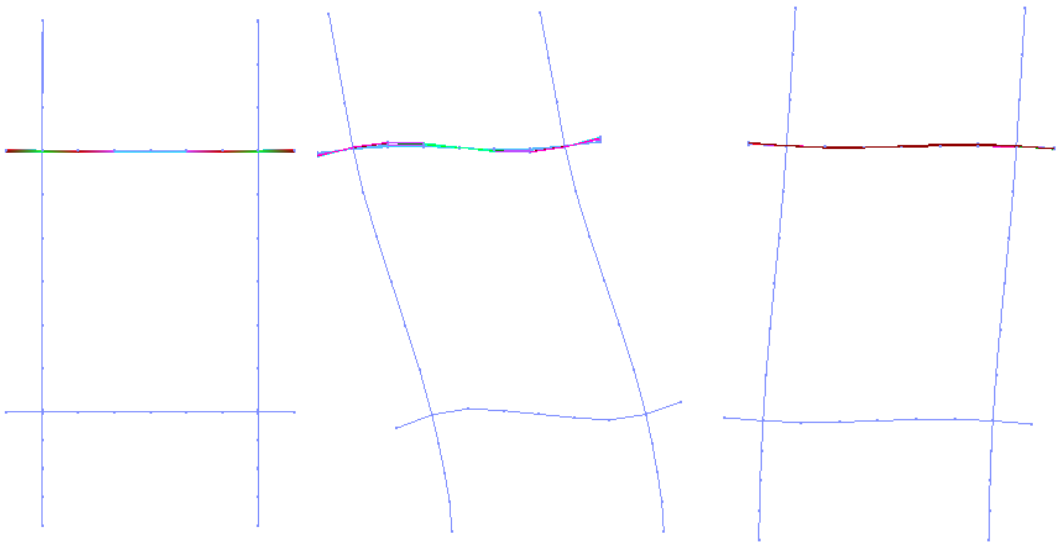


Figure 8.29: Comparison of base shear-roof displacement relationship



(a) (b) (c)

Figure 8.30: deformed shape of the RC frame under cyclic load in OpenSees: (a) initial position under UDL; (b) position at peak displacement in ZX plane (×20); (c) residual after cyclic loading in ZX plane (×20).

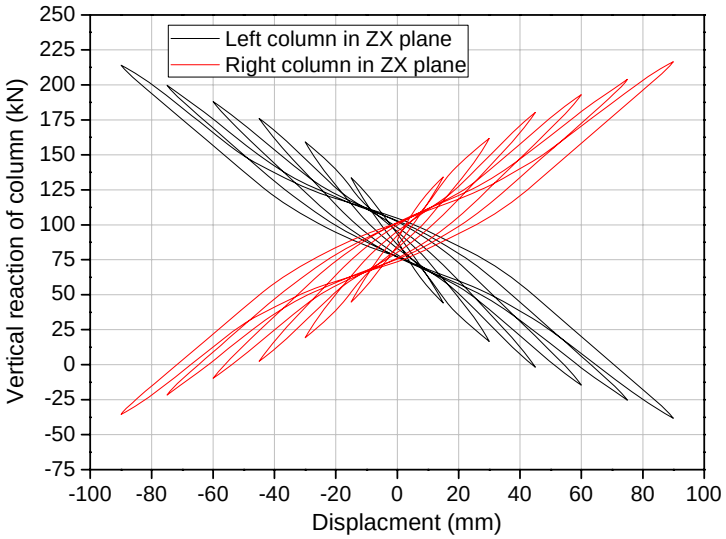


Figure 8.31: Vertical reaction force (Z direction) of columns under cyclic load

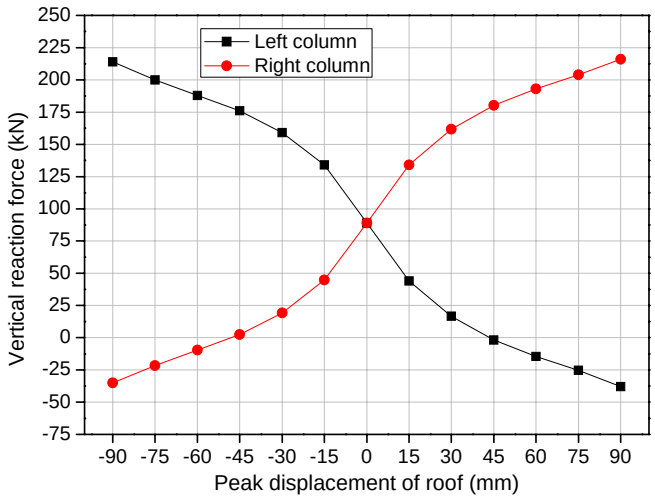


Figure 8.32: Vertical reaction force (Z direction) of columns against peak displacement

After the cyclic loading of the test frame, a fire test was conducted shown in Figure 8.33. The fire compartment of size 3m×3m×3m was constructed with a 1m high opening along the full length of the wall at the bottom of one side (Figure 8.33(a)). According to the temperature distribution in the test frame, the average temperature distributions in the beams, columns and slab used in the OpenSees model are shown in Figures 8.34-8.37.

(a) Pre-damaged frame **(b) Fire test in progress** **(c) Fire damaged frame**

Figure 8.33: Fire test of the pre-damaged frame

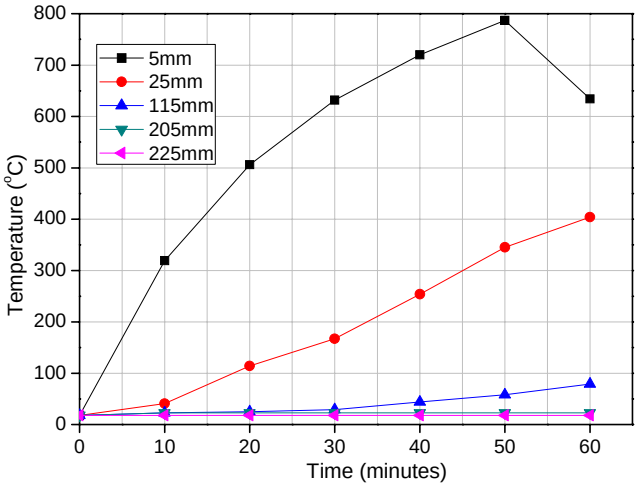


Figure 8.34: Temperature profiles of plinth beam

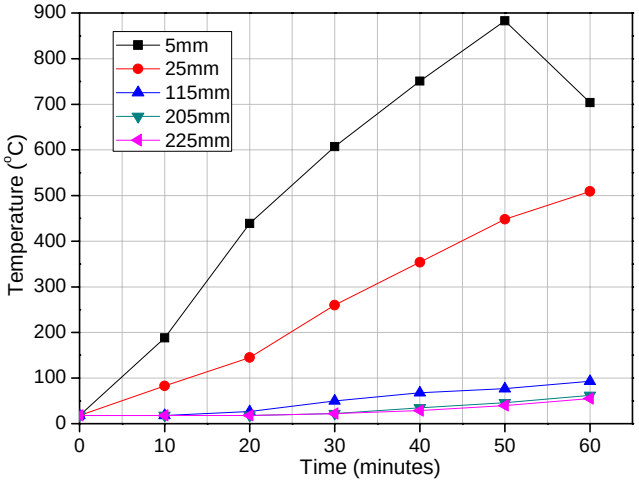


Figure 8.35: Temperature profiles of roof beam

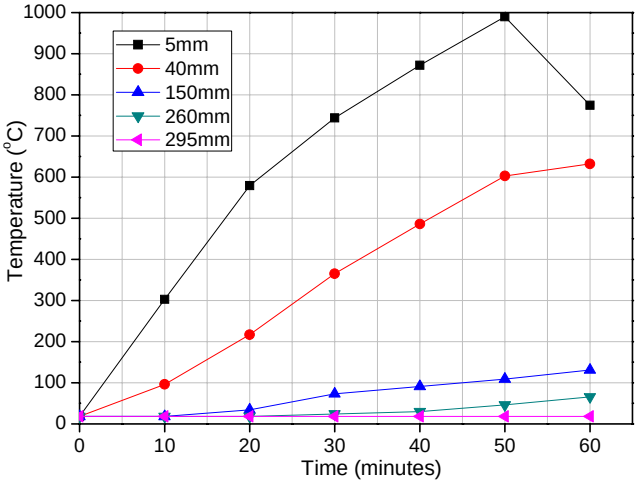


Figure 8.36: Temperature profiles of column

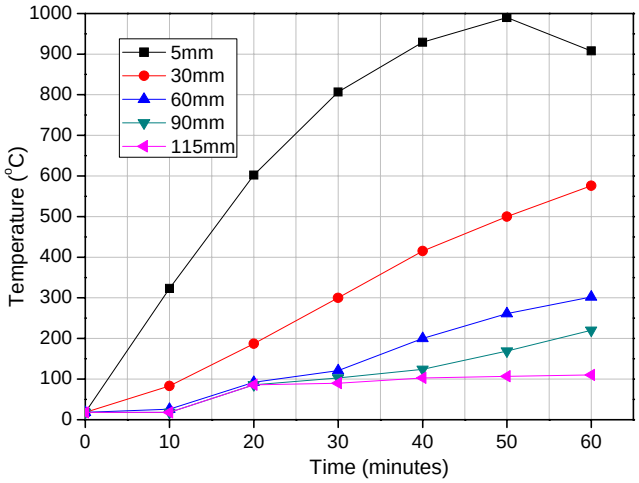
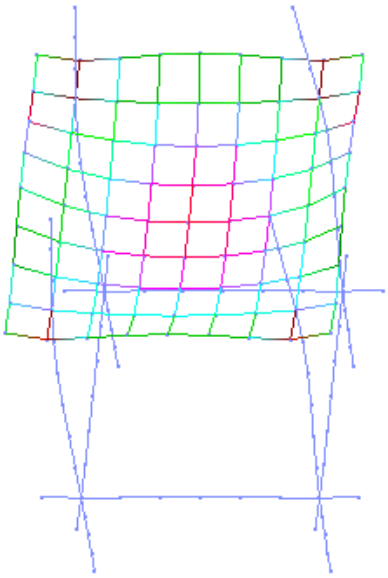
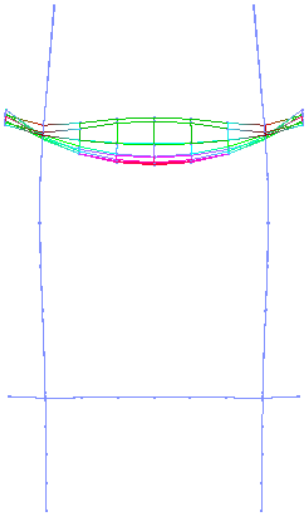


Figure 8.37: Temperature profiles of slab

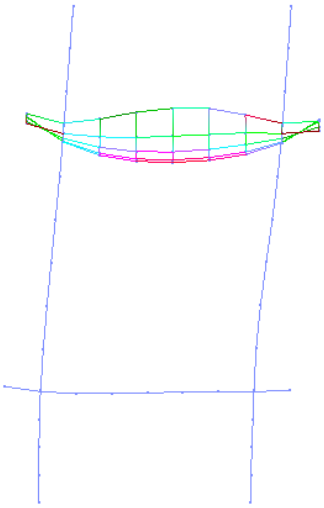
The deformed shapes of the test frame in fire are shown in Figure 8.38. Unfortunately, there is no test data of the structural responses because of the malfunction of displacement measurement due to high temperature. The test is expected to be repeated in August 2012. In the following section only the numerical results are presented to highlight the behaviour of the damaged RC frame in fire. Figure 8.39 shows the deflection at the centre of slab against time. The deflection of the slab keeps increasing until about 50 minutes and thence remains unchanged. At this time the temperature of the bottom surface of the slab begins to decrease as shown in Figure 8.37. Figure 8.40 shows the horizontal movement of the column at the roof level during the fire response simulation. Both nodes moved back (in the positive X direction, thereby recovering some of the permanent residual displacement at the end of the cyclic loading) at the beginning of the heating. This is because initial heating tends to stiffen the frame rather like pre-stressing. The frame starts to move left (in the negative X direction) for higher temperatures. Figures 8.41 and 8.42 show the horizontal and vertical reactions in the columns (ZX plane) during the fire respectively. At the beginning the horizontal reactions in both columns were positive but the thermal elongation at elevated temperatures in the roof slab and beams made the two columns move in opposite directions. There is not much change in the vertical reaction of the columns as shown in Figure 8.42.



(a) 3D



(b) ZY plane



(c) ZX plane

Figure 8.38: Deformation shape of test frame in fire in OpenSees (×20)

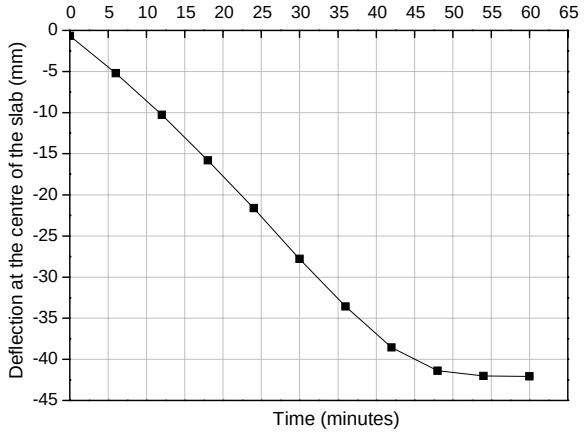


Figure 8.39: Deflection at the centre of slab of OpenSees model

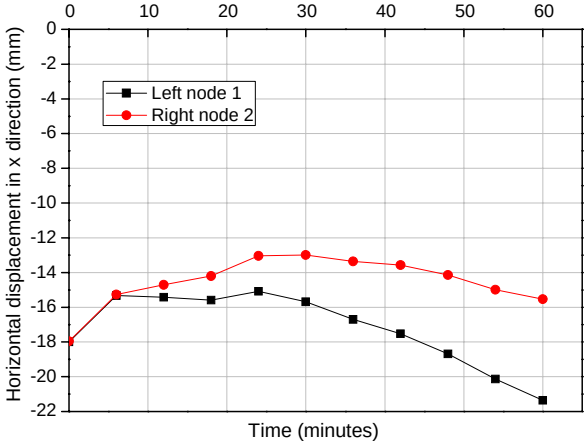


Figure 8.40: Horizontal displacement of column at roof level of OpenSees model

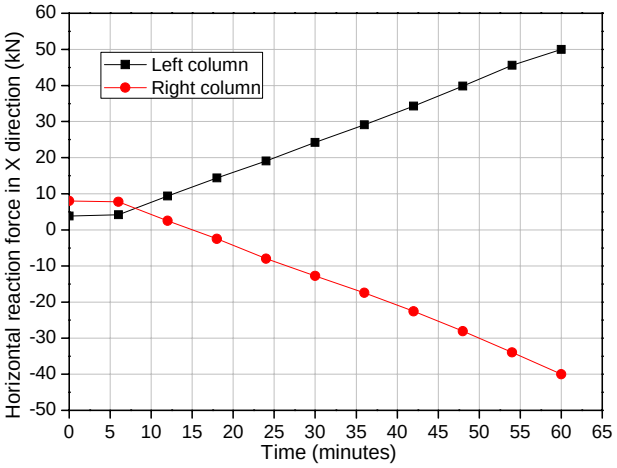


Figure 8.41: Horizontal reaction in columns of OpenSees model

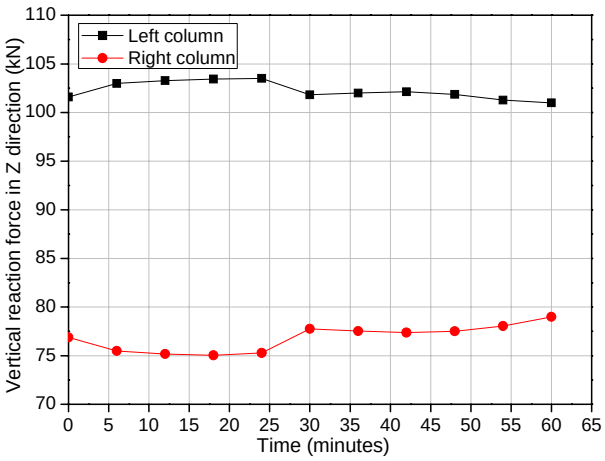


Figure 8.42: Vertical reaction in columns of OpenSees model

Comparisons between OpenSees predictions and experimental measurements show reasonable agreements. The difference for the mid-span slab deflection between OpenSees results and experimental data may be because of the average temperature distribution applied in OpenSees model as well as unability of DruckerPragerThermal material class to model the descending branch of concrete material in the slab. It is proved that the rigid link element should be used to model the connection between the concrete slab, concrete ribs and the beam/columns. It is a pity that there is no experiment data for the Roorkee RC fame test in Section 8.5. A similar test had been repeated and the deflections of the slab as well as the horizontal displacements of columns were obtained. Further numerical modelling of this new test using OpenSees is underway.

Chapter 9

Conclusions and Further Work

9.1 Introduction

The aim of this thesis has been to develop the software framework OpenSees to enable thermomechanical analysis of structures. The development of this capability involves creating new thermal load classes to define the temperature distribution in the element and creating temperature dependent materials, sections and element classes by modifying the existing corresponding classes in OpenSees. The performance of the extended OpenSees framework was tested by series of benchmark cases and experimental data. The agreement of results between OpenSees and other analytical and numerical solutions and also validation against experimental data shows that the extended OpenSees is able to satisfactorily perform thermomechanical analysis of steel framed composite structures.

9.2 Summary and conclusions

The thesis is about the extension of OpenSees to deal with thermomechanical analysis of steel-framed composite structures. The work in this thesis can be mainly classified into two parts: (1) modifying the existing OpenSees framework to include temperature related methods and functions; (2) testing the performance of the extended OpenSees against a good number of benchmarks and experimental data. The most important contribution of my work is that we can model "structures in fire" better with OpenSees. Spread the open source philosophy to a new community (fire community) which will be the future tool of structural engineering research. Further more, the developed OpenSees can be used for research on multi-hazard modelling using existing strength of OpenSees (earthquake simulation modelling). It is part of the whole project which creates a complete tool for fire engineering involving fire dynamic simulation (CFD), heat transfer and mechanical analysis. Further work is planned to link OpenSees to the open source CFD model OpenFOAM (capable of modelling compartment fires) leading to a fully automated software framework for modelling fire, heat transfer and structural response. The following conclusions can be drawn:

- New thermal load classes Beam2dThermalAction and ShellThermalAction were developed to define the temperature distribution in the beam and shell element respectively. At this stage the temperature is defined to be varying through the depth of the section of beam/shell element and uniform along the beam element and in the shell plane. Nine points along the height of the section can be used to model

arbitrary temperature distributions and two points (at bottom and top) used for linearly varying thermal gradient and uniform heating.

- New beam-column element classes DispBeamColumn2dThermal (2D element) and DispBeamColumn3dThermal (3D element) were created by modifying the existing corresponding 2D and 3D beam element in OpenSees. The performance of the new 2D beam element was tested by benchmark problems of a single beam subjected to a range of thermal loading under a variety of boundary conditions. The results show sufficiently satisfactory performance.

- New geometrically nonlinear shell element ShellMITC4GNThermal was developed based on the existing linear shell element in OpenSees. The modifications were based on the Total Lagrangian formulation. The performance of this shell element was verified by solving appropriate benchmark problems and was also validated by analysing large models consisting of beam and shell elements, such as the Cardington tests.

- Temperature dependent material classes were developed by modifying the existing material classes in OpenSees. The material classes Steel01Thermal and Concrete02Thermal were used to model the uniaxial steel and concrete at elevated temperature respectively. Pinching4Thermal class was used to model the damaged steel and concrete under cyclic load. The material class DruckerPragerThermal was used to model concrete in the composite slab. The temperature dependent properties for all these materials were based on the Eurocodes.

- Benchmark problems involving a single beam with different boundary conditions at elevated temperature was analysed. These analyses confirm that the effect of boundary conditions is crucial in determining the response of structural members subjected to thermal actions. If the beam with one end restrained subjected to thermal gradient, it will expands as temperature increases until certain high temperature and then begins to move back. The expansion of the beam is driven by the thermal elongation effect considering the increasing uniform temperature of the beam and moving back is because of increasing thermal gradient which “pulls” the free end back as well as losing resisting capability due to the material degradation at elevated temperature. It is obvious that the translational restraint dominates the

structural behaviour over midrange temperatures resulting in high axial forces and mid-span moment in end-restrained beams.

- For a fixed ended beam subjected to a thermal gradient, an unstable behaviour of the beam can be seen as the beam bends downwards initially to its peak deflection (because of thermal bowing) and then snaps through to the opposite direction. This "thermal snap through" is driven by the additional hogging moment that occurs in the beam resulting from the centre of stiffness of the beam section moving upwards (due to the greater material degradation in the bottom) as the temperature increases, which creates an "eccentricity" for the axial forces and produces a moment opposite to the once caused by thermal bowing.

- The Cardington restrained beam test and corner test were modelled in OpenSees and the results agreed well with the test data. The modelling of the horizontal movement of the columns in these two tests was also presented and reasonably good agreement was obtained for the corner test. The edge columns experienced larger movements than the corner columns because of the higher temperatures in the unprotected internal beams than the protected edge beams.

- A 3D model was set up in OpenSees to simulate the structural behaviour of an RC frame subjected to fire following simulated seismic damage. During the 60 minute fire load step (after the cyclic loading step), the deflection of the slab keeps increasing until about 50 minutes and thence remains unchanged. The two nodes of the columns at the roof level moved right (direction opposite to the position after damage from cyclic loading) at the beginning of the heating due to the fact that initial heating tends to stiffen the frame rather like pre-stressing. The frame starts to move left for higher temperatures.

9.3 Further work

This thesis presents part of the work underway in Edinburgh to add a "structures in fire" modelling capability in OpenSees. Another member in the team is working on the development of OpenSees for heat transfer analysis. Some recommendations for the further work can be made as follows:

- Currently the temperature dependent properties of steel and concrete materials in OpenSees are according to Eurocode 2 and Eurocode 3 respectively. More extensive and flexible material classes should be added with properties at elevated temperature based on advanced theories as well as experimental data to include effects such as LITS (Load Induced Thermal Strain) more explicitly. This effect is considered to model the more obvious creep phenomenon in the concrete material at elevated temperature than ambient temperature.

- The interface between heat transfer analysis and mechanical analysis should be established to enable a fully coupled thermomechanical analysis capability in OpenSees. This means the temperature distribution in the structure resulting from the heat transfer analysis can be automatically transferred to the mechanical analysis by an appropriate mapping between the finite element meshes used in the heat transfer and mechanical analyses. The models for these two fields do not necessarily have the same meshing regime which is helpful for simplifying the model and saving computation cost.

- The newly created geometrically nonlinear shell element is formed by combining the Mindlin bending element and membrane element. Its performance is only verified on some examples of plate like structures and additional curved shell structures like tanks should be modelled in OpenSees.

- No structural response data could be recorded from the damaged RC frame in fir test to compare with OpenSees. It is expected the test will be carried out in the near future and the test results could then be used to verify the performance of the newly developed OpenSees capabilities more rigorously..

Reference

A

Abu-Gazaleh B.N. (1965). Analysis of plate-type prismatic structures Ph.D. Dissertation, Dept. of Civil Engineering, University of California, Berkeley, CA.

Adini A. and Clough R.W. (1960). Analysis of plate bending by the finite element method, NSF report for Grant G-7337, Dept. of Civil Engineering, University of California, Berkeley,

Adini A. (1961). Analysis of shell structures by the finite element method, Ph.D. Dissertation, Department of Civil Engineering, University of California, Berkeley.

Ahmad S., Irons B.M. and Zienkiewicz O.C. (1968). Curved thick shell and membrane elements with particular reference to axisymmetric problems. Proceedings of 2nd Conference on Matrix Methods in Structural Mechanic.7 Wright-Patterson Air Force Base, Ohio.

Ahmad S., Irons B.M. and Zienkiewicz O.C. (1970). Analysis of thick and thin shell structures by curved finite elements. International Journal for Numerical Methods in Engineering, 2: 419-451.

Allman D.J. (1984). A compatible triangular element including vertex rotations for plane elasticity analysis. Computers & Structures, 19: 1-8.

Allman D.J. (1988a). A quadrilateral finite element including vertex rotations for plane elasticity analysis. International Journal for Numerical Methods in Engineering, 26:717-730.

Allman D.J. (1988b). Evaluation of the constant strain triangle with drilling rotations. International Journal for Numerical Methods in Engineering, 26:2645-2655.

Aminpour M.A. (1992). An assumed-stress hybrid 4-node shell element with drilling degrees of freedom. International Journal for Numerical Methods in Engineering, 33: 19-38.

Anderheggen E. (1969). Finite element plate bending equilibrium analysis. Journal of Engineering Mechanics, 841-857.

Anderheggen E. (1970). A conforming triangular finite element plate bending solution. International Journal for Numerical Methods in Engineering, 2 (2): 259-264.

Andelfinger U. and Ramm E. (1993). EAS-elements for two-dimensional, threedimensional, plate and shell structures and their equivalence to HR-elements. International Journal for Numerical Methods in Engineering, 36: 1311-1337.

Argyris J.H. (1965). Triangular elements with linearly varying strain for the matrix displacement method. Journal of Royal Aeronautical, 69: 711-713.

Argyris J.H., Fried I. and Scharpf D.W. (1968). The TUBA family of plate elements for the matrix displacement method. Journal of Royal Aeronautical, 72: 701-709.

Averill R.C. and Reddy J.N. (1990). Behaviour of plate elements based on the first-order shear deformation theory. Engineering Computations, 7: 57-74.

B

Baldwin J.T., Raxxaque A. and Irons B.M. (1973). Shape function subroutines for an isoparametric thin plate element. International Journal for Numerical Methods in Engineering 7 (4): 431- 440.

Bailey C.G. (1995). Simulation of the structural behavior of steel-framed buildings in fire, PhD thesis, Department of Civil and Structural Engineering, University of Sheffield, UK.

Bailey C.G., Burgess I.W., and Plank R.J. (1996). Computer simulation of a full-scale structural fire test. The Structural Engineer, 74(6), 93-100.

Bailey C.G. and Moore D.B. (1999). The Behaviour of Full-Scale Steel Framed Buildings Subject to Compartment Fires. The Structural Engineer, 77(8): 15-21.

Bailey C.G. and Moore D.B. (2000). The structural behaviour of steel frames with composite floors slabs subject to fire—part 1 theory. The Structural Engineer, 78(11): 19-27.

Bailey C.G. (2001a). Steel structures supporting composite floor slabs: design for fire, BRE digest 462. Watford: The Building Research Establishment.

Bailey C.G. (2001b). Membrane action of unrestrained lightly reinforced concrete slabs at large displacements. Engineering Structures, 23: 470-483.

Bailey C.G. (2003). Efficient arrangement of reinforcement for membrane behaviour of composite floor slabs in fire conditions. Journal of Constructional Steel Research, 59:931-949.

Bailey C.G. (2004). Membrane action of slab/beam composite floor systems in fire. Engineering Structrues, 26: 1691-1703.

Bailey C.G. and Toh WS. (2007). Small-scale concrete slab tests at ambient and elevated temperatures. Engineering Structrues, 29: 277-291.

Bathe K.J. and Bolourchi S. (1980). A geometric and material nonlinear plate and shell element. Computers & Structures, 11:23-48.

Bathe K.J. and Dvorkin E.N. (1985). A four-node plate bending element based on Mindlin-Reissner plate theory and a mixed interpolation. International Journal for Numerical Methods in Engineering, 21, 367-383.

Bathe K.J. and Brezzi F. (1985). On the Convergence of a Four-Node Plate Bending Element Based on Mindlin/Reissner Plate Theory and a Mixed Interpolation. Proceedings of the Conference on Mathematics of Finite Elements and Applications V, Academic Press, 491-503.

Bathe K.J. and Dvorkin E. (1986). A formulation of general shell elements- the use of mixed interpolation of tensorial components. International Journal for Numerical Methods in Engineering, 22: 697-722.

Bathe K.J. and Brezzi F. (1987). A Simplified analysis of Two-Plate Bending Elements - the MITC4 and MITC9 Elements. Proceedings of the International Conference on Numerical Methods in Engineering, University College of Swansea, Wales.

Bathe K.J., Cho S.W., Bucalem M.L. and Brezzi F. (1989a). On Our MITC Plate Bending/ Shell Elements. Analytical and Computational Models of Shells, ASME Special Publication, 261-278.

Bathe, K.J., Brezzi F. and Cho S.W. (1989b). The MITC7 and MITC9 Plate Bending Elements. Computers & Structures, 32(3/4): 797-814.

Bathe K.J., Bucalem M.L. and Brezzi F. (1990). Displacement and Stress Convergence of Our MITC Plate Bending Elements. Engineering Computations, 7(4): 291-302.

Bathe K.J. (1996). Finite Element Procedure. Prentice-Hall, Upper Saddle River, New Jersey.

Bathe K.J., Iosilevich A. and Chapelle D. (2000). An evaluation of the MITC shell elements. Computers & Structures, 75:1-30.

Bathe K.J., Lee P.S. and Hiller J.F. (2003). Towards improving the MITC9 shell element. Computers & Structures, 81: 477-489.

Batoz J.L., Bathe K.J. and Ho L.W. (1980). A study of three node triangular plate bending elements. International Journal for Numerical Methods in Engineering, 15: 1771-1812.

Batoz J.L. (1982). An explicit formulation for an efficient triangular plate bending element. International Journal for Numerical Methods in Engineering, 18: 1077-1089.

Batoz J.L. and Tahar M.B. (1982). Evaluation of a new quadrilateral thin plate bending element. International Journal for Numerical Methods in Engineering, 18: 1655-1677.

Battini J.M. (2002). Co-rotational beam elements in instability problems. Report of Royal institute of Technology, Stockholm, Sweden.

Bazeley G.P., Cheung Y.K., Irons B.M. and Zienkiewicz O.C. (1965). Triangular elements in plate bending conforming and non-conforming solutions. Proceedings of 1st Conference on Matrix Methods in Structural Mechanics, Ohio, 547-576.

Bell K. (1969). A refined triangular plate bending finite element. International Journal for Numerical Methods in Engineering, 1(1): 101-122.

Belytschko T. and Tsay C.S. (1983). A stabilization procedure for the quadrilateral plate element with one-point quadrature. International Journal for Numerical Methods in Engineering, 19: 405-419.

Belytschko T., Ong J.S.J. and Liu W.K. (1984a). A consistent control of spurious singles modes in the 9-node Lagrange element for the Laplace and Mindhn plate equations. Computer Methods in Applied Mechanics and Engineering, 44: 269-295.

Belytschko T., Lin J.I. and Tsay C.S. (1984b). Explicit algorithms for the nonlinear dynamics of shells. Computer Methods in Applied Mechanics and Engineering, 42: 225-251.

Belytschko T., Liu W.K., Ong S.J. and Lam D. (1985). Implementation and Application of a 9-node Lagrange Shell Element with Spurious Mode Control. Computers & Structures, 20(3): 121-128.

Belytschko T., Wong B.L. and Stolarski H. (1989). Assumed strain stabilization procedure for the 9-node Lagrange shell element. International Journal for Numerical Methods in Engineering, 28: 385-414.

Belytschko T. and Leviathan I. (1994). Physical stabilization of the 4-node shell with one-point quadrature. Computer Methods in Applied Mechanics and Engineering, 113: 321-350.

Belytschko T., Liu W.K. and Moran B. (2000). Nonlinear Finite Elements for Continua and Structures. John Wiley and Sons, Ltd., New York.

Bercovier M. (1978). Perturbation of a Mixed Variational Problems, Applications to Mixed Finite Element Methods. RAIRO Numerical Analysis, 12: 211-236.

Bergan P.G. (1967). Plane stress analysis using the finite element method. Triangular element with 6 parameters at each node, Division of Structural Mechanics, The Norwegian Institute of Technology, Trondheim, Norway.

Bergan P.G. and Hanseen L. (1975). A new approach for deriving good element stiffness matrices. The Mathematics of Finite Elements and Applications, Academic Press, London, 483-497.

Bergan P.G. (1980). Finite elements based on energy orthogonal functions. International Journal for Numerical Methods in Engineering, 15: 1141-1555.

Bergan P.G. and Nygard M.K. (1984). Finite elements with increased freedom in choosing shape functions. International Journal for Numerical Methods in Engineering 20: 643-664.

Bergan P.G. and Felippa C.A. (1985). A triangular membrane element with rotational degrees of freedom. Computer Methods in Applied Mechanics and Engineering, 50: 25-69.

Bergan P.G. and Nygard M.K. (1986). Nonlinear shell analysis using free formulation finite elements, Proc. Europe-US Symposium on Finite Element Methods for Nonlinear Problems, Springer-Verlag, Berlin.

Bergan P.G. and Felippa C.A. (1986). Efficient implementation of a triangular membrane element with drilling freedoms. Finite Element Methods for Plate and Shell Structures, Pineridge Press, Swansea, 1: 128-152.

Bischoff M. and Ramm E. (1997). Shear Deformable shell elements for large strains and rotations. International Journal for Numerical Methods in Engineering, 40: 4427-4449.

Bischoff M. and Bletzinger K.U. (2001). Stabilized DSG plate and shell elements, Trends in Computational Structural Mechanics, CIMNE, Barcelona, 253-263.

Bischoff M., Koschnick F. and Bletzinger K.U. (2003). Stabilized DSG elements—a new paradigm in finite element technology. Proceedings of 4th European LS-DYNA Users Conference, Ulm, Germany.

Bletzinger K.U., Bischoff M. and Ramm E. (2000). A unified approach for shear-locking free triangular and rectangular shell finite elements. Computers & Structures, 75: 321-334.

Bogner F.K., Fox R.L. and Schmidt L.A. (1965). The generation of interelement compatible stiffness and mass matrices by the use of interpolation formulas, Proceedings of Conference on Matrix Methods in Structural Mechanics, WPAFB, Ohio, in AFFDL TR 66-80, 397-444.

Both C. and Haar P.W. (1994). The mechanical behaviour of three two-span fire-exposed composite steel/concrete slabs with Prins PSV73 steel decking. Test report, Delft University of Technology report 6.94.24/TNO-Building and Construction Research report 94-CVB-R1383.

Bravery P. (1993). Cardington large building test facility, construction details of the first building. Internal Report, British Steel plc.

Brezzi F. Bathe K.J. and Fortin M. (1989). Mixed-Interpolated Elements for Reissner-Mindlin Plates. International Journal for Numerical Methods in Engineering, 28: 1787-1801.

Brezzi F., Fortin M. and Stenberg R. (1991). Error Analysis of Mixed Interpolated Elements for Reissner -Mindlin Plates", Mathematical Models and Methods in Applied Sciences, 1:125-151.

British Steel (BS) plc. (1999). The behaviour of multi-storey steel framed buildings in fire. Report of British Steel, Swinden Technology Center, UK,

British Standard Institution. (1987). BS 476: Parts 20-23: 1987 Fire tests on building materials and structures.

British Standard Institution. (1990). BS 5950 Structural use of steelwork in buildings: Part 8: Code of practice for fire resistant design, London.

Brotchie J.F. and Holley M.J. (1971). Membrane action in slabs cracking, deflection and ultimate load of concrete slab systems. Publication SP-30 In: Detroit, MI: American Concrete Institute, 345-377.

Bucalem M.L. and Bathe K.J. (1993). Higher-order MITC general shell elements. International Journal for Numerical Methods in Engineering, 36: 3729-54.

Bucalem M.L. and Bathe K.J. (1997). Finite element analysis of shell structures. Archives of Computational Methods in Engineering, 4(1): 3-61.

Bucalem M.L. and Nóbrega S.H.S. (2000). A mixed formulation for general triangular isoparametric shell elements based on the degenerated solid approach. Computers & Structures, 78: 35-44.

Building Research Establishment (BRE). (2005). The integrity of compartmentation in building during a fire. BRE Report number 213140(1), Building Research Establishment Ltd.

C

Cameron N. (2003). The behaviour and design of composite floor systems in fire. Ph.D. thesis. University of Edinburgh.

Cardoso R.P.R., Yoon, J.W. and Valente R.A.F. (2006). A new approach to reduce membrane and transverse shear locking for one-point quadrature shell elements: linear formulation. International Journal for Numerical Methods in Engineering, 66(2):214-249.

Cardoso R.P.R., Yoon J.W. and Valente R.A.F. (2007). Enhanced one-point quadrature shell element for nonlinear applications. International Journal for Numerical Methods in Engineering, 69(3): 627-663.

Carpenter N., Stolarski H. and Belytschko T. (1985). A flat triangular shell element with improved membrane interpolation. Communications in Applied Numerical Methods, 1: 161-168.

Carr A.J. (1967). Refined finite element analysis of thin shell structures including dynamic loadings, Ph. D. Dissertation, SESM Report 67-9, Dept. of Civil Engineering, University of California at Berkeley.

Chen J., Wu C. and You Y. (2001). A stabilized conforming nodal integration for Galerkin meshfree method. International Journal for Numerical Methods in Engineering, 50: 435-466.

Chen W.F. and Saleeb A.F. (1994). Constitutive Equations for Engineering Materials Volume I: Elasticity and Modeling. Elsevier Science B.V., Amsterdam.

Chinosi C. (1995). Shell elements as a coupling of plate and drill elements. Computers & Structures, 57(5): 893-902.

Chinosi C., Comodi M.I. and Sacchib G. (1997). A new finite element with ‘drilling’ D.O.F.. Computer Methods in Applied Mechanics and Engineering. 143: 1-11.

Choi C.K. and Lee T.Y. (2003). Efficient remedy for membrane locking of 4-node flat shell elements by non-conforming modes. Computer Methods in Applied Mechanics and Engineering, 192: 1961-1971.

Clough R.W. and Tocher J.L. (1965). Finite element stiffness matrices for analysis of plate bending. Proceedings of 1st Conference on Matrix Methods in Structural Mechanics, WPAFB, Ohio, 515–545.

Cook R.D. (1972). More on reduced integration and isoparametric element. International Journal for Numerical Methods in Engineering, 4:141-148.

Cook R.D. and Zhao H.F. (1982). Control of spurious modes in the nine-node quadrilateral element. International Journal for Numerical Methods in Engineering, 18: 1576-1580.

Cook R.D. (1986). On the Allman triangle and a related quadrilateral element. Computers & Structures, 2: 1065-1067.

Cook R.D. (1987). A plane hybrid element with rotational d.o.f. and adjustable stiffness. International Journal for Numerical Methods in Engineering, 24:1499-1508.

Cook R.D. (1994). Four-node ’flat’ shell element: drilling degrees of freedom, membrane-bending coupling, warped geometry and behaviour. Computers & Structures, 50: 549-555.

Cook, R.D., Malkus, D.S., Plesha, M. E., and Witt, R. J.(2002). Concepts and Applications of Finite Element Analysis,4th edition, John Wiley and Sons publishers.

Cowper G.R., Kosko E., Lindberg G.M. and Olson M.D. (1968). Formulation of a new triangular plate bending element. Trans. Canadian Aeronaut. Space Inst. 1(2): 86-90.

Crisfield M.A. (1991). Non-Linear Finite Element Analysis of Solids and Structures, Volume 1: Essentials. Wiley, Chichester.

Cui X.Y., Liu G.R., Li G.Y., Zhao X., Nguyen T.T. and Sun G. Y. (2008). A smoothed finite element method (sfem) for linear and geometrically nonlinear analysis of plates and shells. CMES: Computer Modelling in Engineering & Sciences, 28(2):109-125.

D

Doherty W.P., Wilson E.L. and Taylor R.L. (1969). Stress analysis of axisymmetric solids utilizing higher order quadrilateral finite elements. SESM Report No. 69-3, Dept. Civil Eng., Univ. California, Berkeley.

Drucker D.C. and Prager W. (1952). Soil mechanics and plastic analysis for limit design. Quarterly of Applied Mathematics, 10(2): 157-165.

Dvorkin E.N. and Bathe K.J. (1984). A continuum mechanics based four-node shell element for general nonlinear analysis. Engineering with Computers, 1: 77-88.

Dungar R. and Severn R.T. (1969). Triangular finite elements of variable thickness and their application to plate and shell problems. Journal of Strain Analysis, 4: l0-21.

E

Elias Z.M. (1968). Duality in finite element methods. Journal of Engineering Mechanics, 94: 931-946.

Eckstein A.and Basar Y. (2000). Ductile damage analysis of elasto-plastic shells at large inelastic strains. International Journal for Numerical Methods in Engineering, 47: 1663-1687.

Elghazouli A.Y. and Izzuddin B.A. (2000). Response of idealised composite beam-slab systems under fire conditions. Journal of Constructional Steel Research, 56: 199-224.

Elghazouli A.Y., Izzuddin B.A., and Richardson A.J. (2000). Numerical modelling of the structural fire behaviour of composite buildings. Fire Safety Journal, 35: 279-297.

Elghazouli A.Y. and Izzuddin B.A. (2001). Analytical assessment of the structural performance of composite floors subject to compartment fires. Fire Safety Journal, 36: 769-793.

Elghazouli A.Y. and Izzuddin B.A. (2004). Realistic modelling of composite and reinforced concrete floor slabs under extreme loading. II: Verification and application. Journal of Structural Engineering, 130(12): 1985-1996.

Eurocode 2 (1992). Design of concrete structures: Part 1.2: General rules, Structural fire design, ENV 1992-1 -2, Brussels, European Committee for Standardisation.

Eurocode 3 (1993). Design of steel structures: Part 1.2: General rules, Structural fire design, ENV 1993-1 -2, Brussels, European Committee for Standardisation.

Eurocode 4 (1994). Design of composite steel and concrete structures: Part 1.2 General rules, Structural fire design, ENV 1994-1 - 2, Brussels, European Committee for Standardisation.

F

Felippa C.A. (1966). Refined finite element analysis of linear and nonlinear two-dimensional structures, Ph. D. Dissertation, SESM Report 66-26, Dept. of Civil Engineering, University of California at Berkeley.

Felippa C.A. and Bergan P.G. (1987). A triangular plate bending element based on energy orthogonal free formulation. Computer Methods in Applied Mechanics and Engineering, 61: 126-60.

Felippa C.A. and Militello C. (1992). Membrane triangles with corner drilling freedoms II. The ANDES element, Finite Elements in Analysis and Design, 12: 189-201.

Felippa C.A. and Alexander S. (1992). Membrane triangles with corner drilling freedoms III. Implementation and performance evaluation, Finite Elements in Analysis and Design, 12: 203-239.

Felippa C.A. (2003). A Study of Optimal Membrane Triangles with Drilling Freedoms , Department of Aerospace Engineering Sciences and Centre for Aerospace structures university of Colorado Boulder, Report CU-CAS-03-02.

Felippa C.A. and Haugen B. (2005). A unified formulation of small-strain corotational finite elements: I. Theory. Computer Methods in Applied Mechanics and Engineering, 194: 2285-2335.

Filippou F.C., Popov E.P. and Bertero V.V. (1983). Effects of bond deterioration on hysteretic behaviour. Report of University of California, Berkeley.

Foster S. (2006). Tensile membrane action of reinforced concrete slabs at ambient and elevated temperatures. Ph.D. thesis. Department of Civil and Structural Engineering, University of Sheffield.

Fraeijs B. Veubeke de. (1665). Bending and stretching of plates-special models for upper and lower bounds. Proceedings of 1st Conference on Matrix Methods in Structural Mechanics, Ohio, 863-886.

Fraeijs B. Veubeke de and Zienkiewicz O.C. (1967). Strain energy bounds in finite element analysis by slab analogy. Journal of Strain Analysis, 2 (4): 265-271.

Franssen J.M., Kodur V.K.R. and Mason J. (2000). User's manual for SAFIR2001, A computer program for analysis of structures submitted to the fire, University of Liege, Belgium.

Franssen J.M. (2003). SAFIR: a thermal/structural program modelling structures under fire. Proceedings of the North American Steel Construction Conference, April, A.I.S.C. Inc., Baltimore.

Fried I. (1973). Shear in C0 and C1 bending finite elements. International Journal of Solids and Structures, 9 (4): 449-460.

Fried I. and Yang S.K. (1973). Triangular nine-degrees-of-freedom, C0 plate bending element of quadratic accuracy. Quarterly of Applied Mathematics, 31: 303-312.

Fried I. (1974). Residual energy balancing technique in the generation of plate bending finite element, Computers & Structures, 4: 771-778.

G

Gallagher R.H. (1969). Analysis of plate and shell structures. Proceedings of Symp. on Application of Finite Element Methods in Civil Engineering, School of Engineering, Vanderbilt University, Nashville, Tennessee, 155-205.

Gallagher R.H. (1975). Finite Element Analysis: Fumiamentals. Prentice Hall.

Geyer S. and Groenwold A.A. (2002). Two hybrid stress membrane finite element families with drilling rotations, International Journal for Numerical Methods in Engineering, 53: 583–601.

Geyer S. and Groenwold A.A. (2003). On reduced integration and locking of flat shell finite elements with drilling rotations, Communications in Applied Numerical Methods, 19: 85-97.

Gillie M. (2000). The behaviour of steel-framed composite structures in fire conditions. PhD thesis, The University of Edinburgh.

Gillie M., Usmani A.S. and Rotter J.M. (2001a). Modelling of heated composite floor slabs with reference to the Cardington experiments. Fire Safety Journal, 36:745-767.

Gillie M., Usmani A.S. and Rotter J.M. (2001b). Structural Analysis of the First Cardington Test. Journal of Constructional Steel Research, 57:581-601,

Gillie, M., Usmani A.S. and Rotter J.M. (2002). A Structural Analysis of the Cardington British Steel Corner Test. Journal of Constructional Steel Research, 58: 427-443.

Greene B.E., Jones R.E., McLay R.W. and Strome D.R. (1969). Generalized variational principles in the finite element method. The American Institute of Aeronautics and Astronautics (AIAA), 7 (7): 1254-1260.

Groenwold A.A. and Stander N. (1995). An efficient 4-node 24 DOF thick shell finite element with 5-point quadrature. Engineering Computations, 12: 723–747.

Groenwold A.A., Xiao Q.Z. and Theron N.J. (2002). Representing traction free boundaries using drilling degrees of freedom. The Sixth International Conference on Computational Structures Technology, Prague, Czech Republic.

Groenwold A.A., Xiao Q.Z. and Theron N.J. (2004). Accurate solution of traction free boundaries using hybrid stress membrane elements with drilling degrees of freedom, Comput. Struct. 82: 2071–2081.

H

Hamerlinck A.F., Twilt L., Brekelmans J.W.P.M. (1990). The mechanical behaviour of fire-exposed composite steel/concrete slabs under negative bending. Test report, Research Report BI-90-118, Eindhoven University of Technology/TNOBuilding and Construction Research.

Harvey J.W. and Kelsey S. (1971). Triangular plate bending elements with enforced compatibility. The American Institute of Aeronautics and Astronautics (AIAA), 9(6): 1023-1026.

Hayes B. (1968). Allowing for membrane action in the plastic analysis of rectangular reinforced concrete slabs. Magazine of Concrete Research, 20(65): 205-212.

Herrmann L.R. (1965). A bending analysis for plates. Proceedings of 1st Conference on Matrix Method in Structural Mechanics, Ohio, 577-602.

Hinton E. and Bicanic N. (1979). A comparison of Lagrangian and serendipity Mindlin plate elements for free vibration analysis. ComputERS & Structures, 10: 483-493.

Hinton E. and Huang H.C. (1986). A family of quadrilateral Mindlin plate elements with substitute shear strain fields. Computers & Structures, 23(3): 409-431.

Holand I. and Bergan P.G. (1968). Higher order finite element for plane stress. Journal of the Engineering Mechanics, 94: 698-702.

Hrabok M.M. and Hrudey T.M. (1984). A review and catalogue of plate bending finite elements. Computers & Structures, 19: 479-495.

Hrabok M.M. (1981). Analysis of stiffened plates by the hybrid stress finite element method. PhD Thesis, Department of Civil Engineering, University of Alberta, Edmonton, Canada.

Huang H.C. and Hinton E. (1984a). A nine node Lagrangian Mindlin plate element with enhanced shear interpolation. Engineering Computations, 1: 369-379.

Huang H.C. and Hinton E. (1984b). A nine node degenerated shell element with enhanced shear interpolation. Research Report, Dept. of Civil Engng, Swansea, U.K.

Huang Z., Burgess I.W. and Plank R.J. (1999a). Nonlinear analysis of reinforced concrete slabs subjected to fire. ACI Structural Journal, 96(1): 127-35.

Huang Z., Burgess I.W. and Plank R.J. (1999b). Influence of shear connectors on the behaviour of composite steel-framed buildings in fire. Journal of Constructional Steel Research, 51(3): 219-237.

Huang Z., Burgess I.W. and Plank R.J. (2000a). Effective stiffness modelling of composite concrete slabs in fire. Engineering Structures, 22(9): 1133-1144.

Huang Z., Burgess I.W. and Plank R.J. (2000b). Three-dimensional analysis of composite steel-framed buildings in fire. Journal of Structural Engineering, 126(3): 389-397.

Huang Z., Burgess I.W. and Plank R.J. (2000c). Non-linear Modelling of Three Full Scale Structural Fire Tests. In: Structures in Fire: Proceedings of the First International Workshop, Copenhagen.

Huang Z., Burgess I.W. and Plank R.J. (2003a). Modelling membrane action of concrete slabs in composite buildings in fire — Part I: Theoretical development. Journal of Structural Engineering, 129(8): 1093-1102.

Huang Z., Burgess I.W. and Plank R.J. (2003b). Modelling membrane action of concrete slabs in composite buildings in fire — Part II: Validations. Journal of Structural Engineering, 129(8): 1103–1112.

Huang Z., Burgess I.W. and Plank R.J. (2004). Fire resistance of composite floors subject to compartment fires. Journal of Constructional Steel Research, 60:339-360.

Huang Z. (2010). The behaviour of reinforced concrete slabs in fire. Fire Safety Journal, 45: 271-282.

Hughes T.J.R., Taylor R.L. and Kanoknukulchai W. (1977). A simple and efficient element for plate bending. International Journal for Numerical Methods in Engineering, 11: 1529-l543.

Hughes T.J.R. and Cohen M. (1978). The “Heterosis” finite element for plate bending. Computers & Structures, 9: 445-450.

Hughes T.J.R., Cohen M. and Haroun M. (1978). Reduced and selective integration techniques in the finite element analysis of plates. Nuclear Engineering and Design, 46:203-222.

Hughes T.J.R. and Tezduyar T.E. (1981). Finite elements based upon Mindlin plate theory with particular reference to the four-node bilinear isoparametric element, Journal of Applied Mechanics, 48: 587-596.

Hughes T.J.R. and Taylor R.L. (1981). The linear triangular bending element, Fourth MAFELAP Conf. Brunel University, Uxbridge.

Hughes T.J.R., Franca L.P. and Hulbert G.M. (1989). A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. Computer Methods in Applied Mechanics and Engineering, 73: 173-189.

Hughes T.J.R. and Brezzi F. (1989). On drilling degrees of freedom. Computer Methods in Applied Mechanics and Engineering, 72: 105-121.

Hughes T.J.R., Brezzi F., Masud A. and Harari I. (1989). Finite element with drilling degrees of freedom: Theory and numerical evaluations. In Proceedings of the fifth international symposium on numerical methods in engineering, Computational mechanics publications, Ashurst, U.K., 3-17.

Hughes T.J.R., Masud A. and Harari I. (1995a). Numerical assessment of some membrane elements with drilling degree of freedom. Computers & structures, 55: 297-314.

Hughes T.J.R. Masud A. and Harari I. (1995b). Dynamic analysis and drilling degrees of freedom. International Journal for Numerical Methods in Engineering, 38: 3193-3210.

Hughes T. J. R. (2000). The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Dover, New York.

I

Ibrahimbegovic A. (1990). A novel membrane finite element with an enhanced displacement interpolation. Finite Elements in Analysis and Design, 7: 167-179.

Ibrahimbegovic A., Taylor R.L. and Wilson E.L. (1990). A robust quadrilateral membrane finite element with drilling degrees of freedom. International Journal for Numerical Methods in Engineering, 30: 445-457.

Ibrahimbegovic A. and Wilson E.L. (1991). A unified formulation for triangular and quadrilateral flat shell finite elements with six nodal degrees of freedom. Communications in Applied Numerical Methods, 7: 1-9.

Ibrahimbegovic A. and Frey F. (1992). Membrane quadrilateral finite elements with rotational degrees of freedom. Engineering Fracture Mechanics, 43(1): 13-24.

Ibrahimbegovic A. (1993). Mixed finite element with drilling rotations for plane problems in finite elasticity. Computer Methods in Applied Mechanics and Engineering, 107: 225-238.

Ibrahimbegovic A. (1994). Stress resultant geometrically nonlinear shell theory with drilling rotations- Part I. A consistent formulation, Computer Methods in Applied Mechanics and Engineering, 118: 265-284.

Ibrahimbegovic A. and Frey F. (1994). Stress resultant geometrically nonlinear shell theory with drilling rotations-Part II. Computational aspects, Computer Methods in Applied Mechanics and Engineering, 118: 285-308.

Irons B.M. and Draper K. (1965). Inadequacy of nodal connections in a stiffness solution for plate bending. The American Institute of Aeronautics and Astronautics (AIAA), 3: 965-966.

Irons B.M. (1969). A conforming quartic triangular element for plate bending. International Journal for Numerical Methods in Engineering, 1(1): 29-45.

Irons B.M. (1976). The semi-loof element. Finite elements for thin shell and curved members. Wiley, New York, 197-222,

Irons B.M. and Razzaque A. (1972). Shape function formulations for elements other than displacement models. the Symp. on Variational Meth. in Engng, University of Southampton, 59-71.

Irons B.M. and Ahmad S. (1980). Techniques of Finte Elements. Ellis Horwood, Chichester.

Izzuddin B.A. (1991). Nonlinear dynamic analysis of framed structures. PhD Thesis, Imperial College, University of London.

Izzuddin B.A. (2002). Advanced Large Displacement Analysis of Composite Floor Slabs, Proc. 15th ASCE Engineering Mechanics Conference, New York.

Izzuddin B.A., Tao X.Y. and Elghazouli A.Y. (2004). Realistic modelling of composite and R/C floor slabs under extreme loading-Part I: Analytical method. Journal of Structural Engineering, 130(12): 1972-1984.

Izzuddin B.A. (2003). Integration of beam–column and shell elements in large displacement structural analysis. In: IASS-APCS.

J

Jaamei S., Frey F. and Jetteur Ph. (1987). Nonlinear thin shell finite element with six degrees of freedom per node, Seventh International Symposium on Computing Methods in Applied Sciences and Engineering, Versailles.

Jang J. and Pinsky P.M. (1987). An Assumed Covariant Strain Based 9-Node Shell Element. International Journal for Numerical Methods in Engineering, 24:2389-2411.

Jetteur Ph. (1986). A shallow shell element with in-plane rotational degrees of freedom, IREM Internal Rept. 86/3, D6partement de G6nie Civil, Institut de Statique et Structures, Ecole Polytechnique F6d6rale de Lausanne, Lausanne.

Jetteur Ph. (1987a). Improvement and large rotations of the JET shell element in nonlinear analysis, IREM Internal Rept. 87/4, Ecole Polytechnique Federale de Lausanne, Lausanne.

Jetteur Ph. (1987b). Improvement of the quadrilateral JET shell element for a particular class of shell problems. Technical Report IREM 87/1, Ecole Polytechnique Federale de Lausanne. Lausanne.

Jones R.E. (1964). A generalization of the direct-stiffness method of structural analysis. The American Institute of Aeronautics and Astronautics (AIAA), 2(5): 821-862.

K

Kemp K.O. (1967). Yield of a square reinforced concrete slab on simple supports allowing for membrane forces, Structural Engineer, 45(7): 235-240.

Kent D.C. and Park R. (1971). Flexural Members with Confined Concrete. Journal of the Structural Division, 97(7):1969-1990.

Kikuchi F. and Ando Y. (1972). Some finite element solutions for plate bending problems by simplified hybrid displacement method. Nuclear Engineering and Design, 23: 155-178.

Kim D.N. and Bathe K.J. (2009). A triangular six-node shell element. Computers & Structures, 87:1451-60.

Kirby B.R. (1997). British steel technical European fire test programme design, construction and results. Fire, static and dynamic tests of building structures. London.

Koschnick F., Bischoff M., Camprubi N. and Bletzinger K. (2005). The discrete strain gap method and membrane locking. Computer Methods in Applied Mechanics and Engineering, 194: 2444-2463.

L

Lamont S. (2001). The behaviour of multi-storey composite steel framed structures in response to compartment fires. PhD thesis, The University of Edinburgh.

Lamont S., Usmani A.S. and Gillie M. (2004). Behaviour of a small composite steel frame structure in a “long-cool” and a “short-hot” fire. Fire Safety Journal, 39:327-357.

Lamont S. and Usmani A.S. (2003). Possible panel instability in composite deck floor systems under fire. Journal of Constructional Steel Research, 59: 1397-1433.

Lee P.S. and Bathe K.J. (2004). Development of MITC isotropic triangular shell finite elements. Computers & Structures, 82: 945-962.

Lee P.S., Noh H.C. and Bathe K.J. (2007). Insight into 3-node triangular shell finite elements: the effects of element isotropy and mesh patterns. Computers & Structures, 85:404-418.

Lee P.S. and Bathe K.J. (2010). The quadratic MITC plate and MITC shell elements in plate bending. Advances in Engineering Software, 41:712-728.

Lee S.C. and Yoo C.H. (1988). A novel shell element including in-plane torque effect. Computers & Structures, 28: 505-522.

Lee S.W. and Pian T.H.H. (1978). Improvement of plate and shell finite elements by mixed formulations. AIAA Journal, 16:29–34.

Li G.Q., Guo S.X. and Zhou H.S. (2007). Modelling of membrane action in floor slabs subjected to fire. Engineering Structures, 29: 880-887.

Lim L.C.S. (2003). Membrane action in fire exposed concrete floor systems. PhD thesis, University of Canterbury, New Zealand.

Lim L.C.S., Buchanan A., Moss P. and Franssen J.M. (2004). Numerical modelling of two-way reinforced concrete slabs in fire. Engineering Structures; 26:1081-91.

Lin T. D., Zwiers R. I., Shirley S.T. and Burg R.G. (1989). Fire test of concrete slab reinforced with epoxy-coated bars. ACI Structural Joural, 86(2): 156-162.

Liu J., Riggs H.R. and Tessler A. (2000). A four-node, shear-deformable shell element developed via explicit Kirchhoff constraints. International Journal for Numerical Methods in Engineering, 49: 1065-1086.

Liu G.R., Dai K.Y. and Nguyen T.T. (2007). A smoothed finite element method for mechanics problems. Computational Mechanics, 39(6): 859-877.

Lowes L.N., Mitra N. and Altoontash A. (2004). A beam-column joint model for simulating the earthquake response of reinforced concrete frames. PEER Report 2003/10, University of California, Berkeley.

Lyons L.P.R. (1970). Elastic plane stress analysis of two-dimensional pierced shear walls by finite element idealisation, CIRIA Report.

Lyons L.P.R. (1977). A general finite element system with special reference to the analysis of cellular structures. PhD Thesis, Department of Civil Engineering, Imperial College of Science Technology, London.

Long C.S., Geyer S. and Groenwold A.A. (2006). A numerical study of the effect of penalty parameters for membrane elements with independent rotation fields and penalized equilibrium. Finite Element in Analysis and Design, 42: 757-765.

M

MacLeod I.A. (1969). New rectangular finite element for shear wall analysis. Journal of the Structural Division, 95(3): 399-409.

MacNeal R.H. (1978a). A simple quadrilateral shell element. Computers & Structures, 8: 175-183.

MacNeal R.H. (1978b). Derivation of element stiffness matrices by assumed strain distribution. Nuclear Engineering and Design, 70: 3-12.

MacNeal R.H. (1987). A theorem regarding the locking of tapered four-noded membrane elements, International Journal for Numerical Methods in Engineering, 24:1793-1799.

MacNeal R.H. and Harter R.L. (1988). A refined four-noded membrane element with rotational degrees of freedom. Computers & Structures, 28:75-88.

Malkus D.S. and Hughes T.J.R. (1978). Mixed finite element methods-reduced and selective integration techniques: a unification of concepts. Computer Methods in Applied Mechanics and Engineering, 15: 63-81.

Mazzoni S., McKenna F., Scott M.H. and Fenves G.L. (2007). OpenSees Command Language Manual. University of California, Berkeley.

McKenna F.T. (1997). Object-Oriented Finite Element Programming: Frameworks for Analysis, Algorithms and Parallel Computing, PhD thesis, University of California, Berkeley.

Melosh R.J. (1961). A stiffness matrix for the analysis of thin plates in bending. Journal of the Aeronautical Sciences, 28 (l): 34-42.

Militello C. and Felippa C.A. (1990a). A variational justification of the assumed natural strain formulation of finite elements: I. Variational principles. Computers & Structures, 34: 431-438.

Militello C. and Felippa C.A. (1990b). A variational justification of the assumed natural strain formulation of finite elements: II. The four-node C0 plate element. Computers & Structures, 34: 439–444.

Militello C. and Felippa C.A. (1991). The first ANDES elements: 9-DOF plate bending triangles. Computer Methods in Applied Mechanics and Engineering, 93:217-246.

Mindhn R.D. (1951). Influence of rotatory inertia and shear in flexural motions of isotropic elastic plates. Journal of Applied Mechanics, 18: 31-38.

Mohr G.A. (1982). Finite element formulation by nested interpolations: application to the drilling freedom problem. Computers & Structures, 15: 185-190.

Morley L.S.D. (1967). A triangular equilibrium element with linearly varying bending moments for plate bending problems. Journal of Royal Aeronautical, 71:715-719.

Morley L.S.D. (1968). The triangular equilibrium element in the solution of plate bending problems. The Aeronautical Quarterly, 19: 149-169.

Morley L.S.D. (1971). The constant-moment plate bending element. Journal of Strain Analysis, 6:20-24.

N

Nagtegaal J.C., Nakazawa S. and Tateishi M. (1986). On the construction of optimal Mindlin type plate and shell elements. In Finite Element Methods for Plate and Shell Structures, Volume 1: Element Technology, Pineridge Press, Swansea, 348-364.

Najjar S.R. (1994). Three-dimensional analysis of steel frames and subframes in fire. PhD thesis, Department of Civil and Structural Engineering, The University of Sheffield, Sheffield, U.K.

Najjar S.R. and Burgess I.W. (1996). A nonlinear analysis for threedimensional steel frames in fire conditions. Engineering Structures, 18(1): 77-89.

Newman G.M., Robinson J.T. and Bailey C.G. (2006). Fire safe design: a new approach to multi-storey steel-framed buildings. 2nd ed. SCI Publication P288. Ascot: The Steel Construction Institute.

Nguyen-Van H., Mai-Duy N. and Tran-Cong T. (2007). A simple and accurate four-node quadrilateral element using stabilized nodal integration for laminated plates. CMC: Computers, Materials & Continua, 6(3): 159-176.

Nguyen-Van H., Mai-Duy N. and Tran-Cong T. (2008a). A node-based element for analysis of planar piezoelectric structures. CMES: Computer Modeling in Engineering & Sciences, 36(1): 65-96.

Nguyen-Van H., Mai-Duy N. and Tran-Cong T. (2008b). A smoothed four-node piezoelectric element for analysis of two-dimensional smart structures. CMES: Computer Modeling in Engineering & Sciences, 23(3): 209-222.

Nguyen-Van H., Mai-Duy N. and Tran-Cong T. (2009). An improved quadrilateral flat element with drilling degrees of freedom for shell structural analysis. Computer Modelling in Engineering and Sciences, 49(2):81-110.

Noor A.K. and Peters J.M. (1981). Mixed models and reduced/selective integration displacement models for nonlinear analysis of curved beams. International Journal for Numerical Methods in Engineering, 17: 615-631.

O

O'Connor M.A. and Martin D.M. (1998). Behaviour of a multi-storey steel framed building subjected to fire attack. Journal of Constructional Steel Research, 46(1): 295-295.

Oliver J. and Onate E. (2005). A total lagrangian formulation for the geometrically nonlinear analysis of structures using finite elements. Part I. Two-dimensional problems: Shell and plate structures. International Journal for Numerical Methods in Engineering, 20(12):2253-2281.

OpenSees website: http://opensees.berkeley.edu/

P

Park K.C. and Flaggs D.L. (1984). An operational proce-dure for the symbolic analysis of the finite element methods. Computer Methods in Applied Mechanics and Engineering, 42: 37-46.

Park K.C. and Flaggs D.L. (1984). A Fourier analysis of spurious mechanism and locking in the finite element method. Computer Methods in Applied Mechanics and Engineering, 46: 65-81.

Park K.C. (1986). Improved strain interpolation for curved C0 elements. International Journal for Numerical Methods in Engineering, 22: 281-288.

Park K.C. and Stanley G.M. (1986). A curved C0 shell element based on assumed natural coordinate strains. Journal of Applied Mechanics, 53: 278-290.

Park R. (1964a). Tensile membrane behaviour of uniformly loaded rectangular reinforced concrete slabs with full restrained edges. Magazine of Concrete Research, 16(46): 39-44.

Park R. (1964b). Ultimate strength of rectangular concrete slabs under short-term uniform loading with edges restrained against lateral movement. Proceedings of Institute of Civil Engineering, 28:125-50.

Pawsey S.F. and Clough R.W. (1971). Improved numerical integration of thick slab finite elements. International Journal for Numerical Methods in Engineering, 3: 575-586.

Peano A. (1979). Conforming approximations for Kirchhoff plates and shells. International Journal for Numerical Methods in Engineering, 14: 1273-1291.

Paknahad M., Noorzaei J., Jaafar M.S. and Thanoon W.A. (2007). Formulation, computation and application of optimal membrane triangular element with drilling degrees of freedom. International Journal of Engineering and Technology, 4(1): 95-105.

Pian T.H.H. (1964). Derivation of element stiffness matrices by assumed stress distribution. Journal of AIAA, 2:1332-1336.

Pian T.H.H. (1965). Element stiffness matrices for boundary compatibility and for prescribed boundary stresses. Proceedings of 1st Conference on Matrix Methods in Structural Mechanics, Ohio, 457-477.

Pian T.H.H. and Tong P. (1968). Rationalization in deriving element stiffness matrix by assumed stress approach. AFFDL-TRdl-150, 441-469.

Pian T.H.H. and Tong P. (1969). Basis of finite element methods for solid continua. International Journal for Numerical Methods in Engineering, 1(1): 3-28.

Pian T.H.H. (1995). State-of-the-art development of hybrid/mixed finite element method. Finite Elements in Analysis and Design, 21: 5-20.

Pimpinelli G. (2004). An assumed strain quadrilateral element with drilling degrees of freedom. Finite Element in Analysis and Design, 41:267-283.

Polak M.A. (1996). Effective stiffness model for reinforced concrete slabs. Journal of Structural Engineering, 122(9): 1025-30.

Pugh E.D.L., Hinton E. and Zienkiewicz O.C. (1978). A study of quadrilateral plate bending elements with reduced integration. International Journal for Numerical Methods in Engineering, 12: 1059-1079.

R

Ramberg W. and Osgood W.R. (1943). Description of stress-strain curves by three parameters. Technical Note No. 902, National Advisory Committee for Aeronautics, Washington DC.

Razzaque A. (1973.) Program for triangular bending elements with derivative smoothing. International Journal for Numerical Methods in Engineering, 6:333-343.

Reissner E. (1945). The effect of transverse shear deformation on the bending of elastic plates. Journal of Applied Mechanics, 12: 69-76.

Reissner E. (1965). A note on variational principles in elasticity. International Journal of Solids and Structures, 1: 93-95.

Rengarajan G., Aminpour M.A. and Knight N.F. (1995). Improved assumed-stress hybrid shell element with drilling degrees of freedom for linear stress buckling and free vibration analyses. International Journal for Numerical Methods in Engineering, 38: 1917-1943.

Robinson J. (1980). Four-node quadrilateral stress membrane element with rotational stiffness. International Journal for Numerical Methods in Engineering, 16: 1567-1569.

Robinson J. and Sheng L.D. (1988). Some single element testing of Cook's quadrilateral membrane with rotational degrees of freedom, Finite Element News, 22-27.

Rose P.S., Bailey C.G., Burgess I.W. and Plank R.J. (1998). The influence of floor slabs on the structural performance of the Cardington frame in fire. Journal of Constructional Steel Research, 46(1): 310-311.

Rossow M.P., Lee J.C. and Chen K.C. (1976). Computer implementation of the constraint method. Computers & Structures, 6: 203-209.

Rossow M.P. and Ibrahimkhail A.K. (1978). Constraint method analysis of stiffned plates. Computers & Structures, 8: 51-60.

Rotter J.M., Sanad A.M., Usmani A.S. and Gillie M. (1999). Structural performance of redundant structures under local fires. In: Proceedings of Interflam'99, 2.

Rumbaugh J., Blaha M. and Premerhani W. (1991). Object-oriented modelling and design. Prentice-Hall, Englewood Cliffs, New Jersey 07632.

Rubert A. and Schaumann P. (1986). Structural steel and plane frame assemblies under fire action. Fire Safety Journal, 10:173-184.

S

Samuelsson A. 1979. The global constant strain condition and the patch test. Energy Methods in Finite Element Analysis, 49-68, John Wiley, Chichester, Sussex.

Sanad A.M., Rotter J.M., Usmani A.S. and O'Connor M. (2000a). Composite beam in large buildings under fire. Fire Safety Journal, 35: 165-188.

Sanad A.M., Lamont S., Usmani A.S. and Rotter J.M. (2000b). Structural behaviour in a fire compartment under different heating regimes—part 1 (slab thermal gradients). Fire Safety Journal, 35: 99-116.

Sanad A.M., Lamont S., Usmani A.S. and Rotter J.M. (2000c). Structural behaviour in a fire compartment under different heating regimes—part 2 (slab mean temperatures). Fire Safety Journal, 35: 117-130.

Sander G. (1970). Applications of the dual analysis principle. Proceedings of IUTAM Svmv. on High Sueed Computing of Elastic Structures, University of Liege, Belgium, 167-207.

Sawczuk A. and Winnicki L. (1965). Plastic behaviour of simply supported reinforced concrete plates at moderately large deflections. International Journal of Solids and Structures, 1:97–111.

Scordelis A.C. (1967). Analysis of continuum box girder bridges. SESM Report 67-2. Department of Civil Engineering, University of California, Berkeley, CA.

Simo J.C. and Rifai M.S. (1990). A class of mixed assumed strain methods and the method of incompatible modes. International Journal for Numerical Methods in Engineering, 29:1595-1638.

Southwell R.V. (1950). On the analogues relating flexure and extension of flat plates. Quarterly of Applied Mathematics, Part 3:257-270.

Spacone E., Ciampi V. and Filippou F.C. (1992). A beam element for seismic damage analysis. Report of University of California, Berkeley.

Spacone E., Ciampi V. and Filippou F.C. (1996). Mixed formulation of nonlinear beam finite element. Computers & Structures, 58(1): 71-83.

Specht B. (1988). Modified functions for the three node plate bending element passing the patch test. International Journal for Numerical Methods in Engineering, 26:705-715.

Spilker R.L. and Munir N.I. (1980). The hybrid-stress model for thin plates. International Journal for Numerical Methods in Engineering, 15(8):1239-1260.

Stolarski H. and Belytschko T. (1983). Shear and membrane locking in curved C0 elements. Computer Methods in Applied Mechanics and Engineering, 41: 279-296.

Stricklin J.A., Haisler W.E., Tisdale P.R. and Gunderson R. (1969). A rapidly converging triangular plate element. Journal of AIAA, 7 (1):180-181.

Structural Fire Engineering Investigation of Broadgate Phase 8 Fire. (1991). Technical Report, The Steel Construction Institute.

Szabo B.A. and Tsai C.T. (1973). The quadratic programming approach to the finite element method. International Journal for Numerical Methods in Engineering, 5: 375-381.

Szabo B.A. and Kassos T. (1975). Linear equality constraints in finite element approximation. International Journal for Numerical Methods in Engineering, 9: 563-580.

Sze K.Y., Wanji C. and Cheung Y.K. (1992). An efficient quadrilateral plane element with drilling degrees of freedom using orthogonal stress modes. Computers & Structures, 42(5): 695-705.

Sze K.Y. and Ghali A. (1993). Hybrid plane quadrilateral element with corner rotations. Journal of Structural Engineering, 119: 2552-2572.

T

Talamona D. and Franssen J.M. (2000). New quadrangular shell element in Safir. Proceedings ofthe First International Symposium of Structures in Fire, Copenhagen, 195-210.

Taucer F.F., Spacone E. and Filippou F.C. (1991). A Fiber Beam-column element for seismic response analysis of reinforced concrete structures. Report of University of California, Berkeley.

Taylor R.L. (1965). A note on a possible basis for a new method of ultimate load design of reinforced concrete slabs, Magazine of Concrete Research, 17(53): 183-186.

Taylor R.L., Zienkiewicz O.C., Simo J.C. and Chan A.H.C. (1986). The patch test – a condition for assessing FEM convergence. International Journal for Numerical Methods in Engineering, 22: 39-62.

Taylor R.L. and Simo J.C. (1985). Bending and membrane elements for analysis of thick and thin shells. Proceedings of NUMETA 85 Conference, Rotterdam, 1: 587-591.

Taylor R.L. (1988). Finite element analysis of linear shell problems. The Mathematics of Finite Elements and Applications, Academic Press, London, 1:191-203.

Tessler A. and Dong S.B. (1981). On a hierarchy of conforming Timoshenko beam elements. Computers & Structures, 14: 335-344.

Tessler A. (1981). On a conforming, Mindlin-type plate element. Fourth MAFELAP Conference, Brunei University, Uxbridge.

Tessler A. (1982). An efficient, conforming axisymmetric shell element including transverse shear and rotary inertia. Computers & Structures, 15: 567-574.

Tessler A. and Hughes T.J.R. (1983). An improved treatment of transverse shear in the Mindlin-type four-node quadrilateral element. Computer Methods in Applied Mechanics and Engineering, 39: 311-335.

Tinawi R. (1972). Behaviour of orthotropic bridge decks Ph.D. Thesis, McGill University. Montreal. Canada.

Tocher J.L. (1962). Analysis of plate bending using triangular elements, PhD Thesis, Dept. of Civil Engineering, University of California, Berkeley, California.

Tocher J.L. and Hartz B.J. (1967). Higher order finite element for plane stress. Journal of Engineering Mechanics, 9(3): 149-172.

Tong P. (1970). New displacement hybrid models for solid continua. International Journal for Numerical Methods in Engineering, 2: 73-83,

Tsach U. (1981). Locking of thin plate shell elements. International Journal for Numerical Methods in Engineering, 17: 633-644.

Turner M.J., Clough R.W., Martin H.C. and Topp L.J. (1956). Stiffness and deflection analysis of complex structures. Journal of Aeronautical Sciences, 23: 805-823.

U

Usmani A.S. (2000). Application of Fundamental Structural Mechanics Principles in Assessing the Cardington Fire Tests. In: Structures in Fire: Proceedings of the First International Workshop, Copenhagen.

Usmani A.S., Rotter J.M., Lamont S., Sanad A.M. and Gillie M. (2001). Fundamental principles of structural behaviour under thermal effects. Fire Safety Journal, 36(8):721-44.

Usmani A.S. and Cameron N.J.K. (2004). Limit capacity of laterally restrained reinforced concrete floor slabs in fire. Cement & Concrete Composites, 26:127-140.

Usmani A.S. (2005). Understanding the response of composite structures to fire. Engineering Journal, AISC, 42(2):83-98.

Ugural A. (1999). Stress in Plates and Shells. McGraw-Hill, New York, 2nd edition.

V

Veubeke B.F. (1965). Displacement and equilibrium models in the finite element method. Stress Analysis, Wiley, New York, 145-197.

Visser W. (1969). A refined mixed type plate bending element. Journal of AIAA, 7(9): 1801-1803.

W

Wang C. and Pian T.H.H. (1988). Hybrid semi-Loof element for buckling of thin-walled structures, Computers & Structures, 30: 811-816.

Wang Y.C., Lennon T., Moore D.B. (1995). The behaviour of steel frames subject to fire. Journal of Constructional Steel Research, 35:291-322.

Wang Y.C. (1996). Tensile membrane action in slabs and its application to the Cardington fire tests. Proceedings of the Second Cardington Conference, March, BRE.

Wempner G.A., Cklen J.T. and Dross D.A. (1968). Finite element analysis of thin shells. Journal of Engineering Mechanics, 94 (6): 1273-1294.

Wempner G., Talaslidis D. and Hwang C.M. (1982). A simple and efficient approximation of shells via finite quadrilateral elements. Journal of Applied Mechanics, 49:115-120

William K.J. (1969). Finite element analysts of cellular structures. PhD Thesis, Dept. of Civil Engineering. University of California. Berkeley, CA .

Wood R.H. (1961). Plastic and elastic design of slabs and plates, with particular reference to reinforced concrete floor slabs. London: Thames and Hudson.

Y

Yassin M.H.M. (1994). Nonlinear analysis of prestressed concrete structures under monotonic and cyclic loads. PhD thesis, University of California, Berkeley.

Yoshida Y. (1974). A hybrid stress element of thin shell analysis. Proc. Intl Conference on Finite Element Methods in Engineering, University of New South Wales, Australia.

Yu X.M., Huang Z., Burgess I.W. and Plank R.J. (2008). Nonlinear analysis of orthotropic composite slabs in fire. Engineering Structures, 30(1): 67-80.

Yu X.M. and Huang Z. (2008). An embedded FE model for modelling reinforced concrete slabs in fire. Engineering Structures, 30: 3228-3238.

Yunus S. M. (1988). A study of different hybrid elements with and without rotational D.O.F. for plane stress/plane strain problems. Computers & Structures, 30: 1127-1133

Yunus S.M., Saigal S. and Cook R.D. (1989). On improved hybrid finite elements with rotational degrees of freedom. International Journal for Numerical Methods in Engineering, 28:785-800.

Z

Zienkiewicz O.C. (1977). The Finite Element Method in Engineering Science. McGraw-Hill, New York.

Zienkiewicz O.C., Too J. and Taylor R.L. (1971). Reduced integration technique in general analysis of plates and shells. International Journal for Numerical Methods in Engineering, 3:275-290.

Zienkiewicz O.C. and Hinton E. (1976). Reduced integration, function smoothing and nonconformity in finite element analysis. Journal of the Franklin Institute, 302(6): 443-461.

Zienkiewicz O.C. and Taylor R.L. (2005). The Finite Element Method for Solid and Structural Mechanics, Butterworth-Heinemann, Oxford.

Appendix A Uniaxial material at elevated temperature

A.1 Steel01Thermal

```
double Steel01Thermal::setThermalTangentAndElongation(double &TempT,
double&ET, double&Elong)
{// EN 1993 pt 1-2-1.
  if (TempT <= 100) {
        fy = fyT;
        E0 = E0T;
  }
  else if (TempT <= 200) {
      fy = fyT;
      E0 = E0T*(1 - (TempT - 100)*0.1/100);
  }
  else if (TempT <= 300) {
      fy = fyT;
      E0 = E0T*(0.9 - (TempT - 200)*0.1/100);
  }
  else if (TempT <= 400) {
        fy = fyT;
        E0 = E0T*(0.8 - (TempT - 300)*0.1/100);
  }
  else if (TempT <= 500) {
      fy = fyT*(1 - (TempT - 400)*0.22/100);
     E0 = E0T*(0.7 - (TempT - 400)*0.1/100);
  }
  else if (TempT <= 600) {
      fy = fyT*(0.78 - (TempT - 500)*0.31/100);
     E0 = E0T*(0.6 - (TempT - 500)*0.29/100);
  }
  else if (TempT <= 700) {
      fy = fyT*(0.47 - (TempT - 600)*0.24/100);
     E0 = E0T*(0.31 - (TempT - 600)*0.18/100);
  }
  else if (TempT <= 800) {
```

```
        fy = fyT*(0.23 - (TempT - 700)*0.12/100);
        E0 = E0T*(0.13 - (TempT - 700)*0.04/100);
    }
    else if (TempT <= 900) {
        fy = fyT*(0.11 - (TempT - 800)*0.05/100);
            E0 = E0T*(0.09 - (TempT - 800)*0.02/100);
    }
    else if (TempT <= 1000) {
        fy = fyT*(0.06 - (TempT - 900)*0.02/100);
            E0 = E0T*(0.0675 - (TempT - 900)*(0.00675 - 0.0045)/100);
    }
    else if (TempT <= 1100) {
        fy = fyT*(0.04 - (TempT - 1000)*0.02/100);
            E0 = E0T*(0.045 - (TempT - 1000)*(0.0045 - 0.00225)/100);
    }
    else if (TempT <= 1200) {
        fy = fyT*(0.02 - (TempT - 1100)*0.02/100);
            E0 = E0T*(0.0225 - (TempT - 1100)*0.0225/100);
    }
    else  {
        opserr << "the temperature is invalid\n";
    }
    // caculation of thermal elongation of reinforcing steel. JZ
    else if (TempT <= 750) {
    ThermalElongation = -2.416e-4 + 1.2e-5 *(TempT) + 0.4e-8
*(TempT)^2);
    }
    else if (TempT <= 1200) {
        ThermalElongation = 11e-3;
    }
    else if (TempT <= 1180) {
        ThermalElongation = -6.2e-3 + 2e-5*TempT;
    }
    else {
        opserr << "the temperature is invalid\n";
    }
    ET = E0;
    Elong = ThermalElongation;
    return 0;}
```

Figure A.1: Function "Steel01Thermal::setThermalTangentAndElongation ()"

A.2 Concrete02Thermal

```
double Concrete02Thermal::setThermalTangentAndElongation(double &TempT,
double&ET, double&Elong)
{//material properties with temperature
  // Temp (TempT) is actural temperature; ambient temperature is 20;
  // Siliceous aggregates
  // The datas are from EN 1992 part 1-2-1

  // Tensile strength at elevated temperature
  if (TempT <= 100) {
          ft = ft0;
      Ets = Ets0;
  }
  else if (TempT <= 600) {
          ft = ft0*(1.0 - 1.0*(TempT -100)/500);
      Ets = Ets0*(1.0 - 1.0*(TempT -100)/500);
  }
  else {
          ft = 0;
          Ets = 0;
  }

  // Compression strength, at elevated TempTerature, Siliceous
aggregates
  // Maximum temperature is 1100
  // Strain at compression strength, at elevated temperature
  // Ultimate (crushing) strain, at elevated temperature
  if (TempT <= 20) {
          fc = fc0;
          epsc0 = -0.0025;
          fcu = fcu0;
          epscu = -0.02;
  }
  else if (TempT <= 100) {
          fc = fc0;
          epsc0 = -(0.0025 + (0.004-0.0025)*(TempT - 20)/80);
          fcu = fcu0;
          epscu = -(0.0200 + (0.0225-0.0200)*(TempT - 20)/80);
  }
  else if (TempT <= 200) {
      fc = fc0*(1 - (TempT - 100)*0.05/100);
          epsc0 = -(0.0040 + (0.0055-0.0040)*(TempT - 100)/100);
      fcu = fcu0*(1 - (Temp - 100)*0.05/100);
          epscu = -(0.0225 + 0.0025*(Temp - 100)/100);
  }
  else if (TempT <= 300) {
```

```
    fc = fc0*(0.95 - (TempT - 200)*0.1/100);
        epsc0 = -(0.0055 + (0.0070-0.0055)*(TempT - 200)/100);
        fcu = fcu0*(0.95 - (TempT - 200)*0.1/100);
        epscu = -(0.0250 + 0.0025*(TempT - 200)/100);
}
else if (TempT <= 400) {
    fc = fc0*(0.85 - (TempT - 300)*0.1/100);
        epsc0 = -(0.0070 + (0.0100-0.0070)*(TempT - 300)/100);
        fcu = fcu0*(0.85 - (TempT - 300)*0.1/100);
        epscu = -(0.0275 + 0.0025*(TempT - 300)/100);
}
else if (TempT <= 500) {
    fc = fc0*(0.75 - (TempT - 400)*0.15/100);
        epsc0 = -(0.0100 + (0.0150-0.0100)*(TempT - 400)/100);
        fcu = fcu0*(0.75 - (TempT - 400)*0.15/100);
        epscu = -(0.03 + 0.0025*(TempT - 400)/100);
}
else if (TempT <= 600) {
    fc = fc0*(0.60 - (TempT - 500)*0.15/100);
        epsc0 = -(0.0150 + (0.0250-0.0150)*(TempT - 500)/100);
        fcu = fcu0*(0.60 - (TempT - 500)*0.15/100);
        epscu = -(0.0325 + 0.0025*(TempT - 500)/100);
}
else if (TempT <= 700) {
    fc = fc0*(0.45 - (TempT - 600)*0.15/100);
        epsc0 = -0.0250;
        fcu = fcu0*(0.45 - (TempT - 600)*0.15/100);
        epscu = -(0.035 + 0.0025*(TempT - 600)/100);
}
else if (TempT <= 800) {
    fc = fc0*(0.30 - (TempT - 700)*0.15/100);
        epsc0 = -0.0250;
        fcu = fcu0*(0.30 - (TempT - 700)*0.15/100);
        epscu = -(0.0375 + 0.0025*(TempT - 700)/100);
}
else if (TempT <= 900) {
    fc = fc0*(0.15 - (TempT - 800)*0.07/100);
        epsc0 = -0.0250;
        fcu = fcu0*(0.15 - (TempT - 800)*0.07/100);
        epscu = -(0.04 + 0.0025*(TempT - 800)/100);
}
else if (TempT <= 1000) {
    fc = fc0*(0.08 - (TempT - 900)*0.04/100);
        epsc0 = -0.0250;
        fcu = fcu0*(0.08 - (TempT - 900)*0.04/100);
        epscu = -(0.0425 + 0.0025*(TempT - 900)/100);
}
else if (TempT <= 1100) {
    fc = fc0*(0.04 - (TempT - 1000)*0.03/100);
        epsc0 = -0.0250;
        fcu = fcu0*(0.04 - (TempT - 1000)*0.03/100);
        epscu = -(0.045 + 0.0025*(TempT - 1000)/100);
}
else  {
    opserr << "the temperature is invalid\n";
}
```

```
// calculation of thermal elongation, Siliceous aggregates
 if (TempT <= 20) {
    ThermalElongation = 0.0;
}
else if (TempT <= 700) {
    ThermalElongation = -1.8e-4 + 9e-6 *TempT + 2.3e-11 *TempT^3
}
else if (TempT <= 1200) {
    ThermalElongation = 14e-3;
}
else {
    opserr << "the temperature is invalid\n";
}
```

Figure A.2: Function "Concrete02Thermal::setThermalTangentAndElongation ()"

A.3 Pinching4ThermalSteel

```
double Pinching4ThermalSteel::setThermalTangentAndElongation(double
&TempT, double&ET, double&Elong)
double
{


   // Update temperature dependent yield stress and Young's modulus
according to
// EN 1993 pt 1-2-1.

   if (TempT <= 400) {

}

else if (TempT <= 500) {
        stress1n = stress3n0*(1 - (TempT - 400)*0.22/100);
        stress2n = stress2n0*(1 - (TempT - 400)*0.22/100);
        stress3n = stress3n0*(1 - (TempT - 400)*0.22/100);

    stress1p = stress3n0*(1 - (TempT - 400)*0.22/100);
        stress2p = stress2n0*(1 - (TempT - 400)*0.22/100);
        stress3p = stress3n0*(1 - (TempT - 400)*0.22/100);

        ET = stress1p/strain1p;

}
else if (TempT <= 600) {
        stress1n = stress3n0*(0.78 - (TempT - 500)*0.31/100);
        stress2n = stress2n0*(0.78 - (TempT - 500)*0.31/100);
        stress3n = stress3n0*(0.78 - (TempT - 500)*0.31/100);

    stress1p = stress3n0*(0.78 - (TempT - 500)*0.31/100);
        stress2p = stress2n0*(0.78 - (TempT - 500)*0.31/100);
```

```
        stress3p = stress3n0*(0.78 - (TempT - 500)*0.31/100);

        ET = stress1n/strain1n;
}
else if (TempT <= 700) {
        stress1n = stress3n0*(0.47 - (TempT - 600)*0.24/100);
        stress2n = stress2n0*(0.47 - (TempT - 600)*0.24/100);
        stress3n = stress3n0*(0.47 - (TempT - 600)*0.24/100);

    stress1p = stress3n0*(0.47 - (TempT - 600)*0.24/100);
        stress2p = stress2n0*(0.47 - (TempT - 600)*0.24/100);
        stress3p = stress3n0*(0.47 - (TempT - 600)*0.24/100);

        ET = stress1n/strain1n;
}
else if (TempT <= 800) {
        stress1n = stress3n0*(0.23 - (TempT - 700)*0.12/100);
        stress2n = stress2n0*(0.23 - (TempT - 700)*0.12/100);
        stress3n = stress3n0*(0.23 - (TempT - 700)*0.12/100);

    stress1p = stress3n0*(0.23 - (TempT - 700)*0.12/100);
        stress2p = stress2n0*(0.23 - (TempT - 700)*0.12/100);
        stress3p = stress3n0*(0.23 - (TempT - 700)*0.12/100);

        ET = stress1n/strain1n;

}
else if (TempT <= 900) {
        stress1n = stress3n0*(0.11 - (TempT - 800)*0.05/100);
        stress2n = stress2n0*(0.11 - (TempT - 800)*0.05/100);
        stress3n = stress3n0*(0.11 - (TempT - 800)*0.05/100);

    stress1p = stress3n0*(0.11 - (TempT - 800)*0.05/100);
        stress2p = stress2n0*(0.11 - (TempT - 800)*0.05/100);
        stress3p = stress3n0*(0.11 - (TempT - 800)*0.05/100);

        ET = stress1n/strain1n;

}
else if (TempT <= 1000) {
        stress1n = stress3n0*(0.06 - (TempT - 900)*0.02/100);
        stress2n = stress2n0*(0.06 - (TempT - 900)*0.02/100);
        stress3n = stress3n0*(0.06 - (TempT - 900)*0.02/100);

    stress1p = stress3n0*(0.06 - (TempT - 900)*0.02/100);
        stress2p = stress2n0*(0.06 - (TempT - 900)*0.02/100);
        stress3p = stress3n0*(0.06 - (TempT - 900)*0.02/100);

        ET = stress1n/strain1n;

}
else if (TempT <= 1100) {
        stress1n = stress3n0*(0.04 - (TempT - 1000)*0.02/100);
        stress2n = stress2n0*(0.04 - (TempT - 1000)*0.02/100);
        stress3n = stress3n0*(0.04 - (TempT - 1000)*0.02/100);

    stress1p = stress3n0*(0.04 - (TempT - 1000)*0.02/100);
        stress2p = stress2n0*(0.04 - (TempT - 1000)*0.02/100);
```

```
            stress3p = stress3n0*(0.04 - (TempT - 1000)*0.02/100);

            ET = stress1n/strain1n;

    }
    else if (TempT <= 1200) {
            stress1n = stress3n0*(0.02 - (TempT - 1100)*0.02/100);
            stress2n = stress2n0*(0.02 - (TempT - 1100)*0.02/100);
            stress3n = stress3n0*(0.02 - (TempT - 1100)*0.02/100);

        stress1p = stress3n0*(0.02 - (TempT - 1100)*0.02/100);
            stress2p = stress2n0*(0.02 - (TempT - 1100)*0.02/100);
            stress3p = stress3n0*(0.02 - (TempT - 1100)*0.02/100);

            ET = stress1n/strain1n;

    }
    else  {
        opserr << "the temperature is invalid\n";
    }

    // Calculate thermal elongation.
    if (TempT <= 20) {
            ThermalElongation = 0.0;
    }
    else if (TempT <= 750) {
        ThermalElongation = -2.416e-4+ 1.2e-5 *TempT + 0.4e-8 *TempT^2;
    }
    else if (TempT <= 860) {
        ThermalElongation = 11e-3;
    }
    else if (TempT <= 1200) {
        ThermalElongation = -6.2e-3 + 2e-5*TempT;
    }
    else {
        opserr << "the temperature is invalid\n";
    }

    Elong = ThermalElongation;

    return 0;
```

Figure A.3: Function "Pinching4ThermalSteel::setThermalTangentAndElongation ()"

A.4 Pinching4ThermalConcrete

```
double Pinching4ThermalConcrete::setThermalTangentAndElongation(double
&TempT, double&ET, double&Elong)
    // Siliceous aggregates
// The datas are from EN 1992 part 1-2-1
```

```
if (TempT <= 20) {

}
else if (TempT <= 100) {

        stress2n = stress2n0;
        stress3n = stress3n0;
        stress1n = 1/3*stress3n;
        stress4n = stress4n0;

        strain2n = -(0.0025 + (0.004-0.0025)*(TempT - 20)/80);
        strain3n = -(0.0025 + (0.004-0.0025)*(TempT - 20)/80);
    strain1n = 1/6*strain3n;
        strain4n = -(0.0200 + (0.0225-0.0200)*(TempT - 20)/80);

        ET = 2*stress3n/strain3n;
}
else if (TempT <= 200) {
        stress2n = stress2n0*(1 - (TempT - 100)*0.05/100);
        stress3n = stress3n0*(1 - (TempT - 100)*0.05/100);
        stress1n = 1/3*stress3n;
        stress4n = stress4n0*(1 - (TempT - 100)*0.05/100);

        strain2n = -(0.0040 + (0.0055-0.0040)*(TempT - 100)/100);
        strain3n = -(0.0040 + (0.0055-0.0040)*(TempT - 100)/100);
    strain1n = 1/6*strain3n;
        strain4n = -(0.0225 + 0.0025*(TempT - 100)/100);

    ET = 2*stress3n/strain3n;

        stress1p = stress1p0*(1.0 - 1.0*(TempT -100)/500);
        stress2p = stress2p0*(1.0 - 1.0*(TempT -100)/500);
        stress3p = stress3p0*(1.0 - 1.0*(TempT -100)/500);

        strain1p = stress1p/ET;
        strain2p = stress2p/ET;
        strain3p = stress3p/ET;
        strain4p = strain3p + 0.002;

}
else if (TempT <= 300) {
        stress2n = stress2n0*(0.95 - (TempT - 200)*0.1/100);
        stress3n = stress3n0*(0.95 - (TempT - 200)*0.1/100);
        stress1n = 1/3*stress3n;
        stress4n = stress4n0*(0.95 - (TempT - 200)*0.1/100);

        strain2n =  -(0.0055 + (0.0070-0.0055)*(TempT - 200)/100);
        strain3n =  -(0.0055 + (0.0070-0.0055)*(TempT - 200)/100);
    strain1n = 1/6*strain3n;
        strain4n = -(0.0250 + 0.0025*(TempT - 200)/100);

    ET = 2*stress3n/strain3n;

        stress1p = stress1p0*(1.0 - 1.0*(TempT -100)/500);
        stress2p = stress2p0*(1.0 - 1.0*(TempT -100)/500);
        stress3p = stress3p0*(1.0 - 1.0*(TempT -100)/500);

        strain1p = stress1p/ET;
```

```
        strain2p = stress2p/ET;
        strain3p = stress3p/ET;
        strain4p = strain3p + 0.002;
}
else if (TempT <= 400) {
        stress2n = stress2n0*(0.85 - (TempT - 300)*0.1/100);
        stress3n = stress3n0*(0.85 - (TempT - 300)*0.1/100);
        stress1n = 1/3*stress3n;
        stress4n = stress4n0*(0.85 - (TempT - 300)*0.1/100);

        strain2n = -(0.0070 + (0.0100-0.0070)*(TempT - 300)/100);
        strain3n = -(0.0070 + (0.0100-0.0070)*(TempT - 300)/100);
    strain1n = 1/6*strain3n;
        strain4n = -(0.0275 + 0.0025*(TempT - 300)/100);

    ET = 2*stress3n/strain3n;

        stress1p = stress1p0*(1.0 - 1.0*(TempT -100)/500);
        stress2p = stress2p0*(1.0 - 1.0*(TempT -100)/500);
        stress3p = stress3p0*(1.0 - 1.0*(TempT -100)/500);

        strain1p = stress1p/ET;
        strain2p = stress2p/ET;
        strain3p = stress3p/ET;
        strain4p = strain3p + 0.002;
}
else if (TempT <= 500) {
    stress2n = stress2n0*(0.75 - (TempT - 400)*0.15/100);
        stress3n = stress3n0*(0.75 - (TempT - 400)*0.15/100);
        stress1n = 1/3*stress3n;
        stress4n = stress4n0*(0.75 - (TempT - 400)*0.15/100);

        strain2n = -(0.0100 + (0.0150-0.0100)*(TempT - 400)/100);
        strain3n = -(0.0100 + (0.0150-0.0100)*(TempT - 400)/100);
    strain1n = 1/6*strain3n;
        strain4n = -(0.03 + 0.0025*(TempT - 400)/100);

    ET = 2*stress3n/strain3n;

        stress1p = stress1p0*(1.0 - 1.0*(TempT -100)/500);
        stress2p = stress2p0*(1.0 - 1.0*(TempT -100)/500);
        stress3p = stress3p0*(1.0 - 1.0*(TempT -100)/500);

        strain1p = stress1p/ET;
        strain2p = stress2p/ET;
        strain3p = stress3p/ET;
        strain4p = strain3p + 0.002;

}
else if (TempT <= 600) {
    stress2n = stress2n0*(0.60 - (TempT - 500)*0.15/100);
        stress3n = stress3n0*(0.60 - (TempT - 500)*0.15/100);
        stress1n = 1/3*stress3n;
        stress4n = stress4n0*(0.60 - (TempT - 500)*0.15/100);

        strain2n = -(0.0150 + (0.0250-0.0150)*(TempT - 500)/100);
        strain3n = -(0.0150 + (0.0250-0.0150)*(TempT - 500)/100);
    strain1n = 1/6*strain3n;
```

```
            strain4n = -(0.0325 + 0.0025*(TempT - 500)/100);

        ET = 2*stress3n/strain3n;

            stress1p = stress1p0*(1.0 - 1.0*(TempT -100)/500);
            stress2p = stress2p0*(1.0 - 1.0*(TempT -100)/500);
            stress3p = stress3p0*(1.0 - 1.0*(TempT -100)/500);

            strain1p = stress1p/ET;
            strain2p = stress2p/ET;
            strain3p = stress3p/ET;
            strain4p = strain3p + 0.002;

    }
    else if (TempT <= 700) {
        stress2n = stress2n0*(0.45 - (TempT - 600)*0.15/100);
            stress3n = stress3n0*(0.45 - (TempT - 600)*0.15/100);
            stress1n = 1/3*stress3n;
            stress4n = stress4n0*(0.45 - (TempT - 600)*0.15/100);

            strain2n = -0.025;
            strain3n = -0.025;
        strain1n = 1/6*strain3n;
            strain4n = -(0.035 + 0.0025*(TempT - 600)/100);

        ET = 2*stress3n/strain3n;

            stress1p = 0.0;
            stress2p = 0.0;
            stress3p = 0.0;

            strain1p = stress1p/ET;
            strain2p = stress2p/ET;
            strain3p = stress3p/ET;
            strain4p = strain3p + 0.002;

    }
    else if (TempT <= 800) {
            stress2n = stress2n0*(0.30 - (TempT - 700)*0.15/100);
            stress3n = stress3n0*(0.30 - (TempT - 700)*0.15/100);
            stress1n = 1/3*stress3n;
            stress4n = stress4n0*(0.30 - (TempT - 700)*0.15/100);

            strain2n = -0.025;
            strain3n = -0.025;
        strain1n = 1/6*strain3n;
            strain4n = -(0.0375 + 0.0025*(TempT - 700)/100);

        ET = 2*stress3n/strain3n;

            stress1p = 0.0;
            stress2p = 0.0;
            stress3p = 0.0;

            strain1p = stress1p/ET;
            strain2p = stress2p/ET;
            strain3p = stress3p/ET;
            strain4p = strain3p + 0.002;
```

```
}
else if (TempT <= 900) {
        stress2n = stress2n0*(0.15 - (TempT - 800)*0.07/100);
        stress3n = stress3n0*(0.15 - (TempT - 800)*0.07/100);
        stress1n = 1/3*stress3n;
        stress4n = stress4n0*(0.15 - (TempT - 800)*0.07/100);

        strain2n = -0.025;
        strain3n = -0.025;
    strain1n = 1/6*strain3n;
        strain4n = -(0.04 + 0.0025*(TempT - 800)/100);

    ET = 2*stress3n/strain3n;

        stress1p = 0.0;
        stress2p = 0.0;
        stress3p = 0.0;

        strain1p = stress1p/ET;
        strain2p = stress2p/ET;
        strain3p = stress3p/ET;
        strain4p = strain3p + 0.002;

}
else if (TempT <= 1000) {
        stress2n = stress2n0*(0.08 - (TempT - 900)*0.04/100);
        stress3n = stress3n0*(0.08 - (TempT - 900)*0.04/100);
        stress1n = 1/3*stress3n;
        stress4n = stress4n0*(0.08 - (TempT - 900)*0.04/100);

        strain2n = -0.025;
        strain3n = -0.025;
    strain1n = 1/6*strain3n;
        strain4n = -(0.0425 + 0.0025*(TempT - 900)/100);

    ET = 2*stress3n/strain3n;

        stress1p = 0.0;
        stress2p = 0.0;
        stress3p = 0.0;

        strain1p = stress1p/ET;
        strain2p = stress2p/ET;
        strain3p = stress3p/ET;
        strain4p = strain3p + 0.002;
}

else  {
    opserr << "the temperature is invalid\n";
}


// calculation of thermal elongation, Siliceous aggregates
 if (TempT <= 20) {
    ThermalElongation = 0.0;
}
else if (TempT <= 700) {
```

```
    ThermalElongation = -1.8e-4 + 9e-6 *TempT + 2.3e-11 *Temp^3;
}
else if (Temp <= 1200) {
    ThermalElongation = 14e-3;
}
else {
    opserr << "the temperature is invalid\n";
}

Elong = ThermalElongation;

return 0;
```

Figure A.4: Function “Pinching4ThermalConcrete::setThermalTangentAndElongation ()”

Appendix B FiberSection2dThermal

```
const Vector&
FiberSection2dThermal::getTemperatureStress(double *dataMixed)
{
  double yBarT = 0;
  sTData[0] = 0.0; sTData[1] = 0.0;
  double fiberLocs[10000];
  double fiberArea[10000];
  if (sectionIntegr != 0) {
    sectionIntegr->getFiberLocations(numFibers, fiberLocs);
    sectionIntegr->getFiberWeights(numFibers, fiberArea);
  }
  else {
    for (int i = 0; i < numFibers; i++) {
      fiberLocs[i] = matData[2*i];
      fiberArea[i] = matData[2*i+1];
    }
  }
  double dataTempe[18];
  for (int i = 0; i < 18; i++) {
     dataTempe[i] = dataMixed[i];
  }

//JJadd, 12/2010, updata yBar = Ai*Ei*yi/(Ai*E*)  start
  double ThermalTangent[100];
  double ThermalElongation[100];
  for (int i = 0; i < numFibers; i++) {
         ThermalTangent[i]=0;
         ThermalElongation[i]=0;
  }

  for (int i = 0; i < numFibers; i++) {
    UniaxialMaterial *theMat = theMaterials[i];
    double yi = fiberLocs[i];
    double Ai = fiberArea[i];
```

```
    double FiberTemperature = 0 ; //JZ
    //if locY1 and locY9 are not less than zero
    if ( fabs(dataTempe[1]) <= 1e-10 && fabs(dataTempe[17]) <= 1e-
10 ) //no tempe load
    {
        FiberTemperature = 0;
    }
    else
    {
    //caculate the fiber tempe, T=T1-(Y-Y1)*(T1-T2)/(Y1-Y2)
    if (  fiberLocs[i] <= dataTempe[1])
    {
        opserr
<<"FiberSection2dThermal::setTrialSectionDeformationTemperature --
fiber loc is out of the section";
    }
    else if (fiberLocs[i] <= dataTempe[3])
    {
FiberTemperature = dataTempe[0] - (dataTempe[1] - fiberLocs[i]) *
(dataTempe[0] - dataTempe[2])/(dataTempe[1] - dataTempe[3]);
    }
    else if (   fiberLocs[i] <= dataTempe[5] )
    {
FiberTemperature = dataTempe[2] - (dataTempe[3] - fiberLocs[i]) *
(dataTempe[2] - dataTempe[4])/(dataTempe[3] - dataTempe[5]);
    }
    else if ( fiberLocs[i] <= dataTempe[7] )
    {
FiberTemperature = dataTempe[4] - (dataTempe[5] - fiberLocs[i]) *
(dataTempe[4] - dataTempe[6])/(dataTempe[5] - dataTempe[7]);
    }
    else if ( fiberLocs[i] <= dataTempe[9] )
    {
FiberTemperature = dataTempe[6] - (dataTempe[7] - fiberLocs[i]) *
(dataTempe[6] - dataTempe[8])/(dataTempe[7] - dataTempe[9]);
    }
    else if (fiberLocs[i] <= dataTempe[11] )
    {
```

```
FiberTemperature = dataTempe[8] - (dataTempe[9] - fiberLocs[i]) *
(dataTempe[8] - dataTempe[10])/(dataTempe[9] - dataTempe[11]);
    }
    else if (fiberLocs[i] <= dataTempe[13] )
    {
FiberTemperature = dataTempe[10] - (dataTempe[11] - fiberLocs[i]) *
(dataTempe[10] - dataTempe[12])/(dataTempe[11] - dataTempe[13]);
    }
    else if (fiberLocs[i] <= dataTempe[15] )
    {
FiberTemperature = dataTempe[12] - (dataTempe[13] - fiberLocs[i]) *
(dataTempe[12] - dataTempe[14])/(dataTempe[13] - dataTempe[15]);
    }
    else if ( fiberLocs[i] <= dataTempe[17] )
    {
FiberTemperature = dataTempe[14] - (dataTempe[15] - fiberLocs[i]) *
(dataTempe[14] - dataTempe[16])/(dataTempe[15] - dataTempe[17]);
    }
    else
    {
        opserr
<<"FiberSection2dThermal::setTrialSectionDeformationTemperature --
fiber loc is out of the section";
    }
    }
    // determine material strain and set it
    double tangent, elongation;
 theMat->setThermalTangentAndElongation(FiberTemperature, tangent,
elongation);

  ThermalTangent[i]=tangent;
  ThermalElongation[i]=elongation;
  }

//calculate centroid of section yBar for composite section,i.e. yBar
is related to tangent E
  double SigmaEAy = 0;
  double SigmaEA = 0;
  for (int i = 0; i < numFibers; i++) {
```

```
      SigmaEAy += ThermalTangent[i]*fiberArea[i]*fiberLocs[i];
          SigmaEA += ThermalTangent[i]*fiberArea[i];
  }
  yBarT = SigmaEAy/SigmaEA;

 // calculate section resisting force due to thermal load
  double FiberForce;
  for (int i = 0; i < numFibers; i++) {
FiberForce = ThermalTangent[i]*fiberArea[i]*ThermalElongation[i];
      sTData[0] += FiberForce;
      sTData[1] -= FiberForce*(fiberLocs[i] - yBarT);
  }

  return *sT;
}
```

Figure B.1: Function of "FiberSection2dThermal::getTemperatureStress()"

Appendix C DispBeamColumn2dThermal

```
DispBeamColumn2dThermal::addLoad(ElementalLoad *theLoad, double
loadFactor)
{
  int type;
  const Vector &data = theLoad->getData(type, loadFactor);
  double L = crdTransf->getInitialLength();
  if (type == LOAD_TAG_Beam2dUniformLoad) {…}
  else if (type == LOAD_TAG_Beam2dPointLoad){…}

  else if (type == LOAD_TAG_Beam2dThermalAction) {
  dataMix = data*loadFactor;
  counterTemperature = 0;
  q0Temperature[0] = 0.0;
  q0Temperature[1] = 0.0;
  q0Temperature[2] = 0.0;
  //const Matrix &pts = quadRule.getIntegrPointCoords(numSections);
  //const Vector &wts = quadRule.getIntegrPointWeights(numSections);
  double xi[maxNumSections];
  beamInt->getSectionLocations(numSections, L, xi);
  double wt[maxNumSections];
  beamInt->getSectionWeights(numSections, L, wt);
  // Loop over the integration points
  for (int i = 0; i < numSections; i++) {
    int order = theSections[i]->getOrder();
    const ID &code = theSections[i]->getType();
    //double xi6 = 6.0*pts(i,0);
    double xi6 = 6.0*xi[i];
    // Get section stress resultant
    const Vector &s = theSections[i]->getTemperatureStress(dataMix);
    // Perform numerical integration on internal force
    //q.addMatrixTransposeVector(1.0, *B, s, wts(i));
    double si;
    for (int j = 0; j < order; j++) {
      //si = s(j)*wts(i);
      si = s(j)*wt[i];
```

```
      switch(code(j)) {
      case SECTION_RESPONSE_P:
    q0Temperature[0] += si; break;
      case SECTION_RESPONSE_MZ:
    q0Temperature[1] += (xi6-4.0)*si; q0Temperature[2] += (xi6-
2.0)*si; break;
      default:
    break;
      }}} }
  return 0;
}
```

Figure C.1: Function of "DisBeamColumn2dThermal::addLoad()"

```
const Vector&
DispBeamColumn2dTemperature::getResistingForce()
{
  double xi[maxNumSections];
  beamInt->getSectionLocations(numSections, L, xi);
  double wt[maxNumSections];
  beamInt->getSectionWeights(numSections, L, wt);
  // Zero for integration
  q.Zero();
  this->update();// added to update stress state before getting
resultant stress to consider material softening.
  // Loop over the integration points
  for (int i = 0; i < numSections; i++) {

    int order = theSections[i]->getOrder();
    const ID &code = theSections[i]->getType();
    //double xi6 = 6.0*pts(i,0);
    double xi6 = 6.0*xi[i];
    // Get section stress resultant
    const Vector &s = theSections[i]->getStressResultant();
    // Perform numerical integration on internal force
```

```
    //q.addMatrixTransposeVector(1.0, *B, s, wts(i));
    double si;
    for (int j = 0; j < order; j++) {
      //si = s(j)*wts(i);
      si = s(j)*wt[i];
      switch(code(j)) {
      case SECTION_RESPONSE_P:
    q(0) += si; break;
      case SECTION_RESPONSE_MZ:
    q(1) += (xi6-4.0)*si; q(2) += (xi6-2.0)*si; break;
      default:
    break;
      }} }
    if (counterTemperature == 0) { //first iteration add thermal
load and no thermal load from second iteration because after this if,
counterTemperature++.
    //thermal force = EA*alpha*DeltaT= EA*alpha*(T^i-T^i-1)
        q(0) -= (q0Temperature[0] - q0TemperatureP[0]);
        q(1) -= (q0Temperature[1] - q0TemperatureP[1]);
        q(2) -= (q0Temperature[2] - q0TemperatureP[2]);

        q0TemperatureP[0] = q0Temperature[0];
        q0TemperatureP[1] = q0Temperature[1];
        q0TemperatureP[2] = q0Temperature[2];
    }
    counterTemperature++;
  // Add effects of element loads, q = q(v) + q0
  q(0) += q0[0];
  q(1) += q0[1];
  q(2) += q0[2];
  // Vector for reactions in basic system
  Vector p0Vec(p0, 3);
  P = crdTransf->getGlobalResistingForce(q, p0Vec);
  return P;
}
```

Figure C.2: Function of "DisBeamColumn2dThermal::getResistingForce()"

Appendix D Scripts of restrained beam

```
# 2D bar element, two elements, subjected to thermal expansion;
# left element thermal expansion, right element ambient temperature
# aimed to use the DispBeamColumn2dThermal to conduct thermal analysis
#-------------------------------------------------------------------------------------------------
# Geometric model
#                                                    _________
#            left ele1     3   right ele2          |         |  (2 fibers in section)
#       1 |----------------o---------------|2       |_________| 0.1m
#         |<-------------2m----------->|            |         |
#                                                   |_________|
#                                                       0.1m
#-------------------------------------------------------------------------------------------------
wipe;
file mkdir Data;                    # create data directory
model BasicBuilder -ndm 2 -ndf 3;        # 2 dimension and 3 dofs per node
source DisplayPlane.tcl;                  # procedure for displaying a plane in model
source DisplayModel2D.tcl;             # procedure for displaying 2D perspective of model

#define node
node 1 0 0;
node 2 2 0;
node 3 1 0;

#define boundary condition; nodes 1 and 2 are fixed; node 3 translational free.
fix 1 1 1 1;
fix 2 1 1 1;
fix 3 0 1 1;

#define an elastic material with Tag=1 and E=2e11.
uniaxialMaterial ElasticThermal 1 2e11;

#define fibred section; Two fibres: fiber $yLoc $zLoc $A $matTag
set secTag 1;
  section FiberThermal $secTag   {
      fiber -0.025 0 0.005 1;
      fiber 0.025 0 0.005 1;
   };

#define coordinate transforamtion: geomTransf $type $TransfTag;
#three transformation types can be chosen: Linear, PDelta, Corotational)
```

```
geomTransf Linear 1  ;

#define element: dispBeamColumnThermal $eleTag $iNode $jNode $numIntgrPts $secTag
$TransfTag;
#"numIntgrPts" is the number of integration points along the element;
#"TransfTag" is pre-defined coordinate-transformation;
element dispBeamColumnThermal 1 1 3 3 1 1;
element dispBeamColumnThermal 2 3 2 3 1 1;

#define output
# displacements of free nodes
recorder Node -file Data/DFree3.out -time -node 3 -dof 1 disp;
recorder Node -file Data/RBase1.out -time -node 1 -dof 1 2 3 reaction;       # support reaction

#view the deformation shape
set ViewScale 2;
# display deformed shape, the scaling factor needs to be adjusted for each model
DisplayModel2D DeformedShape $ViewScale ;

#define thermal load (i.e. temperature distribution in section)
#-beamThermal $T1 $LocY1 $T2 LocY2....; two temperature means uniform or linear
temperature distribution
#T1=bottom temperature;T2=top temperature
pattern Plain 2 Linear {
eleLoad -ele 1 -type -beamThermal 1000 -0.05 1000 0.05
eleLoad -ele 2 -type -beamThermal 0 -0.05 0 0.05
};

constraints Plain;
numberer Plain;
system BandGeneral;
test NormDispIncr 1e-8 10 ;
algorithm Newton;
integrator LoadControl 0.1;
analysis Static;
analyze 10;
loadConst -time 0.0
```

Appendix E Scripts of beam with finite end restraint

```
# Single steel beam subjected to UDL and thermal gradient
# total 8 elements for 6m beam ;
# distributed load UDL=1000N/m; thermal gradient is linear cross section;
# thermal gradient is define by Tbot and Ttop;
# 8 fibers along the height 0.2m of the beam section;
# Material is temperature-dependent elastic; E=2E11 for ambient temperature(20)
# E is referred to the EN 1993 pt 1-2-1.
# expansion coefficient is constant with value 12e-6
# SI unit i.e. meter, newton
#-----------------------------------------------------------------------------
# Geometric model                                        _____
#                                                       |     |
#         |--------------------------------------|      |     | 0.2
#       21&1   2   3   4   5   6   7   8   9&22   |____|
#         |<-----------------6m------------------>|       0.1
#
# SET UP ---------------------------------------------------------------------
wipe;
model BasicBuilder -ndm 2 -ndf 3;       # 2 dimension and 3 dofs per node
set dataDir Data;                  # set up name of data directory (can remove this)
file mkdir $dataDir;                    # create data directory
source DisplayPlane.tcl
source DisplayModel2D.tcl

# nodal coordinates:
node 1 0. 0;                node 2 0.75 0.;  node 3 1.5 0;              node 4 2.25 0; node 5 3 0;
node 6 3.75 0;
node 7 4.5 0;               node 8 5.25 0.; node 9 6 0;

#node 21, 22 for zerolength element to model spring(restraint)
node 21 0 0;
node 22 6. 0.;

# Single point constraints -- Boundary Conditions
fix 1 1 1 0;
```

```
fix 9 1 1 0;
fix 21 1 1 1;
fix 22 1 1 1;

#define an elastic material with Tag=1 and E=2e11.
set matTag 1;
uniaxialMaterial ElasticThermal $matTag 2e11;

#define fibred section; define fibre: fiber $yLoc $zLoc $A $matTag
# origin of section is the center of rectangular
 set SecTag 1;
 section FiberThermal $SecTag  {
#8 fibers
        fiber 0.0125 0 0.0025 $matTag;
                fiber 0.0375 0 0.0025 $matTag;
                fiber 0.0625 0 0.0025 $matTag;
                fiber 0.0875 0 0.0025 $matTag;
        fiber -0.0125 0 0.0025 $matTag;
                fiber -0.0375 0 0.0025 $matTag;
                fiber -0.0625 0 0.0025 $matTag;
                fiber -0.0875 0 0.0025 $matTag;
 };

# define geometric transformation: Linear, PDelta, Corotational
set TransfTag 1;
geomTransf Linear $TransfTag;
#geomTransf PDelta $TransfTag;
#geomTransf Corotational $TransfTag;

#define beam element: dispBeamColumnThermal $eleTag $iNode $jNode $numIntgrPts $secTag
$TransfTag;
#"numIntgrPts" is the number of integration points along the element;
#"TransfTag" is pre-defined coordinate-transformation;
set numIntgrPts 3;
element dispBeamColumnThermal 1 1 2 $numIntgrPts $SecTag $TransfTag;
element dispBeamColumnThermal 2 2 3 $numIntgrPts $SecTag $TransfTag;
element dispBeamColumnThermal 3 3 4 $numIntgrPts $SecTag $TransfTag;
```

```
element dispBeamColumnThermal 4 4 5 $numIntgrPts $SecTag $TransfTag;
element dispBeamColumnThermal 5 5 6 $numIntgrPts $SecTag $TransfTag;
element dispBeamColumnThermal 6 6 7 $numIntgrPts $SecTag $TransfTag;
element dispBeamColumnThermal 7 7 8 $numIntgrPts $SecTag $TransfTag;
element dispBeamColumnThermal 8 8 9 $numIntgrPts $SecTag $TransfTag;

# first define material for zeroLength element;
# elasticity modulus is equal to stiffness of spring
# dir 6 is the rotation around z; dir 1 is the translational dof
uniaxialMaterial Elastic 2 3e6;            #translational spring
uniaxialMaterial Elastic 3 6.67e8;         # rotational spring
#define element for traslational&rotational spring
element zeroLength 121 1 21 -mat 2 -dir 6;    #rotational spring
element zeroLength 922 9 22 -mat 2 -dir 6;    #rotational spring
element zeroLength 933 9 22 -mat 3 -dir 1;    #translational spring

# define output
recorder Node -file Data/DFree5.out -time -node 5 -dof 2 disp;          # displacements of free
nodes
recorder Node -file Data/DFree9.out -time -node 9 -dof 1 disp;
recorder Node -file Data/RBase.out -time -node 1 -dof 1 reaction;             # support reaction
recorder Node -file Data/RBase21.out -time -node 21 22 -dof 1 2 3 reaction;
      #support reaction

#define load
# first add a UDL
set UDL -1000;
pattern Plain 1 Linear {
eleLoad -ele 1 -type -beamUniform $UDL 0. 0.;
eleLoad -ele 2 -type -beamUniform $UDL 0. 0.;
eleLoad -ele 3 -type -beamUniform $UDL 0. 0.;
eleLoad -ele 4 -type -beamUniform $UDL 0. 0.;
eleLoad -ele 5 -type -beamUniform $UDL 0. 0.;
eleLoad -ele 6 -type -beamUniform $UDL 0. 0.;
eleLoad -ele 7 -type -beamUniform $UDL 0. 0.;
eleLoad -ele 8 -type -beamUniform $UDL 0. 0.;
```

```
}
constraints Plain;
numberer Plain;
system BandGeneral;
test NormDispIncr 1e-8 100 ;
algorithm Newton;
integrator LoadControl 1
analysis Static;
analyze 1;
loadConst -time 0.0

#define thermal load (i.e. temperature distribution in section)
#-beamThermal $T1 $LocY1 $T2 LocY2....; two temperature means uniform or linear temperature distribution
# T1 is the bottom temp of beam and T2 is the top
# Y1 is the coordinate of bottom of beam section; Y2 is for top
# the temperature will be interpolated along the section
set T1 1000;set T2 0;
set Y1 -0.1;set Y2 0.1;
pattern Plain 2 Linear {
eleLoad -ele 1 -type -beamThermal $T1 $Y1 $T2 $Y2 ;
eleLoad -ele 2 -type -beamThermal $T1 $Y1 $T2 $Y2 ;
eleLoad -ele 3 -type -beamThermal $T1 $Y1 $T2 $Y2 ;
eleLoad -ele 4 -type -beamThermal $T1 $Y1 $T2 $Y2 ;
eleLoad -ele 5 -type -beamThermal $T1 $Y1 $T2 $Y2 ;
eleLoad -ele 6 -type -beamThermal $T1 $Y1 $T2 $Y2 ;
eleLoad -ele 7 -type -beamThermal $T1 $Y1 $T2 $Y2 ;
eleLoad -ele 8 -type -beamThermal $T1 $Y1 $T2 $Y2 ;
}
integrator LoadControl 0.01
analysis Static;
analyze 100;
loadConst -time 0.0
```

Appendix F DruckerPragerThermal

```
double
DruckerPragerThermal::setThermalTangentAndElongation(double &TempT,
double&ET, double&Elong)
{
  // TempT is actural temperature; ambient temperature is 20;
  // ET is temperature dependent elasticity modulus.
  // Elong is the thermal elongation at elevated temperature
  // The temperature dependent properties are based on concrete with
      Calcareous aggregates EN 1992 part 1-2-1, Page20

    double feT;//elastic compressive strength
    double ee0; // elastic strain corresponding to feT
    double fcT;// compressive strength at elevated temperature
double ecT;//strain at fcT;
fc0 = 5.5*msigma_y0;// compressive strength at ambient temperature
    ec0 = 0.0025; //strain at fcT at ambient temperature

   if (TempT <= 20) {

        ET = 2*fc0/ec0;

  }
       else if (TempT <= 100) {

          mKref = mKref0/(0.0025 + (0.004-0.0025)*(TempT -
20)/80)*ec0;
          mGref = mGref0/(0.0025 + (0.004-0.0025)*(TempT -
20)/80)*ec0;;
          mK = mKref;
          mG = mGref;
          msigma_y = msigma_y0;
          fcT = fc0;
          ecT = (0.0025 + (0.004-0.0025)*(TempT - 20)/80);
          ET = 2*fcT/ecT;
          ee0 = 1/3*fcT/ET;
      mHard = 2/3*fcT/(ecT-ee0)/2.8;

 }
   else if (TempT <= 200) {

      mKref = mKref0*(1 - (TempT - 100)*0.05/100)/(0.0040 + (0.0055-
                        0.0040)*(TempT - 100)/100)*0.0025;
      mGref = mGref0*(1 - (TempT - 100)*0.05/100)/(0.0040 + (0.0055-
                        0.0040)*(TempT - 100)/100)*0.0025;
      mK = mKref;
          mG = mGref;
          msigma_y = msigma_y0*(1 - (TempT - 100)*0.05/100);
          fcT = fc0*(1 - (TempT - 100)*0.05/100);
          ecT = (0.0040 + (0.0055-0.0040)*(TempT - 100)/100);
          ET = 2*fcT/ecT;
          ee0 = 1/3*fcT/ET;
      mHard = 2/3*fcT/(ecT-ee0)/2.8;
```

```
}
 else if (TempT <= 300) {


    mKref  =  mKref0*(0.95  -  (TempT  -  200)*0.1/100)/(0.0055  +
                      (0.0070-0.0055)*(TempT - 200)/100)*0.0025;
    mGref = mGref0*(0.95 - (TempT - 200)*0.1/100)/(0.0055 +
                      (0.0070-0.0055)*(TempT - 200)/100)*0.0025;
    mK = mKref;
    mG = mGref;
        msigma_y = msigma_y0*(0.95 - (TempT - 200)*0.1/100);
        fcT = fc0*(0.95 - (TempT - 200)*0.1/100);
        ecT = (0.0055 + (0.0070-0.0055)*(TempT - 200)/100);
        ET = 2*fcT/ecT;
        ee0 = 1/3*fcT/ET;
    mHard = 2/3*fcT/(ecT-ee0)/2.8;

}
 else if (TempT <= 400) {

    mKref  =  mKref0*(0.85  -  (TempT  -  300)*0.1/100)/(0.0070  +
                      (0.0100-0.0070)*(TempT - 300)/100)*0.0025;
    mGref  =  mGref0*(0.85  -  (TempT  -  300)*0.1/100)/(0.0070  +
                      (0.0100-0.0070)*(TempT - 300)/100)*0.0025;
        mK = mKref;
        mG = mGref;
        msigma_y = msigma_y0*(0.85 - (TempT - 300)*0.1/100);
        fcT = fc0*(0.85 - (TempT - 300)*0.1/100);
        ecT = (0.0070 + (0.0100-0.0070)*(TempT - 300)/100);
        ET = 2*fcT/ecT;
        ee0 = 0.33*fcT/ET;
    mHard = 0.66*fcT/(ecT-ee0)/2.8;
}
 else if (TempT <= 500) {

    mKref  =  mKref0*(0.75  -  (TempT  -  400)*0.15/100)/(0.0100  +
                      (0.0150-0.0100)*(TempT - 400)/100)*0.0025;
    mGref  =  mGref0*(0.75  -  (TempT  -  400)*0.15/100)/(0.0100  +
                      (0.0150-0.0100)*(TempT - 400)/100)*0.0025;
        mK = mKref;
        mG = mGref;
        msigma_y = msigma_y0*(0.75 - (TempT - 400)*0.15/100);
        fcT = fc0*(0.75 - (TempT - 400)*0.15/100);
        ecT = (0.0100 + (0.0150-0.0100)*(TempT - 400)/100);
        ET = 2*fcT/ecT;
        ee0 = 1/3*fcT/ET;
    mHard = 2/3*fcT/(ecT-ee0)/2.8;
}
 else if (TempT <= 600) {

    mKref  =  mKref0*(0.60  -  (TempT  -  500)*0.15/100)/(0.0150  +
                      (0.0250-0.0150)*(TempT - 500)/100)*0.0025;
    mGref  =  mGref0*(0.60  -  (TempT  -  500)*0.15/100)/(0.0150  +
                      (0.0250-0.0150)*(TempT - 500)/100)*0.0025;
        mK = mKref;
        mG = mGref;
        msigma_y = msigma_y0*(0.60 - (TempT - 500)*0.15/100);
```

```
        fcT = fc0*(0.60 - (TempT - 500)*0.15/100);
        ecT = (0.0150 + (0.0250-0.0150)*(TempT - 500)/100);
        ET = 2*fcT/ecT;
        ee0 = 1/3*fcT/ET;
      mHard = 2/3*fcT/(ecT-ee0)/2.8;
}
 else if (TempT <= 700) {

        mKref = mKref0*(0.45 - (TempT - 600)*0.15/100)/10;
        mGref = mGref0*(0.45 - (TempT - 600)*0.15/100)/10;
        mK = mKref;
        mG = mGref;
        msigma_y = msigma_y0*(0.45 - (TempT - 600)*0.15/100);
        fcT = fc0*(0.45 - (TempT - 600)*0.15/100);
        ecT = 0.025;
        ET = 2*fcT/ecT;
        ee0 = 1/3*fcT/ET;
    mHard = 2/3*fcT/(ecT-ee0)/2.8;
}
 else if (TempT <= 800) {

        mKref = mKref0*(0.30 - (TempT - 700)*0.15/100)/10;
        mGref = mGref0*(0.30 - (TempT - 700)*0.15/100)/10;
        mK = mKref;
        mG = mGref;
        msigma_y = msigma_y0*(0.30 - (TempT - 700)*0.15/100);
        fcT = fc0*(0.30 - (TempT - 700)*0.15/100);
        ecT = 0.025;
        ET = 2*fcT/ecT;
        ee0 = 1/3*fcT/ET;
    mHard = 2/3*fcT/(ecT-ee0)/2.8;
}
 else if (TempT <= 900) {

        mKref = mKref0*(0.15 - (TempT - 800)*0.07/100)/10;
        mGref = mGref0*(0.15 - (TempT - 800)*0.07/100)/10;
        mK = mKref;
        mG = mGref;
        msigma_y = msigma_y0*(0.15 - (TempT - 800)*0.07/100);
    fcT = fc0*(0.15 - (TempT - 800)*0.07/100);
        ecT = 0.025;
        ET = 2*fcT/ecT;
        ee0 = 1/3*fcT/ET;
    mHard = 2/3*fcT/(ecT-ee0)/2.8;
}
 else if (TempT <= 1000) {

        mKref = mKref0*(0.08 - (TempT - 900)*0.04/100)/10;
        mGref = mGref0*(0.08 - (TempT - 900)*0.04/100)/10;
        mK = mKref;
        mG = mGref;
        msigma_y = msigma_y0*(0.08 - (TempT - 900)*0.04/100);
        fcT = fc0*(0.08 - (TempT - 900)*0.04/100);
        ecT = 0.025;
        ET = 2*fcT/ecT;
        ee0 = 1/3*fcT/ET;
    mHard = 2/3*fcT/(ecT-ee0)/2.8;
}
```

```
   else  {
      opserr << "the temperature is invalid\n";
  } }
// Calculate thermal elongation
   if (TempT <= 0) {
          ThermalElongation = 0.0;
      }
   else if (TempT <= 700) {
      ThermalElongation = -1.8e-4 + 9e-6*TempT + 2.3e-
                            11*TempT*TempT*TempT;

  }
   else if (TempT <= 1200) {
      ThermalElongation = 14e-3;
  }
   else {
      opserr << "the temperature is invalid\n";
  }

Elong = ThermalElongation;
this->initialize();
this->updateElasticParam( );
this->plastic_integrator( );
  return 0;
}
```

Figure F.1: function "setThermalTangentAndElongation()" in DruckerPragerThermal class

Appendix G Scripts of plate subjected to thermal load

G.1 Cylindrical plate

```
# a 6x30m cyclindrical plate with elastic mateiral; 30m to model the infinite edge
# Geometrically nonlinear shell element ShellMITC4GNThermal used to model plate
# platefiber section is used
# elastic material E=2E11, Poisson ratio v=0.3, thickness t=0.2,
#-----------------------------------------------------------------
#Geometric model
#          _________
#          |        |          ^ y
#          |        |          |
#          |        |          |
#          |        | 30m      |
#          |        |          |-------> x
#          |        |
#          |________|
#              6m
#-----------------------------------------------------------------
wipe;
set dataDir Data;                    # set up name for data directory
file mkdir $dataDir;                 # create data directory

model BasicBuilder -ndm 3 -ndf 6; # 3 dimension and 6 dofs per node
source DisplayPlane.tcl
source DisplayModel3D.tcl

# number of elements per edge; these should both be even.
set nx 16;
set ny 32;

#node at the center of the plate
set mid [expr (($nx+1)*($ny+1)+1) / 2 ];

# create section
```

```
set matTag 2;
set PlateFiberTag 1;
set secTag 1;
set thickness 0.2;
#define 3D elastic material:nDMaterial ElasticIsotropic3D $matTag $E $v 0;
nDMaterial ElasticIsotropic3D $matTag 2e11 0.3 0;
nDMaterial PlateFiberThermal $PlateFiberTag $matTag;

# section: $secTag $matTag $thickness
section PlateFiberThermal $secTag $PlateFiberTag $thickness;

#define shell element (mesh nx X ny)
set eleArgs "1";
block2D $nx $ny 1 1 ShellMITC4GNThermal $eleArgs {
   1   0.   0.  0.
   2   6.   0.  0.
   3   6.   30.  0.
   4   0.   30.  0.
}

#define boundary condition;
#two edges along y direction is translational restrained but rotational free;
#two edges along x direction is free except rotating around x direction;
fixX 0. 1 0 1 0 0 0;
fixX 6. 1 0 1 0 0 0;
fixY 0. 0 0 0 1 0 0;
fixY 30. 0 0 0 1 0 0;

#define output
# displacements of free nodes
recorder Node -file $dataDir/DFreeMid1.out -time -node $mid -dof  3 disp;
recorder Node -file Data/RBase1.out -time -node 1 -dof 1 2 3 4 5 6 reaction;
      #support reaction
#recorder Node -file Data/RBase2.out -time -node 17 -dof 1 2 3 4 5 6 reaction;
#recorder Node -file Data/RBase3.out -time -node 273 -dof 1 2 3 4 5 6 reaction;

#view deformation shape
```

```
set  xPixels 1200;    # height of graphical window in pixels
set  yPixels 800;     # height of graphical window in pixels
set  xLoc1 10; # horizontal location of graphical window (0=upper left-most corner)
set  yLoc1 10; # vertical location of graphical window (0=upper left-most corner)
set ViewScale 0.002;      # scaling factor for viewing deformed shape
DisplayModel3D DeformedShape $ViewScale $xLoc1 $yLoc1  $xPixels $yPixels

#Apply uniformly distributed load(UDL) which is applied by equivalent concentrated load applied on nodes
pattern Plain 1 Linear {
set UDL -1e6;
set Lx 6; #edge length
set Ly 30;
set AreaXY [expr $Lx*$Ly];
set ForcePerEle [expr $UDL*$AreaXY/$nx/$ny];
set CorLoad [expr $ForcePerEle/4];
set YedgeLoad [expr $ForcePerEle/2];
set XedgeLoad [expr $ForcePerEle/2];
set InnerLoad [expr $ForcePerEle];
#add corner node
set corner1 1;
set corner2 [expr $nx+1];
set corner3 [expr $ny*($nx+1)];
set corner4 [expr ($ny+1)*($nx+1)];
load $corner1  0.0  0  $CorLoad  0.0   0.0  0.0;
load $corner2  0.0  0  $CorLoad  0.0   0.0  0.0;
load $corner3  0.0  0  $CorLoad  0.0   0.0  0.0;
load $corner4  0.0  0  $CorLoad  0.0   0.0  0.0;
#add Yedge node
set mY [expr $nx+2];
while {$mY <=(2*$nx+2)} {
set nY $mY;
while {$nY <=($ny*$nx+$ny)} {
load $nY  0.0  0   $YedgeLoad   0.0   0.0  0.0;
incr nY [expr $nx+1];
};
```

```
incr mY $nx;
}
#add Xedge node
set mX 2;
while {$mX <=$nx} {
set nX $mX;
while {$nX <=(($ny+1)*($nx+1)-1)} {
load $nX  0.0  0   $XedgeLoad   0.0   0.0  0.0
incr nX [expr ($nx+1)*$ny]
};
incr mX 1;
}
# add load to inner node
set i [expr $nx+3];
while {$i <=(2*$nx+1)} {
set j $i;
while {$j <=($ny*$nx+$ny-1)} {
load $j  0.0  0   $InnerLoad   0.0   0.0  0.0
incr j [expr $nx+1]
};
incr i 1;
};
};
constraints Plain;
numberer Plain;
system BandGeneral;
test NormDispIncr 1e-8 100;
algorithm Newton;
integrator LoadControl 1;
analysis Static;
analyze 1;
loadConst -time 0.0;

#define thermal load (i.e. temperature distribution in section)
```

```
#-shellThermal $T1 $LocY1 $T2 LocY2....; two temperature means uniform or linear temperature distribution
#T1=bottom temperature;T2=top temperature
pattern Plain 2 Linear {
set numOfEles [expr $nx*$ny];
set T1 1000;
set T2 0;
set LocY1 [expr -$thickness/2];
set LocY2 [expr $thickness/2];
set i 1;
while {$i <=$numOfEles} {
eleLoad -ele $i -type -shellThermal $T1 $LocY1 $T2 $LocY2;
incr i 1
};
};
set stepInter
integrator LoadControl 0.1;
analysis Static;
analyze 10;
loadConst -time 0.0
```

G.2 Square plate

```
# a 6x6m square plate with elastic mateiral
# Geometrically nonlinear shell element ShellMITC4GNThermal used to model plate
# platefiber section is used
# elastic material E=2E11, Poisson ratio v=0.3, thickness t=0.1,
#-------------------------------------------------------------------
#Geometric model
#          ___________
#          |          |              ^ y
#          |          |   6m         |
#          |__________|              |------> x
#               6m
#-------------------------------------------------------------------
```

```
wipe;
set dataDir Data;                    # set up name for data directory
file mkdir $dataDir;                 # create data directory

model BasicBuilder -ndm 3 -ndf 6; # 3 dimension and 6 dofs per node
source DisplayPlane.tcl
source DisplayModel3D.tcl

# number of elements per edge; these should both be even.
set nx 16;
set ny 16;

#node at the center of the plate
set mid [expr (($nx+1)*($ny+1)+1) / 2 ];

# create section
set matTag 2;
set PlateFiberTag 1;
set secTag 1;
set thickness 0.1;
#define 3D elastic material:nDMaterial ElasticIsotropic3D $matTag $E $v 0;
nDMaterial ElasticIsotropic3D $matTag 2e11 0.3 0;
nDMaterial PlateFiberThermal $PlateFiberTag $matTag;

# section: $secTag $matTag $thickness
section PlateFiberThermal $secTag $PlateFiberTag $thickness;

#define shell element (mesh nx X ny)
set eleArgs "1";
block2D $nx $ny 1 1 ShellMITC4GNThermal $eleArgs {
   1  0.  0.  0.
   2  6.  0.  0.
   3  6.  6.  0.
   4  0.  6.  0.
}

#define boundary condition; four edges translational restrained but rotational free
```

```
fixX 0. 1 1 1 0 0 0;
fixX 6. 1 1 1 0 0 0;
fixY 0. 1 1 1 0 0 0;
fixY 6. 1 1 1 0 0 0;

#define output
# displacements of free nodes
recorder Node -file $dataDir/DFreeMid1.out -time -node $mid -dof  3 disp;
recorder Node -file Data/RBase1.out -time -node 1 -dof 1 2 3 4 5 6 reaction;
      #support reaction
#recorder Node -file Data/RBase2.out -time -node 17 -dof 1 2 3 4 5 6 reaction;
#recorder Node -file Data/RBase3.out -time -node 273 -dof 1 2 3 4 5 6 reaction;

#view deformation shape
set  xPixels 1200;    # height of graphical window in pixels
set  yPixels 800;     # height of graphical window in pixels
set  xLoc1 10; # horizontal location of graphical window (0=upper left-most corner)
set  yLoc1 10; # vertical location of graphical window (0=upper left-most corner)
set ViewScale 0.002;      # scaling factor for viewing deformed shape
DisplayModel3D DeformedShape $ViewScale $xLoc1 $yLoc1  $xPixels $yPixels

#Apply uniformly distributed load(UDL) which is applied by equivalent concentrated load applied on nodes
pattern Plain 1 Linear {
set UDL -1e6;
set Lx 6; #edge length
set Ly 6;
set AreaXY [expr $Lx*$Ly];
set ForcePerEle [expr $UDL*$AreaXY/$nx/$ny];
set CorLoad [expr $ForcePerEle/4];
set YedgeLoad [expr $ForcePerEle/2];
set XedgeLoad [expr $ForcePerEle/2];
set InnerLoad [expr $ForcePerEle];
#add corner node
set corner1 1;
set corner2 [expr $nx+1];
```

```
set corner3 [expr $ny*($nx+1)];
set corner4 [expr ($ny+1)*($nx+1)];
load $corner1  0.0  0  $CorLoad  0.0   0.0  0.0;
load $corner2  0.0  0  $CorLoad  0.0   0.0  0.0;
load $corner3  0.0  0  $CorLoad  0.0   0.0  0.0;
load $corner4  0.0  0  $CorLoad  0.0   0.0  0.0;
#add Yedge node
set mY [expr $nx+2];
while {$mY <=(2*$nx+2)} {
set nY $mY;
while {$nY <=($ny*$nx+$ny)} {
load $nY  0.0  0   $YedgeLoad   0.0   0.0  0.0;
incr nY [expr $nx+1];
};
incr mY $nx;
}
#add Xedge node
set mX 2;
while {$mX <=$nx} {
set nX $mX;
while {$nX <=(($ny+1)*($nx+1)-1)} {
load $nX  0.0  0   $XedgeLoad   0.0   0.0  0.0
incr nX [expr ($nx+1)*$ny]
};
incr mX 1;
}
# add load to inner node
set i [expr $nx+3];
while {$i <=(2*$nx+1)} {
set j $i;
while {$j <=($ny*$nx+$ny-1)} {
load $j  0.0  0   $InnerLoad   0.0   0.0  0.0
incr j [expr $nx+1]
};
incr i 1;
```

```
};
};
constraints Plain;
numberer Plain;
system BandGeneral;
test NormDispIncr 1e-8 100;
algorithm Newton;
integrator LoadControl 1;
analysis Static;
analyze 1;
loadConst -time 0.0;

#define thermal load (i.e. temperature distribution in section)
#-shellThermal $T1 $LocY1 $T2 LocY2....; two temperature means uniform or linear temperature distribution
#T1=bottom temperature;T2=top temperature
pattern Plain 2 Linear {
set numOfEles [expr $nx*$ny];
set T1 1000;
set T2 0;
set LocY1 [expr -$thickness/2];
set LocY2 [expr $thickness/2];
set i 1;
while {$i <=$numOfEles} {
eleLoad -ele $i -type -shellThermal $T1 $LocY1 $T2 $LocY2;
incr i 1
};
};
set stepInter
integrator LoadControl 0.1;
analysis Static;
analyze 10;
loadConst -time 0.0
```

G.3 DruckerPragerThermal material

```
# a 6x6m square plate with DruckerPragerThermal mateiral
# Geometrically nonlinear shell element ShellMITC4GNThermal used to model plate
# platefiber section is used
# compressive strength of concrete fc=48MPa, Poisson ratio v=0.2, thickness t=0.1,
#-------------------------------------------------------------------
#Geometric model
#          ___________
#          |          |              ^ y
#          |          |   6m         |
#          |__________|              |------> x
#                6m
#-------------------------------------------------------------------
wipe;
set dataDir Data;                   # set up name for data directory
file mkdir $dataDir;                # create data directory

model BasicBuilder -ndm 3 -ndf 6; # 3 dimension and 6 dofs per node
source DisplayPlane.tcl
source DisplayModel3D.tcl

# number of elements per edge; these should both be even.
set nx 16;
set ny 16;

#node at the center of the plate
set mid [expr (($nx+1)*($ny+1)+1) / 2 ];

# create section
set matTag 2;
set PlateFiberTag 1;
set secTag 1;
set thickness 0.1;
#define Drucker-Prager material
set k 2.13e10;set G 1.6e10;set sigY 7.44e6;set rho 0.437;set rhoBar 0.437;
set Kinf 0;set K0 0;set delta1 1;set H 0.55e10;set theta 1;set delta2 1;set mDen 2400;
```

```
nDMaterial DruckerPragerThermal $matTag $k $G $sigY $rho $rhoBar $Kinf $K0 $delta1
$delta2 $H $theta $mDen;

nDMaterial PlateFiberThermal $PlateFiberTag $matTag;
# section: $secTag $matTag $thickness
section PlateFiberThermal $secTag $PlateFiberTag $thickness;

#define shell element (mesh nx X ny)
set eleArgs "1";
block2D $nx $ny 1 1 ShellMITC4GNThermal $eleArgs {
   1   0.   0.  0.
   2   6.   0.  0.
   3   6.   6.  0.
   4   0.   6.  0.
}

#boundary condition for plate of one edge under compressive load
#fixY 6.0 1 1 1 1 1 1;

#alternative boundary condition: four edges translational restrained but rotational free
fixX 0. 1 1 1 0 0 0;
fixX 6. 1 1 1 0 0 0;
fixY 0. 1 1 1 0 0 0;
fixY 6. 1 1 1 0 0 0;

#define output
# displacements of free nodes
recorder Node -file $dataDir/DFreeMid1.out -time -node $mid -dof  3 disp;
recorder Node -file Data/RBase1.out -time -node 1 -dof 1 2 3 4 5 6 reaction;
      #support reaction
#recorder Node -file Data/RBase2.out -time -node 17 -dof 1 2 3 4 5 6 reaction;
#recorder Node -file Data/RBase3.out -time -node 273 -dof 1 2 3 4 5 6 reaction;

#view deformation shape
set  xPixels 1200;    # height of graphical window in pixels
set  yPixels 800;     # height of graphical window in pixels
set  xLoc1 10; # horizontal location of graphical window (0=upper left-most corner)
set  yLoc1 10; # vertical location of graphical window (0=upper left-most corner)
```

```
set ViewScale 0.002;      # scaling factor for viewing deformed shape
DisplayModel3D DeformedShape $ViewScale $xLoc1 $yLoc1  $xPixels $yPixels

#Apply uniformly distributed load(UDL) which is applied by equivalent concentrated load applied on nodes
pattern Plain 1 Linear {
#load for one edge under compression--start
#set UDL 48e6;
#set Lx 6;
#set Ly 6;
#set AreaXY [expr $Lx*$Ly];
#set ForcePerEle [expr $UDL*0.1*$Lx/$nx];
#set CorLoad [expr $ForcePerEle/2];
#set InnerLoad [expr $ForcePerEle];
#set endNode [expr 1+$nx];
#set i 2;
#load 1  0.0  $CorLoad   0.   0.0   0.0  0.0
#load $endNode  0.0  $CorLoad   0.   0.0   0.0  0.0
#while {$i <=$nx} {
#load $i  0.0  $InnerLoad   0.   0.0   0.0  0.0
#incr i 1;
#};
#load for one edge under compression--end

#Alternative to apply UDL--start
set UDL -5500;
set Lx 6; #edge length
set Ly 6;
set AreaXY [expr $Lx*$Ly];
set ForcePerEle [expr $UDL*$AreaXY/$nx/$ny];
set CorLoad [expr $ForcePerEle/4];
set YedgeLoad [expr $ForcePerEle/2];
set XedgeLoad [expr $ForcePerEle/2];
set InnerLoad [expr $ForcePerEle];
#add corner node
set corner1 1;
```

```
set corner2 [expr $nx+1];
set corner3 [expr $ny*($nx+1)];
set corner4 [expr ($ny+1)*($nx+1)];
load $corner1  0.0  0  $CorLoad  0.0   0.0  0.0;
load $corner2  0.0  0  $CorLoad  0.0   0.0  0.0;
load $corner3  0.0  0  $CorLoad  0.0   0.0  0.0;
load $corner4  0.0  0  $CorLoad  0.0   0.0  0.0;
#add Yedge node
set mY [expr $nx+2];
while {$mY <=(2*$nx+2)} {
set nY $mY;
while {$nY <=($ny*$nx+$ny)} {
load $nY  0.0  0   $YedgeLoad   0.0   0.0  0.0;
incr nY [expr $nx+1];
};
incr mY $nx;
}
#add Xedge node
set mX 2;
while {$mX <=$nx} {
set nX $mX;
while {$nX <=(($ny+1)*($nx+1)-1)} {
load $nX  0.0  0   $XedgeLoad   0.0   0.0  0.0
incr nX [expr ($nx+1)*$ny]
};
incr mX 1;
}
# add load to inner node
set i [expr $nx+3];
while {$i <=(2*$nx+1)} {
set j $i;
while {$j <=($ny*$nx+$ny-1)} {
load $j  0.0  0   $InnerLoad   0.0   0.0  0.0
incr j [expr $nx+1]
};
```

```
incr i 1;
};
#Alternative to apply UDL--start
};
constraints Plain;
numberer Plain;
system BandGeneral;
test NormDispIncr 1e-8 100;
algorithm Newton;
integrator LoadControl 1;
analysis Static;
analyze 1;
loadConst -time 0.0;

#define thermal load (i.e. temperature distribution in section)
#-shellThermal $T1 $LocY1 $T2 LocY2....; two temperature means uniform or linear temperature distribution
#T1=bottom temperature;T2=top temperature
pattern Plain 2 Linear {
set numOfEles [expr $nx*$ny];
set T1 1000;
set T2 0;
set LocY1 [expr -$thickness/2];
set LocY2 [expr $thickness/2];
set i 1;
while {$i <=$numOfEles} {
eleLoad -ele $i -type -shellThermal $T1 $LocY1 $T2 $LocY2;
incr i 1
};
};
set stepInter
integrator LoadControl 0.1;
analysis Static;
analyze 10;
loadConst -time 0.0
```

Appendix H Materials of Cardington test model in OpenSees

The compressive strength of concrete is 48MPa and the yield stress of steel is 280MPa. The determination of the input parameters for material class DruckerPragerThermal, Concrete02, Steel01 can refer to Section 7.3.3, 3.5.1.1 and 3.4 respectively. The results are shown in Table H.1-3.

k (Pa)	G (Pa)	σ_Y (Pa)	ρ	H	theta	Kinf=Ko	delta1=delta2
2.13×10^{10}	1.6×10^{10}	8.72×10^{6}	0.437	5.44×10^{9}	1	0	1

Table H.1: Properties of concrete in slab (DruckerPragerThermal)

fc (Pa)	epsc0	$f$cu	epscu	rat	ft	Ets
-48×10^{7}	-0.0025	-9.6×10^{6}	-0.02	0.1	4.8×10^{6}	2.4×10^{9}

TableH.2: Properties of concrete (Concrete02)

fy (Pa)	E0	rat
2.8×10^{8}	2.0×10^{11}	0.002

TableH.3: Properties of steel (Steel01)

Appendix I Materials of Roorkee RC frame test in OpenSees

The compressive and tensile strengths of concrete at ambient are 34MPa and 3.4MPa, respectively. The yield stress of reinforcing steel is 450MPa. The corresponding input parameters for the material classes DruckerPragerThermal, Concrete02, Steel01, Pinching4MaterialThermalConcrete and Pinching4MaterialThermalSteel are listed in Tables I.1-5 respectively. The determination of these parameters can refer to Section 7.3.3, 3.5.1, 3.4 and 3.5.1.2 respectively.

k (Pa)	G (Pa)	σ_Y (Pa)	ρ	H	theta	Kinf=Ko	delta1=delta2
1.51×10^{10}	1.13×10^{10}	5.27×10^{6}	0.437	3.89×10^{9}	1	0	1

Table I.1: Properties of concrete in slab (DruckerPragerThermal)

fc (Pa)	epsc0	$f$cu	epscu	rat	ft	Ets
-3.4×10^{7}	-0.0025	-6.8×10^{6}	-0.02	0.1	3.4×10^{6}	1.7×10^{9}

Table I.2: Properties of concrete in beam/column (Concrete02)

fy (Pa)	E0	rat
4.5×10^{8}	2.0×10^{11}	0.002

Table I.3: Properties of reinforcement (Steel01)

ePf1 (Pa)	ePf2 (Pa)	ePf3 (Pa)	ePf4 (Pa)	eNf1(Pa)	eNf2(Pa)	eNf3(Pa)	eNf4(Pa)
3.4×10^{6}	3.4×10^{6}	3.4×10^{6}	0	-3.4×10^{7}	-3.4×10^{7}	-3.4×10^{7}	0
ePd1	ePd2	ePd3	ePd4	eNd1	eNd2	eNd3	eNd4
0.000125	0.000125	0.000125	0.002125	-0.0025	-0.0025	-0.0025	-0.02

rDispP=rDispN	rForceP=rForceN	uForceP=uForceN	gE	dmgType
0.5	0.25	0.05	10	Cycle

gK1	gK2	gK3	gK4	gKLim
1.0	0.2	0.3	0.2	0.9
gD1	gD2	gD3	gD4	gDLim
0.5	0.5	2.0	2.0	0.5
gF1	gF2	gF3	gF4	gFLim
1.0	0.0	1.0	1.0	0.9

Table I.4 Properties of pinching concrete in the joint (Pinching4MaterialThermalConcrete)

ePf1 (Pa)	ePf2 (Pa)	ePf3 (Pa)	ePf4 (Pa)	eNf1(Pa)	eNf2(Pa)	eNf3(Pa)	eNf4(Pa)
4.5×10^{8}	4.5×10^{8}	4.5×10^{8}	0	-4.5×10^{8}	-4.5×10^{8}	-4.5×10^{8}	0
ePd1	ePd2	ePd3	ePd4	eNd1	eNd2	eNd3	eNd4
0.00225	0.02	0.15	0.2	-0.00225	-0.02	-0.15	-0.2

Remarks: the other parameters are the same with Table 8.7

Table I.5: Properties of pinching reinforcement in the joint (Pinching4MaterialThermalSteel)