

https://dspace.org/

Simplified method to nonlinear analysis of reinforced concrete in pure flexure

Roberts, Graham Dean

2015-03-22
http://hdl.handle.net/10539/18562

Downloaded from a DSpace repository. https://dspace.org/

SIMPLIFIED METHOD TO NONLINEAR ANALYSIS OF REINFORCED CONCRETE IN PURE FLEXURE

Graham Dean Roberts

A research report submitted to the Faculty of Engineering and the Built Environment, of the University of the Witwatersrand, in partial fulfilment of the requirements for the degree of Master of Science in Engineering.

Johannesburg 2015

DECLARATION

I declare that this research report is my own unaided work. It is being submitted to the Degree of Master of Science to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other University.

……

Graham Dean Roberts

……. day of …… year ……..

ABSTRACT

The use of the finite element method in the design of reinforced concrete slabs and beams has become a generally accepted practice in recent times and when designing structural members, both ultimate and serviceability limit states are required to be considered in the consequent analyses. The nonlinear analysis of reinforced concrete, using plates and shells, may be defined into two broader categories with the first being the layered approach and the second being the effective stiffness approach.

Common commercial finite element software do not all provide the facilities for the nonlinear analysis of reinforced concrete beams and slabs. Although there are currently nonlinear models provided through literature these can be seen as complex to certain engineers and only applicable to the specialist engineer able to understand and implement the theory correctly.

The more complex methods are also aimed at predicting the wider range of failure mechanisms. Unless carrying out forensic engineering, the design engineer might not be interested in the actual failure load but rather, dependant on design philosophy, a cautious yield line load or similar.

This report presents a simplified method, based on an effective stiffness approach, to the nonlinear analysis of reinforced concrete slabs and beams for serviceability and ultimate limit states. The method allows for the use of simple design equations familiar to all structural engineers undertaking reinforced concrete designs.

Using the finite element method, plate elements and simplified constitutive properties a nonlinear algorithm is developed which results in the accurate estimation of the displacements during loading as well as a design ultimate loading. The proposed method is intended for reinforced concrete beams and slabs under transverse loading leading to bending with no axial forces present.

The proposed model and nonlinear algorithm is validated against four experimental case studies which show the accuracy and relevance of the given nonlinear solution. The results provide evidence that the proposed nonlinear model is valid for all loading and boundary conditions considered. The application can be for displacement serviceability checks or the ultimate load design of a slab or beam. The nonlinear model and algorithm presented can be easily integrated into a commercial finite element package, with API capabilities, for use in the design of reinforced concrete slabs and beams.

ACKNOWLEDGEMENTS

To my beautiful and patient wife Stephanie for giving me the opportunity to dedicate the time to my studies throughout all these years and giving me all the support in the world. You are the infinite element in which my world is formulated.

To my loving and supporting family I would just like to thank you for always giving me the upmost support and thought provoking response to any idea I might have.

My deepest gratitude is given to Dr. Kuinian Li for his encouragement, guidance and always allowing me the opportunity to convey my concepts and together, through constructive analytical discussion, reach an effective conclusion.

TABLE OF CONTENTS

DECLARATION ..... **II**

ABSTRACT ..... **III**

ACKNOWLEDGEMENTS ..... **IV**

TABLE OF CONTENTS ..... **V**

LIST OF FIGURES ..... **VII**

LIST OF TABLES ..... **IX**

LIST OF SYMBOLS ..... **X**

CHAPTER 1: INTRODUCTION **1**

1.1 OBJECTIVES OF THE RESEARCH 1
1.2 SCOPE OF THE RESEARCH 2
1.3 RESTRICTIONS OF THE STUDY 2
1.4 OUTLINE OF THE RESEARCH REPORT 2

CHAPTER 2: BACKGROUND AND LITERATURE REVIEW **4**

2.1 PROPERTIES INFLUENCING REINFORCED CONCRETE BEHAVIOUR 4
2.1.1 Concrete 4
2.1.2 Reinforcing Steel 7
2.1.3 Bond between Concrete and Reinforcement 10
2.2 NONLINEAR ANALYSIS USING PLATES AND SHELLS 11

CHAPTER 3: FEM OVERVIEW AND PLATE ELEMENT FORMULATION **15**

3.1 OVERVIEW OF THE GENERALISED FINITE ELEMENT METHOD 15
3.2 PLATE ELEMENT FORMULATION 16

CHAPTER 4: SIMPLIFIED NONLINEAR MATERIAL PROPERTIES **22**

4.1 RATIONALE 22
4.2 MATERIAL PROPERTY CALCULATIONS 22

CHAPTER 5: MATLAB PROGRAM & NONLINEAR SOLVER **27**

5.1 ANALYSIS PROGRAM BASIS 27
5.2 SIMPLIFIED NONLINEAR SOLVER 30

CHAPTER 6: TEST CASE STUDIES **40**

6.1 EXPERIMENTAL TEST CASE OVERVIEW 40
6.2 PROPERTIES OF CASE STUDIES 40
6.2.1 Polak SM1 40
6.2.2 McNeice 43
6.2.3 Ghomein & MacGregor 45
6.2.4 Aghayere & MacGregor 48

CHAPTER 7: RESULTS 51

7.1 INCREMENTAL ASSESSMENT OF NONLINEAR ALGORITHM 51
7.2 POLAK RESULTS 55
7.3 MCNEICE RESULTS 56
7.4 GHOMEIN & MACGREGOR C1 SALB RESULTS 58
7.5 AGHAYERE & MACGREGOR A3 SLAB RESULTS 60

CHAPTER 8: DISCUSSION AND CONCLUSION 62

REFERENCES 64

APPENDIX A: MATERIAL PROPERTY CALCULATIONS 67

APPENDIX B: MATLAB CODE 76

LIST OF FIGURES

Figure 1: Typical compressive strength gain in concrete 5
Figure 2: Typical uniaxial plain concrete stress-strain relationship 6
Figure 3: Elastic modulus of concrete graphical definition 7
Figure 4: Stiffness behaviour of steel reinforcement 8
Figure 5: Stages of stiffness in steel reinforcement 9
Figure 6: (a) Bilinear, (b) Elastic perfectly plastic stress and strain relationship 9
Figure 7: Tension stiffening mechanism 10
Figure 8: Nodal degrees of freedom [5] 16
Figure 9: Shear deformations 17
Figure 10: Deformation assumptions of plate including shear deformation, figure 5.25 [5] 17
Figure 11: Shape interpolation functions, figure 5.4 [5] 18
Figure 12: Typical Moment Curvature Diagram 23
Figure 13: Typical Calculated Stress vs. Strain Diagram 24
Figure 14: Relevant strain distribution to calculate curvature 25
Figure 15: Three major design strain states to be considered 25
Figure 16: Discretisation for McNeice, Ghomein & MacGregor and Aghayere & MacGregor slabs 29
Figure 17: Reinforcement orientation 30
Figure 18: Johansen's stepped yield criteria discretisation 31
Figure 19: (a) Applied section moments, (b) Resisting section moments 31
Figure 20: Acceptance criteria and load increase / reduction 36
Figure 21: Small system nonlinear solver algorithm 38
Figure 22: Large system nonlinear solver algorithm 39
Figure 23: Polak and Vecchio (1994) SM1 41
Figure 24: SM1 finite element model restraints 41
Figure 25: Slab SM1 stress-strain diagram 42
Figure 26: McNeice (1967) [27] slab No. 1 43
Figure 27: McNeice finite element model restraints 44
Figure 28: McNeice slab stress-strain diagram X & Y directions 45
Figure 29: Ghomein & MacGregor (1992) [16] C1 slab 46
Figure 30: C1 and A3 finite element model restraints 46
Figure 31: C1 slab stress-strain diagram X directions 47
Figure 32: C1 slab stress-strain diagram Y directions 48
Figure 33: Aghayere & MacGregor (1990) [3] A3 slab 49
Figure 34: A3 slab stress-strain diagram X directions 50
Figure 35: A3 slab stress-strain diagram Y directions 50
Figure 36: Polak slab Gauss points 52
Figure 37: Stress-strain comparative analysis vs input – Polak slab 52
Figure 38: McNeice slab Gauss points, Reference Figure 16 and Figure 27 54
Figure 39: Stress-strain comparative analysis vs input – McNeice slab 55
Figure 40: Polak SM1 slab moment-curvature comparison 56

Figure 41: McNeice node 6 load vs displacement 57
Figure 42: McNeice slab cracking in the x and y directions 58
Figure 43: McNeice yield line development pattern and extent 58
Figure 44: C1 slab centre point load vs deflection 59
Figure 45: Ghomein & Macgregor slab directional cracking 60
Figure 46: A3 slab centre point load vs deflection 61
Figure 47: Aghayere & Macgregor slab directional cracking..... 61

LIST OF TABLES

Table 1: Geometric Discretisation Convergence 28
Table 2: Experimental Data Summary 40
Table 3: Slab SM1 Material Properties 42
Table 4: McNeice Slab Material Properties 44
Table 5: C1 Slab Material Properties 47
Table 6: A3 Slab Material Properties 49
Table 7: Incremental E-moduli and Poisson's Ratio – Polak slab 53
Table 8: Incremental strain, stress and moment results – Polak slab 53
Table 9: Cumulative design strain, stress and moment with variance – Polak slab 53
Table 10: Incremental E-moduli and Poisson's Ratio – McNeice slab 54
Table 11: Incremental strain, stress and moment results – McNeice slab 54
Table 12: Cumulative design strain, stress and moment with variance – McNeice slab 55

LIST OF SYMBOLS

A_{sc}	Area of steel in compression
A_{st}	Area of steel in tension
B	Gross geometric beam / slab breadth
B_b	Strain interpolation matrix for bending
B_s	Strain interpolation matrix for shear
C_b	Bending constitutive material matrix
C_s	Bending constitutive material matrix
d	Depth to tension steel
d ‘	Depth to compression steel
E	Elastic modulus
E_c	Modulus of elasticity of concrete
E_s	Modulus of elasticity of steel
F_{cu}	Characteristic compressive cube strength of concrete
f'_c	Characteristic compressive cylindrical strength of concrete
f'_t	Tensile strength of concrete, (for this research equal to modulus of rupture, f_r)
f_y	Yield strength of steel
G	Shear modulus
h	Gross geometric beam depth
I_g	Second moment of area of the gross cross-section
J	Jacobian matrix
k_e	Element stiffness matrix
K_{sys}	System stiffness matrix of iteration i
K_g	Modification stiffness matrix of iteration i
K_f	Pre-modification stiffness matrix of iteration i
L	Geometric beam / slab Length
L_n	Load vector for a given iteration
M_{cr}	Cracking moment of the section
M_d	Design moment

$M_{section}$	Section moment capacity at any given strain state
M_y	Yielding moment of the section
M_{ult}	Ultimate moment of the section
x	Depth for top of the section to the neutral axis
w	Vertical displacement
β_x	Rotation in the xz plane
β_y	Rotation in the yz plane
ϕ	Curvature
κ	Shear correction factor 5/6
ε_{con}	Strain control point as defined by a typical stress-strain diagram
ε_{cc}	Compressive concrete strain at the extreme fibre
ε_{sc}	Compressive steel strain
ε_{st}	Tensile steel strain
ε_{cr}	Cracking strain point
ε_{eq}	Equivalent strain
ε_{tx1}	Design strain due to bending action in the x direction
ε_{ty1}	Design strain due to bending action in the y direction
ε_{tx2}	Design strain due to twisting action proportional to the x direction
ε_{ty2}	Design strain due to twisting action proportional to the y direction
ε_{tx}	Total design strain in the x direction
ε_{ty}	Total design strain in the y direction
ε_{ult}	Ultimate strain point
γ	Shear strain
τ_b	Stress due to bending
τ_s	Stress due to shear
σ_{cr}	Modulus of rupture
σ_y	Yield stress point
σ_{ult}	Ultimate stress point
ν	Poisson's Ratio
η	Modular ratio of E_s / E_c

CHAPTER 1: INTRODUCTION

1.1 OBJECTIVES OF THE RESEARCH

Much of current reinforced concrete slab and beam design is undertaken using results obtained through analysis utilising the finite element method with plate or shells elements. In the more common cases, these analyses comprise linear elastic material and linear geometry. As the reinforced concrete members are designed using ultimate loading it can be expected that the flexure members would undergo a certain degree of cracking.

As the structural member undergoes cracking, the flexural stiffness changes and nonlinear behaviour is experienced. There are however commercial finite element software currently available that carry out linear elastic analysis combined with an empirically derived reduced stiffness value based on a cracked section as per design codes.

Nonlinear numerical methods have been established to estimate the nonlinear behaviour of reinforced concrete using plate or shell elements. These methods can be split into two approaches namely the layered approach or effective stiffness approach. The layered approach is seen as numerically complex for the general design engineer to practically be able to use in everyday works. The effective stiffness approach may not be as precise as the layered approach but is efficient, less complex and provides accurate results for design purposes.

The approach proposed in this research report, based on the effective stiffness method, is one which can be used to obtain accurate rotations and displacements based on a user input stress-strain relationship, calculated from the reinforced concrete section configuration. This proposed method is seen as practical and would enable the appropriate simulation of the nonlinear behaviour of conventionally shaped reinforced concrete slabs / beams throughout the loading process.

The proposed method introduces the use of common design and structural mechanics equations which all structural engineers are extensively familiar with and can easily employ. A nonlinear algorithm is established so that the proposed method can be easily implemented into current linear elastic commercial finite element software with API capabilities.

The method established through this research would enable a wider audience of structural engineer to carry out the nonlinear analysis of reinforced concrete slabs and beams with confidence and with full knowledge of the numerical process, design assumptions and analysis limitations.

1.2 SCOPE OF THE RESEARCH

The research will cover aspects relevant to the understanding of where the proposed method fits into the greater picture of nonlinear analysis of reinforced concrete using plate and shells.

Following this, the technical aspects of how the proposed method is formulated is covered, including the plate element formulation and theoretical approach to the development of the proposed nonlinear material model.

To validate the proposed method, well known experimental test cases are studied to be analysed using the proposed method. Results obtained in the current research are compared to that of the experimental data. Discussion and interpretation of the results is covered and concluding recommendations provided.

1.3 RESTRICTIONS OF THE STUDY

Firstly it is noted that the research is restricted to reinforced concrete members subjected to flexure with negligible in-plane action. There are two reasons for this, firstly in the formulation of plate elements no consideration to axial stiffness is given and secondly no axial forces are considered present when calculating the nonlinear material properties to be used in analysis.

The proposed method is intended for short term behaviour and does not include creep, shrinkage and temperature effects. The proposed method is also not intended for forensic engineering, but can rather be seen as a method of design assuming tension steel yielding in flexure with no in-plane forces considered.

With regards to the nonlinear analysis, material nonlinearity is considered and nonlinear geometry is not considered. This is a valid assumption for standard analysis and design of reinforced concrete beams and slabs.

It should be noted that the proposed method is also intended for and may be applied to continuous slabs and beams although this form of structural system has not been covered in this research and may need to be and validated against experimental data where required.

1.4 OUTLINE OF THE RESEARCH REPORT

The overall objective of the research is to present a simplified method for the nonlinear analysis of reinforced concrete beams and slabs which produces results suitable for design and serviceability checks. The research report is structured such that:

Chapter 2 gives a background to the various material aspects involved in the nonlinear assessment of reinforced concrete. A brief literature review on the topic of nonlinear analysis of reinforced concrete using plates and shells is also presented.

Chapter 3 provides a brief history and background to the generalised finite element method. As in this research plate elements are used to represent the reinforced concrete, the formulation thereof is established.

Chapter 4 provides a detailed description of the derivation of the material properties represented using a stress-strain diagram for use in the present research.

Chapter 5 runs through the methodology used to formulate the finite element program written in MatLab. The development of the yield criteria and nonlinear solver is also explained.

Chapter 6 gives details of the test cases used to validate the proposed nonlinear model. Brief descriptions of the experimentation process as well as important values and observations are provided.

Chapter 7 provides the results obtained from analytical studies carried out using the present proposed nonlinear model.

Chapter 8 comprises a discussion and conclusion of the results obtained from the studies compared to that of the experimental results.

Pre-processing calculations and finite element program code are provided as annexes to the research report.

CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

2.1 PROPERTIES INFLUENCING REINFORCED CONCRETE BEHAVIOUR

The nonlinear behaviour of reinforced concrete is complex in nature and has been under consideration since mid-1960 through till today. When analysing the nonlinear behaviour of reinforced concrete using the finite element method either solid, shell or plate elements are generally used.

The following discussion is intended to give an overview of the aspects influencing to the nonlinear behaviour of reinforced concrete to enable the reader to understand the positioning of proposed method in light of the overall subject. This section will include only a high level discussion of the various aspects of nonlinear behaviour of reinforced concrete.

2.1.1 Concrete

In general concrete comprises three parts namely cement, water, and aggregate. Aggregate is further subdivided into two parts, fine aggregate (sand) and course aggregate (stone). As can be seen there is a number of different materials making up concrete and therefore concrete itself is in actual fact not homogenous but rather heterogeneous. Despite the fact that concrete is in essence not homogenous, it is assumed for convince' sake to be homogenous in analysis with relatively little error in results. This assumption is valid during stages of low to moderate loading prior to the onset of discrete cracking.

2.1.1.1 Creep and shrinkage

In the long term analysis of concrete one would also need to consider the effect of creep and shrinkage. Creep is a time dependent variable which attributed to an increase in strain under a constant applied stress. Shrinkage can broadly be defined as the change in volume due to loss of moisture in the concrete which is also a time dependent variable. This research only considers the short term behaviour of concrete but creep and shrinkage are mentioned as they too hold a large degree of importance in overall long term behaviour of concrete.

2.1.1.2 Temperature

Another factor to consider is thermal effects. Thermal effects could result from the exothermic heat of hydration process, environmental exposure or heat due to fire loading. The thermal properties of concrete include thermal conductivity, specific heat capacity and thermal diffusivity.

Stresses from temperature effects can be due to heat gradients, the concrete structural system's ability to react without cracking and the structural system restraint conditions. As with creep and shrinkage thermal effects are not considered in this research but are mentioned as important factors to consider in design.

2.1.1.3 Concrete compressive strength

Concrete strength is directly dependant on time and in general it can be stated that the strength increases over time. The compressive strength of concrete is dependent on many variables including water:cement ratio, aggregate strength, cement paste aggregate interface, porosity, admixtures etc. As it can be seen that the concrete strength and strength development is dependent on many variables Figure 1 can only depict a typical relationship of concrete compressive strength gain as a ratio of strength to 28 day strength against time.

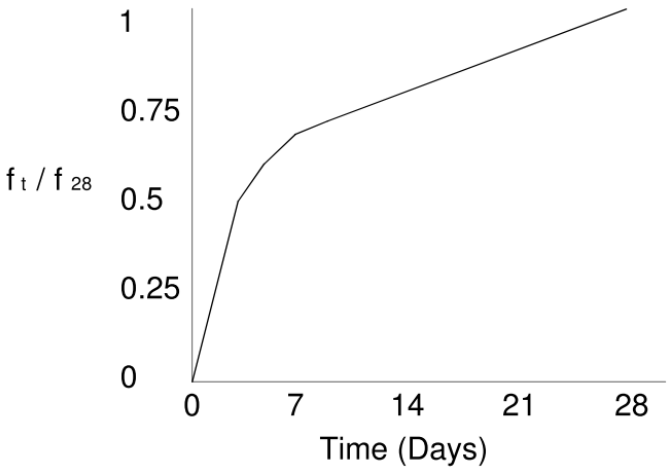


Figure 1: Typical compressive strength gain in concrete

For design purposes the strength of concrete is termed the characteristic strength and is defined as the compressive strength of concrete below which no more than 5% of the valid test results obtained from test cubes / cylinders taken from the same mixture should fall.

A typical tress-strain diagram for concrete under uniaxial loading is given in Figure 2. In compression the concrete behaves linearly till what one could see as yield stress. Following this stiffness softening and permanent plastic deformation occurs and the loading continues till peak stress is reached. Following peak stress compression softening occurs whereby the concrete is significantly damaged but can still take loading till ultimate strain.

Concrete tensile strength is minimal compared to the compressive strength and not ductile in failure. The concrete in tension can be defined by a peak tensile strength and tension softening gradient which then produces the uniaxial tension zone of the stress-strain diagram. The

tensile strength given in the diagram can either be the direct tensile strength or modulus of rupture dependant on the required application.

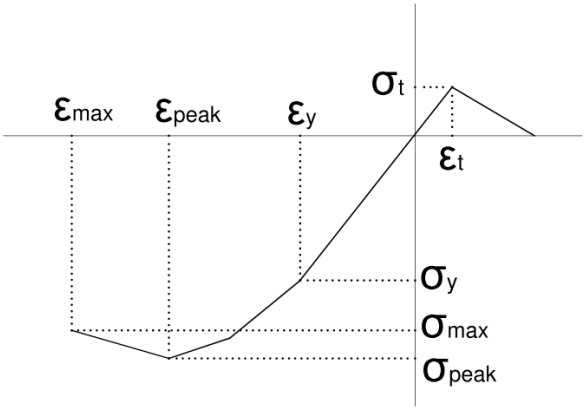


Figure 2: Typical uniaxial plain concrete stress-strain relationship

Under most circumstances concrete structures can be seen to be under a state of biaxial or triaxial stress. There are various nonlinear failure models that have been successfully established for analyses of concrete structures exposed to these stress state conditions.

2.1.1.4 Tensile strength

Plain concrete as a material is in all cases stronger in compression than tension. Tensile strength can be seen as either direct tensile strength or modulus of rupture which latter relates to the flexural strength of concrete. In standard beam and slab design the tensile strength is taken equal to zero although the tensile flexural capacity of concrete could be taken equal to the modulus of rupture. Many equations exist as to the modulus of rupture but for this research an equation given in ACI 318M-05 [1] is used, equation 1-1.

$$f'_t = 0{,}62 \cdot \sqrt{f'_c} \qquad \textbf{1-1}$$

2.1.1.5 Elastic modulus

One of the more influential factors in determining the structural behaviour of concrete is the Elastic modulus of the hardened concrete mixture. The Elastic modulus of harden concrete can be defined as the ration of uniaxial stress to the resultant axial strain [13]. The Elastic modulus is a direct measure of the concrete's stiffness properties but is highly variable from one mix to another. The Elastic modulus of concrete is manly affected by three components namely the concrete strength, type of aggregate and aggregate paste interface connection.

Concrete Elastic modulus can be given in three ways, Figure 3. The Elastic modulus can be defined as the initial tangent modulus, where concrete behaviour in a linear manner up to approximately 30-40% of ultimate strength. The secant modulus is the slope of the stress-strain diagram from the origin to a specific point and usually taken at about 0.45 x f '$_c$. The tangent modulus is the slope taken at any specified point along the curve.

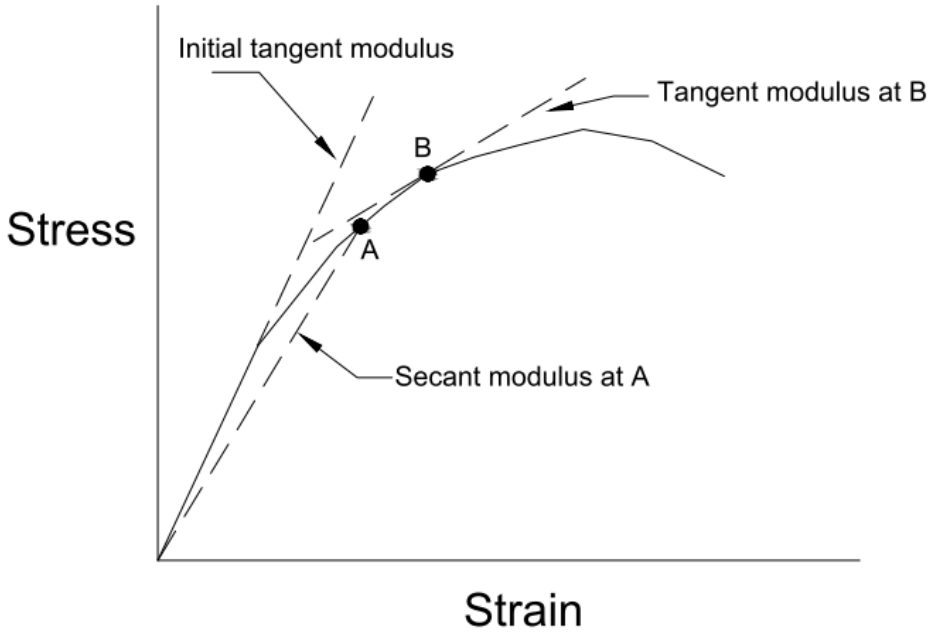


Figure 3: Elastic modulus of concrete graphical definition

A simplified formula for the secant Elastic modulus at 0.45 x f '$_c$ as given in ACI 318M-05 [1] for concrete with an assumed density of 2300 kg/m^3 is given in equation 1-2 in MPa.

$$E_c = 4700.\sqrt{f'_c} \qquad \textbf{1-2}$$

2.1.1.6 Poisson's ratio

Poisson's ratio is the ratio of lateral strain to the axial strain of a specimen caused by uniaxial strain alone. Poison's ratio is effected by factors including aggregate:cement ratio and type of aggregate. Through test it has been shown that the Poisson's ratio can vary from 0.11 to 0.23 but for design purposes is usually taken as 0.2.

2.1.2 Reinforcing Steel

Reinforcement with regard to reinforced concrete is meant to provide sufficient tensile strength to the resisting mechanism which is lacking in the concrete material itself. Tensile reinforcement in theory may be any material that provides the tensile strength, stiffness,

bondage and movement behaviour to sufficiently resist the applied loading conditions. It has been established that steel reinforcement has the properties to form a strong and robust structural mechanism with concrete.

Standard steel reinforcement used in reinforced concrete can be separated into mild and high-yield bars. Where, in South Africa, the steel has a manufacturer guaranteed minimum yield strength of 250 MPa and 450 MPa for mild and high-yield bars respectively. Figure 4 shows the difference in strength and stiffness behaviour between mild and high-yield steel reinforcement.

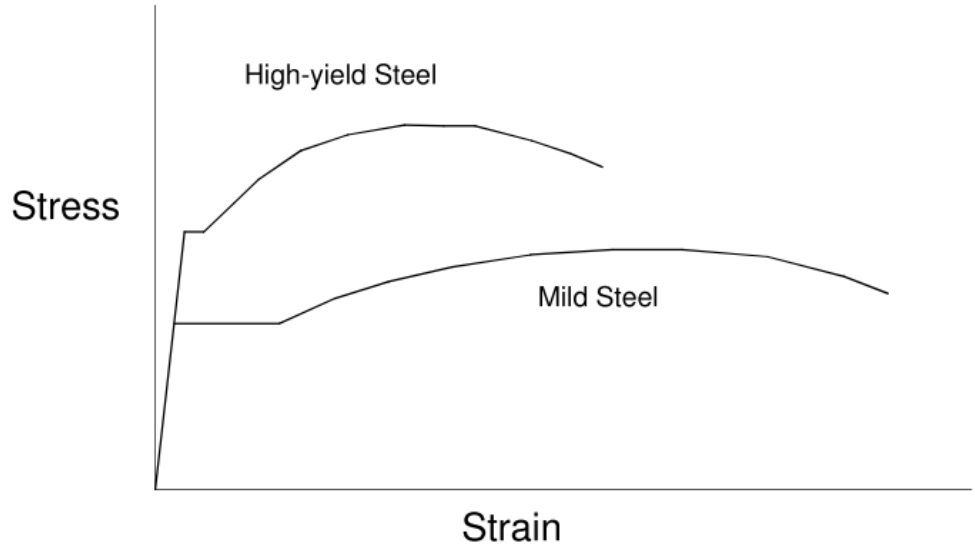


Figure 4: Stiffness behaviour of steel reinforcement

The nonlinear behaviour of steel reinforcement can be identified to comprise four main stages, Figure 5. In the first stage the steel behaves in a linear manner according to Hooke's law prior to point A. Between points A and B is stage two where the steel undergoes permanent plastic deformation and, although not always the case, can be seen to behave in a perfectly plastic state. Stage three between points B and C is termed strain hardening where the steel is strengthened due to plastic deformation.

The strengthening occurs because of movements caused by dislocation of different planes within the crystal structure of the material. The dislocations pile up against one another, and can become interwoven thereby preventing further deformation and "strengthening" the steel. Following strain hardening stage four begins, between points C and D, where necking occurs. Necking can simply be defined as the visible reduction of cross-sectional area of the reinforcement due to extreme axial strain.

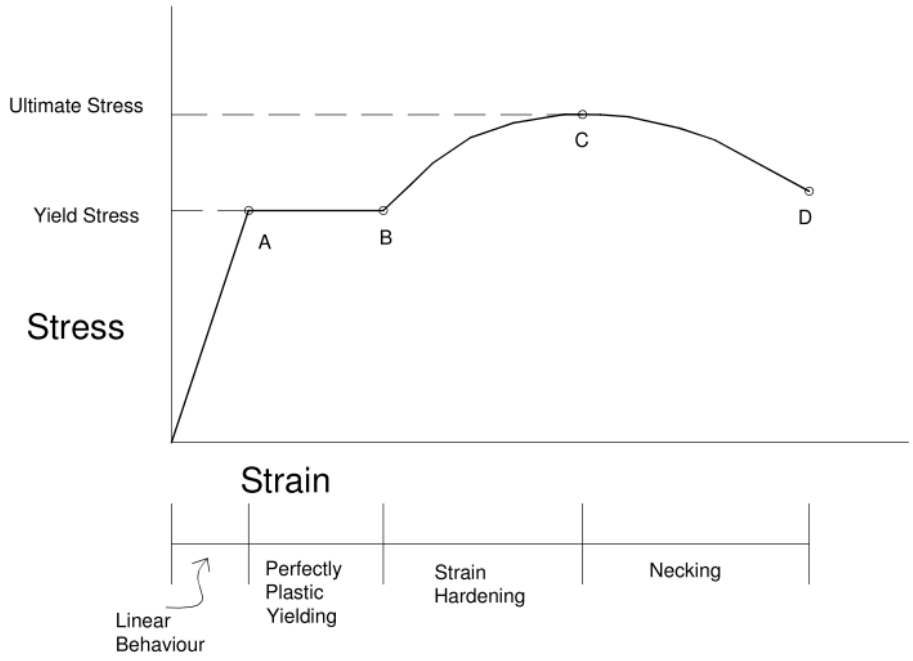


Figure 5: Stages of stiffness in steel reinforcement

Although steel comprises different proportions of metal alloys, it can theoretically be seen as an isotropic material. It is therefore much simpler to model the nonlinear behaviour of the steel reinforcement compared to modelling the nonlinear behaviour of concrete. Two common methods of representing the nonlinear behaviour of steel reinforcement is with either a bilinear (a) or elastic perfectly plastic (b) stress and strain relationship as shown in Figure 6.

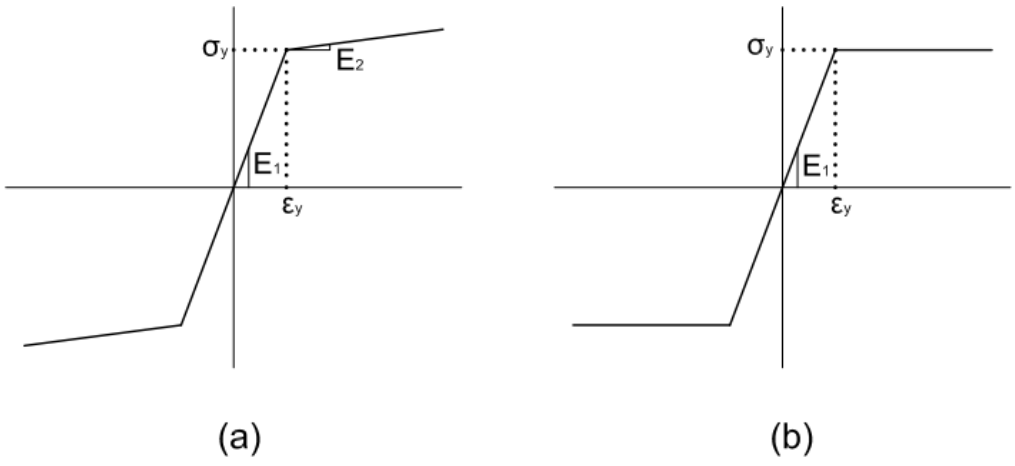


Figure 6: (a) Bilinear, (b) Elastic perfectly plastic stress and strain relationship

2.1.3 Bond between Concrete and Reinforcement

For the reinforced concrete section to behave as one mechanism, transfer of the tensile stresses in the reinforcement to the concrete needs to take place. This transfer is done through bond stress. In reinforced concrete design it is assumed that there is a perfect bond between the reinforcement and the concrete. To validate this assumption one needs to ensure that there is a sufficient bond stress development length and / or mechanical anchorage.

Bond stress can be transferred through friction, adhesion or bearing on deformations of the steel reinforcing bar. Generally friction and adhesion have limited transfer ability due to Poisson's effect causing a reduction in bar diameter when the steel bar is in tension. The majority of bond stress is therefore developed through bearing on deformation of the bar. As the deformations are angled the stress transfer comprises a radial component as well as a longitude bearing component. The bond stress is not constant with the length of the beam but follows the bending moment magnitudes.

To understand true bond stress one must review the process of crack development. When a simply supported beam is loaded past the cracking moment capacity the concrete tensile stress capacity is exceeded which causes cracking. When the beam cracks there are still however portions between the cracks where the concrete still carries tension. As the loading increases, the tensile stress in the concrete between the existing cracks increases until the discrete portion of concrete's tensile capacity is reached and secondary cracks are formed. The true bond stress distribution for a central portion of a cracked beam is given in Figure 7.

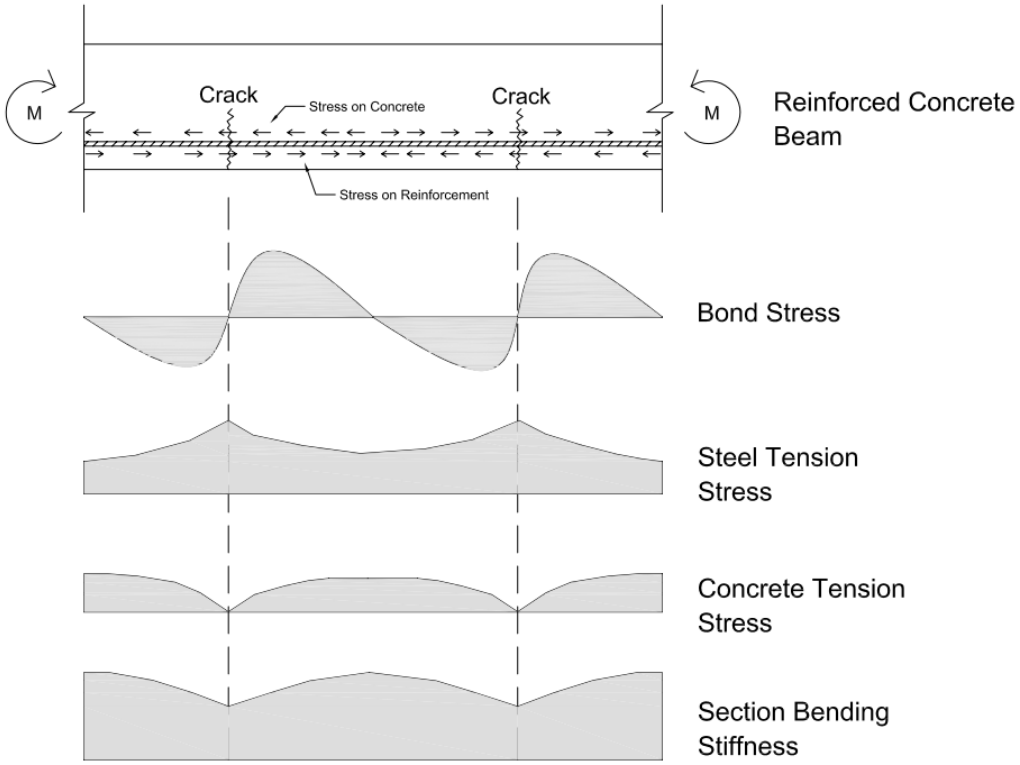


Figure 7: Tension stiffening mechanism

This action of true bond stress distribution is important when considering the effect on the nonlinear displacement behaviour. The increased rigidity due to these uncracked portions between the primary and, with increased loading, secondary cracks is termed tension stiffening. Tension stiffening is not explicitly taken into account under the proposed nonlinear model.

2.2 NONLINEAR ANALYSIS USING PLATES AND SHELLS

As previously mentioned reinforced concrete can be modelled using solid, shell or plate elements but the following paragraphs are focussed on the nonlinear analysis applicable to plate and shell representation of reinforced concrete.

The formulation of plate and shell elements began to be established in the mid-1960 and formulations improved rapidly through till late 1980. During this period where the formulation of the plate and shell elements evolved so did the investigation and application of nonlinear reinforced concrete behaviour and analysis.

As the development of both topics have taken place over a similar period and with the increase in computing power over the late 1980 early 1990 many researchers have developed separate constitutive numerical relations combined with plate and shell element formulations to mimic the nonlinear behaviour of reinforced concrete slabs.

When carrying out analysis using plate or shell elements the formulation can be dived into two main categories namely the layered and effective stiffness approaches. The layered method subdivides the cross-section of the element into imaginary discrete layers isolating the concrete and steel materials enabling the assessment of the concrete and reinforcement to be carried out separately. The effective stiffness approach seeks to combine the behaviour of the concrete and reinforcement into one material only capturing changes in structural stiffness as a whole which allows for a more simple solution.

McNeice (1967) [27] carried out a series of experimentations on transversely loaded square slabs supported vertically at the corners. McNeice employed plate elements and bilinear moment curvature graph to depict the nonlinear reinforced concrete behaviour. He included a derived yield criteria to enable the assessment of two-way spanning slabs. His approach is in essence an effective stiffness approach which was used to correlate the analytical results to the experimental study comparing elastic-plastic behaviour of the experimental specimens. McNeice's innovative work proved that the nonlinear response of reinforced concrete could be represented using the effective stiffness approach.

Jofriet and McNeice (1971) [21] established an effective stiffness analytical method taking into account the orientation of the cracks, the rigidity of a cracked regions and the effect on rigidity of steel orientation with respect to the crack direction. They derived a yield criteria and established two analytical models using two methods of calculating effective stiffness empirically derived by Beeby (1968) and Branson (1963). This paper proved that empirically derived stiffness equations can be combined with numerical methods using an appropriate yield criteria to effectively capture nonlinear behaviour.

Hand, Pecknold and Schnobrich (1972) [19] used the layered method approach to determine the load-deflection history of reinforced concrete slabs up to failure represented by plates and shells. As the layered method allows for the steel and concrete to be assessed separately an elastic-plastic model was used for the steel and modified the yield criteria proposed by Kupfer, Hilsdorf and Rusch (1969) to be used for concrete. The proposed method did not include geometric nonlinearity. This research established the layered method approach. The reinforcement and concrete behaviour could now be evaluated using separate material models. Final results were not completely accurate but did adequately capture the nonlinear behaviour including in-plane loading.

Rahman (1982) [37] established nonlinear models based on the layered approach with plate elements. The models allow for cracking in one or two directions and accounts for tension stiffening effects. Other aspects of nonlinearity such as yielding and crushing of concrete in compression and yielding or strain hardening of steel reinforcement were also considered. Rahman also used the Kupfer, Hilsdorf and Rusch (1969) concrete yield criteria but utilised a bilinear steel stress-strain relationship. The research also evaluated a number nonlinear algorithms to establish the stability and efficiency of each. When considering solid slabs, analytical results were compared to Cardenas (1968), Duddeck (1978), McNeice (1971) and Taylor (1966) experiments. An important aspect to this research is that the models could be applied to solid slabs, T – Beams, Bridge composites and prestressed concrete void slabs. In general, appropriate agreement with experimental results were obtained.

Milford and Schnobrich (1984) [28] used the layered approach with shell elements to establish nonlinear behaviour necessary for the analysis of reinforced concrete cooling towers. The objective of the work was to establish a model which deals with the failure analysis of reinforced concrete cooling towers subjected to wind loading, studies include the effect of cracking and yielding of reinforcement at failure and failure mode. The study also covered effects of large displacement geometric nonlinearity. A different approach to the representation of cracked concrete is formed using a rotating crack model developed by Gupta and Habibollah (1982) in which the direction of the crack is not fixed after the formation of the crack. It was assumed that the crack direction is normal to the current direction of the maximum principal strain caused by a change in stiffness and load redistribution. The cracks thus defined are not in essence

cracks, but can be seen as a mechanism defining the average crack direction. The rotating crack model produced significantly better results than the conventional fixed crack model. This research was able to accurately capture the nonlinear behaviour of a complex structural system such as a cooling tower using the layered approach and an updated concrete crack model.

Hu and Schnobrich (1990) [19] carried out similar research to the work carried out by Milford and Schnobrich (1984) [28]. This research however concentrated on extending the work of the nonlinear concrete rotating crack model. Due to an improved concrete model, excellent results were obtained when analytical values were compared to Cardenas-Sozen (1968) experimental values which included out of plane and in-plane loading.

Polak and Vecchio (1993) [35] undertook research which included an experimental program to a better understanding of the nonlinear behaviour of reinforced concrete slabs subjected to out of plane and in-plane loading. The objectives included the development of a simple and sufficiently accurate method for predicting the capacity of simply supported slabs including tension stiffening as well as geometric and material nonlinear effects. Polak and Vecchio used a layered finite element approach combined with shell elements. The concrete cracking was represented by modified compression field theory established by Vecchio and Collins (1986). Modified compression field theory is based on a smeared, rotating crack logic that, based on average stresses and average strains, considers equilibrium and compatibility conditions. When carrying out the analyses positive stability and convergence was achieved when using the iterative, full-load secant stiffness solution technique. Considering the complex structural response being evaluated, satisfactory results were obtained when comparing behaviour history of the analytical and experimental specimens.

Polak (1996) [32] used the effective stiffness approach to establish a simple procedure to calculate serviceability deflections in reinforced concrete one and two-way spanning slabs using the finite element method. The proposed model used the empirically derived effective stiffness equation established by Branson (1963). Polak proposed the use of simple yield criteria equations to incorporate the twisting moments experienced in two-way spanning slabs. Results obtained were accurate to the experimental results of Polak (1993), McNeice (1967), Ghoneim (1992) and Aghayere & MacGregor (1991). The disadvantage of the proposed method is that a design ultimate loading or actual failure mode load is not established and the method can only be used for serviceability checks.

Agbossou and Mougin (2005) [2] established a method for the nonlinear dynamic analyses of reinforced concrete slabs. The method was based on the layered approach. The research was specifically carried out to create models to analyse and design slabs required to resist impact from rock falls. The results from these models compared well with those of

experimental. The analyses revealed the influence of anisotropy effects on natural frequencies and mode shapes of slabs which are highly reinforced in one direction only. The research into nonlinear dynamic analysis using plates and shells was new and therefore more work should be carried out prior to any final conclusions being draw on the adequateness of the representation of the nonlinear dynamic behaviour of reinforced concrete slabs.

Zhang and Bradford and Gilbert (2007) [45] undertook to develop a comprehensive and numerically stable 4-node, 24-degree of freedom rectangular shell element formulation. The formulation is carried out using the layered approach. The formulation is appropriate for thin to moderately thick reinforced concrete slabs. Both geometric nonlinearity and material nonlinearity are included. Material nonlinearity combines effects due to concrete cracking and tension stiffening. Results obtained are considered accurate and numerically stable but the formulation thereof is considered fairly complex. The research was able to provide an element formulation capable of accurately capturing nonlinear response but with a computationally more efficient element as compared to earlier studies. The proposed formulation should be validated against and used for more complex structural systems which includes out of plane and in-plane forces.

The development of the nonlinear analysis using plate and shells has followed the philosophy of being able to, as accurately as possibly, replicate the nonlinear response of reinforced concrete using the finite element method. This philosophy is correct if one would like to capture actual failure modes. For design however the exact failure mode might not actually be required but would rather require a conservative analysis and design is preferred. Another aspect to consider is that to capture the nonlinear response accurately, complex mathematical models which not all design engineers are familiar with would be required. These mathematical models would also be required to form part of a commercial finite element software as few engineers would practically be able to program it themselves. It can therefore be seen that the proposed methods requiring complex formulation have limited scope as to their applicability. They are however invaluable when carrying out forensic analysis and design of complex structures which are not common.

Reinforced concrete slabs and beams can be seen as an extremely common structural systems which are familiar to all design engineers. There is therefore a need for a simplified design approach to the nonlinear analysis and design of reinforced concrete slabs and beams. The approach proposed allows for standard stress-strain relationships to be established using common reinforced concrete design equations and assumptions which all design engineers are familiar with. These stress-strain relations can be defined for each direction of reinforcement and used in the nonlinear analysis as would be done for typical orthotropic ductile materials. The method will allow for easy implementation into most commercial finite element software packages with API capability.

CHAPTER 3: FEM OVERVIEW AND PLATE ELEMENT FORMULATION

3.1 OVERVIEW OF THE GENERALISED FINITE ELEMENT METHOD

The generalised finite element method has become a standard method of structural analysis in recent times. The finite element method is based on the principle of virtual work, (virtual displacements), and states that the equilibrium of a body requires that for any compatible small virtual displacement imposed on a body in its state of equilibrium, the total internal virtual work is equal to the total external virtual work [5]. When using the finite element method with the required computing power, solutions to structural problems that were previously too complicated to solve are now effortlessly attained.

The beginnings of the finite element method are not completely known but it is generally accepted that the application of the method in practical engineering was initially developed by M.J. Turner carrying out work for Boeing from 1950-1962. Two persons working under Turner at Boeing had a significant contribution namely M.B. Irons and R.J. Melosh with the former having primary input into the development of isoparametric schemes, shape functions, the patch test and frontal solvers. Melosh went on to show that conforming displacement based formulations are a form of Rayleigh-Ritz which is based on the minimum potential energy principal.

As this work was carried out for Boeing it is not unexpected that the developments were first made in the aerospace industry. The transferal of the generalised finite element method to other fields of engineering can mainly be attributed to four persons involved in academia namely J.H. Argyris, R.W. Clough, H.C. Martin, and O.C. Zienkiewicz. This transfer took place between the 1950s and 1960s. The finite element method began to be introduced to the civil engineering field in the early 1960s mainly through the workings of R.W. Clough having the link between Boeing and University of California, Berkley.

From 1970 the existing method began to be consolidated and improved upon rather than formulated. Publications by T.J.R Huges and K.J. Bathe further established and improved upon the foundations of the generalised finite element method. Along with this establishment and improvement of formulations the rapid development of computing ability, on which the finite element method is so reliant upon, also took place. Commercial finite element code began to be established for use in industry.

Recent developments have moved more toward usability and involve the increased efficiency of the mathematical solutions, automeshing capabilities, automation and graphical user

interface capabilities. These advances enable the relatively complex method of analysis to be readily available to a wider audience of user. This is both a good and a bad thing.

In spite of the great power of the finite element method, the disadvantages of automated computer solutions must be kept in mind. The analysis is in modern times a generally graphical method and does not implicitly show how the analysis results are affected by a variation in structural properties like materials and geometric features to name but a few. Errors in input data which are not picked up by the analyst might not be evident in the final result and can easily be overlooked by the person carrying out the analysis.

The finite element method should be viewed as a method to give an accurate approximation of the real life situation rather than an exact solution of the real life situation. All solutions obtained should firstly be reviewed to see if the results "make sense" and therefore no matter what the application is, all analyses require a degree of engineering judgement. Where possible, verification results should be used to "calibrate" models which can then be expanded upon for use in applications of similar conditions.

3.2 PLATE ELEMENT FORMULATION

As in this study in-plane axial stiffness is not to be considered due to the assumption of pure bending, plate elements are used having 3 degrees of freedom per node as can be seen in Figure 8. The MATLAB program created to demonstrate the simplified method of analysis enables analysis via 8-noded plate elements using Mindlin-Reissner theory.

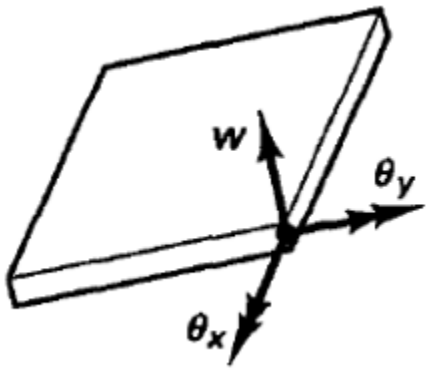


Figure 8: Nodal degrees of freedom [5]

The displacements and rotations of interest for this study are the out of plane z displacement and θ_x, θ_y rotations. The Mindlin-Reissner theory assumes that the particles of the plate element that are in a straight line normal to the mid-plane of the plate prior to deformation remain in a straight line but not necessarily normal to the mid-plane during deformation as shown in Figure 9.

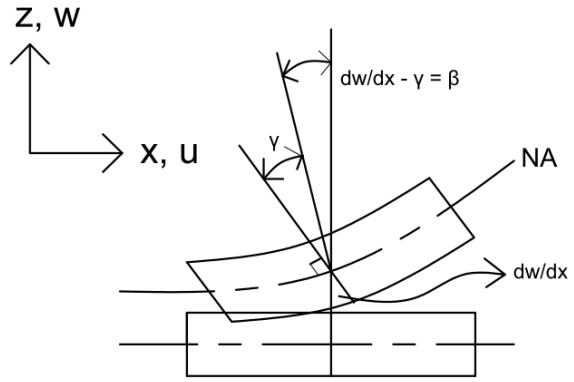


Figure 9: Shear deformations

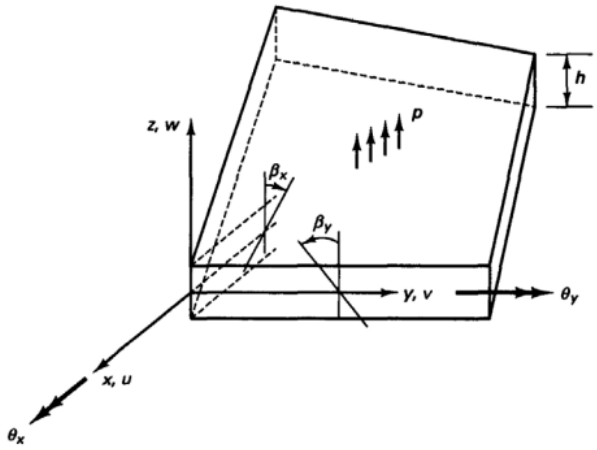


Figure 10: Deformation assumptions of plate including shear deformation, figure 5.25 [5]

The displacement components of co-ordinates x, y and z obtained from the plate element including shear deformation and small displacement theory are as follows:

$$u = -z.\beta_x(x,y)$$

$$v = -z.\beta_y(x,y)$$

$$w = w(x,y)$$

3-1

The βx and βy are the rotations of the normal to the mid-plane in the xz and yz planes respectively, Figure 10. In the formulation of the isoparametric plate elements the x and y components are represented by r and s respectively. The displacement interpolation functions are represented by h_i, with i being the applicable node number. The vertical displacement and rotations in the xz and yz planes are given by:

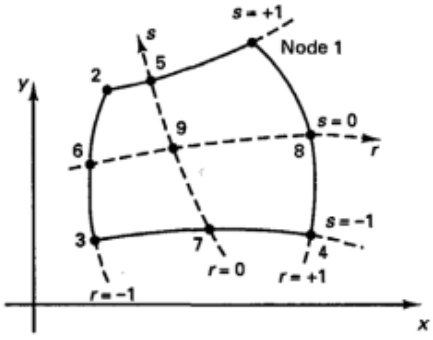
$$w(r,s) = \sum_{i=1}^{n} h_i(r,s) w_i$$

$$\beta_x(r,s) = -\sum_{i=1}^{n} h_i(r,s) \theta_{yi}$$

$$\beta_y(r,s) = \sum_{i=1}^{n} h_i(r,s) \theta_{xi}$$

3-2

The interpolation functions are obtained from literature by K.J. Bathe (1996) [5]. Shape functions used in the 8-noded plate elements are shown in Figure 11.



(a) 4 to 9 variable-number-nodes two-dimensional element

Include only if node i is defined

		$i=5$	$i=6$	$i=7$	$i=8$	$i=9$
$h_1 =$	$\frac{1}{4}(1+r)(1+s)$	$-\frac{1}{2}h_5$			$-\frac{1}{2}h_8$	$-\frac{1}{4}h_9$
$h_2 =$	$\frac{1}{4}(1-r)(1+s)$	$-\frac{1}{2}h_5$	$-\frac{1}{2}h_6$			$-\frac{1}{4}h_9$
$h_3 =$	$\frac{1}{4}(1-r)(1-s)$		$-\frac{1}{2}h_6$	$-\frac{1}{2}h_7$		$-\frac{1}{4}h_9$
$h_4 =$	$\frac{1}{4}(1+r)(1-s)$			$-\frac{1}{2}h_7$	$-\frac{1}{2}h_8$	$-\frac{1}{4}h_9$
$h_5 =$	$\frac{1}{2}(1-r^2)(1+s)$					$-\frac{1}{2}h_9$
$h_6 =$	$\frac{1}{2}(1-s^2)(1-r)$					$-\frac{1}{2}h_9$
$h_7 =$	$\frac{1}{2}(1-r^2)(1-s)$					$-\frac{1}{2}h_9$
$h_8 =$	$\frac{1}{2}(1-s^2)(1+r)$					$-\frac{1}{2}h_9$
$h_9 =$	$(1-r^2)(1-s^2)$					

(b) Interpolation functions

Figure 11: Shape interpolation functions, figure 5.4 [5]

The element bending strains are assumed to vary linearly through the element thickness and the transverse shear strains to remain constant through the element thickness. The curvature, bending strain and shear strains are defined by each of the following equations respectively:

$$\phi = \begin{pmatrix} \phi_{xx} \\ \phi_{yy} \\ \phi_{xy} \end{pmatrix} = \begin{pmatrix} \partial\beta_x/\partial x \\ \partial\beta_y/\partial y \\ \partial\beta_x/\partial y + \partial\beta_y/\partial x \end{pmatrix}$$

3-3

$$\varepsilon = \begin{pmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{pmatrix} = -z.\emptyset$$

3-4

$$\gamma = \begin{pmatrix} \gamma_{xz} \\ \gamma_{yz} \end{pmatrix} = \begin{pmatrix} \partial w/\partial x - \beta_x \\ \partial w/\partial y - \beta_y \end{pmatrix}$$

3-5

Using the strain matrices, element stresses can be determined. The in-plane and shear stresses are calculated as described in the following equations:

$$\tau_b = C_b.\varepsilon$$

3-6

$$\tau_s = \kappa.C_s.\gamma$$

3-7

Where C_b and C_s are the bending and shear constitutive material relationships respectively. These values are obtained using equations 3-8 and 3-9 where in equation 3-7 κ = 5/6 is the commonly used shear correction factor to convert the parabolic shear distribution to a uniform distribution.

$$C_b = \begin{bmatrix} \frac{E_x}{(1-\nu_x.\nu_y)} & \frac{E_y.\nu_x}{(1-\nu_x.\nu_y)} & 0 \\ \frac{E_x.\nu_y}{(1-\nu_x.\nu_y)} & \frac{E_y}{(1-\nu_x.\nu_y)} & 0 \\ 0 & 0 & G \end{bmatrix}$$

3-8

$$C_s = \begin{bmatrix} G & 0 \\ 0 & G \end{bmatrix}$$

3-9

$$G = \frac{E_x . E_y}{E_x . (1 + \nu_x) + E_y . (1 + \nu_y)}$$

3-10

To obtain the nodal displacements and resulting strains, stresses and forces, the global stiffness matrix is required to solve the equation K.u = R. The global stiffness matrix is merely an assemblage of the systems element stiffness matrices.

The element matrix comprises two parts, namely the contribution from bending and secondly the contribution from shear. The element stiffness matrix is thus summarised as:

$$k_e = \int_{A_e} h^3/12 \; . [B_b]^T . C_b . B_b dA + \int_{A_e} \kappa . h . [B_s]^T . C_s . B_s dA$$

3-11

The strain interpolation matrices for bending and shear shown in equation 3-15 are given in equations 3-12 and 3-13 respectively.

$$\mathrm{B_b} = \left[\begin{array}{ccc|c|ccc} 0 & 0 & -\partial \mathrm{h_1}/\partial \mathrm{x} & & 0 & 0 & -\partial \mathrm{h_n}/\partial \mathrm{x} \\ 0 & \partial \mathrm{h_1}/\partial \mathrm{y} & 0 & \ldots & 0 & \partial \mathrm{h_n}/\partial \mathrm{y} & 0 \\ 0 & \partial \mathrm{h_1}/\partial \mathrm{x} & -\partial \mathrm{h_1}/\partial \mathrm{y} & & 0 & \partial \mathrm{h_n}/\partial \mathrm{x} & -\partial \mathrm{h_n}/\partial \mathrm{y} \end{array}\right]$$

3-12

$$\mathrm{B_s} = \left[\begin{array}{ccc|c|ccc} \partial \mathrm{h_1}/\partial \mathrm{x} & 0 & \mathrm{h_1} & & \partial \mathrm{h_n}/\partial \mathrm{x} & 0 & \mathrm{h_n} \\ \partial \mathrm{h_1}/\partial \mathrm{y} & -\mathrm{h_1} & 0 & \cdots & \partial \mathrm{h_n}/\partial \mathrm{y} & -\mathrm{h_n} & 0 \end{array}\right]$$

3-13

To enable ease of numerical integration from -1 to 1, the Jacobian matrix, J, is used which maps a differential component from the isoparametric coordinates to the global coordinates. Using the Jacobian operator $(\partial h_n/\partial x)$, $(\partial h_n/\partial y)$ are calculated by:

$$\begin{bmatrix} \partial h_n/\partial x \\ \partial h_n/\partial y \end{bmatrix} = J^{-1} \begin{bmatrix} \partial h_n/\partial r \\ \partial h_n/\partial s \end{bmatrix}$$

3-14

Where the Jacobian matrix is defined as:

$$J = \begin{bmatrix} \partial x/\partial r & \partial y/\partial r \\ \partial x/\partial s & \partial y/\partial s \end{bmatrix}$$

3-15

with

$$\partial x/\partial r = \sum_{i=1}^{n} \frac{\partial h_i}{\partial r} * x_i$$

$$\partial x/\partial s = \sum_{i=1}^{n} \frac{\partial h_i}{\partial s} * x_i$$

$$\partial y/\partial r = \sum_{i=1}^{n} \frac{\partial h_i}{\partial r} * y_i$$

$$\partial y/\partial s = \sum_{i=1}^{n} \frac{\partial h_i}{\partial s} * y_i$$

3-16

With the strain interpolation matrices now determined the element stiffness matrix is formulated as a function of natural co-ordinates r and s. The integral is mapped to the global co-ordinate system using the determinant of the Jacobian.

$$k_e = \int_{-1}^{1}\int_{-1}^{1} h^3/12 \ .[B_b(r,s)]^T.C_b.B_b(r,s).|J|\,drds + \int_{-1}^{1}\int_{-1}^{1} \kappa.h.[B_s(r,s)]^T.C_s.B_s(r,s).|J|.drds$$

3-17

Although the pure displacement based Mindlin-Reissner plate element is subject to “shear locking”, the simplest means of circumventing this is through reduced or selective reduced integration. With guidance from documentation by Bletzinger (2001) [8], it was established that for the 8-noded plate elements selective reduced integration be utilised with the bending terms integrated using 3x3 Gauss point integration and the shear terms integrated using 2x2 Gauss point integration. By using selective reduced integration errors in the integration scheme lead to a “softening” of the shear term thus avoiding the problem of “shear locking”. This also has the added benefit of lower computational cost but a large disadvantage is that reduced integration may lead to zero-energy displacement modes.

CHAPTER 4: SIMPLIFIED NONLINEAR MATERIAL PROPERTIES

4.1 RATIONALE

The nonlinear material constitutive relations are established with two objectives in mind. Firstly the material properties presented must have the required level of accuracy so as to validate its use in design methods and secondly to enable ease of formulation for the design engineer. The simplified method of analysis presented in this report is based on provisions given in the American concrete design code, ACI 318–05 [1] to calculate section capacity. The tensile strength of concrete is therefore not taken into account. In the derivation of the nonlinear material properties of reinforced concrete, with regard to this research, the following assumptions are made:

- *Simple beam theory*: Cross sections normal to the mid-plane of an element that are plane prior to loading remain plane during stress development process. This assumption is made for the formulation of thin plates. Thick plate theory is used in the finite element formulation in this research and this assumption may affect final results. But as the research is carried out for use in slab elements, shear deformations are not expected to contribute significantly to the overall deformation of an element.
- *Tension controlled reinforced concrete*: Structural elements considered in the scope of this report are tension controlled whereby the tensile reinforcement will reach yield strain prior to the onset of maximum concrete compressive strain. The assumption is valid as slab designs should be carried out so as to ensure ductile failure modes. Tension control should in any case be checked in the design process.
- *Linear strain distribution*: Strains throughout the cross sectional depth are assumed to have a linear distribution. This has been shown to be a valid assumption when carrying out the analysis and design of reinforced concrete structures.
- *Compatible strains*: the reinforcement is assumed to have a perfect bond with the surrounding concrete. This is a common assumption in the design of reinforced concrete slabs.

4.2 MATERIAL PROPERTY CALCULATIONS

Moment curvature relationships are a convenient way to represent the nonlinear behaviour of reinforced concrete. Figure 12 depicts a typical moment curvature relationship for a reinforced concrete structural element designed so as to give ductile yield failure. Moment curvature relationships can, with the available computing facilities, effortlessly be formulated for a given

arrangement of a reinforced concrete section. For this study three significant points are identified to form the basis of the nonlinear evaluation.

The first the point is where the concrete section reaches its tensile capacity at the extreme outer fibre. Prior to this point the section acts as a conventional linear elastic element.

The second point is where the reinforced concrete section reaches yield. It is assumed that the tension reinforcement reaches yield strain prior to the concrete reaching ultimate compressive strain.

The third point is where the reinforced concrete section reaches the ultimate section moment capacity. This is generally the strain state which is used in design under ultimate loading conditions.

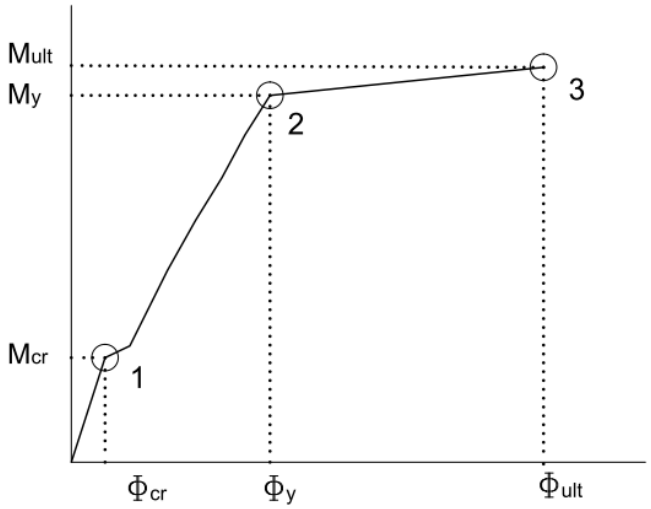


Figure 12: Typical Moment Curvature Diagram

For the purposes of this research, it is required that an equivalent stress-strain diagram be used to represent the nonlinear behaviour of the reinforced concrete section. This however cannot be carried out naturally.

In the real behaviour of reinforced concrete members the neutral axis shifts in depth as the section is exposed to increasing loads and, as the strain distribution is assumed to be linear throughout the section, strains will not be equal at either side of the element.

When carrying out finite element analysis using plates to represent the reinforced concrete element, strains are also assumed to vary linearly throughout the section but the element does not physically mimic the vertical crack propagation and therefore the “neutral axis” remains at the mid-plane of the element confirming that strains are equal at either edge of the element section.

It is proposed that the equivalent stress-strain diagram is formulated by ensuring strain compatibility using curvature relations and that the associated stress point is calculated using the moment related to the relevant curvature value.

Points 1, 2 and 3, given in Figure 13, can be thought of as a nonlinear behaviour begin point, yield point and ultimate strength point respectively. It is noted that extra sub-points are to be calculated between points 1 and 2 as this will give a much more accurate description of the nonlinear behaviour.

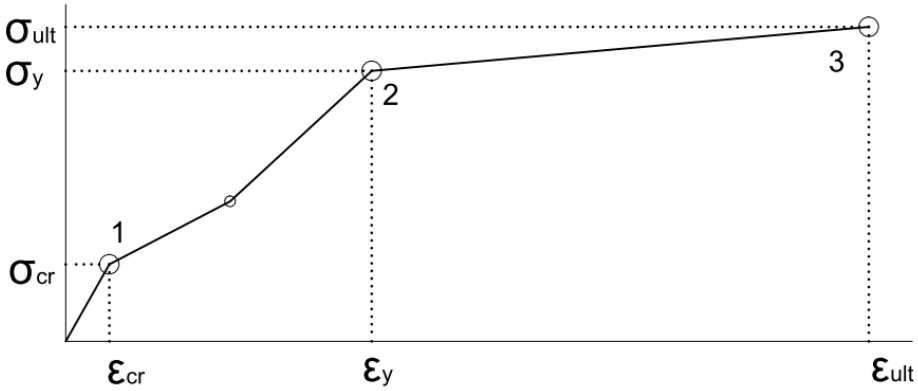


Figure 13: Typical Calculated Stress vs. Strain Diagram

To determine point 1 on the stress-strain diagram a simple equation to determine the cracking stress is used as given in 4-1 and 4-2.

$$\sigma_{cr} = 0{,}62 \,.\sqrt{F_{cu} \,.\, 0{,}8}$$

4-1

$$\varepsilon_{cr} = \sigma_{cr} \,/\, E_c$$

4-2

Following point 1 of the stress-strain diagram, the member is considered to be cracked and nonlinear behaviour begins. Curvature for the finite element program can be estimated using equation 4-3.

$$\emptyset_{fem} = \left(\varepsilon_{eq} \,.\, 2\right) / \, h$$

4-3

As can be seen in Figure 14 curvature of the actual behaviour may be calculated using equation 4-4.

$$\emptyset_{design} = \varepsilon_{cc} \,/\, x$$

4-4

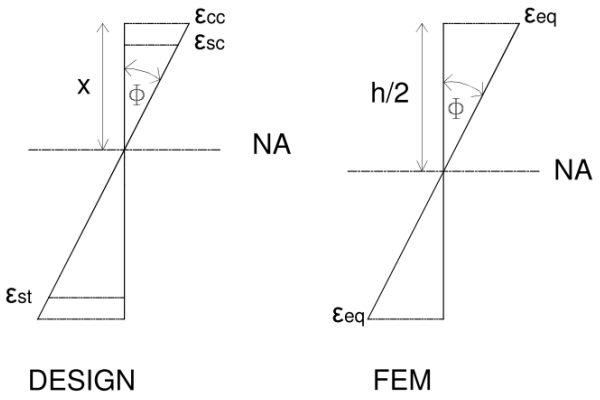


Figure 14: Relevant strain distribution to calculate curvature

To ensure curvature compatibility between the finite element program and actual behaviour 4-3 and 4-4 are equated giving rise to equation 4-5 which is used to calculate the equivalent strain point (ε_{eq}) on the a typical stress-strain diagram.

$$\varepsilon_{eq} = (\varepsilon_{cc} \,.\, h) \,/\, (2 \,.\, x)$$

4-5

When calculating the strain point using 4-5 the neutral axis depth (x) is required. This neutral axis value is associated to the section moment at that particular strain distribution. The equivalent stress σ_{eq} is estimated using 4-6.

$$\sigma_{eq} = \frac{M_{section} \,.\, h}{2 \,.\, I_g \,.\, (1 - \nu^2)} = \frac{M_{section} \,.\, 6}{h^2 \,.\, (1 - \nu^2)}$$

4-6

The three main strain states which should be considered in the formulation of a typical stress-strain diagram are shown in Figure 15.

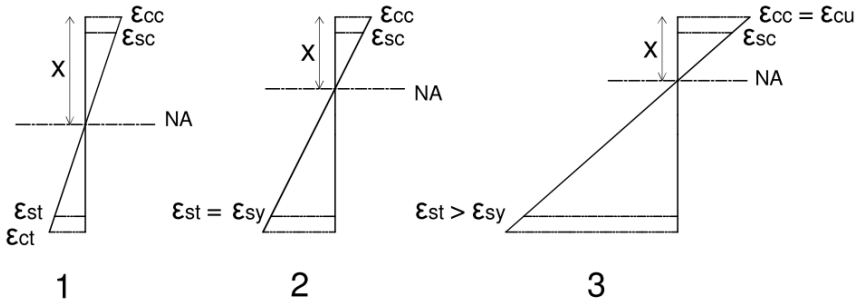


Figure 15: Three major design strain states to be considered

The proposed method for constructing the typical stress-strain diagram is set out in the steps given below:

1. Calculate ultimate moment and ensure tension controlled member.
2. Calculate Cracking Stress. – *Point 1*
3. Calculate associated Cracking Strain. – *Point 1*
4. Calculate section equilibrium where strain in the tension steel is at yield strain.
5. Calculate equivalent strain at yield. – *Point 2*
6. Calculate equivalent stress at yield. – *Point 2*
7. Using information from step 1 calculate equivalent strain at ultimate. – *Point 3*
8. Calculate equivalent stress at ultimate. – *Point 3*
9. Using information from step 4 calculate section equilibrium where curvature = 1/2 of yield called Point 1a.
10. Calculate equivalent strain. – *Point 1a**
11. Calculate equivalent stress. – *Point 1a**

*It is noted that closely following the reaching of cracking stress a large degree of nonlinearity is experienced. Between the cracking moment and the yielding moment the upward crack propagation has a significant nonlinear effect on the bending stiffness and, dependant on reinforcement ratios, therefore this portion of the stress-strain graph is represented using a bilinear relationship with an intermediate point being defined as the point whereby the curvature is 1/2 that of the yielding moment state curvature. Through this research it has been seen that for members with a tension steel area to gross cross sectional area ratio < 1% only one extra point between points 1 and 2 is required to be calculated.

CHAPTER 5: MATLAB PROGRAM & NONLINEAR SOLVER

5.1 ANALYSIS PROGRAM BASIS

To validate the proposed method a custom written finite element program is developed using MatLab 2011b. The program considers only 8-noded plate elements as given in Section 3.2. This section gives a brief description on the high level approach to a finite element solution using any given programming language.

To perform a complete finite element analysis, certain stages are identified which are necessary for carrying out the finite element analysis. The following stages were considered in the development of the finite element program:

- *Physical Problem Definition*

  As the finite element method is in essence an approximation of a real life situation, the analyst is required to understand the real life situation and express this in a finite element simulation. For the current research it is identified that the behaviour in question is predominantly bending against traverse loading and therefore any loading and stiffness contributions in the in-plane direction are considered negligible.

  With regards to the required nonlinear behaviour it is assumed that large displacement nonlinearity is not of concern as in the design of a reinforced concrete beam / slab it is assumed that the beam / slab not deflect excessively. Therefore when carrying out the nonlinear analysis only material nonlinearity, using Cauchy stress tensors, is considered.

- *Geometric Modelling*

  Once one has carried out the problem definition, the question of how the physical problem will be represented geometrically. For instance should one model a reinforced concrete beam / slab under bending using 1D, 2D or 3D elements? This all depends on what behaviour the analyst is trying to capture, how accurate the results need to be and the level of computational cost. For this research it is decided that the modelling of the reinforced concrete beams / slabs will be done using 2D plate elements and is seen as sufficient for the behaviour, accuracy and computational cost in question.

- *Element type*

  Refers to the nature of element used to represent the structural behaviour of the identified simulation required to denote the real life problem. Elements can vary from truss, beam, plane strain, plane stress, plate, shell, solid and many others. The element type is also defined by the interpolation functions which are commonly defined by the number of nodes. For example a 4-noded plate element is defined by linear interpolation functions whereas

an 8-noded plate element is defined by a quadratic polynomial interpolation function. For the current research, and as mentioned in Section 3.2, 8-noded plate elements are used.

- *Discretization*

  To correctly capture the necessary displacement behaviour of a simulated problem the correct fineness of discretization should be employed. One of the statements relevant to the degree of discretization is that the finer the discretization the higher the computational cost, the more coarse the discretization the lower the computational cost.

  The point at which the level of accuracy achieved compared with computational cost incurred is satisfactory will be termed geometric discretisation convergence. There are many ways to decide on this point of convergence but for this research the relationship of displacement and associated mesh density is considered. As a secondary convergence measure, for the McNeice and C1 slabs, the change in moment result is also evaluated to ensure a satisfactory mesh density is used. For the one way spanning Polak slab, the applied moment at the ends of the slab is compared to that of the moment result obtained through analysis to establish a satisfactory mesh density.

  Table 1 gives the percentage of change in deflection associated with an increase in mesh density as well as stipulated moment result comparison. Only boundary conditions and loading type for the experimental case studies used in this research are considered. Slab A3 has the same loading and boundary conditions as slab C1 and therefore discretisation convergence is judged from results of slab C1.

Table 1: Geometric Discretisation Convergence

Test Specimen	Mesh 1	Mesh 2	Displ 1	Displ 2	Δ Displ %	My (kN.m) 1	My (kN.m) 2	Δ My %
McNeice	3x3	4x4	-0.0065	-0.0065	0.17%	13.637	14.164	3.70%
C1	3x3	4x4	-0.0079	-0.0079	0.01%	14.838	14.777	0.42%
Test Specimen	**Mesh 1**	**Mesh 2**	**Displ 1**	**Displ 2**	**Δ Displ %**	**1x1 M (kN.m) Applied**	**1x1 M (kN.m) Result**	**Variance**
SM1	1x1	1x3	-0.0020	-0.0021	4.18%	10.000	10.053	0.51%

For the McNeice, Ghomein & MacGregor and Aghayere & MacGregor slabs only a quarter symmetry model is used and the finite element discretisation is shown in Figure 16.

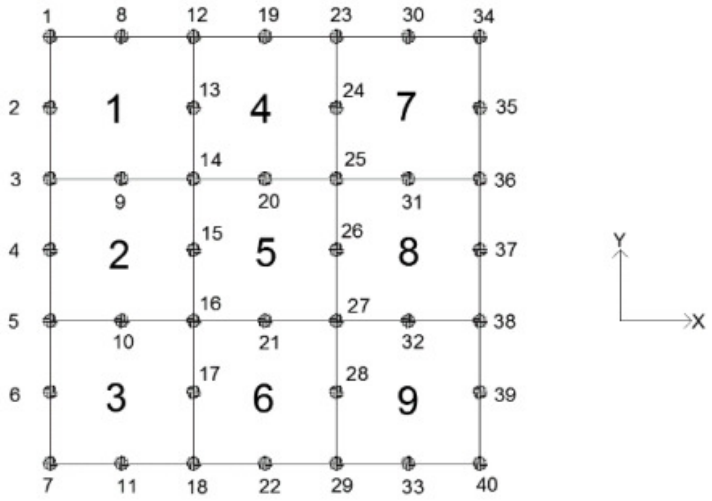


Figure 16: Discretisation for McNeice, Ghomein & MacGregor and Aghayere & MacGregor slabs

- *Constitutive relations*

  The strains induced by externally applied loads are required to be converted into stresses. The transformation of these strains into stresses are done by using the correct constitutive relations. The required material properties for this particular analysis are assigned in the form of an initial E-Modulus, stress-strain graph as per Section 4.2, plate thickness and Poisson's ratio.

- *Boundary conditions*

  Two types of boundary conditions are identified namely, essential and natural boundary conditions. The naming convention originates from the weak form formulation whereby the essential boundary, displacement, conditions are required prior to the derivation of the system equations. Natural, force, boundary conditions are not needed for the formulation process but are *naturally* derived during the formulation process.

  For the current research plate elements having three degrees of freedom per node are used. Only essential boundary conditions are considered by restraining either vertical displacement or rotations or both dependant on the support conditions of each respective test case.

- *Loading*

  There is a large variation of load types including transverse surface pressure, nodal point, side traction, body force loads to name but a few. All loads are eventually converted into equivalent nodal loads and load vector either using consistent or lumped methods. Lumping loads is not recommended as numerical errors are generally incurred dependant on element and analysis type. For the current research only transverse nodal point loads and nodal moments are required which are simple to apply.

- *Solution*

  Can be defined as the method of obtaining the required analysis outputs given the system boundary, loading, initial and stiffness input conditions. Solutions can vary in type from a simple linear A.x = b solution to nonlinear implicit or explicit solutions. Solution methods can vary further in terms of the mathematical procedure employed in each type of solution process.

  One of the strengths of the current method is the required solution process is simple in nature taking the form of a repetitive A.x = b solution. By using this method, a finite element program only having a standard A.x = b solution will be capable of carrying out the, herein presented, nonlinear problem.

- *Result output*

  Refers to the type of results and how these results are given to the analyst / engineer following the completion of the analysis. For the current research analysis results were output into Microsoft Excel where the required post-processing was carried out.

5.2 SIMPLIFIED NONLINEAR SOLVER

The proposed nonlinear solution considers only material nonlinearities and does not consider geometric nonlinearities which is valid as large displacements are not of concern for this particular application.

As discussed in Section 4.2 the E-modulus of the material can be changed to give an equivalent bending stiffness to the plate formulation as discussed in section 3.2. Using the reinforcement properties of the concrete member, control points making up a typical stress-strain diagram, Figure 13, can be established. This is assuming that the local x and y co-ordinates of the plate elements are aligned with the two directions of reinforcement which are orthogonal to one another as shown in Figure 17.

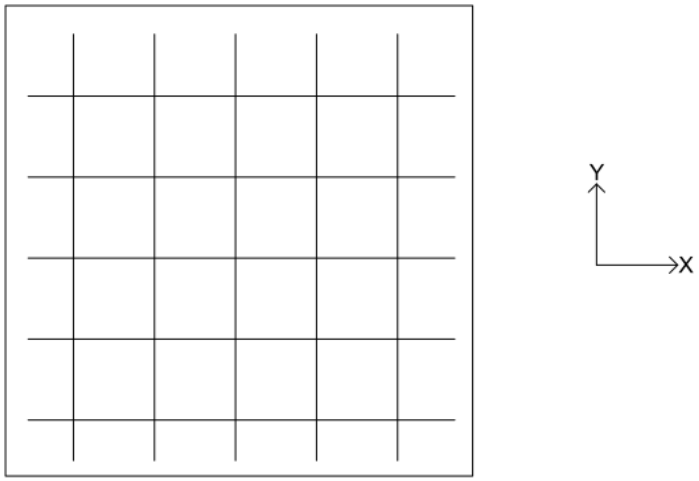


Figure 17: Reinforcement orientation

If one assesses a pure one way spanning system only moments in the bending direction are to be evaluated, but for a two way spanning system the twisting moments are required to be considered. Methods established by Wood and Armer (1968) [44] allow for the accounting of the twisting moments in the design. Johansen (1962) [20] established modern day Yield-Line theory which is widely used today.

Using this theory as a basis, similar design equations have been be established for the x and y directions independently. Take Figure 18 (a) depicting portion of slab showing Johansen's stepped yield criteria. Figure 18 (b) takes an infinitesimal portion of that stepped yield criteria. It will be considered that the crack developed in the reinforced concrete slab will being taken at a direction normal to the principal moment defined as θ.

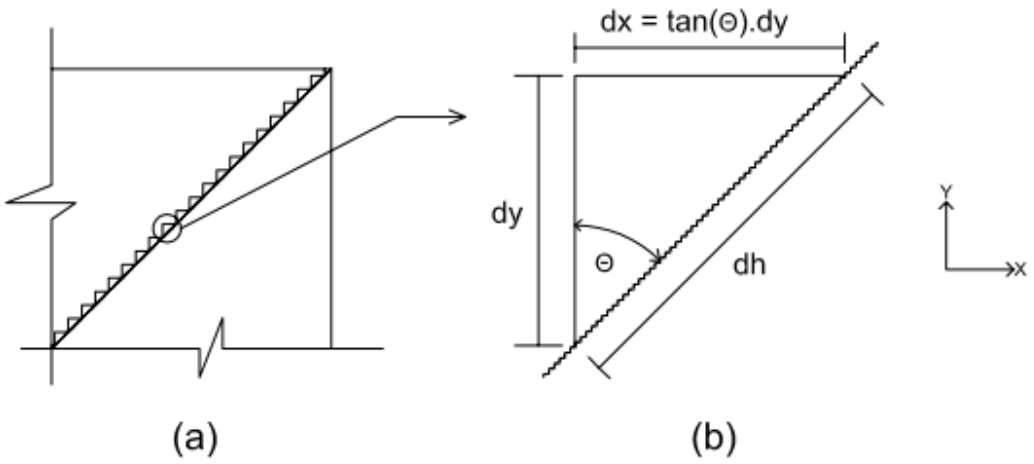


Figure 18: Johansen's stepped yield criteria discretisation

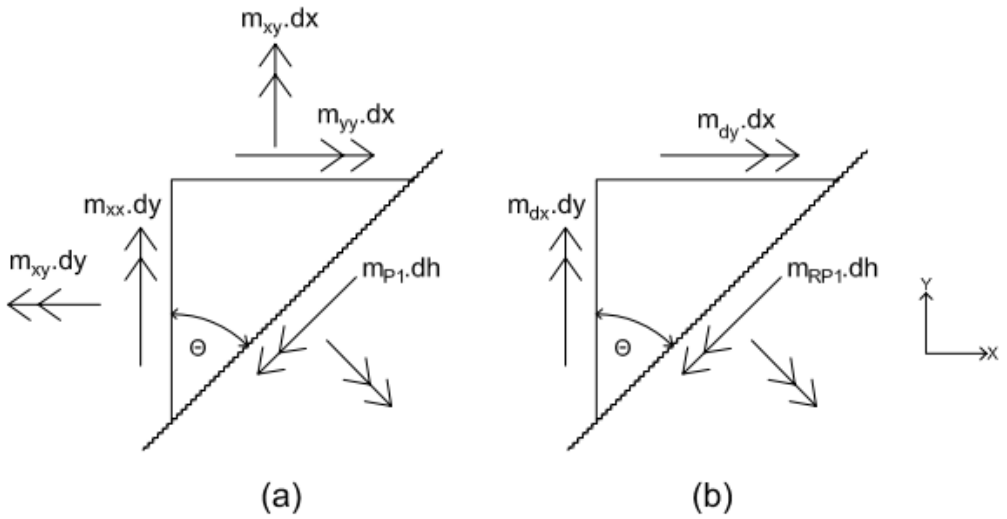


Figure 19: (a) Applied section moments, (b) Resisting section moments

Figure 19 (a) shows a discretised portion intersecting the yield line of a given slab. Using this configuration of applied moments an equilibrium equation is established and as given in equation 5-1. Simplifying equation 5-1 leads to the development of an equation expressing the principal 1 moment in terms of m_x, m_y, m_{xy} and θ, equation 5-2.

$$m_{P1}.dh = (m_{xx}.dy + m_{xy}.tan\theta.dy).cos\theta + (m_{yy}.tan\theta.dy + m_{xy}.dy).sin\theta$$

5-1

$$m_{P1} = \frac{dy}{dh}.cos\theta.[(m_{xx} + m_{xy}.tan\theta) + (m_{yy}.tan\theta + m_{xy}).tan\theta]$$

$$= \left(\frac{dy}{dh}\right)^2.(m_{xx} + m_{yy}.tan^2\theta + 2.m_{xy}.tan\theta)$$

5-2

An equation to evaluate the reinforced concrete slab moment capacity needs to be established to ensure the slab has sufficient capacity to resist the applied moments. By using the condition as illustrated in Figure 19 (b), orthogonal reinforcement, and following a similar method as previously equation 5-3 is developed giving the principal 1 moment resistance capacity in terms of moment capacity in the xx and yy directions.

$$m_{RP1} = \left(\frac{dy}{dh}\right)^2.(m_{dx} + m_{dy}.tan^2\theta)$$

5-3

Solving for the conditions at the point where the applied moment equals the resisting moment produces the common 'normal' yield criterion for slabs. Equations 5-2 and 5-3 are set equal to one another and solved for the minimum by differentiating with respect to theta.

To ensure that the applied moment is never greater than that of the design moment the sign of the twisting moment is assigned that of the associated in-plane moment thus producing equations 5-4 and 5-5. For design theta is generally taken as equal to 45° to cater for a wide range of moment values thus making tan(θ) equal to unity. These equations are only suitable for reinforcement arrangements which are orthogonal in the x and y directions as shown in Figure 17.

$$M_{dx} = M_{xx} + tan\theta.|M_{xy}| * |M_{xx}|/M_{xx}$$

5-4

$$M_{dy} = M_{yy} + \frac{1}{tan\theta}.|M_{xy}| * |M_{yy}| / M_{yy}$$

5-5

The proposed algorithm requires for the design moments given by equations 5-4 and 5-5 to be represented by a strain quantity. The strain quantity will then be compared to the strain as given in the typical stress-strain diagram. The stress contributing to the moment in the xx and yy directions can be expressed using equations 5-6 and 5-7.

$$\sigma_{xx} = \varepsilon_{xx}.\left(\frac{E_x}{(1-\nu_x.\nu_y)}\right) + \varepsilon_{yy}.\left(\frac{E_y.\nu_x}{(1-\nu_x.\nu_y)}\right)$$

5-6

$$\sigma_{yy} = \varepsilon_{xx}.\left(\frac{E_x.\nu_y}{(1-\nu_x.\nu_y)}\right) + \varepsilon_{yy}.\left(\frac{E_y}{(1-\nu_x.\nu_y)}\right)$$

5-7

The twisting stress contributing to the twisting moment is calculated using equation 5-8.

$$\sigma_{xy} = \varepsilon_{xy}.\left(\frac{E_x.E_y}{E_x.(1+\nu_x)+E_y.(1+\nu_y)}\right)$$

5-8

Using these equations the total strain contribution in the xx and yy directions can be calculated in terms of ε_{xx} and ε_{yy} respectively. Firstly the strain contributions for σ_{xx} and σ_{yy} in terms of ε_{xx} and ε_{yy} respectively are defined by equations 5-9 and 5-10.

$$\varepsilon_{tx1} = \left(1 + \frac{E_y.\nu_x.|\varepsilon_{yy}|}{E_x.|\varepsilon_{xx}|}\right).\varepsilon_{xx}$$

5-9

$$\varepsilon_{ty1} = \left(1 + \frac{E_x.v_y.|\varepsilon_{xx}|}{E_y.|\varepsilon_{yy}|}\right).\varepsilon_{yy}$$

5-10

The contribution of twisting strain is required to be considered and is carried out by calculating the twisting strain as a portion of the strain contributions given in equations 5-9 and 5-10. The contribution of twisting strain can be expressed using equations 5-11 and 5-12 .

$$\varepsilon_{tx2} = \left(\frac{G.|\varepsilon_{xy}|.(1 - v_x.v_y)}{|E_x.\varepsilon_{xx} + E_y.v_x.\varepsilon_{yy}|}\right).\varepsilon_{tx1}$$

5-11

$$\varepsilon_{ty2} = \left(\frac{G.|\varepsilon_{xy}|.(1 - v_x.v_y)}{|E_x.v_y.\varepsilon_{xx} + E_y.\varepsilon_{yy}|}\right).\varepsilon_{ty1}$$

5-12

The total design strain quantity for the xx direction is established by combining equations 5-9 and 5-11, where the total design strain quantities for the yy direction is obtained by combining equations 5-10 and 5-12. The total design strain quantities for the xx and yy directions are given in equations 5-13 and 5-14.

$$\varepsilon_{tx} = \left[\left(1 + \frac{E_y.\nu_x.\varepsilon_{yy}}{E_x.\varepsilon_{xx}}\right).\left(1 + \left(\frac{G.|\varepsilon_{xy}|.(1 - \nu_x.\nu_y)}{|E_x.\varepsilon_{xx} + E_y.\nu_x.\varepsilon_{yy}|}\right)\right)\right].\varepsilon_{xx}$$

5-13

$$\varepsilon_{ty} = \left[\left(1 + \frac{E_x.\nu_y.\varepsilon_{xx}}{E_y.\varepsilon_{yy}}\right).\left(1 + \left(\frac{G.|\varepsilon_{xy}|.(1 - \nu_x.\nu_y)}{|E_y.\varepsilon_{yy} + E_x.\nu_y.\varepsilon_{xx}|}\right)\right)\right].\varepsilon_{yy}$$

5-14

Following the cracking strain and under the assumption of a cracked Poisson's ratio equal to zero equations 5-13 and 5-14 reduce to equations 5-15 and 5-16 respectively.

$$\varepsilon_{tx} = \left[1 + \left(\frac{E_y.|\varepsilon_{xy}|}{\left(E_x + E_y.(1 + \nu_y)\right).|\varepsilon_{xx}|}\right)\right].\varepsilon_{xx}$$

5-15

$$\varepsilon_{ty} = \left[1 + \left(\frac{E_x.|\varepsilon_{xy}|}{\left(E_y + E_x.(1 + \nu_x)\right).|\varepsilon_{yy}|}\right)\right].\varepsilon_{yy}$$

5-16

Now that the design strain quantities at any given load step have been established the method of nonlinear analysis is considered. Although the analysis process presented in this report considers the nonlinearity of the material properties, the solution technique is in essence a repetitive linear static solution updating the stiffness matrix after each converged load step.

During the analysis and solution, the strain at each converged load step and associated results are calculated. For the first increment, the ratio of stress over strain, E_1, must be equal to E_C. The subsequent steps are linear between the points on the curve and a generalised convergence check to evaluate adherence can be defined as:

$$0.01 \geq \left(\varepsilon_{con} - \left|\left(\sum_{g=1}^{k} \varepsilon_g\right) + \varepsilon_n\right|\right) \Big/ \varepsilon_{con} \geq -0.01$$

5-17

Where ε_{con} is controlling strain point given by points making up the graph in Figure 13 and ε_g is the cumulative calculated design strain quantity for all current converged load steps, (k), and ε_n is current iteration calculated design strain quantity. Considering equation 5-17 cannot be completely satisfied due to rounding off errors or formulated design strain quantity equation, an acceptance error chosen by the user but defaulted to 1.0% for this research is employed.

Should any point in the system fall below -0.01 using the convergence criteria given in equation 5-17 it can be seen that one or more strain control points have been exceeded and load reduction would be required to take place.

Should no points in the system fall below 0.01 using the convergence criteria it can be seen that the load is required to be increased. Should the system not pass the criteria given in

equation 5-17, the load increment can be increased / reduced and a new loading magnitude assigned by using equation 5-18 with logic defined using Figure 20.

$$L_{n+1} = L_n . \left| \left(\varepsilon_{con} - \left| \left(\sum_{g=1}^{k} \varepsilon_g \right) + \varepsilon_n \right| \right) \Big/ \varepsilon_n \right|$$

5-18

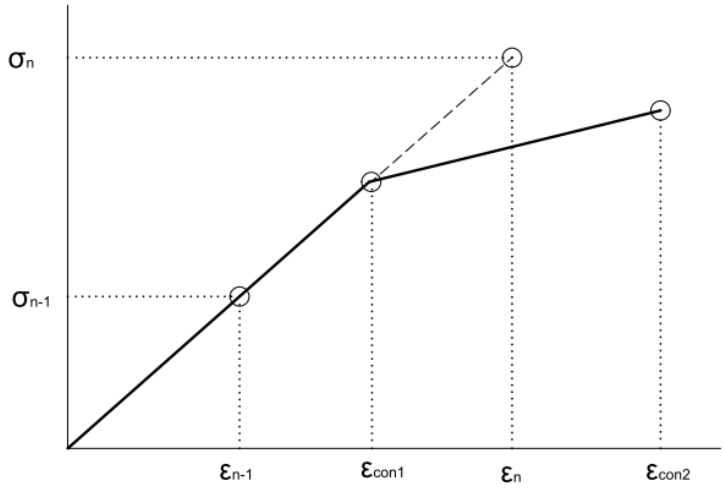


Figure 20: Acceptance criteria and load increase / reduction

This check can be carried out for all elements at each gauss point in the ε_x and ε_y in a given problem. From checking the strain limits at Gauss points and if one or more of these points have passed convergence requirements, stresses and other required results can be calculated and saved for the current load step, L_n. Following the saving of the incremental results, the new modified E-modulus must be assigned for the Gauss points within convergence limits or equal to strain limits as dictated by the input stress-strain graph.

The modified E-modulus can be calculated using equation 5-19:

$$E_t = \frac{\sigma_t - \sigma_{t-1}}{\varepsilon_t - \varepsilon_{t-1}}$$

5-19

Where *t* is the number of the control points given in the calculated input stress-strain graph. All required E-modulus values are calculated prior to analysis, the element stiffness matrices

can be reformulated and global stiffness matrix reassembled. With smaller systems, the total global stiffness matrix is reformulated as there is little computational cost.

The reassembly process can become time-consuming with a large model and it would be a great advantage to simplify this step and manipulate the global stiffness matrix with minimal computational cost. A simple method to reduce the cost of reformulation is proposed for larger systems.

The global stiffness matrix can be updated directly by only considering the elements which have reached yield. This can also be further extended to only include elements of a greater model which have been identified as critical for nonlinear design checks.

The idea is to formulate three separate stiffness matrices, the first being the current global system stiffness matrix, K_{sys}, at load step x and iteration i. The second matrix would be the current stiffness matrix, K_f, of only the elements which have been identified as yielding under load step x and contributing to the global system stiffness matrix. The third stiffness matrix, K_g, is the updated stiffness matrix of those elements which have been considered as yielding.

The three matrices are combined to give the updated global system stiffness matrix. The summation of the three stiffness matrices which allow for updated global stiffness matrix is given by equation 5-20.

$$K_{sys}^{i} = K_{sys}^{i-1} - K_{f}^{i} + K_{g}^{i} \qquad \textbf{5-20}$$

Where:

K_{sys} = System stiffness matrix of iteration i

K_f = Pre-modification stiffness matrix of iteration i

K_g = Modification stiffness matrix of iteration i

The above method is seen as efficient, especially for the large assemblies, as not all element stiffness matrices are reformulated and reassembled. The algorithm flow charts for the small and large systems are given in Figure 21 and Figure 22 respectively.

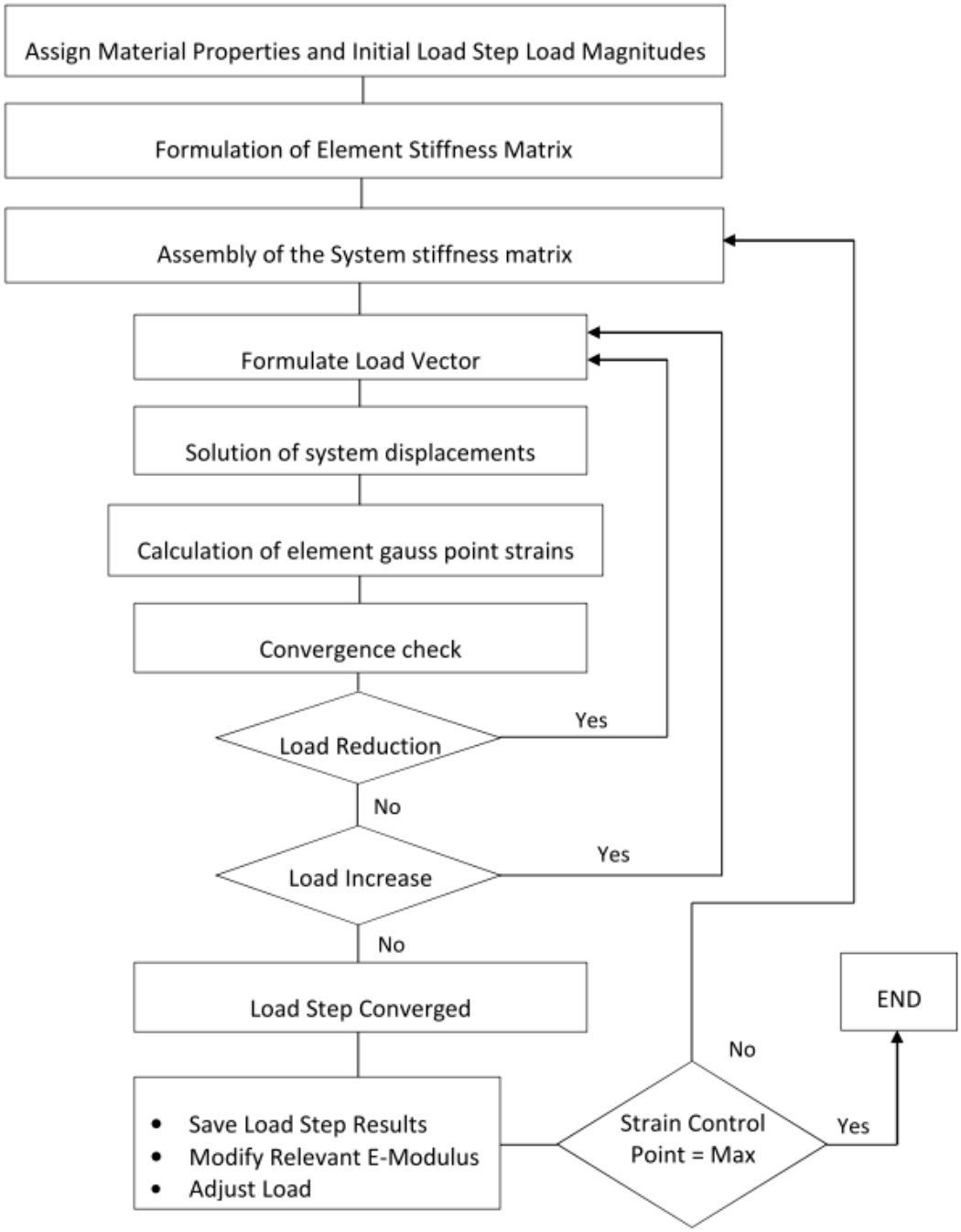


Figure 21: Small system nonlinear solver algorithm

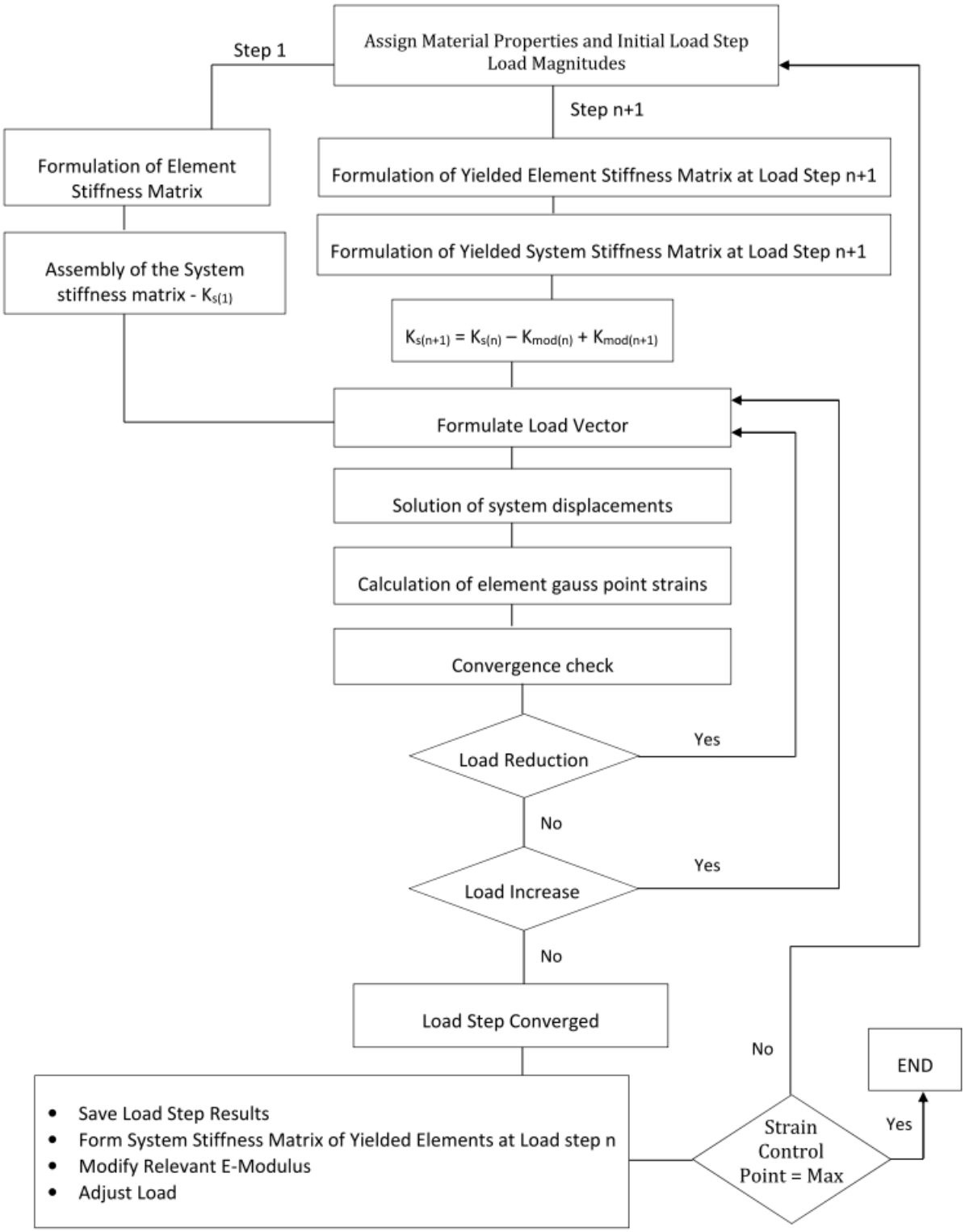


Figure 22: Large system nonlinear solver algorithm

CHAPTER 6: TEST CASE STUDIES

6.1 EXPERIMENTAL TEST CASE OVERVIEW

To validate the simplified analysis procedure four experimental test cases were evaluated. The four cases presented were the same as used by Polak (1996) [32]. A summary of the test cases is given in Table 2.

Table 2: Experimental Data Summary

Testers	Date	Test Specimen	Loading Type	Action	Support	Support Type
Polak	1994	SM1	Edge Moment	One-Way	Edge	Simple
McNeice	1967	TWS	Middle Point Load	Two-Way	Corner	Simple
Ghomein & MacGregor	1994	C1	UDL	Two-Way	Edge	Simple
Aghayere & MacGregor	1990	A3	UDL	Two-Way	Edge	Simple

6.2 PROPERTIES OF CASE STUDIES

The following section gives a brief description of the test cases considered when validating the proposed methodology. The test specimens' properties are given along with a brief description on the test methodology. Certain important aspects of the experimental findings are pointed out which are used for later comparison.

6.2.1 Polak SM1

Polak and Vecchio (1994) [34] carried out a series of experiments on reinforced concrete slabs subject to bending and in-plane forces. One of these experiments was carried out on a slab test specimen SM1. In the experiment, a simply supported one-way spanning slab is subjected to an increasing edge moment loading in one direction only, Figure 23.

During the experiment cracking was reported at an applied moment of 75 kN.m/m with first yielding at a moment of 440 kN.m/m and the testing stopped at moment of 464 kN.m/m due to excessive deflections. It is further noted that a maximum ultimate moment loading of 477 kN.m/m was observed. Moment curvature results were established and are used for the current evaluation. The validity of the present effective stiffness approach can be established by comparing these experimental curvature results against results obtained through analysis.

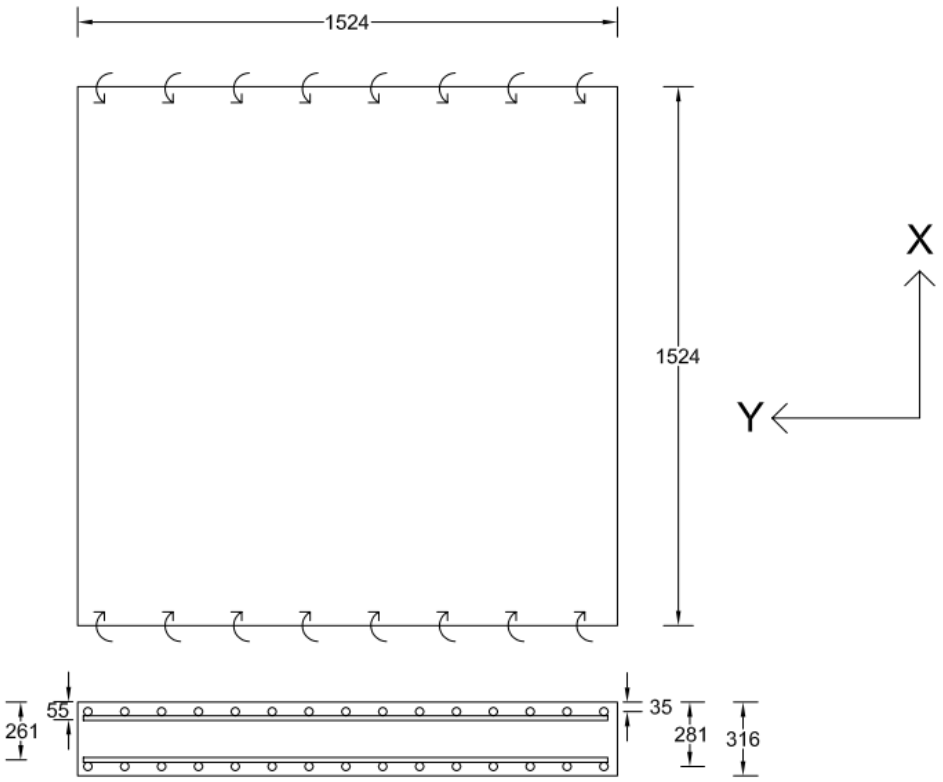


Figure 23: Polak and Vecchio (1994) SM1

For the nonlinear analysis, the SM1 slab was modelled using one 8-noded element. The element is modelled as simply supported at its ends as shown in Figure 24. The moment loading was applied at the simply supported ends as a nodal moment and proportioned based on a parabolic interpolation function.

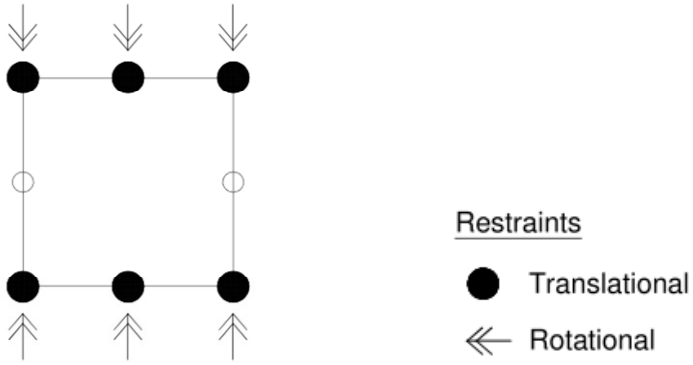


Figure 24: SM1 finite element model restraints

Using the proposed method given in Section 4.2 and material properties given in Table 3 a typical stress-strain diagram is developed for the SM1 Slab and shown in Figure 25.

Table 3: Slab SM1 Material Properties

Polak & Vecchio	
h (mm)	316
L (mm)	1 524
B (mm)	1 524
v	0.20
Ec (MPa)	34 278
Es (MPa)	200 000
fc' (MPa)	47
ft' (MPa)	4.46**
fy (MPa)	425
fu (MPa)	611
Ast-x (mm^2/m)	3 950
Asc-x (mm^2/m)	3 950
Ast-y (mm^2/m)	1 327
Asc-y (mm^2/m)	1 327
d'x (mm)	35
d'y (mm)	55
dx (mm)	281
dy (mm)	261

**Back calculated from recorded experimental values

Cracking is calculated to occur at an equivalent stress of 4.46 MPa. The yielding equivalent stress is calculated at 25.37 MPa and the ultimate equivalent stress is calculated at 27.30 MPa. The intermediate stress point is calculated at a stress of 12.69 MPa.

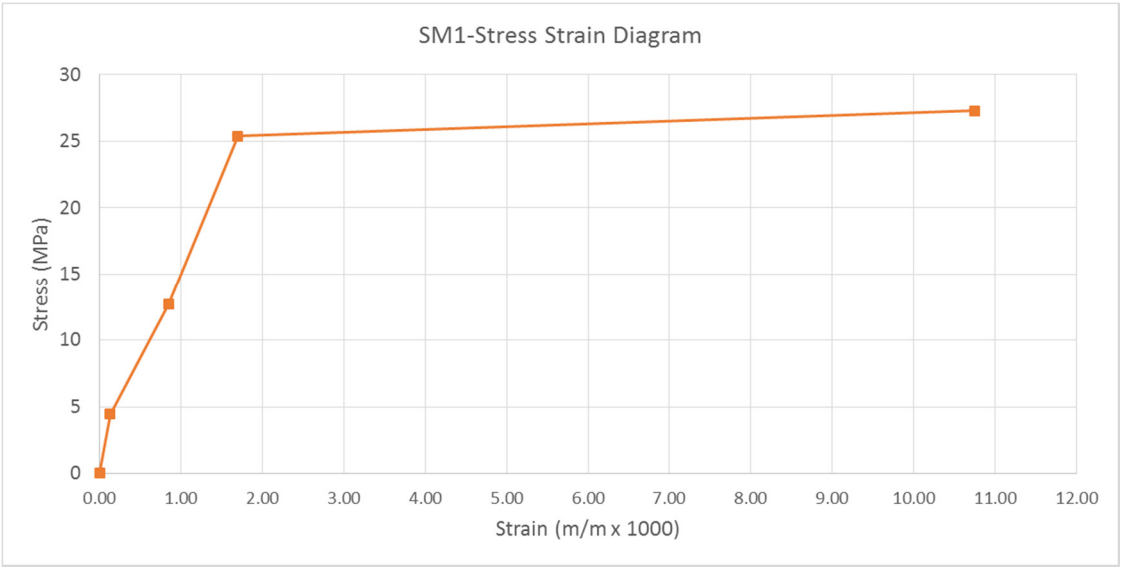


Figure 25: Slab SM1 stress-strain diagram

6.2.2 McNeice

McNeice (1967) [27] carried out a series of experiments on four reinforced concrete slabs. From these studies, Slab No. 1 was chosen for use in the research. The experiment comprised a square slab simply supported at each corner and subjected to a point load in the centre of the slab, Figure 26. As the point load magnitude is increased, deflection measurements are taken at four different points of the slab.

It is noted that the measurements are taken prior to the onset of yielding. Crack patterns were drawn at elastic limit stages of 0.557 and 0.855 of the analytical collapse load calculated by the method presented by McNeice. These patterns give insight as to the pattern and extent of cracking between the real behaviour and computer simulation behaviour.

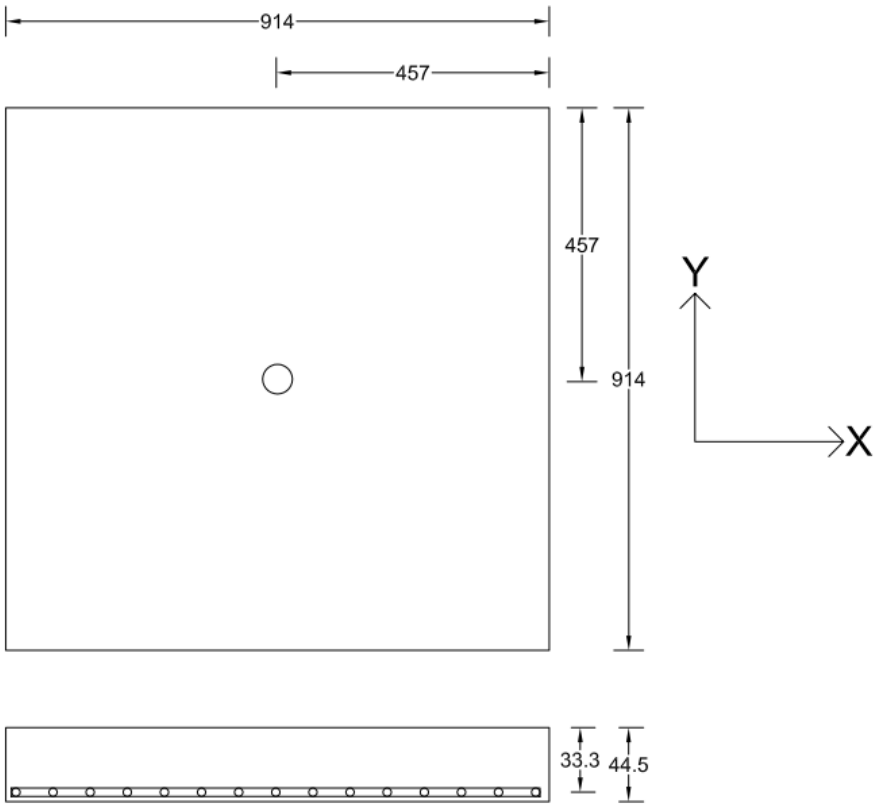


Figure 26: McNeice (1967) [27] slab No. 1

For the nonlinear analysis, the McNeice slab was modelled using quarter symmetry model. The quarter symmetry model comprises a 3x3 8-noded plate element mesh. The McNeice No. 1 slab is modelled with restraints as shown in Figure 27. The out of plane nodal point loading was applied at node 7 as given in Figure 16.

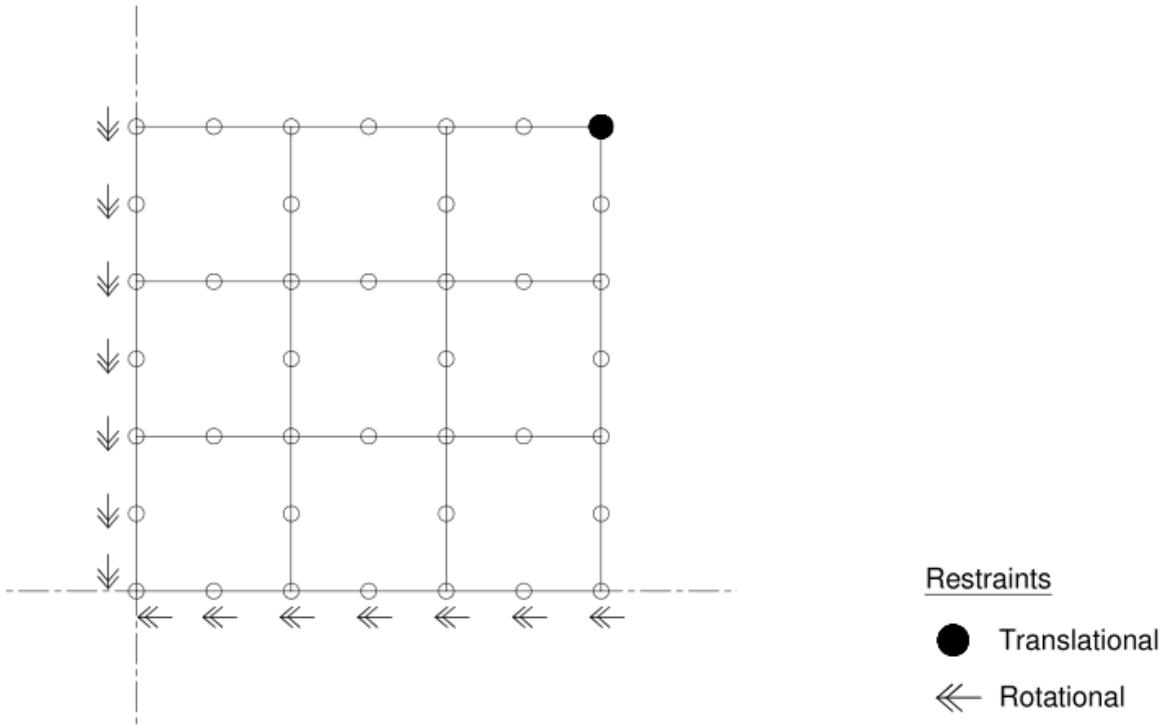


Figure 27: McNeice finite element model restraints

Material properties used in the analysis are given in Table 4 with the associated stress-strain diagram is for the McNeice Slab and shown in Figure 28.

Table 4: McNeice Slab Material Properties

McNiece Slab No. 1	
h (mm)	44
L (mm)	914
B (mm)	914
v	0.15
Ec (MPa)	28 613
Es (MPa)	199 948
fc' (MPa)	37.92
ft' (MPa)	3.82
fy (MPa)	380**
fu (MPa)	485**
Ast-x (mm^2/m)	380
Asc-x (mm^2/m)	0
Ast-y (mm^2/m)	380
Asc-y (mm^2/m)	0
d'x (mm)	0
d'y (mm)	0
dx (mm)	33
dy (mm)	33

**Back calculated from recorded experimental values

Cracking is calculated to occur at an equivalent stress of 3.82 MPa. The yielding equivalent stress is calculated at 13.00 MPa and the ultimate equivalent stress is calculated at 17.02 MPa. The intermediate stress point is calculated at a stress of 6.50 MPa.

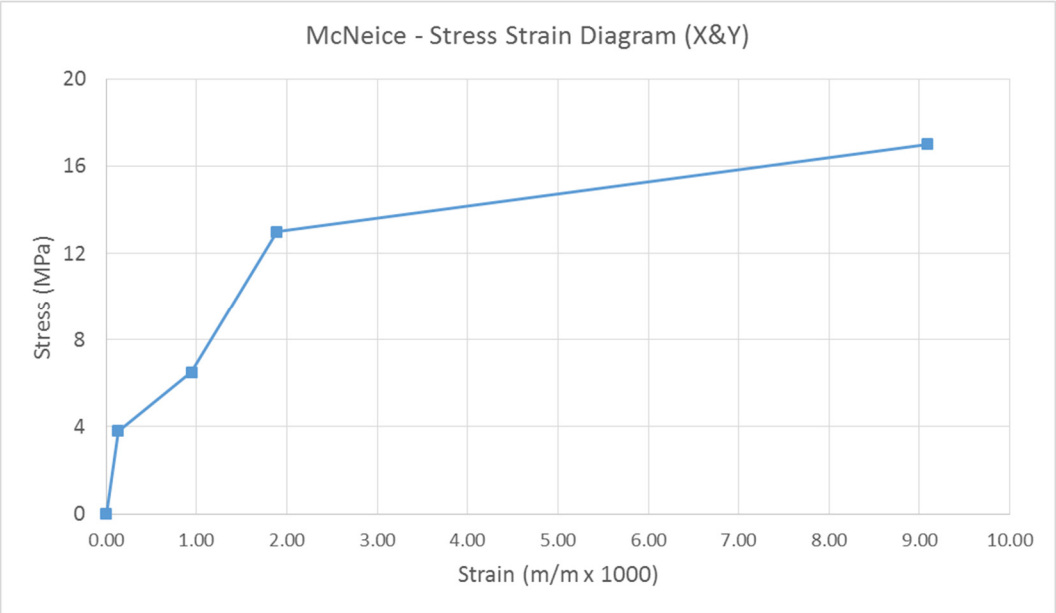


Figure 28: McNeice slab stress-strain diagram X & Y directions

6.2.3 Ghomein & MacGregor

Another experimental test case is that of a two-way spanning slab, specimen C1, tested by Ghomein and MacGregor (1992) [16]. The slab was simply supported along all four edges with the uniformly distributed loading being simulated using nine concentrated loads over the slab surface as shown in Figure 29. Cracking was first observed at a load of 9.1 kPa.

Fully developed yield lines were predicted to occur at a loading of 42.8 kPa. The fully developed yield lines predictions were based on a 0.2% offset yield neglecting the effects of corner levers.

It was reported that although the yield lines were present, load carrying capacity was not inhibited. The increase in load carrying capacity was seen to be attributed to an increased combined bending and tensile membrane actions. An experimental ultimate loading of approximately 73.9 kPa was observed.

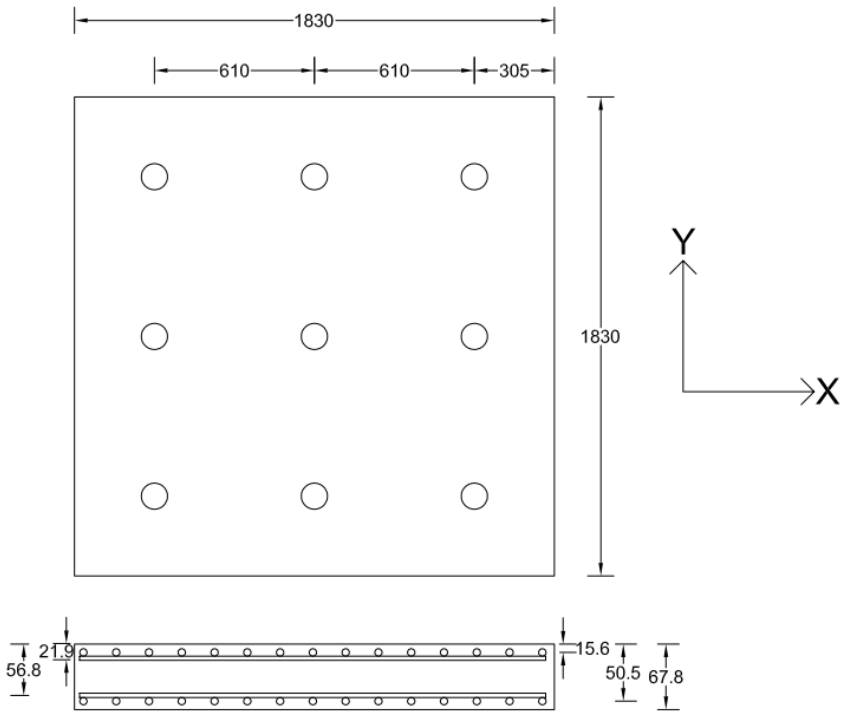


Figure 29: Ghomein & MacGregor (1992) [16] C1 slab

For the nonlinear analysis, the C1 slab was modelled using quarter symmetry model. The quarter symmetry model comprises a 3x3 8-noded plate element mesh. The C1 slab is modelled with restraints as shown in Figure 30. The experimental loading is represented by applying an out of plane proportional nodal point loading at nodes 7, 3, 25 and 29 as given in Figure 16.

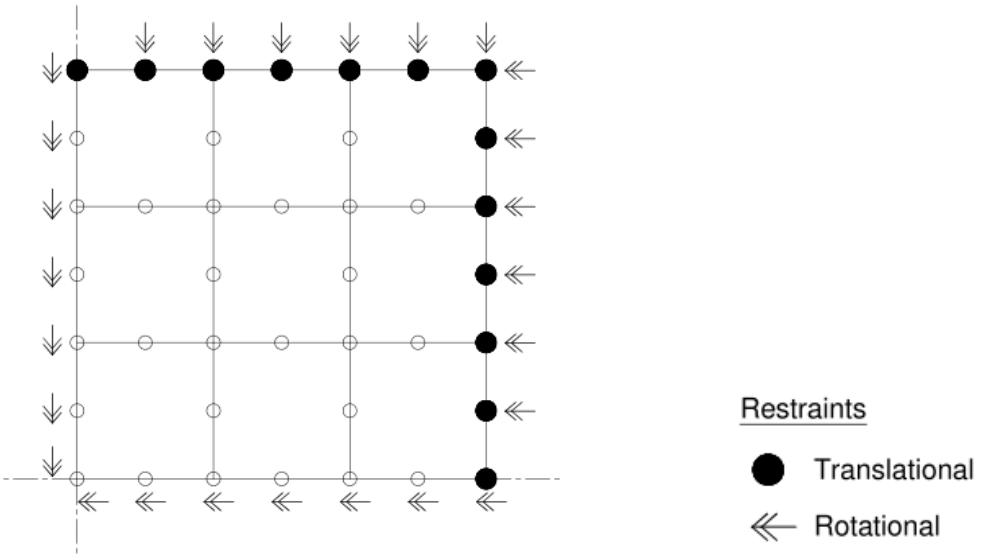


Figure 30: C1 and A3 finite element model restraints

Material properties used in the analysis are given in Table 5 with the associated stress-strain diagrams for the x and y directions of reinforcement of the C1 Slab are shown in Figure 31 and Figure 32 respectively.

Table 5: C1 Slab Material Properties

Ghomein & MacGregor (C1)	
h (mm)	68
L (mm)	1 830
B (mm)	1 830
v	0.20
Ec (MPa)	21 300
Es (MPa)	181 500
fc' (MPa)	25.21
ft' (MPa)	1.76**
fy (MPa)	450
fu (MPa)	620
Ast-x (mm^2/m)	260
Asc-x (mm^2/m)	260
Ast-y (mm^2/m)	260
Asc-y (mm^2/m)	260
d'x (mm)	22
d'y (mm)	16
dx (mm)	57
dy (mm)	51

**Back calculated from recorded experimental values

When evaluating the X-direction, cracking is calculated to occur at an equivalent stress of 1.69 MPa. The yielding equivalent stress is calculated at 8.36 MPa and the ultimate equivalent stress is calculated at 10.61 MPa. The intermediate stress point is calculated at a stress of 4.18 MPa.

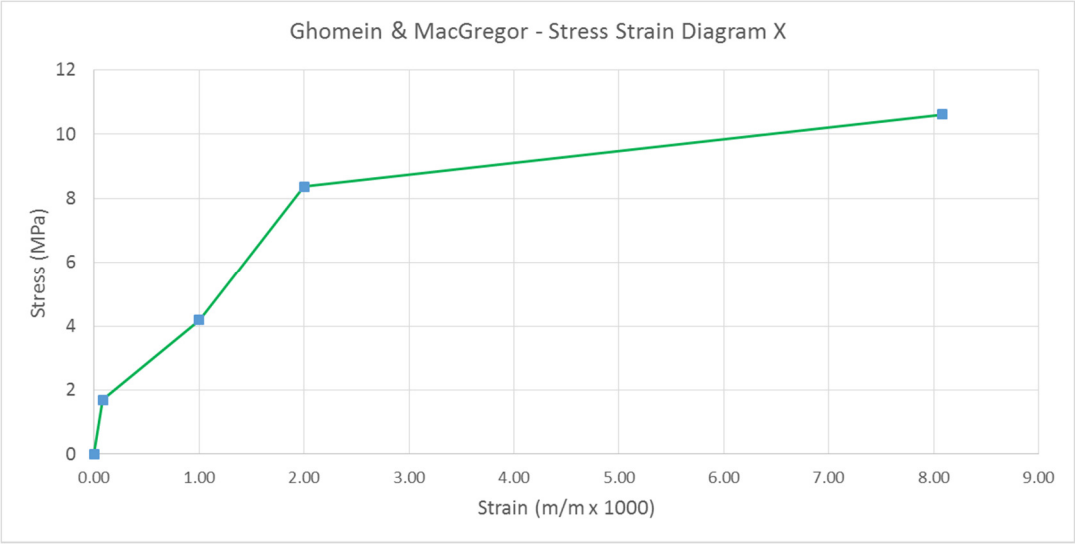


Figure 31: C1 slab stress-strain diagram X directions

When evaluating the Y-direction, cracking is calculated to occur at an equivalent stress of 1.69 MPa. The yielding equivalent stress is calculated at 7.14 MPa and the ultimate equivalent stress is calculated at 8.48 MPa. The intermediate stress point is calculated at a stress of 3.57 MPa.

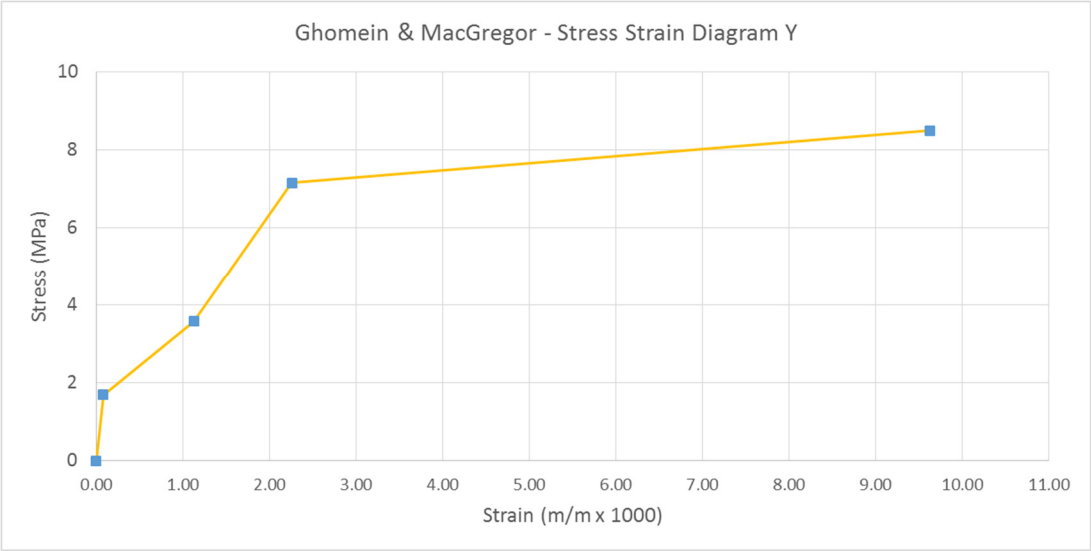


Figure 32: C1 slab stress-strain diagram Y directions

6.2.4 Aghayere & MacGregor

Prior to the experimentation carried out by Ghomein and MacGregor (1992) [16], Aghayere & MacGregor (1990) [3] carried out similar experimentation also to evaluate effects of in-plane forces. Slab specimen A3 from these experiments was chosen as the final test case becuase this slab was tested with no in-plane forces applied. The testing equipment and methodology is the same as given in Section 6.2.3 as these were carried out in the same laboratory. The slab dimensions were the same but reinforcement details were different. This then serves as a good method of comparison of the proposed nonlinear model to two separate slabs tested under the same conditions but with differing properties.

As with the C1 slab, the A3 slab was simply supported along all four edges with uniformly distributed loading being simulated using nine concentrated loads over the slab surface as shown in Figure 33. Through the evaluation of the load deflection graph given from the experimentation, cracking was first observed at a load of approximately 10.2 kPa.

It is noted that the A3 slab was not tested to final failure. The experimentation was terminated before the ultimate loading due to excessive deflections and rotations of the supports. A maximum experimental centre point deflection of 95 mm was obtained. The yield line load of the slab is calculated to be 43.6 kPa. The experimental loading reached a maximum of approximately 56.7 kPa.

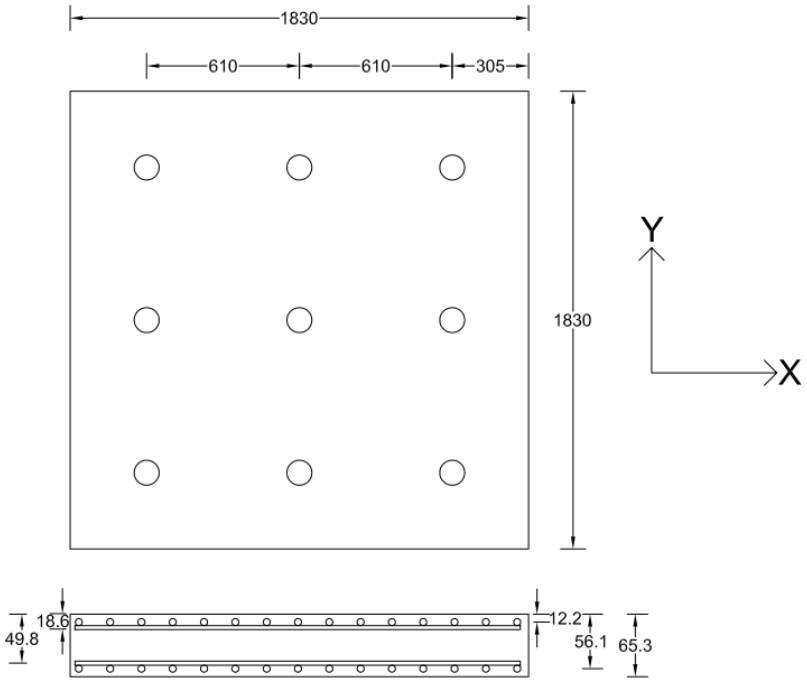


Figure 33: Aghayere & MacGregor (1990) [3] A3 slab

Material properties used in the analysis are given in Table 6 with the associated stress-strain diagrams for the x and y directions of reinforcement of the A3 Slab are shown in Figure 34 and Figure 35 respectively.

Table 6: A3 Slab Material Properties

Aghayere & MacGregor (A3)	
h (mm)	65
L (mm)	1 830
B (mm)	1 830
v	0.20
Ec (MPa)	23 150
Es (MPa)	197 300
fc' (MPa)	32.20
ft' (MPa)	2.27**
fy (MPa)	504
fu (MPa)	670
Ast-x (mm^2/m)	225
Asc-x (mm^2/m)	225
Ast-y (mm^2/m)	260
Asc-y (mm^2/m)	260
d'x (mm)	19
d'y (mm)	12
dx (mm)	50
dy (mm)	56

**Back calculated from recorded experimental values

The A3 slab was modelled the same as the C1 slab and is shown in Figure 30.

When evaluating the X-direction, cracking is back calculated to occur at an equivalent stress of 2.27 MPa. The yielding equivalent stress is calculated at 7.61 MPa and the ultimate equivalent stress is calculated at 10.03 MPa. The intermediate stress point is calculated at a stress of 3.81 MPa.

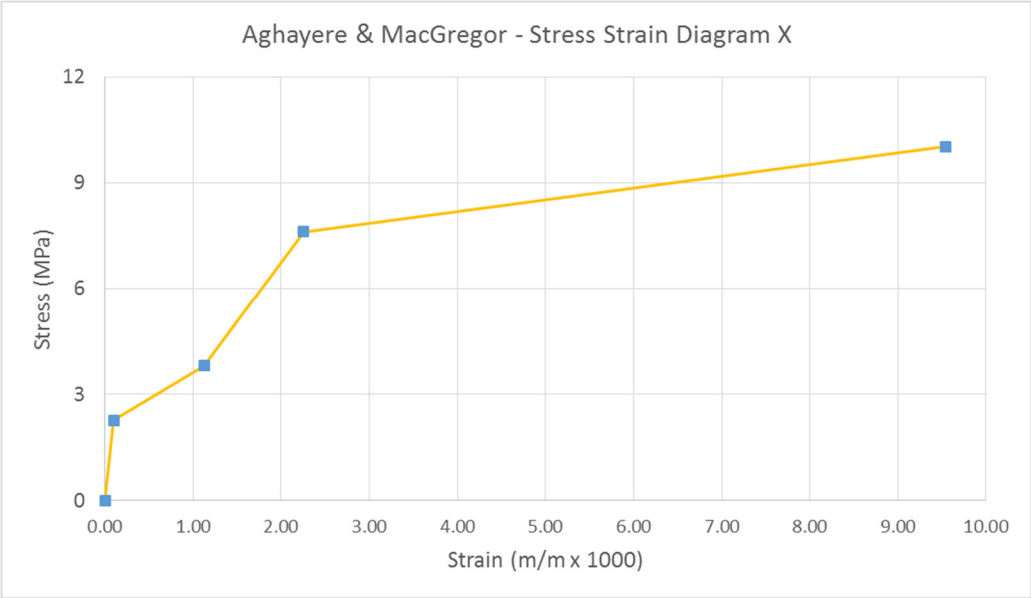


Figure 34: A3 slab stress-strain diagram X directions

When evaluating the Y-direction, cracking is back calculated to occur at an equivalent stress of 2.27 MPa. The yielding equivalent stress is calculated at 9.51 MPa and the ultimate equivalent stress is calculated at 11.49 MPa. The intermediate stress point is calculated at a stress of 4.76 MPa.

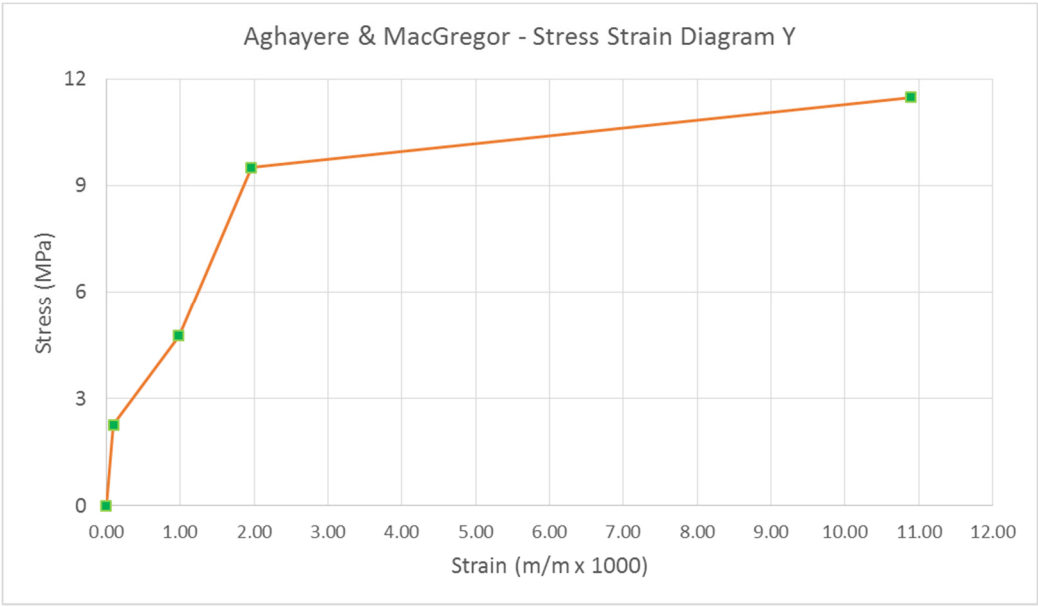


Figure 35: A3 slab stress-strain diagram Y directions

CHAPTER 7: RESULTS

7.1 INCREMENTAL ASSESSMENT OF NONLINEAR ALGORITHM

Incremental results are extracted and hand calculations performed to evaluate whether the algorithm has been correctly implemented according to the proposed method. The hand calculations solve for the relevant equations as given in Section 5.2 using the incremental results as inputs. The resulting solution is then compared to the input strain and stress values as well as the expected moment.

The hand calculations were carried out for the one-way spanning Polak slab and two-way spanning McNeice slab in the X direction only. This is seen as sufficient to evaluate the correct implementation of the proposed method. To enable the incremental evaluation the effective E-modulus in the X and Y directions as well as Poisson's ratio are required to be extracted at the relevant stages of analysis.

The expected design strain value is then calculated according to equation 5-13 and compared to the input strain control point as given in the input stress-strain data. The design moments are calculated using equation 5-4 with tan(θ) equal to unity. These design moments are then compared to the section moment capacity as used in equation 4-6. The design stress is calculated in the same manner as the design moments and compared to the stress value as given in the input stress-strain data. The hand calculated design strain and stress values are also plotted and compared to the input stress-strain diagram to obtain a visual on the variance under the proposed algorithm.

Results for the evaluation of the Polak slab are taken from Gauss point number 5 as shown in Figure 36. When reviewing the analysis of the Polak slab the incremental E-modulus for the X and Y directions are shown only to change in the direction of loading, X direction. The change in effective E-modulus and Poisson's ratio is given in Table 7.

Table 8 shows the incremental strain, stress and moment results for the Polak slab as given by the finite element analysis. These results are then used as inputs for the calculation of design values. The design strain, stress and moment quantities are given in columns (1), (4) and (7) of Table 9 respectively. The design quantities are compared to the input strain, stress and moments quantities given in columns (2), (5) and (8) of Table 9 respectively. The variance of the strain, stress and moment values using the input quantities as a base are given in columns (3), (6) and (9) of Table 9 respectively.

The variance of the strain values can also be seen as the convergence value as defined by equation 5-17. It can therefore be expected that this value should never exceed the input strain value or be less than the input strain value by more than 1%.

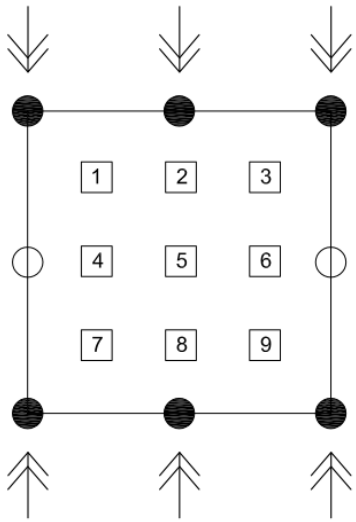


Figure 36: Polak slab Gauss points

For the Polak slab the strain variance has a peak of 0.406% which indicates the automatic convergence check equation 5-17 and loading equation 5-18 are being carried out correctly. The design stress and design moment quantities do not vary significantly from the expected input stress and moments values. For the Polak slab Figure 37 shows an illustrative comparative of the variance between the input stress-strain values and those calculated from the results obtained from the nonlinear analysis. The variance is seen to be insignificant in a global evaluation.

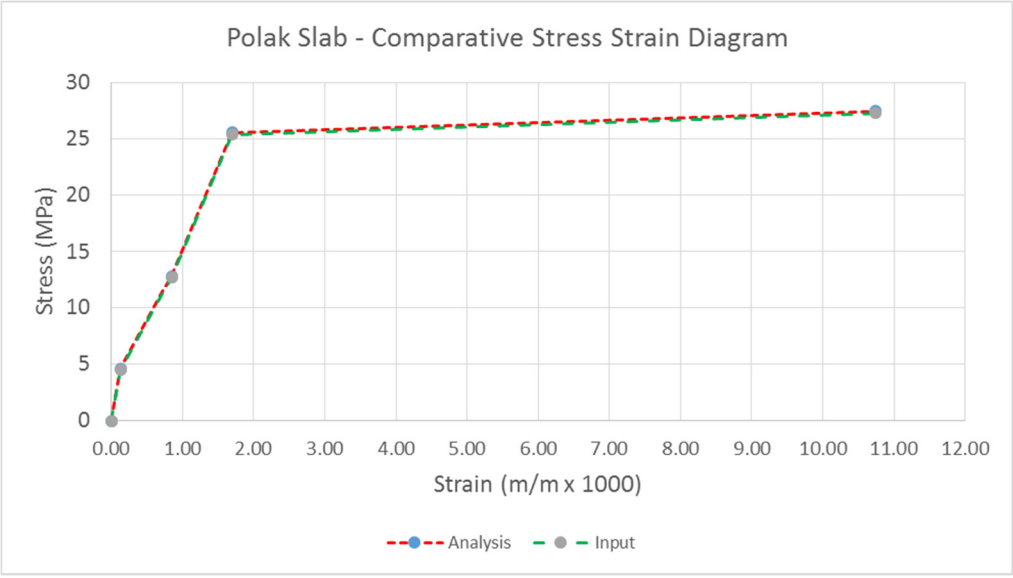


Figure 37: Stress-strain comparative analysis vs input – Polak slab

Table 7: Incremental E-moduli and Poisson's Ratio – Polak slab

Polak XX-Direction (SI Units)				
	(1)	(2)	(3)	(4)
	E_x	E_y	v_x	v_y
Cracking	3.43E+10	3.43E+10	0.20	0.20
Half Point	1.15E+10	3.43E+10	0	0.20
Yield	1.50E+10	3.43E+10	0	0.20
Ultimate	2.13E+08	3.43E+10	0	0.20

Table 8: Incremental strain, stress and moment results – Polak slab

Polak XX-Direction (SI Units)							
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
	ε_{xx}	ε_{yy}	ε_{xy}	σ_{xx}	σ_{xy}	m_{xx}	m_{xy}
Cracking	1.30E-04	-4.39E-06	8.60E-21	4.62E+06	1.23E-10	7.69E+04	2.36E-12
Half Point	7.18E-04	-1.35E-05	1.33E-19	8.23E+06	9.92E-10	1.37E+05	1.65E-11
Yield	8.49E-04	-1.75E-05	1.71E-19	1.27E+07	1.58E-09	2.11E+05	2.63E-11
Ultimate	9.05E-03	-9.56E-06	9.04E-18	1.93E+06	1.52E-09	3.21E+04	2.53E-11

Table 9: Cumulative design strain, stress and moment with variance – Polak slab

Polak XX-Direction (SI Units)									
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	$\Sigma\varepsilon_{tx}$	ε_{ix}	$\%\Delta\varepsilon_x$	$\Sigma\sigma_{tx}$	σ_{ix}	$\%\Delta\sigma_x$	Σm_{tx}	m_{ix}	$\%\Delta m_x$
Cracking	1.29E-04	1.30E-04	0.406	4.62E+06	4.46E+06	3.744	7.69E+04	7.73E+04	0.405
Half Point	8.47E-04	8.48E-04	0.094	1.29E+07	1.27E+07	1.348	2.14E+05	2.11E+05	1.349
Yield	1.70E-03	1.70E-03	0.004	2.55E+07	2.54E+07	0.659	4.25E+05	4.22E+05	0.660
Ultimate	1.07E-02	1.07E-02	0.002	2.75E+07	2.73E+07	0.623	4.57E+05	4.54E+05	0.624

The McNeice slab results were also inspected to confirm the algorithm was operating correctly for the two-way spanning slab cases. Results are extracted from Gauss point 9, element 3, for the evaluation of the algorithm as shown in Figure 38. This point is chosen as it was expected that yielding of the slab would initially begin at this location.

The same method of result evaluation was followed for the McNeice as was followed for the Polak slab. The incremental E-modulus and Poisson's ratio is shown in Table 10 with incremental analysis strain, stress and moment results given in Table 11. The hand calculated design strains, stresses and moments with associated input values and variances are shown in Table 12.

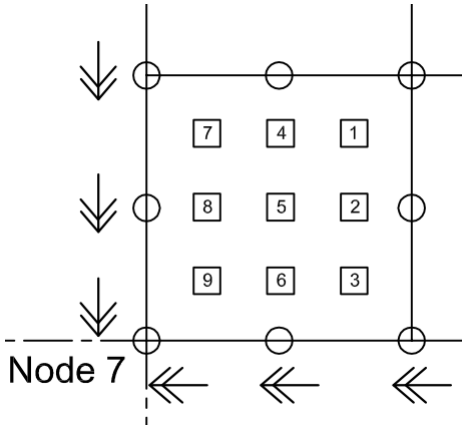


Figure 38: McNeice slab Gauss points, Reference Figure 16 and Figure 27

The results from the McNeice slab confirm conclusions drawn from the Polak slab incremental analysis and evaluation. The design strain is seen to never exceed the input strain values with the related variance never greater than 0.605%. The design stress and moments values again do not vary considerably from the expected input values. The incremental evaluation of the McNeice slab therefore also shows that the nonlinear algorithm is operating as expected according to the proposed methodology.

Table 10: Incremental E-moduli and Poisson's Ratio – McNeice slab

McNeice XX-Direction (SI Units)				
	(1)	(2)	(3)	(4)
	E_x	E_y	v_x	v_y
Cracking	2.86E+10	2.86E+10	0.15	0.15
Half Point	3.31E+09	3.31E+09	0	0
Yield	6.88E+09	6.88E+09	0	0
Ultimate	5.59E+08	5.59E+08	0	0

Table 11: Incremental strain, stress and moment results – McNeice slab

McNeice XX-Direction (SI Units)							
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
	ε_{xx}	ε_{yy}	ε_{xy}	σ_{xx}	σ_{xy}	m_{xx}	m_{xy}
Cracking	1.13E-04	1.13E-04	-8.08E-06	3.79E+06	-1.01E+05	1.25E+03	-3.32E+01
Half Point	7.64E-04	7.64E-04	-8.20E-05	2.52E+06	-1.35E+05	8.33E+02	-4.47E+01
Yield	9.16E-04	9.16E-04	-5.19E-05	6.35E+06	-1.81E+05	2.10E+03	-5.97E+01
Ultimate	6.88E-03	6.88E-03	-6.39E-04	3.85E+06	-1.79E+05	1.27E+03	-5.89E+01

Table 12: Cumulative design strain, stress and moment with variance – McNeice slab

McNeice XX-Direction (SI Units)									
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	$\Sigma\varepsilon_{tx}$	ε_{ix}	$\%\Delta\varepsilon_x$	$\Sigma\sigma_{tx}$	σ_{ix}	$\%\Delta\sigma_x$	Σm_{tx}	m_{ix}	$\%\Delta m_x$
Cracking	1.33E-04	1.33E-04	0.323	3.79E+06	3.82E+06	0.663	1.25E+03	1.29E+03	2.601
Half Point	9.38E-04	9.44E-04	0.605	6.32E+06	6.50E+06	2.801	2.09E+03	2.14E+03	2.489
Yield	1.88E-03	1.89E-03	0.431	1.27E+07	1.30E+07	2.542	4.18E+03	4.28E+03	2.253
Ultimate	9.08E-03	9.08E-03	0.014	1.65E+07	1.70E+07	2.990	5.45E+03	5.61E+03	2.701

The results for the McNeice slab are also expressed graphically to determine the effectiveness of the algorithm in implementing the proposed nonlinear model. Figure 39 shows a comparison of the input stress-strain values to that of the values calculated from results obtained through analysis. It is noted that there is not a large variation between the two instances.

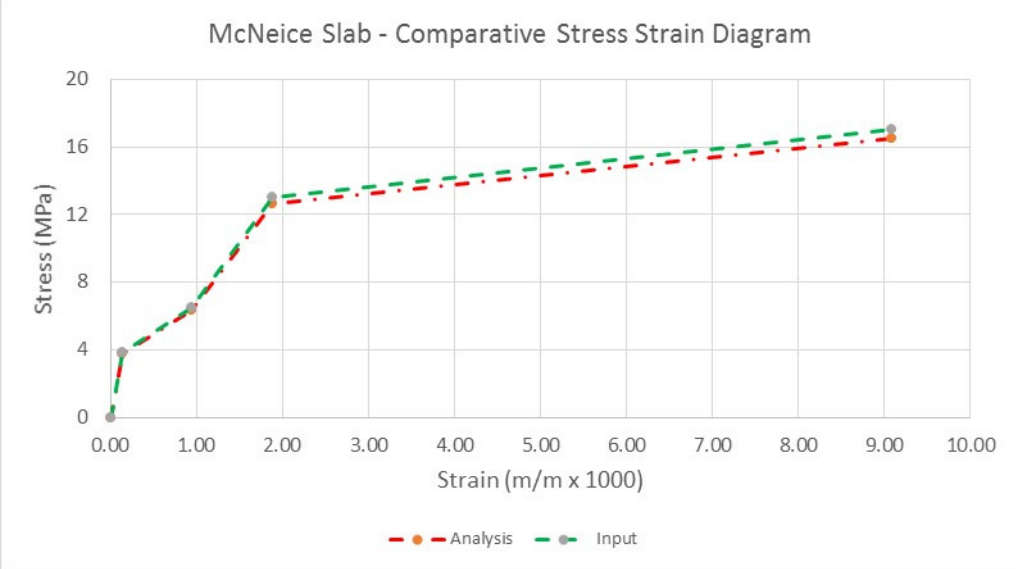


Figure 39: Stress-strain comparative analysis vs input – McNeice slab

7.2 POLAK RESULTS

Following the analysis of the SM1 slab the moment and curvature results are extracted and plotted as shown in Figure 40. Through the analysis initial cracking occurred at a moment of 77 kN.m/m varying by 2.6% when compared to the experimental value of 75 kN.m/m. The analytical yielding moment was seen to occur at a moment of 425 kN.m/m. The analytical yielding moment, 425 kN.m/m, varies by 3.4% as that given by the Polak slab experiment, 440 kN.m/m. The analytical model continues to take loading until the design ultimate moment capacity, 457 kN.m/m, is reached. Curvatures for the Polak slab analysis are seen to be accurate throughout the analysis.

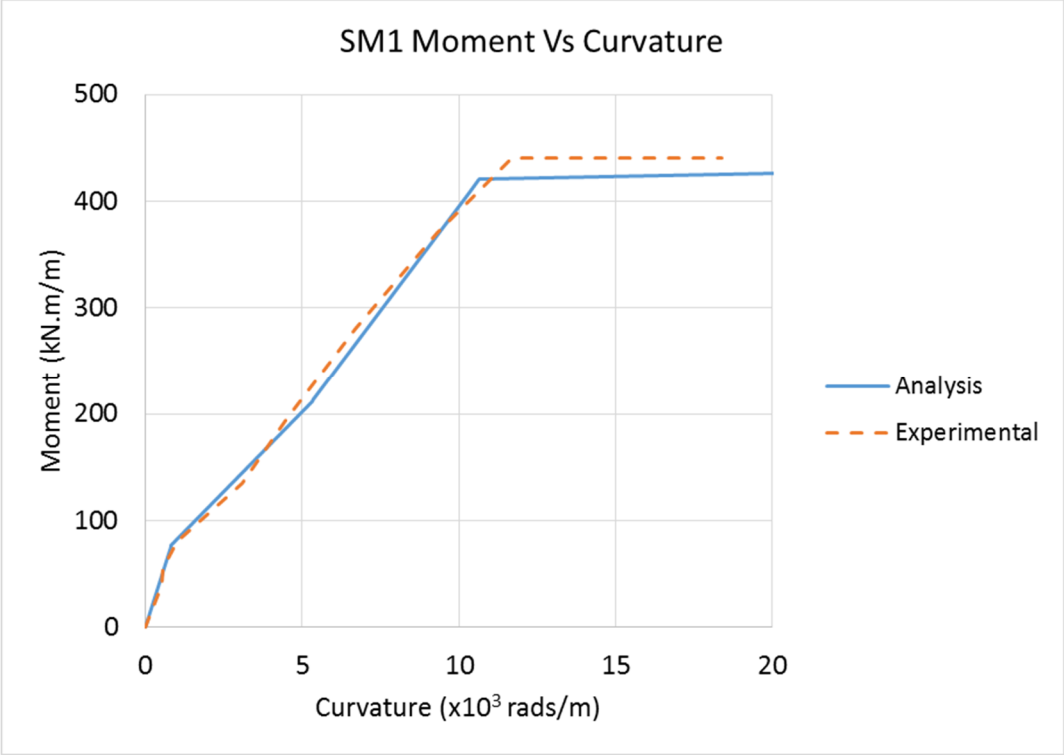


Figure 40: Polak SM1 slab moment-curvature comparison

7.3 MCNEICE RESULTS

The McNeice slab analysis was carried out assuming an ultimate steel stress almost equal to that of the yield steel stress producing a slight increase in moment capacity and therefore a close to perfectly plastic behaviour following the reaching of slab yield moment capacity. Point load magnitudes and vertical nodal translation at node 6 is given in Figure 41.

Significant initial cracking is seen to have occurred at a point load of 5.3 kN with ultimate loading being reached at 22.4 kN. Displacement accuracy is assessed at serviceability points where L/w equals 180 and 360.

In both cases the analytical model predicted these points at a loading approximately 12% lower than that of the experiment. The theoretical model therefore exhibited a less stiff behaviour to that of the experiment. At higher loading the theoretical model predicts the displacements of the McNeice slab more closely.

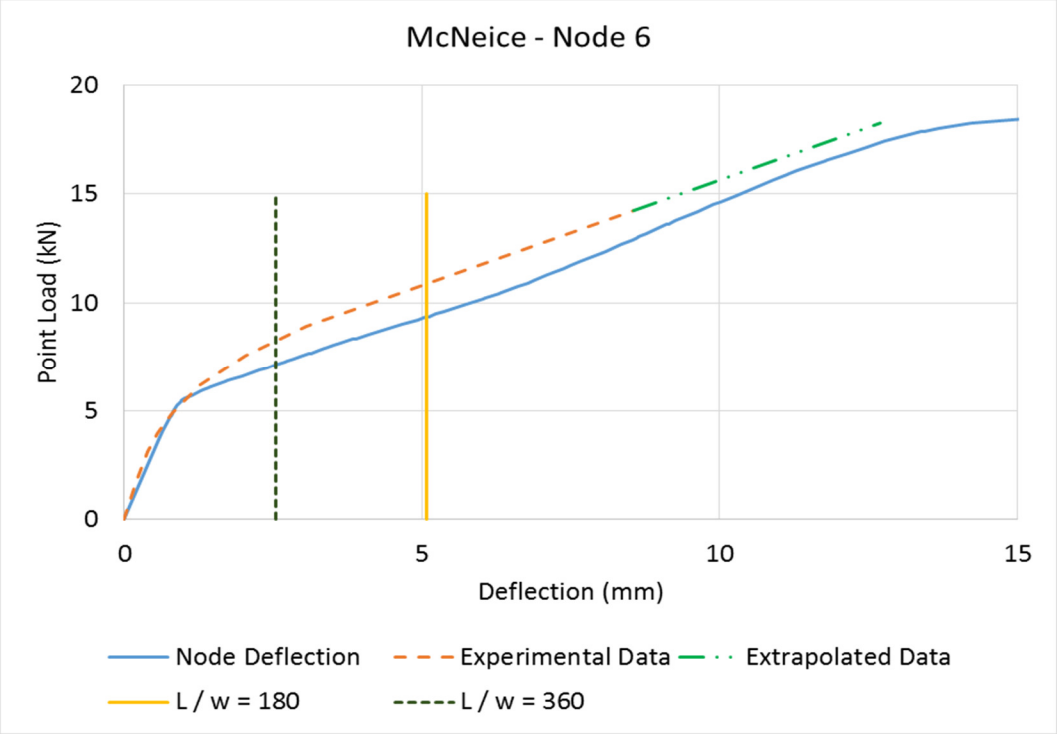


Figure 41: McNeice node 6 load vs displacement

Another aspect which is evaluated is the pattern and extent of yield line development. McNeice established a sketch of the crack pattern at the end of elastic behaviour where the ratio of loading to predicted ultimate loading was 0.56. This sketch, from his research, is compared to the current model at the end of elastic behaviour which was through this study calculated at a loading to predicted ultimate loading ratio equal to 0.52.

As each of the Gauss points are given a unique equivalent E-modulus, the cracking pattern can be established by inspecting the equivalent E-modulus at the end of elastic behaviour where ultimate loading has occurred. When interpreting the crack patterns, a value of 1 indicates to E-modulus given by points 0 and 1 on the typical stress-strain graph, Figure 13, a value of 2 therefore indicates an E-modulus given by points 1 and 1a on the typical stress-strain graph, a value of 3 indicates a point between 1a and 2. When a gauss point has reached ultimate strength, i.e. the point 3 on the typical stress-strain diagram, a value of 4 will be specified.

Figure 42 gives the equivalent E-modulus for Gauss points in the slab in the x and y directions respectively. This equivalent E-modulus for Gauss points is indicative of the actual crack pattern to be expected. The cracking will be normal to the specified axis. Therefore moments in the x-axis will cause cracking normal to this direction and vice versa.

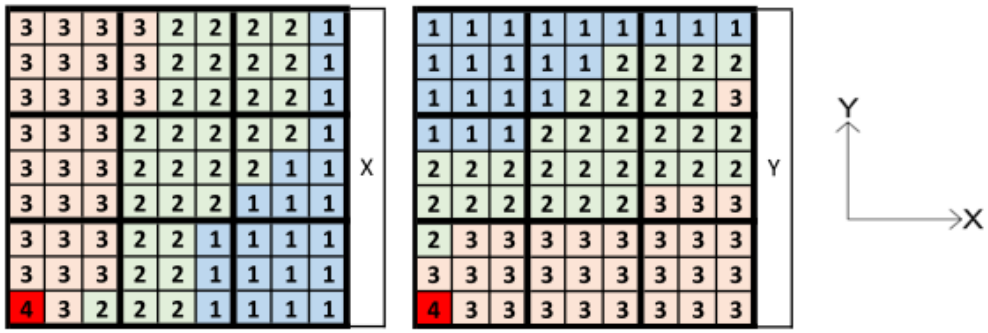


Figure 42: McNeice slab cracking in the x and y directions

Figure 43 is formulated by superimposing the crack patterns in the x and y directions and comparing this to the observed crack pattern established by McNeice and given in Figure 5.1 of reference [27].

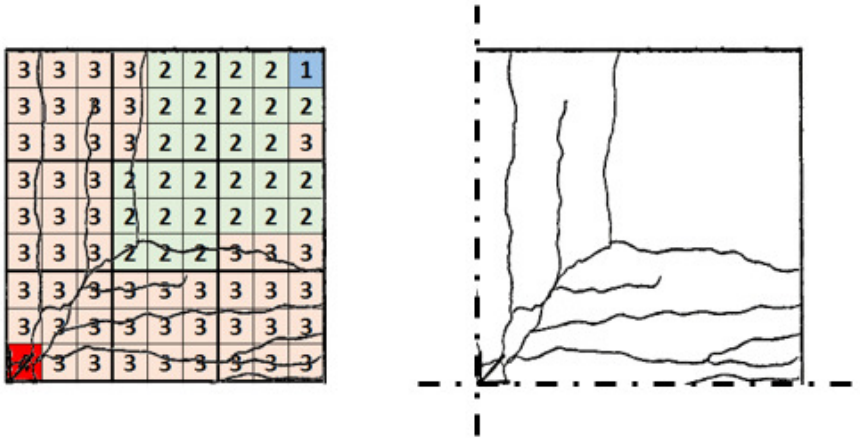


Figure 43: McNeice yield line development pattern and extent

As can be seen the cracking pattern and extent of cracking at the end of elastic behaviour in the x and y directions is seen to be accurate when compared to the McNeice experiment.

7.4 GHOMEIN & MACGREGOR C1 SALB RESULTS

The C1 slab experimentation was carried out to onset of possible failure, the C1 slab analysis produced interesting but expected results which require further explanation.

Two analyses were carried out, where the first is carried out by considering the section ultimate capacity equal to that of the yield moment capacity for evaluation against predicted yield line load given by Ghomein and MacGregor (1992) [15]. The second analysis was carried out by considering the section ultimate capacity equal to that of the design ultimate moment capacity. Figure 44 shows the load displacement results of the two C1 slab analyses.

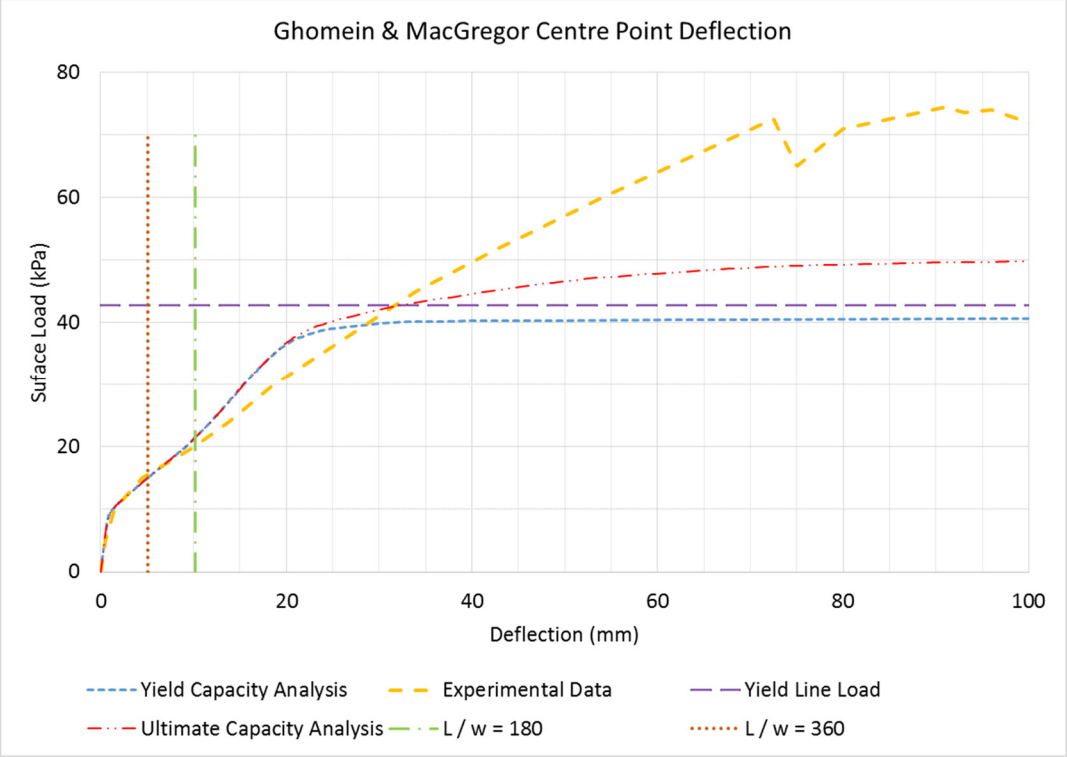


Figure 44: C1 slab centre point load vs deflection

For both analyses the substantial initial cracking and change in stiffness was shown to occur at a loading of approximately 9 kPa. When reviewing the first analysis loading continued to increase and analytical slab yielding was seen and eventual ultimate yield line loading identified as 40.95 kPa. The predetermined yield line load was calculated as 42.80 kPa which gives a difference of 4.3% when compared with the current model's results.

When reviewing the results of the second analysis it should be restated that in the C1 slab experimentation it was reported that the load carrying capacity was not inhibited following the reaching of the predicted yield line load due to an increase in combined bending and tensile membrane actions.

The current analytical theory is based on the premise that the system failure mechanism is considered to be section yielding. It is therefore expected that a difference in the predicted ultimate load and actual ultimate load exist. As can be seen in Figure 44, through the analysis the predicted ultimate load is established at 49.5 kPa where the experimental ultimate load was recorded as 73.9 kPa

Differences in results are considered acceptable when considering displacements at serviceability limits of L / w equal to 180 and 360 with differences in loading at each displacement ratio being 6.5% and 3.6% respectively. The current model can therefore be seen as an accurate representation of the experimental behaviour in these ranges.

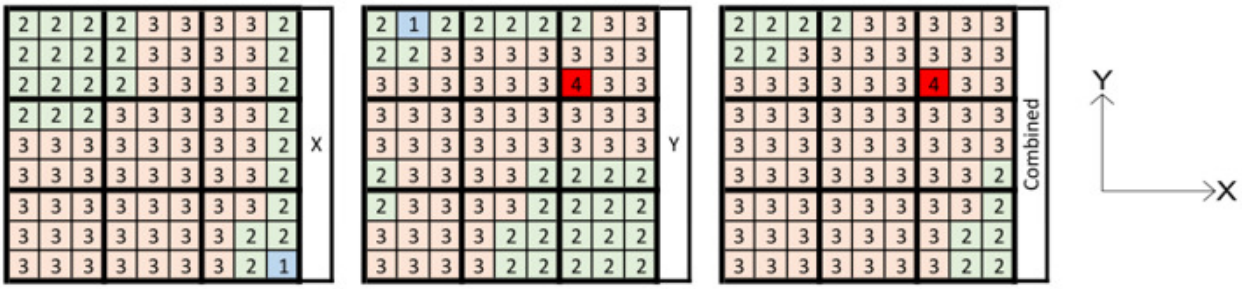


Figure 45: Ghomein & Macgregor slab directional cracking

Although the experimental crack pattern was not visually recorded as with McNeice, an evaluation of the analytical crack pattern is still useful. Through the analysis it was established that the end of elastic behaviour occurred at a loading of 28.26 kPa. The extracted crack pattern in the x and y directions is representative of what would be expected in an experiment comprising the same loading and boundary conditions, Figure 45.

7.5 AGHAYERE & MACGREGOR A3 SLAB RESULTS

The A3 slab was assessed using the same analysis and results evaluation methodology as that of the C1 slab. The resulting behaviour patterns and conclusions follow closely to what was observed with the C1 slab. As with the C1 slab the same two types of analyses were carried out with the first considering the ultimate section capacity and the second considering only the yield line capacity. Figure 46 shows the load displacement results of the two A3 slab analyses.

For both analyses the substantial initial cracking and change in stiffness was shown to occur at a loading of approximately 10 kPa. When reviewing the first analysis loading continued to increase and analytical slab yielding was seen and eventual ultimate yield line loading identified as 41.10 kPa. The predetermined yield line load was calculated as 43.60 kPa which gives a difference of 5.7% when compared with the current model's results.

As is the case with the C1 slab, the A3 slab continued to carrying loading beyond the predicted yield line load. Using the current theory which considers section yielding to be the failure mode an ultimate loading capacity of 51 kPa is determined whereas the experimental ultimate loading can be seen as 56.7 kPa as shown in Figure 46.

Consistency in the accuracy of the analytical model when considering serviceability limits is established when reviewing results obtained from the A3 slab and C1 slab. As with the C1 slab, differences in results obtained from the A3 slab compared to the experimental results are considered acceptable when considering displacements at serviceability limits of L / w equal

to 180 and 360 with differences in loading at each displacement ratio being 1.7% and 4.8% respectively.

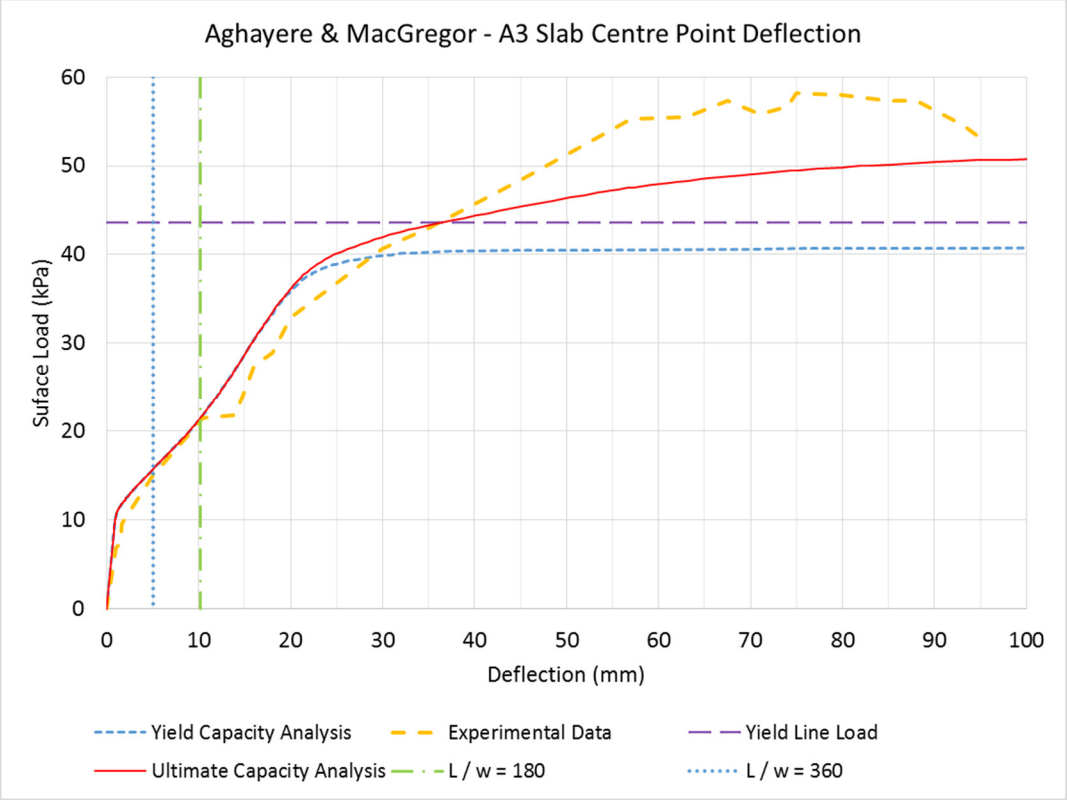


Figure 46: A3 slab centre point load vs deflection

As with the C1 slab, the evaluation of the analytical crack pattern is also carried out for the A3 slab to evaluate the expected pattern to that of the analysis. Through the analysis it was established that the end of elastic behaviour occurred at a loading of 27.65 kPa. The extracted crack pattern in the x and y directions is representative of what would be expected in an experiment comprising the same loading and boundary conditions, Figure 47.

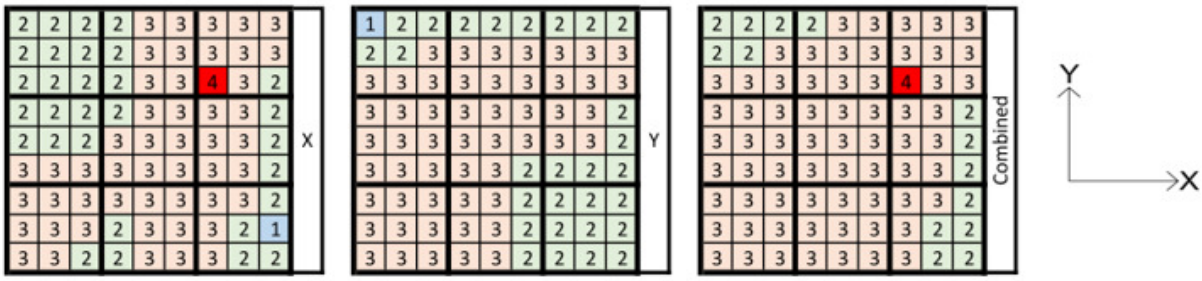


Figure 47: Aghayere & Macgregor slab directional cracking

CHAPTER 8: DISCUSSION AND CONCLUSION

Through reviewing previous studies and establishing the current theory, many important aspects of the proposed method are identified. This section will summarise the noteworthy findings and conclusions with regard to the current proposed nonlinear model.

The reinforced concrete slab nonlinear material properties given in this research are based on design code calculations. The proposed method is intended for reinforced concrete beams and slabs under transverse loading leading to bending with no axial forces present or axial forces being considered negligible. The nonlinear material properties are established for the x and y orthogonal directions of reinforcement independent of one another. Using these quantities a typical stress-strain relationship is formulated for use in a nonlinear finite element analysis.

A solution algorithm considering material nonlinearity is presented giving a simple linear finite element program, with API capabilities, the ability to solve the nonlinear problem described in this report. Although any existing nonlinear algorithm may be utilised for the solution, it should ensure that the updating of the stiffness matrix is carried out using the methods and criteria described in this report.

The validation is carried out considering four experimental case studies. These experimental case studies ensured that a variety of loading types and boundary conditions are tested against the proposed analytical model. The overall displacement results obtained through the various analyses can be considered sufficiently accurate for serviceability checks under design conditions.

The nonlinear behaviour under the current model is significantly influenced by the assumption of the cracking stress of the concrete. As concrete properties from different batches of concrete are not homogenous, one equation covering all types of concrete for the estimation of the modulus of rupture may not be applicable. It is therefore suggested that this be taken into account when carrying out design serviceability checks by possibly carrying out a sensitivity analysis using varying values of modulus of rupture.

If the design of reinforced concrete slabs is to be carried out assuming ductile yield failure as would usually be done, the proposed model would produce valuable and accurate moment distribution and displacement predictions. The method is however not suitable for forensic structural analysis where failure mechanisms other than tension steel yielding under bending alone is required.

Although results and stiffness values were checked and assigned at element Gauss points independently, a different approach may be followed whereby the Gauss point results are checked and an average element stiffness value calculated and assigned at element level for each direction of reinforcement respectively. This decision would depend on the fineness of the discretisation which should be able to sufficiently represent the distribution of yielding throughout the slab.

Although not considered in this report, continuous slabs can be seen to be analysed by extending the typical stress-strain graph to include the reinforced concrete slab section negative moment capacities for cracking, yield and ultimate in the same manner as described in the report. By extending the typical stress-strain diagram any negative moments experienced by the slab, whereby the top of the slab is in tension, can be considered independently and with ease. By using this method accurate moment redistribution results can be established. This should however still be validated with experimental data prior to implementation.

The benefit of the proposed model is that the nonlinear behaviour of a slab is based on a design code approach which closely relates analytical results to the real-life structural system behaviour. The simplified method can therefore easily form part of a design report with engineer's calculations and not merely a theoretical model built into a finite element program which the design engineer is not fully comfortable with or knowledgeable about.

Further research could be carried out covering the following topics:

- Application of the current method to continuous indeterminate slabs / beams;
- Application to the current method irregular shaped slabs;
- Application to the current method slabs with openings;
- Application to the current method slabs supported by beams;
- Development of the current approach to enable the consideration of in-plane forces;
- Include a yield criteria for slabs with reinforcement which is not orthogonal in the x and y directions.

REFERENCES

[1] American Concrete Institute (2005), Building Code Requirements for Structural Concrete and Commentary, ACI 318M-05

[2] Agbossou, A and Mougin, J.P. (2005), A layered approach to the non-linear static and dynamic analysis of rectangular reinforced concrete slabs, International Journal of Mechanical Sciences, Vol. 48 No. 3, pp. 294-306.

[3] Aghayere, A.O and MacGregor J.G (1990), Tests of Reinforced Concrete Plates under Combined In-Plane and Transverse Loads, ACI Structural Journal, Vol. 87 No. 6, pp. 615-622

[4] Armer, G.S.T (1968), Ultimate Load Tests of Slabs Designed by the Strip Method, ICE Proceedings, Vol. 41 No. 2, pp. 313-331

[5] Bathe, K.J (1996), Finite Element Procedures, Prentice Hall, New Jersey.

[6] Bischoff, P.H (2005), Reevaluation of Deflection Prediction for Concrete Beams Reinforced with Steel and Fiber Reinforced Polymer Bars, Journal of Structural Engineering ASCE, Vol. 131 No. 5, pp. 752-767

[7] Bischoff, P.H (2007), Deflection Calculation of FRP Reinforced Concrete Beams Based on Modifications to the Existing Branson Equation, Journal of Composites for Construction ASCE, Vol. 11 No. 1, pp. 4-14

[8] Bletzinger K.U, (2001), Theory of Plates Lecture Notes, Part III: Finite elements for plates in bending, Technische Universitat Munchen

[9] Branson, D.E (1968), Design Procedures for Computing Deflections, ACI Journal, Vol. 65 No. 9, pp. 730-742

[10] Cerioni R. and Iori I. and Michelini E. and Bernardi P. (2007), Multi-directional modeling of crack pattern in 2D R/C members, Engineering Fracture Mechanics, Vol. 75 No. 3-4, pp. 615-628

[11] Cloete, R. (2004), A Simplified Finite Element Model for Time-Dependent Deflections of Flat Slabs, MEng Thesis, University of Pretoria

[12] Department of Aerospace Engineering Sciences (2014), The Origins of the Finite Element Method, Appendix O, ASEN 5007, University of Colorado at Boulder

[13] Edited by Gill Owens (2009), Fulton's Concrete Technology 9th Edition, Cement and Concrete Institute, Midrand, South Africa

[14] Foster, S.J and Marti, P. and Mojsilovic, N. (2003), Design of Reinforced Concrete Solids Using Stress Analysis , ACI Structural Journal, Vol. 100 No. 6, pp. 758-764

[15] Ghoniem, M.G and MacGregor J.G (1994), Tests of Reinforced Concrete Plates under Combined Inplane and Lateral Loads, ACI Structural Journal, Vol. 91 No. 1, pp. 19-30

[16] Ghoniem, M.G (1992), Strength and Stability of Reinforced Concrete Plates Under Combined Inplane and Lateral Loads, Doctoral Thesis, Department of Civil Engineering, University of Alberta

[17] Hand, Pecknold and Schnobrich (1973), A Layered Finite Element Nonlinear Analysis of Reinforced Concrete Plates and Shells, SRS-389, Civil Engineering Studies, University of Illinois, Urbana, Illinois

[18] Hu, H.T and Lin, F.M and Jan, Y.Y (2004), Nonlinear Finite Element Analysis of Reinforced Concrete Beams Strengthened by Fiber-reinforced Plastics, Composite Structures, Vol. 63 No. 3-4, pp. 271-281

[19] Hu, H.T and Schnobrich, W.C (1991), Nonlinear Finite Element Analysis of Reinforced Concrete Plates and Shells Under Monotonic Loading, Computers and Structures, Vol. 38 No. 5/6, pp. 637-651

[20] Johansen, K.W. (1962), Yield-Line Theory (English translation from German), Cement and Concrete Association, London

[21] Jofriet, J.C and McNeice, M (1971), Finite Element Analysis of Reinforced Concrete Slabs, Journal of the Struct. Div, Proceedings of the ASCE, ST 3, pp. 785-806

[22] Kwak, H.G and Filippou, F.C (1990), Finite Element Analysis of Reinforced Concrete Structures Under Monotonic Loads, Report No. UCB/SEMM-90/14, Department of Civil Engineering, University of California, Berkeley

[23] Kalkan, I (2010), Deflection Prediction for Reinforced Concrete Beams Through Different Effective Moment of Inertia Expressions, Int. J. of Eng. Research and Development

[24] Kwon, Y.W and Bang, H (1997), The Finite Element Method using MATLAB, CRC Press

[25] Lui, G.R and Quek S.S (2003), The Finite Element Method, a practical course

[26] MacGregor, J.G. (1997), Reinforced Concrete Mechanics and Design, Sixth Edition, Pearson Education, New Jersey.

[27] McNeice, G. M.(1967), Elastic-Plastic Bending of Plates and Slabs by the Finite Element Method, Partial Fulfilment of The Requirements For The Degree of Doctor of Philosophy, University of London, England

[28] Milford, R.V. and Schnobrich, W.C. (1984), Nonlinear Behavior of Reinforced Concrete Cooling Towers, SRS-514, Civil Engineering Studies, University of Illinois, Urbana, Illinois

[29] Myoungsu, S. and Bommer, A. and Deaton, J.B. and Alemdar, B.N. (2009), Twisting Moments in Two-way Slabs, Concrete International, Vol. 31 No. 7, pp. 35

[30] Ngo, D. and Scordelis, A.C. (1967), Finite Element Analysis of Reinforced Concrete Beams, Journal of ACI, Vol. 64 No. 3, pp. 152 – 163.

[31] Oliveira, R.S and Ramalho, M.A and Correa, M.R.S (2008), A Layered Finite Element for Reinforced Concrete Beams with Bond-Slip Effects, Cement and Concrete Composites, Vol. 30 No. 3, pp 245 – 252

[32] Polak, M.A (1996), Effective Stiffness Model for Reinforced Concrete Slabs, Journal of Structural Engineering, Vol. 122 No. 9, pp. 1025 – 1030

[33] Polak, M.A (1997), Effective Stiffness Model for Reinforced Concrete Slabs Discussion by A. Bensalem and Closure by Maria Anna Polak, Journal of Structural Engineering, Vol. 123, pp. 1695-1696

[34] Polak, M.A and Vecchio, F.J (1994), Reinforced Concrete Shell Elements Subjected to Bending and Membrane Loads, ACI Structural Journal, Vol. 91 No. 3, pp. 261-268

[35] Polak, M.A and Vecchio, F.J (1993), Nonlinear Analysis of Reinforced-Concrete Shells, Journal of Structural Engineering, Vol. 119 No. 12, pp. 3439 - 3462

[36] Rahal, K.N (2010), Post-cracking Shear Modulus Of Reinforced Concrete Membrane Elements, Engineering Structures, Vol. 32 No.1, pp. 218-225

[37] Rahman, H.H (1982), Computational Models For The Nonlinear Analysis Of Reinforced Concrete Flexural Slab Systems, Doctoral Thesis, Department of Civil Engineering, University of Swansea.

[38] Šculac, P. and Jelenic´, G. and Škec, L. (2014), Kinematics of layered reinforced-concrete planar beam finite elements with embedded transversal cracking, International Journal of Solids and Structures, Vol. 51 No.1, pp. 74-92.

[39] Vecchio, F.J and Collins, M.P (1986), Modified Compression-Field Theory for Reinforced Concrete Elements Subjected to Shear, ACI Journal, Vol. 83 No. 2, pp. 219-231

[40] Vecchio, F.J (1989), Nonlinear Finite Element Analysis of Reinforced Concrete Membranes, ACI Structural Journal, Vol. 86 No. 4, pp. 26-35

[41] Vecchio, F.J and Tata, M. (1999), Approximate Analyses of Reinforced Concrete Slabs, Structural Engineering and Mechanics, Vol. 8 No. 1, pp. 1-18

[42] Wight J.K. and MacGregor, J.G. (2012), Reinforced Concrete Mechanics and Design, Sixth Edition, Pearson Education, New Jersey.

[43] Wood, R.H and Armer, G.S.T (1968), The Theory of the Strip Method for Design of Slabs, ICE Proceedings, Vol. 41 No. 2, pp. 285-311

[44] Wood, R.H. (1968), The Reinforcement of Slabs in Accordance with a Pre-Determined Field of Moments, Concrete, Vol. 2 No. 2, pp 69-76. (discussion by Armer)

[45] Zhang, Y.X. and Bradford, M.A. and Gilbert, R.I. (2007), A layered shear-flexural plate/shell element using Timoshenko beam functions for nonlinear analysis of reinforced concrete plates, Finite Elements in Analysis and Design, Vol 43. No.11, pp. 888-900.

[46] Zienkiewicz, O.C. and Taylor R.L. and Zhu, J.Z. (2005), Finite Element Method: Its Basics and Fundamentals, Sixth Edition

[47] Zienkiewicz, O.C. and Taylor R.L. (2005), The Finite Element Method for Solid and Structural Mechanics, Sixth Edition

APPENDIX A: MATERIAL PROPERTY CALCULATIONS

Polak SM1 Slab:

Specimen Data		
h	316	mm
b	1 000	mm
E_s	200 000.00	MPa
$E_{c\text{-}Actual}$	34 278.00	MPa
$E_{c\text{-}Theory}$	32 447.81	MPa
η	5.83	
A_{sc}	3 950.00	mm^2/m
A_{st}	3 950.00	mm^2/m
f_{ult}	611.00	MPa
f_y	425.00	MPa
$f_{c'}$	47.00	MPa
$f_{t'}$	4.46	MPa
d'	35.00	mm
d	281.00	mm
M_{cr}	77.253	kN.m
I_{cr}	1.323E+09	mm^4
$I_{g\text{-}v}$	2.739E+09	mm^4
I_g	2.630E+09	
I_{cr}/I_g	0.483	
Φ_{cr}	0.823	
v	0.2	

Curvature	NA	Strain (ε)			Stress (σ)		
Φ	x	Conc	Sc	St	Conc	Sc	St
5.369	83.10	0.000446	0.000258	-0.001063	15.29	51.64	-212.50
10.738	83.10	0.000892	0.000516	-0.002125	30.58	103.29	-425.00
68.011	44.11	0.003000	0.000620	-0.016111	39.95	123.93	-442.59

Force				Lever Arm		
Conc	Sc	St	ΣF = 0	Conc	Sc	St
635 379	203 996	-839 375	0	0.253	0.246	0.000
1 270 758	407 992	-1 678 750	0	0.253	0.246	0.000
1 258 732	489 506	-1 748 238	0	0.265	0.246	0.000

Moment			Total Moment	
Conc	Sc	St	β_1	M
160.94	50.18	0.00	1	211.13
321.88	100.37	0.00	1	422.25
333.87	120.42	0.00	0.71	454.29

σ	ε
0.00	0.00000
4.46	0.00013
12.69	0.00085
25.37	0.00170
27.30	0.01075

McNeice Slab:

Specimen Data		
h	44.45	mm
b	1 000	mm
E_s	200 000.00	MPa
$E_{c\text{-Actual}}$	28 613.24	MPa
$E_{c\text{-Theory}}$	29 145.44	MPa
η	6.99	
A_{sc}	0.00	mm^2/m
A_{st}	380.05	mm^2/m
f_{ult}	485.00	MPa
f_y	380.00	MPa
$f_{c'}$	37.92	MPa
$f_{t'}$	3.82	MPa
d'	0.01	mm
d	33.27	mm
M_{cr}	1.286	kN.m
I_{cr}	1.917E+06	mm^4
$I_{g\text{-}v}$	7.487E+06	mm^4
I_g	7.319E+06	mm^4
I_{cr}/I_g	0.256	
Φ_{cr}	6.004	
v	0.15	

Curvature	NA	Strain (ε)			Stress (σ)		
Φ	x	Conc	Sc	St	Conc	Sc	St
42.470	10.90	0.000463	0.000463	-0.000950	13.25	92.51	-190.00
84.941	10.90	0.000926	0.000925	-0.001900	26.50	185.03	-380.00
408.739	7.34	0.003000	0.002996	-0.010599	32.23	485.00	-485.00

Force				Lever Arm		
Conc	Sc	St	ΣF = 0	Conc	Sc	St
72 209	0	-72 209	0	0.030	0.033	0.000
144 418	0	-144 418	0	0.030	0.033	0.000
184 323	0	-184 323	0	0.030	0.033	0.000

Moment			Total Moment	
Conc	Sc	St	β_1	M
2.14	0.00	0.00	1	2.14
4.28	0.00	0.00	1	4.28
5.61	0.00	0.00	0.78	5.61

σ	ε
0.00	0.00000
3.82	0.00013
6.50	0.00094
13.00	0.00189
17.02	0.00908

Ghomein & MacGregor X Direction:

Specimen Data		
h	67.8	mm
b	1 000	mm
E_s	181 500.00	MPa
$E_{c\text{-}Actual}$	21 300.00	MPa
$E_{c\text{-}Theory}$	23 764.19	MPa
η	8.52	
A_{sc}	260.00	mm^2/m
A_{st}	260.00	mm^2/m
f_{ult}	620.00	MPa
f_y	450.00	MPa
$f_{c'}$	25.21	MPa
$f_{t'}$	1.76	MPa
d'	21.90	mm
d	56.80	mm
M_{cr}	1.346	kN.m
I_{cr}	5.186E+06	mm^4
$I_{g\text{-}v}$	2.705E+07	mm^4
I_g	2.597E+07	mm^4
I_{cr}/I_g	0.200	
Φ_{cr}	2.434	
v	0.2	

Curvature	NA	Strain (ε)			Stress (σ)		
Φ	x	Conc	Sc	St	Conc	Sc	St
29.489	14.76	0.000435	-0.000211	-0.001240	9.27	-38.21	-225.00
58.978	14.76	0.000871	-0.000421	-0.002479	18.54	-76.41	-450.00
238.467	12.58	0.003000	-0.002222	-0.010545	21.43	-403.37	-477.95

Force				Lever Arm		
Conc	Sc	St	ΣF = 0	Conc	Sc	St
68 434	-9 934	-58 500	0	0.052	0.035	0.000
136 868	-19 868	-117 000	0	0.052	0.035	0.000
229 142	-104 876	-124 266	0	0.051	0.035	0.000

Moment			Total Moment	
Conc	Sc	St	β_1	M
3.55	-0.35	0.00	1	3.20
7.10	-0.69	0.00	1	6.41
11.79	-3.66	0.00	0.85	8.13

σ	ε
0.00	0.00000
1.69	0.00008
4.18	0.00100
8.36	0.00200
10.61	0.00808

Ghomein & MacGregor Y Direction:

Specimen Data		
h	68	mm
b	1 000	mm
E_s	181 500.00	MPa
$E_{c\text{-}Actual}$	21 300.00	MPa
$E_{c\text{-}Theory}$	23 764.19	MPa
η	8.52	
A_{sc}	260.00	mm^2/m
A_{st}	260.00	mm^2/m
f_{ult}	620.00	MPa
f_y	450.00	MPa
$f_{c'}$	25.21	MPa
$f_{t'}$	1.76	MPa
d'	15.60	mm
d	50.50	mm
M_{cr}	1.346	kN.m
I_{cr}	3.981E+06	mm^4
$I_{g\text{-}v}$	2.705E+07	
I_g	2.597E+07	mm^4
I_{cr}/I_g	0.153	
Φ_{cr}	2.434	
v	0.2	

Curvature	NA	Strain (ε)			Stress (σ)		
Φ	x	Conc	Sc	St	Conc	Sc	St
33.277	13.25	0.000441	-0.000078	-0.001240	9.39	-14.21	-225.00
66.555	13.25	0.000882	-0.000157	-0.002479	18.78	-28.42	-450.00
283.945	10.57	0.003000	-0.001430	-0.011339	21.43	-259.46	-480.70

Force				Lever Arm		
Conc	Sc	St	ΣF = 0	Conc	Sc	St
62 195	-3 695	-58 500	0	0.046	0.035	0.000
124 389	-7 389	-117 000	0	0.046	0.035	0.000
192 441	-67 460	-124 981	0	0.046	0.035	0.000

Moment			Total Moment	
Conc	Sc	St	β_1	M
2.87	-0.13	0.00	1	2.74
5.73	-0.26	0.00	1	5.47
8.85	-2.35	0.00	0.85	6.50

σ	ε
0.00	0.00000
1.69	0.00008
3.57	0.00113
7.15	0.00226
8.48	0.00963

Aghayere & MacGregor X Direction:

Specimen Data		
h	65	mm
b	1 000	mm
E_s	197 300.00	MPa
$E_{c\text{-}Actual}$	23 150.00	MPa
$E_{c\text{-}Theory}$	26 857.43	MPa
η	8.52	
A_{sc}	224.63	mm^2/m
A_{st}	224.63	mm^2/m
f_{ult}	670.00	MPa
f_y	504.00	MPa
$f_{c'}$	32.20	MPa
$f_{t'}$	2.27	MPa
d'	18.60	mm
d	49.80	mm
M_{cr}	2.520	kN.m
I_{cr}	3.484E+06	mm^4
$I_{g\text{-}v}$	2.417E+07	mm^4
I_g	2.320E+07	mm^4
I_{cr}/I_g	0.144	
Φ_{cr}	3.003	
v	0.2	

Curvature	NA	Strain (ε)			Stress (σ)		
Φ	x	Conc	Sc	St	Conc	Sc	St
34.521	12.80	0.000442	-0.000200	-0.001277	10.23	-39.50	-252.00
69.042	12.80	0.000884	-0.000400	-0.002554	20.46	-78.99	-504.00
292.402	10.26	0.003000	-0.002439	-0.011562	27.37	-481.15	-543.93

Force				Lever Arm		
Conc	Sc	St	ΣF = 0	Conc	Sc	St
65 479	-8 872	-56 607	0	0.046	0.031	0.000
130 959	-17 744	-113 215	0	0.046	0.031	0.000
230 266	-108 082	-122 184	0	0.046	0.031	0.000

Moment			Total Moment	
Conc	Sc	St	β_1	M
2.98	-0.28	0.00	1	2.70
5.96	-0.55	0.00	1	5.41
10.50	-3.37	0.00	0.82	7.13

σ	ε
0.00	0.00000
2.27	0.00010
3.81	0.00113
7.61	0.00225
10.03	0.00955

Aghayere & MacGregor Y Direction:

Specimen Data		
h	65	mm
b	1 000	mm
E_s	197 300.00	MPa
$E_{c\text{-}Actual}$	23 150.00	MPa
$E_{c\text{-}Theory}$	26 857.43	MPa
η	8.52	
A_{sc}	261.20	mm^2/m
A_{st}	261.20	mm^2/m
f_{ult}	670.00	MPa
f_y	504.00	MPa
$f_{c'}$	32.20	MPa
$f_{t'}$	2.27	MPa
d'	12.20	mm
d	56.10	mm
M_{cr}	2.520	kN.m
I_{cr}	5.203E+06	mm^4
$I_{g\text{-}v}$	2.417E+07	mm^4
I_g	2.320E+07	mm^4
I_{cr}/I_g	0.215	
Φ_{cr}	4.504	
v	0.2	

Curvature	NA	Strain (ε)			Stress (σ)		
Φ	x	Conc	Sc	St	Conc	Sc	St
30.014	13.55	0.000407	0.000040	-0.001277	9.41	7.97	-252.00
60.028	13.55	0.000813	0.000081	-0.002554	18.82	15.93	-504.00
333.452	9.00	0.003000	-0.001068	-0.015707	27.37	-210.74	-562.30

Force				Lever Arm		
Conc	Sc	St	ΣF = 0	Conc	Sc	St
63 742	2 081	-65 822	0	0.052	0.044	0.000
127 483	4 162	-131 645	0	0.052	0.044	0.000
201 919	-55 045	-146 874	0	0.052	0.044	0.000

Moment			Total Moment	
Conc	Sc	St	β_1	M
3.29	0.09	0.00	1	3.38
6.58	0.18	0.00	1	6.76
10.58	-2.42	0.00	0.82	8.17

σ	ε
0.00	0.00000
2.27	0.00010
4.76	0.00098
9.51	0.00196
11.49	0.01089

Yield line load calculation:

Experimental Slab	Mx	My	Length = Breadth	Internal Work	External Work / qf	qf	kN
Aghayere & MacGregor (A3)	5.41	6.76	1.83	48.67	1.12	43.60	146.02
Ghomein & MacGregor (C1)	6.41	5.47	1.83	47.53	1.12	42.58	142.58

APPENDIX B: MATLAB CODE

Polak Input:

```
%Input Data for the analysis of the slab/beam (All in SI Units)
%
%
clear;
%Mesh / Material Settings
totl=1.5;                    %Total Length of member
totb=1.5;                    %Total Breadth of member
h=0.316;                     %Depth of member
numdivl=1;                   %Number of Divisions Length
numdivb=1;                   %Number of Divisions Breadth
elel=totl/numdivl;           %Element length
eleb=totb/numdivb;           %Element breadth
aspectr=elel/eleb;           %Aspect ratio
eltype=8;                    %Element type
Exl=34.278e9;                %Ex-Modulus
Eyl=34.278e9;                %Ey-Modulus
vxl=0.2;                     %Poisson Ratio x
vyl=0.2;                     %Poisson Ratio y
numgpr=3; numgps=3;          %3x3 Gauss Integration for bending
%-------------------------------------------------------------
nel=numdivl*numdivb;         %Total number of elements
nnel=8;                      %Total number of nodes per element & total Number of nodes
nnode=(numdivl+1)*(numdivb+1)+(numdivl)*(numdivb+1)...
+(numdivl+1)*(numdivb);
ndof=3;                      %Number of degrees of freedom
sdof=nnode*ndof;             %System degrees of freedom
edof=nnel*ndof;              %Degrees of freedom per element
%-------------------------------------------------------------
%Obtain Reduced Integration Points for shear
%-------------------------------------------------------------
ngprshear=2;
ngpsshear=2;

%-------------------------------------------------------------
%Obtain nodes co-ordinates for all nodes in the system
%-------------------------------------------------------------
[gcoord] = nodecoord(numdivb,numdivl,eleb,elel,eltype,nnode);

%-------------------------------------------------------------
%Obtain nodal conectivity for elements in the system
%-------------------------------------------------------------
[nodes, index] = connect(numdivb,numdivl,eltype,nel,ndof);

%-------------------------------------------------------------
%Input boundary conditions
%-------------------------------------------------------------
bcdof=[1,2,4,5,7,8,16,17,19,20,22,23];
bcval=[0,0,0,0,0,0,0,0,0,0,0,0];
```

```
%-----------------------------------------------------------------
%Obtain local element co-ordinates
%-----------------------------------------------------------------
[elcoord] = localcoord(eltype,nel,gcoord,nodes);

%-----------------------------------------------
%Load Vector (Nodal Moment)
%-----------------------------------------------
fvec=zeros(sdof,1);
TMom=-100;
Mom=TMom/2;

fvec(6,1)=Mom*4/3;
fvec(21,1)=-Mom*4/3;
fvec(3,1)=Mom/3;
fvec(9,1)=Mom/3;
fvec(18,1)=-Mom/3;
fvec(24,1)=-Mom/3;
```

McNeice Input:

```
%Input Data for the analysis of the slab/beam (All in SI Units)
%
%
clear;
%Mesh / Material Settings
totl=0.457;                    %Total Length of member
totb=0.457;                    %Total Breadth of member
h=0.0445;                      %Depth of member
numdivl=3;                     %Number of Divisions Length
numdivb=3;                     %Number of Divisions Breadth
elel=totl/numdivl;             %Element length
eleb=totb/numdivb;             %Element breadth
aspectr=elel/eleb;             %Aspect ratio
eltype=8;                      %Element type
ExI=28.613e9;                  %Ex-Modulus
EyI=28.613e9;                  %Ey-Modulus
vxI=0.15;                      %Poisson Ratio x
vyI=0.15;                      %Poisson Ratio y
numgpr=3; numgps=3;            %3x3 Gauss Integration for bending
%-----------------------------------------------------------------
nel=numdivl*numdivb;           %Total number of elements
nnel=8;                        %Total number of nodes per element & total Number of nodes
nnode=(numdivl+1)*(numdivb+1)+(numdivl)*(numdivb+1)...
+(numdivl+1)*(numdivb);
ndof=3;                        %Number of degrees of freedom
sdof=nnode*ndof;               %System degrees of freedom
edof=nnel*ndof;                %Degrees of freedom per element
%-----------------------------------------------------------------
%Obtain Reduced Integration Points for shear
%-----------------------------------------------------------------
ngprshear=2;
```

```
ngpsshear=2;

%-------------------------------------------------------------
%Obtain nodes co-ordinates for all nodes in the system
%-------------------------------------------------------------
[gcoord] = nodecoord(numdivb,numdivl,eleb,elel,eltype,nnode);

%-------------------------------------------------------------
%Obtain nodal conectivity for elements in the system
%-------------------------------------------------------------
[nodes, index] = connect(numdivb,numdivl,eltype,nel,ndof);

%-------------------------------------------------------------
%Input boundary conditions
%-------------------------------------------------------------
bcdof=[3,6,9,12,15,18,20,21,32,53,65,86,98,100,119];
bcval=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];

%-------------------------------------------------------------
%Obtain local element co-ordinates
%-------------------------------------------------------------
[elcoord] = localcoord(eltype,nel,gcoord,nodes);

%-----------------------------------------------
%Load Vector (Nodal Moment)
%-----------------------------------------------
fvec=zeros(sdof,1);
fvec(19,1)=-10;
```

Ghomein & MacGregor Input:

```
%Input Data for the analysis of the slab/beam (All in SI Units)
%
%
clear;
%Mesh / Material Settings
totl=0.915;                    %Total Length of member
totb=0.915;                     %Total Breadth of member
h=0.068;                       %Depth of member
numdivl=3;                      %Number of Divisions Length
numdivb=3;                      %Number of Divisions Breadth
elel=totl/numdivl;              %Element length
eleb=totb/numdivb;                %Element breadth
aspectr=elel/eleb;              %Aspect ratio
eltype=8;                      %Element type
ExI=21.3e9;                     %Ex-Modulus
EyI=21.3e9;                     %Ey-Modulus
vxI=0.2;                       %Poisson Ratio x
vyI=0.2;                       %Poisson Ratio y
numgpr=3; numgps=3;                %3x3 Gauss Integration for bending
%-------------------------------------------------------------
nel=numdivl*numdivb;               %Total number of elements
nnel=8;                        %Total number of nodes per element & total Number of nodes
nnode=(numdivl+1)*(numdivb+1)+(numdivl)*(numdivb+1)...
```

```
+(numdivl+1)*(numdivb);
ndof=3;                         %Number of degrees of freedom
sdof=nnode*ndof;                  %System degrees of freedom
edof=nnel*ndof;                   %Degrees of freedom per element
%-------------------------------------------------------------------
%Obtain Reduced Integration Points for shear
%-------------------------------------------------------------------
ngprshear=2;
ngpsshear=2;

%-------------------------------------------------------------------
%Obtain nodes co-ordinates for all nodes in the system
%-------------------------------------------------------------------
[gcoord] = nodecoord(numdivb,numdivl,eleb,elel,eltype,nnode);

%-------------------------------------------------------------------
%Obtain nodal conectivity for elements in the system
%-------------------------------------------------------------------
[nodes, index] = connect(numdivb,numdivl,eltype,nel,ndof);

%-------------------------------------------------------------------
%Input boundary conditions
%-------------------------------------------------------------------
bcdof=[1,22,34,55,67,88,3,24,36,57,69,90,6,9,12,15,18,21,20,32,53,65,86,98,100,101,102,103,104,106,107,109,110,112,113,1
15,116,118,119];
bcval=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];

%-------------------------------------------------------------------
%Obtain local element co-ordinates
%-------------------------------------------------------------------
[elcoord] = localcoord(eltype,nel,gcoord,nodes);

%-----------------------------------------------
%Load Vector (Nodal Force)
%-----------------------------------------------
fvec=zeros(sdof,1);
fvec(7,1)=-50;
fvec(19,1)=-25;
fvec(73,1)=-100;
fvec(85,1)=-50;
```

Aghayere & MacGregor Input:

```
%Input Data for the analysis of the slab/beam (All in SI Units)
%
%
clear;
%Mesh / Material Settings
totl=0.915;                     %Total Length of member
totb=0.915;                      %Total Breadth of member
h=0.0653;                        %Depth of member
numdivl=3;                       %Number of Divisions Length
numdivb=3;                        %Number of Divisions Breadth
elel=totl/numdivl;                %Element length
```

```
eleb=totb/numdivb;                  %Element breadth
aspectr=elel/eleb;                  %Aspect ratio
eltype=8;                           %Element type
Exl=23.15e9;                        %Ex-Modulus
Eyl=23.15e9;                        %Ey-Modulus
vxl=0.2;                            %Poisson Ratio x
vyl=0.2;                            %Poisson Ratio y
numgpr=2; numgps=2;                 %3x3 Gauss Integration for bending
%-------------------------------------------------------------
nel=numdivl*numdivb;                %Total number of elements
nnel=8;                             %Total number of nodes per element & total Number of nodes
nnode=(numdivl+1)*(numdivb+1)+(numdivl)*(numdivb+1)...
+(numdivl+1)*(numdivb);
ndof=3;                             %Number of degrees of freedom
sdof=nnode*ndof;                    %System degrees of freedom
edof=nnel*ndof;                     %Degrees of freedom per element
%-------------------------------------------------------------
%Obtain Reduced Integration Points for shear
%-------------------------------------------------------------
ngprshear=2;
ngpsshear=2;

%-------------------------------------------------------------
%Obtain nodes co-ordinates for all nodes in the system
%-------------------------------------------------------------
[gcoord] = nodecoord(numdivb,numdivl,eleb,elel,eltype,nnode);

%-------------------------------------------------------------
%Obtain nodal conectivity for elements in the system
%-------------------------------------------------------------
[nodes, index] = connect(numdivb,numdivl,eltype,nel,ndof);

%-------------------------------------------------------------
%Input boundary conditions
%-------------------------------------------------------------
bcdof=[1,22,34,55,67,88,3,24,36,57,69,90,6,9,12,15,18,21,20,32,53,65,86,98,100,101,102,103,104,106,107,109,110,112,113,1
15,116,118,119];
bcval=[0,0,0,
```

Linear Solver:

```
%-------------------------------------------------------------
%Linear solver
%-------------------------------------------------------------

%-------------------------------------------------------------
%Obtain material matrices
%-------------------------------------------------------------
[Cb,Cs] = Lmatmatrix(ExI,EyI,vx,vy,nel,numgpr,numgps);

%-------------------------------------------------------------
%Obtain Strain Interpolation matrices
%-------------------------------------------------------------
[Bb,Bs,BbRes,BsRes] = bmatrix(elcoord,eltype,nel,numgpr,numgps,ndof,ngprshear,ngpsshear);

%-------------------------------------------------------------
%Obtain elements stiffness matrix
%-------------------------------------------------------------
[Kel] = kmatrix(elcoord,eltype,nel,numgpr,numgps,ndof,Cb,Cs,Bb,Bs,h,ngprshear,ngpsshear);

%-------------------------------------------------------------
%Obtain system stiffness matrix
%-------------------------------------------------------------
[Ksys] = assemble(sdof,index,Kel,nel);

%-------------------------------------------------------------
%Obtain modified system stiffness matrix and force vector
%-------------------------------------------------------------
[Kmod, fvecmod] = applybc(bcdof,fvec,Ksys);

%-------------------------------------------------------------
%Obtain increment displacement solution
%-------------------------------------------------------------
d=Kmod\fvecmod;

%-------------------------------------------------------------------
%Obtain iteration curvatures, in-plane strain & out-plane shear strain
%-------------------------------------------------------------------
[del,curvstr,inplstr,shearstr] = esmatrix(BbRes,BsRes,d,nel,nnel,ndof,index...
    ,numgpr,numgps,ngprshear,ngpsshear,h);

%-------------------------------------------------------------
%Obtain load step stresses, moments and forces
%-------------------------------------------------------------
[inplstress,shearstress,pltmom,pltshear] = stressmatrix(inplstr,shearstr,nel...
    ,Cb,Cs,numgpr,numgps,h,ngprshear,ngpsshear);

%-------------------------------------------------------------
%Write out results
%-------------------------------------------------------------
for t=1:nnode
 dw(t,1)=d(t*ndof-2,1);
 rxw(t,1)=d(t*ndof-1,1);
```

```
 ryw(t,1)=d(t*ndof,1);
end
pltmomres=zeros(nel,1);
for dd=1:nel
 for x=1:(numgpr*numgps)
  pltmomresx(dd,x)=pltmom(1,1,x,dd);
  pltmomresy(dd,x)=pltmom(2,1,x,dd);
  ipstressX(dd,x)=inplstress(1,1,x,dd);
  ipstressY(dd,x)=inplstress(2,1,x,dd);
  ipstressXY(dd,x)=inplstress(3,1,x,dd);
 end
end

xlswrite('LResults.xls', pltmomresx, 'mx_Result', 'A2');
xlswrite('LResults.xls', pltmomresy, 'my_Result', 'A2');
xlswrite('LResults.xls', ipstressX, 'SX_Result', 'A2');
xlswrite('LResults.xls', ipstressY, 'SY_Result', 'A2');
xlswrite('LResults.xls', ipstressXY, 'SXY_Result', 'A2');
xlswrite('LResults.xls', ryw, 'ryw_Result', 'A2');
xlswrite('LResults.xls', rxw, 'rxw_Result', 'A2');
xlswrite('LResults.xls', dw, 'dw_Result', 'A2');
```

Nonlinear Solver:

```
%-------------------------------------------------------------
%Non-Linear solver
%-------------------------------------------------------------
%strconx=[0.000133,0.000944,0.001888,0.009084,0.1,1];              %McNeice
%stressconx=[3.82e6,6.4986e6,12.99725e6,17.02211e6,17.03e6,17.2e6];
%strcony=[0.000133,0.000944,0.001888,0.009084,0.1,1];
%stresscony=[3.82e6,6.4986e6,12.99725e6,17.02211e6,17.03e6,17.2e6];

%strconx=[0.00013,0.000848,0.001697,0.010746,0.1,0.15];            %Polak
%stressconx=[4.45618e6,12.69e6,25.372e6,27.3e6,27.4e6,27.5e6];
%strcony=[0.00013,0.000848,0.001697,0.010746,0.1,0.15];
%stresscony=[4.45618e6,12.69e6,25.372e6,27.3e6,27.4e6,27.5e6];

%strconx=[0.000083,0.001,0.001999,0.1,1];                          %Ghomein & MacGregor - Yield
%stressconx=[1.687e6,4.181509e6,8.363017e6,8.4e6,10e6];
%strcony=[0.000083,0.001128,0.002256,0.1,1];
%stresscony=[1.687e6,3.572779e6,7.14556e6,7.2e6,10e6];

%strconx=[0.000083,0.001,0.001999,0.008084,0.1,1];                 %Ghomein & MacGregor - Ult
%stressconx=[1.687e6,4.181509e6,8.363017e6,10.612e6,10.65e6,10.7e6];
%strcony=[0.000083,0.001128,0.002256,0.009626,0.1,1];
%stresscony=[1.687e6,3.572779e6,7.14556e6,8.4838e6,8.5e6,10e6];

%strconx=[0.000098,0.001127,0.002254,0.1,1];                       %Aghayere & MacGregor - Yield
%stressconx=[2.27e6,3.81e6,7.61e6,7.65e6,7.7e6];
%strcony=[0.000098,0.00098,0.00196,0.1,1];
%stresscony=[2.27e6,4.76e6,9.51e6,9.55e6,9.6e6];

%strconx=[0.000098,0.001127,0.002254,0.009547,0.1,1];              %Aghayere & MacGregor - Ult
%stressconx=[2.27e6,3.81e6,7.61e6,10.03e6,10.1e6,10.2e6];
```

```
%strcony=[0.000098,0.00098,0.00196,0.01089,0.1,1];
%stresscony=[2.27e6,4.76e6,9.51e6,11.49e6,11.55e6,12e6];


lx=numgpr*numgps;
lxx=ngprshear*ngpsshear;
k=ndof*eltype;
tol=0.01;                                %Convergence tolerance
n=1;                                     %Initialise increment
ncp=length(strconx);                          %Get the number of control points
maxls=500;                               %Maximum number of load steps
maxitr=50;                               %Maximum number of iterations
Lstp=0;

%-------------------------------------------------------------
%Initialise Matrices
%-------------------------------------------------------------
econtx=zeros(lx,nel);
econty=zeros(lx,nel);

strchkTx=zeros(lx,nel);
strchkTy=zeros(lx,nel);

del=zeros(k,1,nel,10);
curvstr=zeros(3,1,lx,nel,10);
inplstr=zeros(3,1,lx,nel,10);
shearstr=zeros(2,1,lxx,nel,10);

delf=zeros(k,1,nel,500);
curvstrf=zeros(3,1,lx,nel,500);
inplstrf=zeros(3,1,lx,nel,500);
shearstrf=zeros(2,1,lxx,nel,500);

inplstressf=zeros(3,1,lx,nel,500);
shearstressf=zeros(2,1,lxx,nel,500);
pltmomf=zeros(3,1,lx,nel,500);
pltshearf=zeros(3,1,lxx,nel,500);

%-------------------------------------------------------------
%Initial val of E-Modulus to initial
%-------------------------------------------------------------
Ex=zeros(nel,lx);
Ey=zeros(nel,lx);
vx=zeros(nel,lx);
vy=zeros(nel,lx);
logX=zeros(lx,nel);
logY=zeros(lx,nel);
Ex(:,:)=ExI;
Ey(:,:)=EyI;
vx(:,:)=vxI;
vy(:,:)=vyI;

%-------------------------------------------------------------
```

```
%Initial val of e-control point = 1
%-------------------------------------------------------------------
for s=1:nel
 econtx(:,s)=1;
 econty(:,s)=1;
end

%-------------------------------------------------------------------
%Get various E-Modulus values for non-linear material properties
%-------------------------------------------------------------------
[Econx,Econy] = econ(strconx,stressconx,strcony,stresscony);

%-------------------------------------------------------------------
%Obtain Strain Interpolation matrices
%-------------------------------------------------------------------
[Bb,Bs,BbRes,BsRes] = bmatrix(elcoord,eltype,nel,numgpr,numgps,ndof,ngprshear,ngpsshear);

%-------------------------------------------------------------------
%Start non-linear solver as a while loop
%-------------------------------------------------------------------
ex=0;
maxecon=0;
fvec1=fvec;      %Baseline load vector
Ls=fvec;         %Initialise load vector

while Lstp<=maxls && n<=maxitr && maxecon<ncp
 ex=ex+1;
 fvec=Ls;

 %-------------------------------------------------------------------
 %Obtain initial material matrices
 %-------------------------------------------------------------------
 [Cb,Cs] = NLmatmatrix(Ex,Ey,vx,vy,nel,numgpr,numgps);


 %-------------------------------------------------------------------
 %Obtain elements stiffness matrix
 %-------------------------------------------------------------------
 [Kel] = kmatrix(elcoord,eltype,nel,numgpr,numgps,ndof,Cb,Cs,Bb,Bs,h,ngprshear,ngpsshear);

 %-------------------------------------------------------------------
 %Obtain system stiffness matrix
 %-------------------------------------------------------------------
 [Ksys] = assemble(sdof,index,Kel,nel);

 %-------------------------------------------------------------------
 %Obtain modified system stiffness matrix and force vector
 %-------------------------------------------------------------------
 [Kmod, fvecmod] = applybc(bcdof,fvec,Ksys);

 %-------------------------------------------------------------------
 %Obtain increment displacement solution
 %-------------------------------------------------------------------
```

```
d=Kmod\fvecmod;

%-------------------------------------------------------------------
%Obtain iteration curvatures, in-plane strain & out-plane shear strain
%-------------------------------------------------------------------
[~,~,inplstrt,~]...
    = esmatrix(BbRes,BsRes,d,nel,nnel,ndof,index,numgpr,numgps,ngprshear,ngpsshear,h);
  inplstr=inplstrt;
%-------------------------------------------------------------------
%Check for convergence and increment non-linear material properties
%-------------------------------------------------------------------
[itr,L,n,maxecon,Ex,Ey,econtx,econty,vx,vy,logX,logY]                                                                    =
matchk(Lstp,Ls,inplstr,nel,numgpr,numgps,strconx,strcony,tol,n,econtx,econty,Ex,Ey,fvec1,vx,vy,strchkTx,strchkTy,logX,logY);

if itr==0
  %-----------------------------------------------------------
  %Load step has converged and increment load step
  %-----------------------------------------------------------
  Lstp=Lstp+1;
  if Lstp==1
   dres(:,1)=d(:,1);
  else
   dres(:,Lstp)=d(:,1)+dres(:,Lstp-1);
  end
  %-------------------------------------------------------------------
  %Obtain load step curvatures, in-plane strain & out-plane shear strain
  %-------------------------------------------------------------------
  [delt,curvstrt,inplstrt,shearstrt]...
      = esmatrix(BbRes,BsRes,d,nel,nnel,ndof,index,numgpr,numgps,ngprshear,ngpsshear,h);

      %-----------------------------------------------------------
      %Obtain load step stresses, moments and forces
      %-----------------------------------------------------------
      [inplstresst,shearstresst,pltmomt,pltsheart]...
            = stressmatrix(inplstrt,shearstrt,nel,Cb,Cs,numgpr,numgps,h,ngprshear,ngpsshear);

      inplstressf(:,:,:,:,Lstp)=inplstresst;
      shearstressf(:,:,:,:,Lstp)=shearstresst;
      pltmomf(:,:,:,:,Lstp)=pltmomt;
      pltshearf(:,:,:,:,Lstp)=pltsheart;

      if Lstp>1
         inplstressf(:,:,:,:,Lstp)=inplstressf(:,:,:,:,Lstp)+inplstressf(:,:,:,:,Lstp-1);
         shearstressf(:,:,:,:,Lstp)=shearstressf(:,:,:,:,Lstp)+shearstressf(:,:,:,:,Lstp-1);
         pltmomf(:,:,:,:,Lstp)=pltmomf(:,:,:,:,Lstp)+pltmomf(:,:,:,:,Lstp-1);
         pltshearf(:,:,:,:,Lstp)=pltshearf(:,:,:,:,Lstp)+pltshearf(:,:,:,:,Lstp-1);
      end
      %-----------------------------------------------------------

  delf(:,:,:,Lstp)=delt;
  curvstrf(:,:,:,:,Lstp)=curvstrt;
  inplstrf(:,:,:,:,Lstp)=inplstrt;
  shearstrf(:,:,:,:,Lstp)=shearstrt;
```

```
for yg=1:nel
gg=0;
 for xg=1:numgpr
  for zg=1:numgps
   gg=gg+1;

   Gx=(Ex(yg,gg)*Ey(yg,gg))/(Ex(yg,gg)*(1+vx(yg,gg))+Ey(yg,gg)*(1+vy(yg,gg)));
   Gy=Gx;
   exx=inplstrt(1,1,gg,yg);
   eyy=inplstrt(2,1,gg,yg);
   exy=inplstrt(3,1,gg,yg);

   portx1= Ey(yg,gg)*vx(yg,gg)*eyy/(Ex(yg,gg)*exx);
   portx2= abs((Gx*exy*(1-vx(yg,gg)*vy(yg,gg))/...
        (Ex(yg,gg)*exx+Ey(yg,gg)*vx(yg,gg)*eyy))*(1+portx1));

   porty1= Ex(yg,gg)*vy(yg,gg)*exx/(Ey(yg,gg)*eyy);
   porty2= abs((Gy*exy*(1-vx(yg,gg)*vy(yg,gg))/...
        (Ey(yg,gg)*eyy+Ex(yg,gg)*vy(yg,gg)*exx))*(1+porty1));

   strchkTx(gg,yg)= exx*(1+portx1+portx2)+strchkTx(gg,yg);
   strchkTy(gg,yg)= eyy*(1+porty1+porty2)+strchkTy(gg,yg);

   if logX(gg,yg)==1
    econtx(gg,yg)=econtx(gg,yg)+1;
    Ex(yg,gg)=Econx(econtx(gg,yg));
    vx(yg,gg)=0;
   end

   if logY(gg,yg)==1
    econty(gg,yg)=econty(gg,yg)+1;
    Ey(yg,gg)=Econy(econty(gg,yg));
    vy(yg,gg)=0;
   end
  end
 end
end

if Lstp>1
 delf(:,:,:,Lstp)=delf(:,:,:,Lstp)+delf(:,:,:,Lstp-1);
 curvstrf(:,:,:,:,Lstp)=curvstrf(:,:,:,:,Lstp)+curvstrf(:,:,:,:,Lstp-1);
 inplstrf(:,:,:,:,Lstp)=inplstrf(:,:,:,:,Lstp)+inplstrf(:,:,:,:,Lstp-1);
 shearstrf(:,:,:,:,Lstp)=shearstrf(:,:,:,:,Lstp)+shearstrf(:,:,:,:,Lstp-1);
end


%-----------------------------------------------------------------
%Recalculate material matrices with updated E-modulus
%-----------------------------------------------------------------
if Lstp==1
 fvecT(:,1,Lstp)=fvec;
else
```

```
 fvecT(:,1,Lstp)=fvec+fvecT(:,1,Lstp-1);
end
%------------------------------------------------------------
%Recalculate material matrices with updated E-modulus
%------------------------------------------------------------
[Cb,Cs] = NLmatmatrix(Ex,Ey,vx,vy,nel,numgpr,numgps);

%------------------------------------------------------------
%Initialise load and increment values
%------------------------------------------------------------
Ls=L;
n=1;

else
%------------------------------------------------------------
%Load step has not converged and increment load iteration
%------------------------------------------------------------
Ls=L;
n=n+1;

end

end
```

Function Nonlinear Material Check:

```
function [itr,L,n,maxecon,Ex,Ey,econtx,econty,vx,vy,logX,logY] =
matchk(Lstp,Ls,inplstr,nel,numgpr,numgps,strconx,strcony,tol,n,econtx,econty,Ex,Ey,fvec1,vx,vy,strchkTx,strchkTy,logX,logY)
%Check for convergence and reduce stiffness

%------------------------------------------------------------
%Initialise values
%------------------------------------------------------------
cmin=0;
cmax=1;
itr=2;
L=Ls;
logX=zeros(numgpr*numgps,nel);
logY=zeros(numgpr*numgps,nel);

for y=1:nel
 xx=0;
 for x=1:numgpr
  for z=1:numgps
   xx=xx+1;

   %------------------------------------------------------------
   %Calculate convergence criteria value
   %------------------------------------------------------------
   Gx=(Ex(y,xx)*Ey(y,xx))/(Ex(y,xx)*(1+vx(y,xx))+Ey(y,xx)*(1+vy(y,xx)));
   Gy=Gx;
   exx=inplstr(1,1,xx,y);
   eyy=inplstr(2,1,xx,y);
   exy=inplstr(3,1,xx,y);
```

```
portx1= Ey(y,xx)*vx(y,xx)*eyy/(Ex(y,xx)*exx);
portx2= abs((Gx*exy*(1-vx(y,xx)*vy(y,xx))/...
       (Ex(y,xx)*exx+Ey(y,xx)*vx(y,xx)*eyy))*(1+portx1));

porty1= Ex(y,xx)*vy(y,xx)*exx/(Ey(y,xx)*eyy);
porty2= abs((Gy*exy*(1-vx(y,xx)*vy(y,xx))/...
       (Ey(y,xx)*eyy+Ex(y,xx)*vy(y,xx)*exx))*(1+porty1));


strchkx= exx*(1 + portx1 + portx2);
strchky= eyy*(1 + porty1 + porty2);

if Lstp>0
 strconpx=strconx(1,econtx(xx,y))-abs(strchkTx(xx,y));
 strconpy=strcony(1,econty(xx,y))-abs(strchkTy(xx,y));
else
 strconpx=strconx(1,econtx(xx,y));
 strconpy=strcony(1,econty(xx,y));
end

cvergx = (strconpx-abs(strchkx))...
    /(strconx(1,econtx(xx,y)));
cvergy = (strconpy-abs(strchky))...
    /(strcony(1,econty(xx,y)));

%------------------------------------------------------------------
%Check if loading has caused exceededance of strain control point
%------------------------------------------------------------------
if cvergx < -tol || cvergy < -tol
  if cvergx < cvergy
   if cmin > cvergx
    cmin = cvergx;
    itr=1;
    stconmin=strconpx;
    strn= exx*(1 + portx1 + portx2);

   end
  elseif cvergy < cvergx
   if cmin > cvergy
    cmin = cvergy;
    itr=1;
    stconmin=strconpy;
    strn= eyy*(1 + porty1 + porty2);

   end
  end

%------------------------------------------------------------------
%Can the load be automatically increased to increse convergence rate
%------------------------------------------------------------------
elseif cvergx > tol && cvergy > tol && itr~=1
  if cvergx < cvergy
   if cmax > cvergx
```

```
        cmax = cvergx;
        stconmax=strconpx;
        strn= exx*(1 + portx1 + portx2);

       end

      elseif cvergy < cvergx
       if cmax > cvergy
        cmax = cvergy;
        stconmax=strconpy;
        strn= eyy*(1 + porty1 + porty2);

       end
      end

    %-----------------------------------------------------------------
    %Convergence for particular guass point and stiffness reduction
    %-----------------------------------------------------------------
    elseif itr~=1
      if abs(cvergx) <= tol
       itr=0;
       logX(xx,y)=1;
      end

      if abs(cvergy) <= tol
       itr=0;
       logY(xx,y)=1;
      end

    end

   end
 end
end

%-----------------------------------------------------
%Total Convergence not achieved and load reduction
%-----------------------------------------------------
if itr==1
 mecon(1)=max(econtx(:));
 mecon(2)=max(econty(:));
 maxecon=max(mecon(:));
 L=L*abs(stconmin/strn);

%-----------------------------------------------------
%Total Convergence not achieved and load increase
%-----------------------------------------------------
elseif itr==2
 mecon(1)=max(econtx(:));
 mecon(2)=max(econty(:));
 maxecon=max(mecon(:));
 L=L*abs(stconmax/strn);
```

```
%-----------------------------------------------------
%Total Convergence is achieved and load initialised
%-----------------------------------------------------
else
 mecon(1)=max(econtx(:));
 mecon(2)=max(econty(:));
 maxecon=max(mecon(:));
 L=fvec1;
end
```

Function Linear Material Matrix:

```
function [Cb,Cs] = Lmatmatrix(Ex,Ey,vx,vy,nel,numgpr,numgps)
%Obtain the stiffness matrix, shape and strain interpolation matrix for the four and nine noded
%Ex, Ey, vx & vy are matrices setup by (Element Number, Guass point)
%Cb is matrix setup by (:,:,Guass point, Element Number)
%Cs is matrix setup by (:,:,Element Number)
%-------------------------------------------------------------
%Initialise matrices
%-------------------------------------------------------------

tnumgp=numgpr*numgps;
Cb=zeros(3,3,tnumgp,nel);
Cs=zeros(2,2,nel);

%-------------------------------------------------------------
%Bending and shear material matrix
%-------------------------------------------------------------
for y=1:nel
   xx=0;
  %-----------------------------------------------------------
  %Bending material matrix
  %-----------------------------------------------------------
  for x=1:numgpr
   for z=1:numgps
    vxt=vx;
    vyt=vy;
    xx=xx+1;
    G=(Ex*Ey)/(Ex*(1+vxt)+Ey*(1+vyt));
    Cb(:,:,xx,y)=[Ex/(1-vxt*vyt),Ey*vxt/(1-vxt*vyt),0;...
    Ex*vyt/(1-vxt*vyt),Ey/(1-vxt*vyt),0;0,0,G];
   end
  end
  %-----------------------------------------------------------
  %Shear material matrix
  %-----------------------------------------------------------
  Cs(:,:,y)=[G,0;0,G];
end
```

Function Nonlinear Material Matrix:

```
function [Cb,Cs] = NLmatmatrix(Ex,Ey,vx,vy,nel,numgpr,numgps)
%Obtain the stiffness matrix, shape and strain interpolation matrix for the four and nine noded
%Ex, Ey, vx & vy are matrices setup by (Element Number, Guass point)
%Cb is matrix setup by (:,:,Guass point, Element Number)
%Cs is matrix setup by (:,:,Element Number)
%-------------------------------------------------------------
%Initialise matrices
%-------------------------------------------------------------

tnumgp=numgpr*numgps;
Cb=zeros(3,3,tnumgp,nel);
Cs=zeros(2,2,nel);
Exnet=zeros(nel);
Eynet=zeros(nel);
Vxnet=zeros(nel);
Vynet=zeros(nel);

%-------------------------------------------------------------
%Bending and shear material matrix
%-------------------------------------------------------------
for y=1:nel
  xx=0;
 %-------------------------------------------------------------
 %Bending material matrix
 %-------------------------------------------------------------
 for x=1:numgpr
  for z=1:numgps
   xx=xx+1;
   vxt=vx(y,xx);
   vyt=vy(y,xx);

   G=(Ex(y,xx)*Ey(y,xx))/(Ex(y,xx)*(1+vxt)+Ey(y,xx)*(1+vyt));
   Cb(:,:,xx,y)=[Ex(y,xx)/(1-vxt*vyt),Ey(y,xx)*vxt/(1-vxt*vyt),0;...
   Ex(y,xx)*vyt/(1-vxt*vyt),Ey(y,xx)/(1-vxt*vyt),0;0,0,G];

   Exnet(y)=Exnet(y)+Ex(y,xx);
   Eynet(y)=Eynet(y)+Ey(y,xx);
   Vxnet(y)=Vxnet(y)+vxt;
   Vynet(y)=Vynet(y)+vyt;

  end
 end

 Exnet(y)=Exnet(y)/tnumgp;
 Eynet(y)=Eynet(y)/tnumgp;
 Vxnet(y)=Vxnet(y)/tnumgp;
 Vynet(y)=Vynet(y)/tnumgp;
%-------------------------------------------------------------
%Shear material matrix
%-------------------------------------------------------------
 vxt=Vxnet(y);
 vyt=Vynet(y);
```

```
  G=(Exnet(y)*Eynet(y))/(Exnet(y)*(1+vxt)+Eynet(y)*(1+vyt));
  Cs(:,:,y)=[G,0;0,G];
end
```

Function Boundary Conditions:

```
function [Kmod, fvecmod] = applybc(bcdof,fvec,Ksys)
%Obtain modified stiffness matrix and force vector after aplication of
%... zero boundary conditions

%----------------------------------------------------------------
%Modify System Stiffness Matrices and Force Vector
%----------------------------------------------------------------
q=length(bcdof);        %Length of zero boundary condition matrix
Kmod=Ksys;              %Use a modified stifness matrix
fvecmod=fvec;           %give the modified force vector the value of input force vector
for y=1:q
  i=bcdof(1,y);         %Extract degree of freedom
    Kmod(i,:)=0;        %Change the row in the n-th row to zero
    Kmod(:,i)=0;        %Change the column in the n-th column to zero
  Kmod(i,i)=1;          %Change the diagonal in the n-th row to zero
  fvecmod(i,1)=0;       %Change to force vector to 0
end
```

Function Assemble System Stiffness Matrix:

```
function [Ksys] = assemble(sdof,index,Kel,nel)
%Obtain system stiffness matrix
%Assemble element matrix into system stifness matrix

%----------------------------------------------------------------
%Initialise Matrices
%----------------------------------------------------------------
Ksys=zeros(sdof,sdof);
i=size(index,2);

%----------------------------------------------------------------
%Assemble System Stiffness Matrices
%----------------------------------------------------------------
 for y=1:nel
  for x=1:i
    a=index(y,x);
     for w=1:i
      b=index(y,w);
      Ksys(a,b)=Kel(x,w,y)+Ksys(a,b);
     end
  end
 end
```

Function Strain Interpolation Matrix:

```
function [Bb,Bs,BbRes,BsRes] = bmatrix(elcoord,eltype,nel,numgpr,numgps,ndof,ngprshear,ngpsshear)
%Obtain the stiffness matrix, shape and strain interpolation matrix for the four and eight noded
%Bb, Bs and H Matrix is size [nodf x ndof*eltype x numpgr*numgps x nel]
```

```
%Kel Matrix is size [nodf*eltype x ndof*eltype x nel]

%-------------------------------------------------------------------
%Obtain Intergration Points for bending
%-------------------------------------------------------------------
[gpvBb, ~] = ginter2(numgpr,numgps);

%-------------------------------------------------------------------
%Obtain Jacobian Matrix bending
%-------------------------------------------------------------------
[~, ijacobb] = jacobian(elcoord,eltype,nel,numgpr,numgps);

%-------------------------------------------------------------------
%Obtain Reduced Intergration Points for shear
%-------------------------------------------------------------------
[gpvBs, ~] = ginter2(ngprshear,ngpsshear);

%-------------------------------------------------------------------
%Obtain Jacobian Matrix shear
%-------------------------------------------------------------------
[~, ijacobs] = jacobian(elcoord,eltype,nel,ngprshear,ngpsshear);

%-------------------------------------------------------------------
%Initialise Matrices
%-------------------------------------------------------------------
j=ndof*eltype;
k=numgpr*numgps;

Bb=zeros(3,j,k,nel);                %Bending Strain interpolation Matrix
Bs=zeros(2,j,k,nel);                %Shear Strain interpolation Matrix
BbRes=zeros(3,j,k,nel);             %Results Bending Strain interpolation Matrix
BsRes=zeros(2,j,k,nel);             %Results Shear Strain interpolation Matrix
dhdrj=zeros(eltype,1);
dhdsj=zeros(eltype,1);

%-------------------------------------------------------------------
%Obtain bending and shear strain interpolation matrices
%-------------------------------------------------------------------
for y=1:nel
 q=0;
 u=0;
 for x=1:numgpr
   r=gpvBb(x,1);
   for z=1:numgps
   s=gpvBb(z,2);
   q=q+1;
    [~, dhr, dhs] = shapefunctions(eltype,r,s);

    for i=1:eltype
     %-------------------------------------------------------------
     %Transfer derivatives natural to global
     %-------------------------------------------------------------
     dhdrj(i)=ijacobb(1,1,q,y)*dhr(i,1)+ijacobb(1,2,q,y)*dhs(i,1);
```

```
    dhdsj(i)=ijacobb(2,1,q,y)*dhr(i,1)+ijacobb(2,2,q,y)*dhs(i,1);
    g=(i-1)*ndof;

    %------------------------------------------------------------------
    %Bending Strain Matrix
    %------------------------------------------------------------------
    Bb(2,2+g,q,y)=dhdsj(i);
    Bb(3,2+g,q,y)=dhdrj(i);
    Bb(1,3+g,q,y)=-dhdrj(i);
    Bb(3,3+g,q,y)=-dhdsj(i);
    BbRes(:,:,q,y)=Bb(:,:,q,y);
   end
  end
end


for x=1:ngprshear
 rshear=gpvBs(x,1);                    %replace with gauss point isoparametric r value
 for z=1:ngpsshear
   sshear=gpvBs(z,2);                   %replace with gauss point isoparametric s value
   u=u+1;
   [~, dhr, dhs] = shapefunctions(eltype,rshear,sshear);

   [hs, ~, ~] = shapefunctions(eltype,rshear,sshear);

   for i=1:eltype
    %------------------------------------------------------------------
    %Transfer derivatives natural to global
    %------------------------------------------------------------------
    dhdrj(i)=ijacobs(1,1,u,y)*dhr(i,1)+ijacobs(1,2,u,y)*dhs(i,1);
    dhdsj(i)=ijacobs(2,1,u,y)*dhr(i,1)+ijacobs(2,2,u,y)*dhs(i,1);
    g=(i-1)*ndof;

    %------------------------------------------------------------------
    %Shear Strain Matrix
    %------------------------------------------------------------------
    Bs(1,1+g,u,y)=dhdrj(i);
    Bs(2,1+g,u,y)=dhdsj(i);
    Bs(2,2+g,u,y)=-hs(i);
    Bs(1,3+g,u,y)=hs(i);
    BsRes(:,:,u,y)=Bs(:,:,u,y);
   end
 end
end

end
```

Function Connectivity Matrix:

```
function [nodes, index] = connect(numdivb,numdivl,eltype,nel,ndof)
%Detail connectivity between elements
%nodes Matrix is a nel x (4 / 8) matrix rows holding elment number &
%...global node number relating to local node number

%-------------------------------------------------------------
%Connectivity for quad eight
%-------------------------------------------------------------
elseif eltype==8
 nodes=zeros(nel,8);
 index=zeros(1,8*ndof);

 %-------------------------------------------------------------
 %Relate local element side node numbers to global nodal numbers
 %-------------------------------------------------------------
 for x=1:numdivb
  for y=1:numdivl
   i=(x)+(numdivb*(y-1));
   inc=3*numdivb+2;
   nodes(i,1)=(inc*(y))+((x*2)-1);
   nodes(i,2)=(inc*(y-1))+((x*2)-1);
   nodes(i,3)=(inc*(y-1))+((x*2)+1);
   nodes(i,4)=(inc*(y))+((x*2)+1);
   nodes(i,6)=(inc*(y-1))+((x*2));
   nodes(i,8)=(inc*(y))+((x*2));
  end
 end

 %-------------------------------------------------------------------------
 %Relate local element upper/lower node numbers to global nodal numbers
 %-------------------------------------------------------------------------
 for x=1:numdivb
  for y=1:numdivl
   inc=3*numdivb+2;
   strt=2*numdivb+1;
   i=(x)+((numdivb)*(y-1));
   nodes(i,5)=(inc*(y-1))+(x+strt);
   nodes(i,7)=(inc*(y-1))+(x+strt+1);
  end
  i=i+(2*numdivb+1);
 end

 %-------------------------------------------------------------
 %Create index matrix for future use
 %-------------------------------------------------------------
 for y=1:nel
  for x=1:8
   i=nodes(y,x);          %Assign index
   for df=1:ndof
    j=(i*ndof-(ndof))+df;   %Obtain Dof
    l=(x*ndof-(ndof))+df;   %Dof to index ref
    index(y,l)=j;          %Add value to index Matrix
```

```
    end
   end
  end

end
```

Function Calculate Strain Control Points:

```
function [Econx,Econy] = econ(strconx,stressconx,strcony,stresscony)
%find the e-modulus of the control points

%-------------------------------------------------------------------
%Obtain young's modulus of the control points
%-------------------------------------------------------------------
nump=length(strconx);
Econx=zeros(1,nump);
Econy=zeros(1,nump);

Econx(1)=0;
Econy(1)=0;
for x=2:nump
 Econx(x)=(stressconx(1,x)-stressconx(1,(x-1)))/(strconx(1,x)-strconx(1,(x-1)));
 Econy(x)=(stresscony(1,x)-stresscony(1,(x-1)))/(strcony(1,x)-strcony(1,(x-1)));
end
```

Function Calculate Curvatures and Strains:

```
function [del,curvstr,inplstr,shearstr] = esmatrix(BbRes,BsRes,d,nel,nnel,ndof,index,numgpr,numgps,ngprshear,ngpsshear,h)
%Obtain the strains from backsubstitution
%curvstr,inplstr,shearstr Matrix is size [1 x ndof*eltype x numpgr*numgps x nel]

%-------------------------------------------------------------
%Initialise Matrices
%-------------------------------------------------------------
j=numgpr*numgps;
jj=ngprshear*ngpsshear;
k=ndof*nnel;

del=zeros(k,1,nel);
curvstr=zeros(3,1,j,nel);
inplstr=zeros(3,1,j,nel);
shearstr=zeros(2,1,jj,nel);

%-------------------------------------------------------------
%Obtain element local displacement values
%-------------------------------------------------------------
for x=1:nel
 a=0;
 for y=1:nnel
  for z=1:ndof
   a=a+1;
   i=index(x,a);
   del(a,1,x)=d(i);
  end
```

```
 end
end

for y=1:nel
 q=0;
 u=0;
 for x=1:numgpr
   for z=1:numgps
    q=q+1;

    %---------------------------------------------------------------
    %Obtain gauss point curvatures
    %---------------------------------------------------------------
    curvstr(1,1,q,y)=BbRes(1,:,q,y)*del(:,1,y);
    curvstr(2,1,q,y)=BbRes(2,:,q,y)*del(:,1,y);
    curvstr(3,1,q,y)=BbRes(3,:,q,y)*del(:,1,y);

    %---------------------------------------------------------------
    %Obtain gauss point in-plane strains
    %---------------------------------------------------------------
    inplstr(1,1,q,y)=BbRes(1,:,q,y)*del(:,1,y)*h/2;
    inplstr(2,1,q,y)=BbRes(2,:,q,y)*del(:,1,y)*h/2;
    inplstr(3,1,q,y)=BbRes(3,:,q,y)*del(:,1,y)*h/2;
   end
 end
  for x2=1:ngprshear
   for z2=1:ngpsshear
    u=u+1;

    %---------------------------------------------------------------
    %Obtain gauss point shear strains
    %---------------------------------------------------------------
    shearstr(1,1,u,y)=BsRes(1,:,u,y)*del(:,1,y);
    shearstr(2,1,u,y)=BsRes(2,:,u,y)*del(:,1,y);
   end
  end


end
```

Function Calculate Surface Pressure to Nodal Load:

```
function [fvecS] = fvecS(elcoord,eltype,nel,numgpr,numgps,ndof,press,eleNo,sdof,index)
%Obtain the surface force vector
%H Matrix is size [nodf x ndof*eltype x numpgr*numgps x nel]
%fvecS Matrix is size [nodf*eltype x 1 x nel]

%---------------------------------------------------------------
%Obtain Intergration Points for force vector
%---------------------------------------------------------------
[gpvBf, gpwBf] = ginter2(numgpr,numgps);

%---------------------------------------------------------------
%Obtain Jacobian Matrix Det
```

```
%-------------------------------------------------------------
[djacobf, ~] = jacobian(elcoord,eltype,nel,numgpr,numgps);

%-------------------------------------------------------------
%Initialise Matrices
%-------------------------------------------------------------
j=ndof*eltype;
k=numgpr*numgps;

H=zeros(3,j,k,nel);                    %Shape Function Matrix
fvecSe=zeros(j,1,nel);                  %Force Vector Element Matrix
fvecS=zeros(sdof,1);                    %Force Vector Total Matrix
%-------------------------------------------------------------
%Obtain H Matrix
%-------------------------------------------------------------

for y=1:length(eleNo)
 q=0;
  for x=1:numgpr
   r=gpvBf(x,1);
   for z=1:numgps
   s=gpvBf(z,2);
   q=q+1;
    [hs, ~, ~] = shapefunctions(eltype,r,s);
    for i=1:eltype
     %-------------------------------------------------------------
     %Construct H Matrix
     %-------------------------------------------------------------
     g=(i-1)*ndof;

     H(1,1+g,q,y)=hs(i);
     H(2,2+g,q,y)=hs(i);
     H(3,3+g,q,y)=hs(i);
    end
   end
  end

 u=0;
%-------------------------------------------------------------
%Surface Force Vector Component
%-------------------------------------------------------------
for x=1:numgpr
 rwgtf=gpwBf(x,1);                      %replace with gauss point isoparametric r weight
 for z=1:numgps
  swgtf=gpwBf(z,2);                     %replace with gauss point isoparametric s weight
  u=u+1;
  fvecSe(:,1,y)=fvecSe(:,1,y)+transpose(H(1,:,u,y))*(press)...
    *swgtf*rwgtf*djacobf(:,:,u,y);
 end
end
%-------------------------------------------------------------
%Modified Global Force Vector Component
%-------------------------------------------------------------
```

```
 for rr=1:eltype
  yy=rr*3-2;
  xx=index(eleNo(y),yy);
  fvecS(xx,1)=fvecS(xx,1)+fvecSe(yy,1);
 end
end
```

Function Gauss Integration 1D:

```
function [gpval1, gpweight1] = ginter1(numgp)
%Obtain guass point values and associated weights

%---------------------------------------------------------------
%Initialise Matrices
%---------------------------------------------------------------
gpval1=zeros(numgp,1);        %Inital Gauss point value matrix
gpweight1=zeros(numgp,1);     %Inital Gauss point weight matrix

%---------------------------------------------------------------
%One point gauss integration
%---------------------------------------------------------------
if numgp==1
 gpval1(1)=0;
 gpweight1(1)=2;

%---------------------------------------------------------------
%Two point gauss integration
%---------------------------------------------------------------
elseif numgp==2
 gpval1(1)=0.577350269189626;
 gpweight1(1)=1;
 gpval1(2)=-0.577350269189626;
 gpweight1(2)=1;

%---------------------------------------------------------------
%Three point gauss integration
%---------------------------------------------------------------
elseif numgp==3
 gpval1(1)=0.774596669241483;
 gpweight1(1)=0.555555555555556;
 gpval1(2)=0;
 gpweight1(2)=0.888888888888889;
 gpval1(3)=-0.774596669241483;
 gpweight1(3)=0.555555555555556;

end
```

Function Gauss Integration 2D:

```
function [gpval2, gpweight2] = ginter2(numgpr,numgps)
%2D gauss point integration

%---------------------------------------------------------------
%Find larger of r and s directional integeration
%---------------------------------------------------------------
if numgpr > numgps
 numgp=numgpr;
else
 numgp=numgps;
end

%---------------------------------------------------------------
%Initialise Matrices
%---------------------------------------------------------------
gpval2=zeros(numgp,2);
gpweight2=zeros(numgp,2);

[gpvalr,gpweightr]=ginter1(numgpr);
[gpvals,gpweights]=ginter1(numgps);

%---------------------------------------------------------------
%Obtain values and weights for r direction
%---------------------------------------------------------------
for intr=1:numgpr
 gpval2(intr,1)=gpvalr(intr);
 gpweight2(intr,1)=gpweightr(intr);
end

%---------------------------------------------------------------
%Obtain values and weights for s direction
%---------------------------------------------------------------
for ints=1:numgps
 gpval2(ints,2)=gpvals(ints);
 gpweight2(ints,2)=gpweights(ints);
end
```

Function Gauss Integration 3D:

```
function [gpval3, gpweight3] = ginter3(numgpr,numgps)
%3D gauss point integration

%---------------------------------------------------------------
%Find larger of r and s directional integeration
%---------------------------------------------------------------
if numgpr > numgps
 numgp=numgpr;
else
 numgp=numgps;
end

%---------------------------------------------------------------
```

```
%Initialise Matrices
%------------------------------------------------------------------
gpval3=zeros(numgp,2);
gpweight3=zeros(numgp,2);

[gpvalr,gpweightr]=ginter1(numgpr);
[gpvals,gpweights]=ginter1(numgps);

%------------------------------------------------------------------
%Obtain values and weights for r direction
%------------------------------------------------------------------
for intr=1:numgpr
 gpval3(intr,1)=gpvalr(intr);
 gpweight3(intr,1)=gpweightr(intr);
end

%------------------------------------------------------------------
%Obtain values and weights for s direction
%------------------------------------------------------------------
for ints=1:numgps
 gpval3(ints,2)=gpvals(ints);
 gpweight3(ints,2)=gpweights(ints);
end
```

Function Jacobian Values:

```
function [djacob, ijacob] = jacobian(elcoord,eltype,nel,numgpr,numgps)
%Obtain the jacobian inverse and determinant matrix for the four and
%...eight noded plate elements

%------------------------------------------------------------------
%Initialise Matrices
%------------------------------------------------------------------
[gpval2, ~] = ginter2(numgpr,numgps);
k=numgpr*numgps;
jacob=zeros(2,2,k,nel);         %jacobian matrix is a 2x2xGPxN matrix
ijacob=zeros(2,2,k,nel);        %inverse jacobian matrix is a 2x2xGPxN matrix
djacob=zeros(1,1,k,nel);        %inverse jacobian matrix is a 2x2xGPxN matrix

%------------------------------------------------------------------
%Obtain Jacobian inverses and determinants for gauss points
%------------------------------------------------------------------
for y=1:nel
 q=0;
 for x=1:numgpr
  r=gpval2(x,1);    %replace with gauss point isoparametric r value
  for z=1:numgps
   s=gpval2(z,2);   %replace with gauss point isoparametric s value
   q=q+1;
   [~, dhr, dhs] = shapefunctions(eltype,r,s);
   for i=1:eltype
    jacob(1,1,q,y)=jacob(1,1,q,y)+dhr(i)*elcoord(1,i,y);
    jacob(1,2,q,y)=jacob(1,2,q,y)+dhr(i)*elcoord(2,i,y);
    jacob(2,1,q,y)=jacob(2,1,q,y)+dhs(i)*elcoord(1,i,y);
```

```
    jacob(2,2,q,y)=jacob(2,2,q,y)+dhs(i)*elcoord(2,i,y);
   end
   ijacob(:,:,q,y)=(jacob(:,:,q,y))^-1;
   djacob(1,1,q,y)=det(jacob(:,:,q,y));
  end
 end
end
```

Function Element Stiffness Matrix:

```
function [Kel] = kmatrix(elcoord,eltype,nel,numgpr,numgps,ndof,Cb,Cs,Bb,Bs,h,ngprshear,ngpsshear)
%Obtain the stiffness matrix for the four and nine noded
%Kel Matrix is size [nodf*eltype x ndof*eltype x nel]

%-------------------------------------------------------------
%Shear Correction Factor
%-------------------------------------------------------------
ks=5/6;

%-------------------------------------------------------------
%Obtain Jacobian Matrix bending
%-------------------------------------------------------------
[djacobb, ~] = jacobian(elcoord,eltype,nel,numgpr,numgps);

%-------------------------------------------------------------
%Obtain Intergration Points for bending
%-------------------------------------------------------------
[~, gpwBb] = ginter2(numgpr,numgps);

%-------------------------------------------------------------
%Obtain Reduced Intergration Points for shear
%-------------------------------------------------------------
[~, gpwBs] = ginter2(ngprshear,ngpsshear);

%-------------------------------------------------------------
%Obtain Jacobian Matrix shear
%-------------------------------------------------------------
[djacobs, ~] = jacobian(elcoord,eltype,nel,ngprshear,ngpsshear);

%-------------------------------------------------------------
%Initialise Matrices
%-------------------------------------------------------------
j=ndof*eltype;
Kel=zeros(j,j,nel);                    %Element Stiffness Matrix
Kelb=zeros(j,j,nel);                   %Element Bending Stiffness Matrix
Kels=zeros(j,j,nel);                   %Element Shear Stiffness Matrix

%-------------------------------------------------------------
%Obtain bending and shear stiffness matrices
%-------------------------------------------------------------
for y=1:nel
 q=0;
 u=0;
 %-------------------------------------------------------------
```

```
%Bending Component
%---------------------------------------------------------------
for x=1:numgpr
 rwgtb=gpwBb(x,1);                    %replace with gauss point isoparametric r weight
 for z=1:numgps
  swgtb=gpwBb(z,2);                    %replace with gauss point isoparametric s weight
  q=q+1;
  Kelb(:,:,y)=Kelb(:,:,y)+(h^3)/(12)*transpose(Bb(:,:,q,y))*Cb(:,:,q,y)*Bb(:,:,q,y)...
    *swgtb*rwgtb*djacobb(:,:,q,y);
 end
end

%---------------------------------------------------------------
%Shear Component
%---------------------------------------------------------------
for x=1:ngprshear
 rwgts=gpwBs(x,1);                    %replace with gauss point isoparametric r weight
 for z=1:ngpsshear
  swgts=gpwBs(z,2);                    %replace with gauss point isoparametric s weight
  u=u+1;
  Kels(:,:,y)=Kels(:,:,y)+(h*ks)*transpose(Bs(:,:,u,y))*Cs(:,:,y)*Bs(:,:,u,y)...
    *swgts*rwgts*djacobs(:,:,u,y);
 end
end
 %---------------------------------------------------------------
 %Total element stiffness matrix
 %---------------------------------------------------------------
 Kel(:,:,y)=Kelb(:,:,y)+Kels(:,:,y);
End
```

Function Local Element Co-ords:

```
function [elcoord] = localcoord(eltype,nel,gcoord,nodes)
%Assign local co-ord in terms of x,y per node 1 to n
%elcoord Matrix is a 2 x(4 / 8)x nel matrix rows holding x & y

%---------------------------------------------------------------
%Initialise Matrices
%---------------------------------------------------------------
avex=zeros(1,nel);
avey=zeros(1,nel);
g=eltype;
elglcoord=zeros(2,g,nel);      %2x4/8xN Matrix holding global [x1 ... xn; y1 ... yn]
elcoord=zeros(2,g,nel);        %2x4/8xN Matrix holding local [x1 ... xn; y1 ... yn]

%---------------------------------------------------------------
%Global node co-ordinates for element and centroid
%---------------------------------------------------------------
 for y=1:nel
  for x=1:g
    i=nodes(y,x);             %Assign index
    a=gcoord(i,1);            %Obtain element global nodal coord
    b=gcoord(i,2);
    elglcoord(1,x,y)=a;       %Assign to global element coord Matrix
```

```
    elglcoord(2,x,y)=b;
    avex(1,y)=avex(1,y)+a;
    avey(1,y)=avey(1,y)+b;
   end
  avex(1,y)=avex(1,y)/g;      %Get x value of the element centroid
  avey(1,y)=avey(1,y)/g;      %Get y value of the element centroid
 end

%-------------------------------------------------------------------
%Obtain local node co-ordinates for element and local centroid
%-------------------------------------------------------------------
 for w=1:nel
  for z=1:g
    i=nodes(w,z);             %Assign index
    a=gcoord(i,1);            %Obtain element global nodal coord
    b=gcoord(i,2);
    a1= a-avex(1,w);          %Obtain element global nodal coord
    b1= -b+avey(1,w);
    elcoord(1,z,w)=a1;        %Assign to local element coord Matrix
    elcoord(2,z,w)=-b1;
  end
 end
```

Function Node Global Co-ords:

```
function [gcoord] = nodecoord(numdivb,numdivl,eleb,elel,eltype,nnode)
%Assign global co-ord in terms of x,y per node 1 to n
%gcoord Matrix is a 2x(4 / 8) matrix rows holding x & y

%-------------------------------------------------------------------
%Initialise Matrices
%-------------------------------------------------------------------
gcoord=zeros(nnode,2);

%-------------------------------------------------------------------
%Global node co-ordinates for quad four
%-------------------------------------------------------------------
if eltype==4
 i=0;
 for x=1:numdivl+1
   for y=1:numdivb+1
    i=i+1;
    gcoord(i,1)=(x-1)*elel;
    gcoord(i,2)=-(y-1)*eleb;
   end
 end

%-------------------------------------------------------------------
%Global node co-ordinates for quad eight
%-------------------------------------------------------------------
elseif eltype==8
 i=0;
 for x=1:numdivl+1        %Assign side node coord quad8
   for y=1:2*numdivb+1
```

```
    i=i+1;
    gcoord(i,1)=(x-1)*elel;
    gcoord(i,2)=-(y-1)*eleb/2;
   end
   i=i+(numdivb+1);
 end

 i=2*numdivb+1;
 for x=1:numdivl          %Assign upper and lower mid node coord quad8
  for y=1:numdivb+1
   i=(i+1);
   gcoord(i,1)=(x-1)*elel+elel/2;
   gcoord(i,2)=-(y-1)*eleb;
  end
  i=i+(2*numdivb+1);
 end
end
```

Function Shape Functions:

```
function [hs, dhr, dhs] = shapefunctions(eltype,r,s)
%Obtain the shape function matrix for the four and eight noded
%...plate elements

%-------------------------------------------------------------
%Initialise Matrices
%-------------------------------------------------------------
hs=zeros(eltype,1);
dhr=zeros(eltype,1);
dhs=zeros(eltype,1);

%-------------------------------------------------------------
%Shape functions and derivatives for eight noded element
%-------------------------------------------------------------

%-------------------------------------------------------------
%Shape functions
%-------------------------------------------------------------
 hs(1)=  1/4*(s+1)*(r+1)*(r+s-1);
 hs(2)=  1/4*(s+1)*(r-1)*(r-s+1);
 hs(3)= -1/4*(s-1)*(r-1)*(r+s+1);
 hs(4)=  1/4*(s-1)*(r+1)*(s-r+1);
 hs(5)= -1/2*(r-1)*(r+1)*(s+1);
 hs(6)=  1/2*(r-1)*(s-1)*(s+1);
 hs(7)=  1/2*(r-1)*(r+1)*(s-1);
 hs(8)= -1/2*(r+1)*(s-1)*(s+1);

%-------------------------------------------------------------
%Derivatives in terms of r
%-------------------------------------------------------------
 dhr(1)=  1/4*(s+1)*(2*r+s);
 dhr(2)=  1/4*(s+1)*(2*r-s);
 dhr(3)= -1/4*(s-1)*(2*r+s);
 dhr(4)= -1/4*(s-1)*(2*r-s);
```

```
 dhr(5)= -r*(s+1);
 dhr(6)=  1/2*(s-1)*(s+1);
 dhr(7)=  r*(s-1);
 dhr(8)= -1/2*(s-1)*(s+1);

%-------------------------------------------------------------
%Derivatives in terms of s
%-------------------------------------------------------------
 dhs(1)=  1/4*(2*s+r)*(r+1);
 dhs(2)=  1/4*(-2*s+r)*(r-1);
 dhs(3)= -1/4*(2*s+r)*(r-1);
 dhs(4)= -1/4*(-2*s+r)*(r+1);
 dhs(5)= -1/2*(r-1)*(r+1);
 dhs(6)=  s*(r-1);
 dhs(7)=  1/2*(r-1)*(r+1);
 dhs(8)= -s*(r+1);
```

Function Calculate Stresses, Forces and Moments:

```
function [inplstress,shearstress,pltmom,pltshear] =
stressmatrixNL(inplstr,shearstr,nel,Cb,Cs,numgpr,numgps,h,ngprshear,ngpsshear)
%Obtain inplane stresses, moments and out of plane shear stress, force

%-------------------------------------------------------------
%Shear Correction Factor
%-------------------------------------------------------------
ks=5/6;

%-------------------------------------------------------------
%Initialise Matrices
%-------------------------------------------------------------
j=numgpr*numgps;
jj=ngprshear*ngpsshear;
a=nel;

inplstress=zeros(3,1,j,a);
shearstress=zeros(2,1,jj,a);
pltmom=zeros(3,1,j,a);
pltshear=zeros(3,1,jj,a);

for y=1:nel
 q=0;
 u=0;
 for x=1:numgpr
   for z=1:numgps
    q=q+1;

    %-------------------------------------------------------------
    %Obtain gauss point in-plane stresses
    %-------------------------------------------------------------
    inplstress(1,1,q,y)=Cb(1,:,q,y)*inplstr(:,1,q,y);
    inplstress(2,1,q,y)=Cb(2,:,q,y)*inplstr(:,1,q,y);
    inplstress(3,1,q,y)=Cb(3,:,q,y)*inplstr(:,1,q,y);
```

```
    %-----------------------------------------------------------------
    %Obtain gauss point moments / m
    %-----------------------------------------------------------------
    pltmom(1,1,q,y)=Cb(1,:,q,y)*inplstr(:,1,q,y)*(h^2)/6;
    pltmom(2,1,q,y)=Cb(2,:,q,y)*inplstr(:,1,q,y)*(h^2)/6;
    pltmom(3,1,q,y)=Cb(3,:,q,y)*inplstr(:,1,q,y)*(h^2)/6;
   end
  end

for x2=1:ngprshear
   for z2=1:ngpsshear
    u=u+1;
    %-----------------------------------------------------------------
    %Obtain gauss point out of plane stresses
    %-----------------------------------------------------------------
    shearstress(1,1,u,y)=ks*Cs(1,:,y)*shearstr(:,1,u,y);
    shearstress(2,1,u,y)=ks*Cs(2,:,y)*shearstr(:,1,u,y);

    %-----------------------------------------------------------------
    %Obtain gauss point out of plane force / m
    %-----------------------------------------------------------------
    pltshear(1,1,u,y)=shearstress(1,1,u,y)*h;
    pltshear(2,1,u,y)=shearstress(2,1,u,y)*h;
   end
  end
end
```

Function Write Results for Nonlinear Solver:

```
%Write Results to file

xx=0;

if Lstp<254
mx=zeros((numgpr*numgps*nel),Lstp);
my=zeros((numgpr*numgps*nel),Lstp);
mxy=zeros((numgpr*numgps*nel),Lstp);

sx=zeros((numgpr*numgps*nel),Lstp);
sy=zeros((numgpr*numgps*nel),Lstp);
sxy=zeros((numgpr*numgps*nel),Lstp);

cx=zeros((numgpr*numgps*nel),Lstp);
cy=zeros((numgpr*numgps*nel),Lstp);
cxy=zeros((numgpr*numgps*nel),Lstp);

ex=zeros((numgpr*numgps*nel),Lstp);
ey=zeros((numgpr*numgps*nel),Lstp);
exy=zeros((numgpr*numgps*nel),Lstp);

dw=zeros(nnode,Lstp);
drx=zeros(nnode,Lstp);
```

```
dry=zeros(nnode,Lstp);

fw=zeros(nnode,Lstp);
frx=zeros(nnode,Lstp);
fry=zeros(nnode,Lstp);

for z=1:Lstp
 xx=0;
 for x=1:nel
  for y=1:(numgpr*numgps)
   xx=xx+1;

   mx(xx,z)=pltmomf(1,1,y,x,z);
   my(xx,z)=pltmomf(2,1,y,x,z);
   mxy(xx,z)=pltmomf(3,1,y,x,z);

   sx(xx,z)=inplstressf(1,1,y,x,z);
   sy(xx,z)=inplstressf(2,1,y,x,z);
   sxy(xx,z)=inplstressf(3,1,y,x,z);

   cx(xx,z)=curvstrf(1,1,y,x,z);
   cy(xx,z)=curvstrf(2,1,y,x,z);
   cxy(xx,z)=curvstrf(3,1,y,x,z);

   ex(xx,z)=inplstrf(1,1,y,x,z);
   ey(xx,z)=inplstrf(2,1,y,x,z);
   exy(xx,z)=inplstrf(3,1,y,x,z);

  end
 end
end


for t=1:nnode
 dw(t,:)=dres((t*3-2),:);
 drx(t,:)=dres((t*3-1),:);
 dry(t,:)=dres((t*3-0),:);

 fw(t,:)=fvecT((t*3-2),:);
 frx(t,:)=fvecT((t*3-1),:);
 fry(t,:)=fvecT((t*3-0),:);
end
else
mx=zeros((numgpr*numgps*nel),254);
my=zeros((numgpr*numgps*nel),254);
mxy=zeros((numgpr*numgps*nel),254);

sx=zeros((numgpr*numgps*nel),254);
sy=zeros((numgpr*numgps*nel),254);
sxy=zeros((numgpr*numgps*nel),254);

cx=zeros((numgpr*numgps*nel),254);
cy=zeros((numgpr*numgps*nel),254);
```

```
cxy=zeros((numgpr*numgps*nel),254);

ex=zeros((numgpr*numgps*nel),254);
ey=zeros((numgpr*numgps*nel),254);
exy=zeros((numgpr*numgps*nel),254);

dw=zeros(nnode,254);
drx=zeros(nnode,254);
dry=zeros(nnode,254);

fw=zeros(nnode,254);
frx=zeros(nnode,254);
fry=zeros(nnode,254);

for z=1:254
 xx=0;
 for x=1:nel
  for y=1:(numgpr*numgps)
   xx=xx+1;

   mx(xx,z)=pltmomf(1,1,y,x,z);
   my(xx,z)=pltmomf(2,1,y,x,z);
   mxy(xx,z)=pltmomf(3,1,y,x,z);

   sx(xx,z)=inplstressf(1,1,y,x,z);
   sy(xx,z)=inplstressf(2,1,y,x,z);
   sxy(xx,z)=inplstressf(3,1,y,x,z);

   cx(xx,z)=curvstrf(1,1,y,x,z);
   cy(xx,z)=curvstrf(2,1,y,x,z);
   cxy(xx,z)=curvstrf(3,1,y,x,z);

   ex(xx,z)=inplstrf(1,1,y,x,z);
   ey(xx,z)=inplstrf(2,1,y,x,z);
   exy(xx,z)=inplstrf(3,1,y,x,z);

  end
 end
end


for t=1:nnode
 dw(t,1:254)=dres((t*3-2),1:254);
 drx(t,1:254)=dres((t*3-1),1:254);
 dry(t,1:254)=dres((t*3-0),1:254);

 fw(t,1:254)=fvecT((t*3-2),1:254);
 frx(t,1:254)=fvecT((t*3-1),1:254);
 fry(t,1:254)=fvecT((t*3-0),1:254);
end
end

xlswrite('NLResults.xls', mx, 'mx_Result', 'B2');
```

```
xlswrite('NLResults.xls', my, 'my_Result', 'B2');
xlswrite('NLResults.xls', mxy, 'mxy_Result', 'B2');

xlswrite('NLResults.xls', sx, 'sx_Result', 'B2');
xlswrite('NLResults.xls', sy, 'sy_Result', 'B2');
xlswrite('NLResults.xls', sxy, 'sxy_Result', 'B2');

xlswrite('NLResults.xls', cx, 'cx_Result', 'B2');
xlswrite('NLResults.xls', cy, 'cy_Result', 'B2');
xlswrite('NLResults.xls', cxy, 'cxy_Result', 'B2');

xlswrite('NLResults.xls', ex, 'ex_Result', 'B2');
xlswrite('NLResults.xls', ey, 'ey_Result', 'B2');
xlswrite('NLResults.xls', exy, 'exy_Result', 'B2');

xlswrite('NLResults.xls', dw, 'dw_Result', 'B2');
xlswrite('NLResults.xls', drx, 'drx_Result', 'B2');
xlswrite('NLResults.xls', dry, 'dry_Result', 'B2');

xlswrite('NLResults.xls', fw, 'fw_Result', 'B2');
xlswrite('NLResults.xls', frx, 'frx_Result', 'B2');
xlswrite('NLResults.xls', fry, 'fry_Result', 'B2');
```