Sudan University of Science and Technology

College of Graduate Studies and Scientific Research

Linear Static Finite Element Analysis of Thin Beam under Bending Action using MATLAB

التحليل الخطي الاستاتيكي لطريقة العناصر المحددة لتحليل العارضات النحيفة المعرضة لعزوم الانحناء باستخدام الماتلاب

A thesis submitted in partial fulfillment of the requirement for the degree of Master of Science in Civil Engineering (Construction Engineering) to Civil Engineering Department, College of Engineering

By:

Murwan Elsir Mohamed Khalil Ali

B.Sc (Hon). Sudan University of Science and Technology. 2008

Supervised by:

Dr. Nuha Moawia Akasha Hilal

February 2022

الآيـــــــــــه

بسم الله الرحمن الرحيم

قال تعالى:-

(وَجَعَلَ فِيهَا رَوَٰسِىَ مِن فَوْقِهَا وَبَٰرَكَ فِيهَا وَقَدَّرَ فِيهَآ أَقْوَٰتَهَا فِىٓ أَرْبَعَةِ أَيَّامٍ سَوَآءً لِّلسَّآئِلِينَ)

صدق الله العظيم

سورة فصلت (10)

DEDICATION

Every challenging works needs self-efforts as well as guidance of elders especially those who very close to you.

My humble effort I dedicate to

My Mother,

Father,

Family,

Teachers,

And Friends.

Murwan Elsir Mohamed Khalil Ali

ACKNOWLEDGEMENT

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the thesis successfully.

I would like to express my deep and sincere gratitude to my research supervisor, **Dr. Nuha Moawia Akasha Hilal**, for giving me the opportunity to do research and providing invaluable guidance throughout this research. Her dynamism, vision, sincerity and motivation have deeply inspired me. She has taught me the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under her guidance. I am extremely grateful for what she has offered me.

I would also like to express my grateful appreciation to **Prof. Abdel-Rahman Elzuber Mohamed, Dr. Mudthir Bakri Hassan Almakki**, and **Dr. Fatah Elrahman Mohamed** for their valuable suggestions, and good comments.

My thanks also extend to my family and friends for their continuous encouragement and support to finish this research.

المستخلص

العارضات هي أكثر الأعضاء الانشائية شيوعاً، لهذا فان التحليل الانشائي للعارضات النحيفة قد نال اهتماماً كبيراً بغرض التصميم. العارضات معرضة لأحمال محورية تسبب تشوهات محورية، وأحمال عمودية على المحور تسبب تشوهات عمودية على المحور، هذه التشوهات العمودية على المحور ناتجة بسبب قوة القص التي تسبب تشوهات القص وعزوم الانحناء والتي تسبب تشوهات الانحناء. يمكن دراسة أنماط هذه التشوهات بشكل مستقل من الاخر عند التحليل الانشائي الخطي للعارضات المرنة.

في هذه الاطروحة تمت دراسة نمط تشوه الانحناء للعارضات النحيفة باستخدام نظرية أولر برنولي (Euler-Bernoulli theory) وذلك لتحليل الابيام النحيفة باستخدام طريقة العناصر المحددة للإزاحة بغرض تطوير برنامج للتحليل الخطي الاستاتيكي للعارضات النحيفة وللتنبؤ بالسلوك الإنشائي لانحناء العارضات، وتشوهات الانحناء هي مصنفة بانها مشكلة استمرارية من فئة الدرجة الاولي (Class C^1 continuity problem) والتي تتطلب استمرارية لكل من الإزاحة الرأسية والتفاضل الأول للإزاحة الرأسية وهو الميل في منحني التشوه (الدوران).

تمت صياغة العنصر المحدد على أساس عنصر محدد للعارضات النحيفة حيث يوجد به عقدتان بالأطراف وعدد اثنان درجة حرية لكل عقدة وهما الإزاحة الراسية والدوران. تم استخدام دالة الإزاحة المكعبة متعددة الحدود لهيرمت (Hermit) لدرجات الحرية الغير معلومة للعنصر المحدد وأيضا تم استخدامها لحل مشكلة الاستمرارية لكل من الازاحة العرضية والميل في منحني التشوه (الدوران). وتم تطوير برنامج حاسوبي بطريقة العناصر المحددة باستخدام الماتلاب اصدار (MATLAB R2019b) وايضاً تم اعتماد نمط البرمجة باستخدام نظام الملفات (m-file)، كما تم التحقق من نتائج البرنامج ومقارنتها مع تنائج مرجعية منشورة، كما تم تحليل تسع عارضات مختلفة معرضة لأحمال ومساند مختلفة بغرض التأكد من نتائج البرنامج، وكذلك تم عقد مقارنة لنتائج البرنامج التي تم الحصول عليها مع النتائج المرجعية المنشورة وكانت شبة متطابقة وحيث ان هنالك توافق تام بين نتائج البرنامج والنتائج المرجعية، وعند التحليل لوحظ ان العنصر المحدود الذي به عقدتان كان أداءه جيد عند تحليل العارضات النحيفة، كما تم التأكد من هذا عند مقارنة نتائجه مع النتائج المرجعية المنشورة.

ABSTRACT

The analysis of thin beam structures become of great interest for designing purposes, because beams are most common type of structural component. Beams are generally subjected to both, axial loads causes (axial deformation) and transverse loads causes (transverse deformation). The latter is resulted from both shear forces (shear deformation) and bending moment (flexural deformation). For linearly elastic beams, these modes of deformation can be examined independently from one another.

In this research, Euler-Bernoulli beam flexural mode of deformation has been examined using finite element displacement methods (FEDM) to develop a linear static finite element computer program to predict bending behavior of thin beams under transverse loading. This bending deformation is classified as class C^1 problem (continuity C^1), that require continuity of both the transverse displacement and the fist derivatives of transverse displacement (slope).

The finite element formulation is based on two-node thin beam bending element with two degrees of freedom per each node (transverse vertical displacement w_i^e and rotation θ_i^e). Hermite cubic polynomial displacement function used for the element unknown degrees of freedom and to satisfy this class of continuity problem, the resulting element is conforming element and the convergence will be monotonic convergence.

The approach used in the deviation of the equation adopting the principle of minimum potential energy and the generalized coordinate approach.

A finite element computer program was developed and implemented using MATAB R2019b adopting m-file mode of programming.

The verification of the generated developed program results was compared with known published analytical exact solution results and published finite element analysis results for thin beam bending. Nine different numerical beam examples were conducted for this purpose with different loading and supporting conditions.

The developed program results obtained are in good agreement with published, and there is no significance difference between the results.

We conclude that using two nodes Euler-Bernoulli beam element was showed good performance after it used to analyze thin straight beams for bending. This was conformed when comparing results obtained with known published exact analytical solution.

TABLE OF CONTENTS

ABSTRACT……………………………………………………………v
TABLE OF CONTENTS…………………………………………...……vi
LIST OF TABLES……………………………………………..………xi
LIST OF FIGURES……………………………………….……….xiv
LIST OF ABBREVIATIONS ……………………………..………xix

CHAPTER ONE: INRODUCTION 1

1.1 Introduction *1*

1.2 Problem Statement *3*

1.3 Research Objective *4*

1.4 Methodology of Study *4*

1.5 Outline of Thesis *5*

CHAPTER TWO: LITERATURE REVIEW 6

2.1 General Introduction *6*

2.2 Linear Structural Analysis *6*

2.3 Beam Introduction *7*

2.4 Classical Beam Theories *11*

2.5 Euler-Bernoulli Beam Theory *12*
2.5.1 Introduction to Euler-Bernoulli Beam Theory 12
2.5.2 Kinematics of Deformation for Euler-Bernoulli Beam 12
2.5.2.1 Axial Displacement of Euler-Bernoulli Beam 13
2.5.2.2 Axial Strain of Euler-Bernoulli Beam 14
2.5.2.3 Axial Stress of Euler-Bernoulli Beam 14
2.5.2.4 Axial Force of Euler-Bernoulli Beam 14
2.5.2.5 Bending Moment of Euler-Bernoulli Beam 15
2.5.2.6 Generalized Constative Equation of Euler-Bernoulli Beam 15

2.6 The Governing Equilibrium PDE and BCs of Euler-Bernoulli Beam... 16
2.6.1 The Governing Equilibrium PDE 16
2.6.2 The Essential and Natural BCs 16

2.7 Analytical Methods *18*

2.8 Computational Methods *18*
2.8.1 Mathematical Models 18
2.8.2 Numerical Simulation 19
2.8.3 Numerical Methods 19
2.9 Finite Element Methods (FEM) *20*
2.9.1 General Description of FEM 20
2.9.2 Advantages of FEM 22
2.9.3 The Application of the FEM 22
2.9.4 General FEM Steps for Structural Mechanics Problem 23
2.9.5 The Weak-Integral Form of Equilibrium PDE 23
2.9.6 The Approaches for the Derivation of Finite Element Equations ... 24
2.9.6.1 Direct Approach 24
2.9.6.2 Weight Residual Approach 24
2.9.6.3 Variational Approach 25
2.9.7 The Two Selection Attributes for the FEM Choices 26
2.9.8 Finite Element Displacement Methods (FEDM) 26
2.10 Interpolation Displacement Function *26*
2.10.1 Guidelines for Selecting Polynomial Form of Displacement Function 27
2.11 Convergence Requirement for Polynomial Displacement *27*
2.11.1 Completeness Requirement 28
2.11.2 Continuity or Compatibility Requirement 28
2.12 Conforming and Non-Conforming Elements *29*
2.12.1 Conforming Elements 29
2.12.2 Non-Conforming Elements 30
2.13 Finite Element Solution *31*
2.13.1 The Error Concept in Finite Element Analysis 31
2.13.2 The Lowest Limit Solution of Minimum Potential Energy 31
2.13.3 Methods to Improve Convergence of Finite Element Results 31
2.13.4 Checking FEA Results 32
2.14 The Use of Computer for Numerical Computation *32*
2.14.1 Role of Computer in the Development of FEM 32
2.14.2 Commercial Finite Element Packages 33
2.14.3 General and Special Purpose Finite Element Program 34

2.14.4 The Standard Capabilities of General-Purpose Finite Element Program 35
2.14.5 The Basic Structure of a Finite Element Program 36

2.15 Computer Programming 37
2.15.1 Programming Languages 37

2.16 MATALB Program 38
2.16.1 Introduction to MATLAB Program 38
2.16.2 MATLAB Desktop Layout 39

2.17 Programming the FEM using MATLAB 41

CHAPTER THREE: LINEAR FINITE ELEMENT FORMULATION OF THIN BEAMS FOR FLEXURAL DEFORMATION 42

3.1 Introduction 42

3.2 The Governing PDE and BCs of (EB) Beam for Flexural Deformation 44
3.2.1 The Governing Equilibrium PDE OF EB Beam for Flexural Deformation 44
3.2.2 The Essential and Natural BCs of EB Beam for Flexural Deformation 45

3.3 Formulation of EB Beam for Flexural Deformation 45
3.3.1 Geometric Definition of Beam Element 45
3.3.2 Sign Convention of Beam Element 46
3.3.3 Hermite Interpolation Function 47
3.3.4 Element Displacement Field for Flexural Deformation 50
3.3.5 Element Strain Field for Flexural Deformation 50
3.3.6 Element Bending Moment or Stress Resultant Curvature Matrix 51
3.3.7 Element Transverse Shear Force 52
3.3.8 Element Stiffness Matrix for Flexural Deformation 52
3.3.8.1 Element Stiffness Matrix without Internal Hinge Considered 53
3.3.8.2 Element Stiffness Matrix with Internal Hinge Considered 53
3.3.9 Element Nodal Load Vector 54
3.3.9.1 The Work-Equivalence Method 54
3.3.9.2 Distributed Loading 56
3.3.9.3 Concentrated Loading 56
3.3.10 Coordinate Transformation 57
3.3.11 Global System of Equations 58
3.3.12 Support Nodal Reactions 58

CHAPTER FOUR: DISCRIPTION OF LINEAR FINITE ELEMENT COMPUTER PROGRAM 60

4.1 Introduction 60

4.2 The Structure of Finite Element Program 61

4.3 The Main Program 62

4.3.1 Sub-Program Sub_Input_Modal_Data.m 66
4.3.1.1 Geometrical Data 66
4.3.1.2 Nodal Coordinate Data 66
4.3.1.3 Element Connectivity Data 66
4.3.1.4 Element Materials and Geometrical Properties Data 67
4.3.1.5 Boundary Condition Data 67
4.3.1.6 Internal Hinges Data 67
4.3.1.7 Joint Nodal Load Data 68
4.3.1.8 Element Nodal Loads Data 68
4.3.2 Sub-Program Sub_Print_Display_Modal_Data.m 68
4.3.3 Sub-program Sub_Print_Solution_Results.m 68
4.3.4 Function $g = beam\text{-}g\ (i)$ 68
4.3.5 Function $F = form\_beam\text{-}F\ (F)$ 69
4.3.6 Function $KK = form\text{-}KK\ (KK, kg, g)$ 69
4.3.7 Function $kl = beam\text{-}k\ (i)$ 69
4.3.8 Function $F = Assem\_Elem\_loads\ (F, fg, g)$ 69
4.3.9 Function $F = Assem\_Joint\_loads\ (F)$ 69

4.4 Program Implementation 69

CHAPTER FIVE: VERIFICATION OF PROGRAM RESULTS 70

5.1 Introduction 70

5.2 Application to Numerical Examples 70

5.2.1 Example (1): Cantilever Beam Subjected to UDL. 70
5.2.1.1 Example (1): For One Element Solution at Nodal points......... 71
5.2.1.2 Example (1): For One Element Solution at Nodal points and Mid-Length 73
5.2.1.3 Example (1): For Two Elements Solution at Nodal Points 74
5.2.2 Example (2) Fixed-Fixed Beam Subjected to Concentrated Loads 76
5.2.3 Example (3): Overhanging Beam Subjected to UDL. 80
5.2.4 Example (4): Simple Supported Beam with varying Cross-Section Subjected to Concentrated Loads 83

5.2.5 Example (5): Cantilever Beam Subjected to UDL and Concentrated Loads. 87
5.2.6 Example (6): Continuous Beam Subjected to UDL and Concentrated Loads. 91
5.2.7 Example (7): Beam with Internal Hinge Subjected to UDL. 94
5.2.8 Example (8): Cantilever Beam Subjected to Linearly varying Distributed Load. 96
5.2.9 Example (9): Beam with Concentrated Load, UDL and Linearly Varying Distributed Load. 98

5.3 Discussion of The Results 100

CHAPTER SIX: CONCLUSIONS AND RECOMMENDATIONS 106

6.1 Conclusion 106

6.2 Recommendations for Further Studies 106

REFERENCES 107

APPENDIX (I) FINITE ELEMENT COMPUTER PROGRAM 109

APPENDIX (II) PROGRAM OUTPUT RESULTS 112

LIST OF TABLES

Table 2.1: The representation of trail function and its derivatives for each class of problem. 28
Table 2.2: The continuity of trail function and its derivatives for each class of problem. 29
Table 2.3: Comparison between (general and special) purpose finite element program. 34
Table 3.1: The properties of Hermit cubic interpolation function [23]. 49
Table 4.1: The name and description of each supplementary subprograms. . 61
Table 4.2: The name and description of each function. 62
Table 4.3: Input modal for geometrical data. 66
Table 4.4: Input modal for nodal coordinate data. 66
Table 4.5: Input modal for element connectivity data. 66
Table 4.6: Input modal for element materials and geometrical properties data. 67
Table 4.7: Input modal for boundary conditions data. 67
Table 4.8: Input modal for internal hinges data. 67
Table 4.9: Input modal for joint nodal loads data. 68
Table 4.10: Input modal for element nodal loads data. 68
Table 5.1: Example (1) one element solution for vertical deflection at nodal points. 71
Table 5.2: Example (1) one element solution for rotation at nodal points. ... 71
Table 5.3: Example (1) one element solution for vertical deflection at nodal points and mid-length of a beam at distance ($x = 50$ in). 73
Table 5.4: Example (1) two elements solution for vertical deflection at nodal points. 74
Table 5.5: Example (1) two elements solution for rotation at nodal points... 74
Table 5.6: Example (2) two elements solution for vertical deflection at nodal points. 76
Table 5.7: Example (2) two elements solution for rotation at nodal points... 77
Table 5.8: Example (2) two elements solution for shear force at nodal points. 77
Table 5.9: Example (2) two elements solution for bending at nodal points. . 77
Table 5.10: Example (3) two elements solution for vertical deflection at nodal points. 80

Table 5.11: Example (3) two elements solution for rotation at nodal points. 80
Table 5.12: Example (3) two elements solution for shear force at nodal points.. 81
Table 5.13: Example (3) two elements solution for bending at nodal points. 81
Table 5.14: Example (4) three elements solution for vertical deflection at nodal points.. 84
Table 5.15: Example (4) three elements solution for rotation at nodal points. 84
Table 5.16: Example (4) three elements solution for shear force at nodal points.. 84
Table 5.17: Example (4) three elements solution for bending at nodal points. 85
Table 5.18: Example (5) three elements solution for vertical deflection at nodal points.. 88
Table 5.19: Example (5) three elements solution for rotation at nodal points. 88
Table 5.20: Example (5) three elements solution for shear force at nodal points.. 88
Table 5.21: Example (5) three elements solution for bending at nodal points. 89
Table 5.22: Example (6) five elements solution for vertical deflection at nodal points.. 92
Table 5.23: Example (6) five elements solution for rotation at nodal points. 92
Table 5.24: Example (7) three elements solution for vertical deflection at nodal points.. 94
Table 5.25: Example (7) three elements solution for rotation at nodal points. 94
Table 5.26: Example (8) two elements solution for vertical deflection at nodal points.. 96
Table 5.27: Example (8) two elements solution for rotation at nodal points. 96
Table 5.28: Example (9) three elements solution for vertical deflection at nodal points.. 98
Table 5.29: Example (9) three elements solution for rotation at nodal points. 99
Table 5.30: Discussion on the location of maximum values of Example (1). 101

Table 5.31: Discussion on the location of maximum values of Example (1).
.... 101
Table 5.32: Discussion on the location of maximum values of Example (1).
.... 102
Table 5.33: Discussion on the location of maximum values of Example (2).
.... 102
Table 5.34: Discussion on the location of maximum values of Example (3).
.... 103
Table 5.35: Discussion on the location of maximum values of Example (4).
.... 103
Table 5.36: Discussion on the location of maximum values of Example (5).
.... 104
Table 5.37: Discussion on the location of maximum values of Example (6).
.... 104
Table 5.38: Discussion on the location of maximum values of Example (7).
.... 104
Table 5.39: Discussion on the location of maximum values of Example (8).
.... 105
Table 5.40: Discussion on the location of maximum values of Example (9).
.... 105

LIST OF FIGURES

Figure 2.1: Tonti diagram for linear relationships of solid and structure continuum mechanics. 7
Figure 2.2: Example of beam in building structure. 7
Figure 2.3: Continuous beam bridge under loading. 8
Figure 2.4: Cantilever beam in Airplane wings is clamped at one end and free at the other. 8
Figure 2.5: A simply supported beam bridge structure. 8
Figure 2.6: Straight beam with straight longitudinal axis. 8
Figure 2.7: Curved beam with curved longitudinal axis. 9
Figure 2.8: Examples of beam supporting conditions. 9
Figure 2.9: Examples of cross-section shapes for a beam member. 9
Figure 2.10: beam two-modes of deformation (axial deformation and deflection). 10
Figure 2.11: (a) A beam with generic transverse loads (b) cross-sectional area. 10
Figure 2.12: Bending action for a beam subjected to transverse loads. 11
Figure 2.13: Three-dimensional beam geometry in local axis. 11
Figure 2.14: Beam deformation is represented by one-dimension reference line. 12
Figure 2.15: Euler-Bernoulli beam segment, plane section remain plane. 13
Figure 2.16: Determination of axial displacement at a point P. 13
Figure 2.17: Contribution of “small piece” of cross-section to axial force and bending moment. 15
Figure 2.18: PDE solution methods. 18

Figure 2.19: Typical conceptual steps to obtain the finite element solution of a problem....21
Figure 2.20: The increase FEA accuracy and the convergence to the exact solution for displacement as the number of elements of a finite element solution is increased. 30
Figure 2.21: Basic structural phases of a finite element computer program for structural mechanics problem....37
Figure 2.22: The default MATAB desktop layout. 39
Figure 2.23: The Command Window, after entering commands and see responses here. 40
Figure 2.24: MATLAB toolstrip, which allows you to select from a wide variety of tools and commands....40
Figure 3.1: The Tonti diagram for the governing equations of the Bernoulli-Euler model....44
Figure 3.2: Two-nodes beam element in local coordinate systems, nodal transverse displacement wie, rotation θie, force Fyie , and moment Mie.....46
Figure 3.3: Sign convention for positive nodal transverse displacements wie and rotations θie nodal forces Fyie and moments Mie....46
Figure 3.4: Sign convention for positive nodal reaction forces FyiR and reaction moments MiR....46
Figure 3.5: Plots of properties of Hermite cubic translational shape function. 49
Figure 3.6: Beam with Internal hinge....53
Figure 3.7: Beam with internal hinge considered either at right or left end of the element. 53
Figure 3.8: Curved beam with local axes not colinear with global axes [29]. 57

Figure 3.9: Frame structure with different members orientation in global coordinate system [1]. 57
Figure 4.1: Flow chart of main program. 63
Figure 4.2: Flow chart of the steps to obtain global loads vector F. 64
Figure 4.3: Flow chart of the steps to obtain global stiffness matrix K. 65
Figure 5.1: Cantilever beam subjected to UDL. 70
Figure 5.2: Example (1) one finite elements solution for vertical deflection along beam length at nodal points. 72
Figure 5.3: Example (1) one finite elements solution for rotation along beam length at nodal points. 72
Figure 5.4: Example (1) one finite elements solution for vertical deflection along beam length at nodal points and mid-length of a beam ($x = 50$ in). 73
Figure 5.5: Example (1) two finite elements solution for vertical deflection along beam length at nodal points. 75
Figure 5.6: Example (1) two finite elements solution for rotation along beam length at nodal points. 75
Figure 5.7: Fixed-fixed beam subjected to concentrated loads. 76
Figure 5.8: Nodal forces and moments acting on each element. 76
Figure 5.9: Example (2) two finite elements solution for vertical deflection along beam length at nodal points. 78
Figure 5.10: Example (2) two finite elements solution for rotation along beam length at nodal points. 78
Figure 5.11: Example (2) two finite elements solution for shear force along beam length at nodal points. 79
Figure 5.12: Example (2) two finite elements solution for bending moment along beam length at nodal points. 79
Figure 5.13: Overhanging beam subjected to UDL. 80

Figure 5.14: Example (3) two finite elements solution for vertical deflection along beam length at nodal points. 81
Figure 5.15: Example (3) two finite elements solution for rotation along beam length at nodal points. 82
Figure 5.16: Example (3) two finite elements solution for shear force along beam length at nodal points. 82
Figure 5.17: Example (3) two finite elements solution for bending moment along beam length at nodal points. 83
Figure 5.18: Simple supported beam with varying cross-section subjected to concentrated loads. 83
Figure 5.19: Example (4) three finite elements solution for vertical deflection along beam length at nodal points. 85
Figure 5.20: Example (4) three finite elements solution for rotation along beam length at nodal points. 86
Figure 5.21: Example (4) three finite elements solution for shear force along beam length at nodal points. 86
Figure 5.22: Example (4) three finite elements solution for bending moment along beam length at nodal points. 87
Figure 5.23: Cantilever beam subjected to UDL and concentrated loads...... 87
Figure 5.24: Example (5) three finite elements solution for vertical deflection along beam length at nodal points. 89
Figure 5.25: Example (5) three finite elements solution for rotation along beam length at nodal points. 90
Figure 5.26: Example (5) three finite elements solution for shear force along beam length at nodal points. 90
Figure 5.27: Example (5) three finite elements solution for bending moment along beam length at nodal points. 91
Figure 5.28: Continuous beam subjected to UDL and concentrated loads.... 91

Figure 5.29: Finite element modal for Example (6) 92
Figure 5.30: Example (6) five finite elements solution for vertical deflection along beam length at nodal points. 93
Figure 5.31: Example (6) five finite elements solution for rotation along beam length at nodal points. 93
Figure 5.32: Beam with internal hinge and uniformly distributed loading....94
Figure 5.33: Example (7) three finite elements solution for vertical deflection along beam length at nodal points. 95
Figure 5.34: Example (7) three finite elements solution for rotation along beam length at nodal points. 95
Figure 5.35: Cantilever beam with linearly varying distributed loads...........96
Figure 5.36: Example (8) two finite elements solution for vertical deflection along beam length at nodal points. 97
Figure 5.37: Example (8) two finite elements solution for rotation along beam length at nodal points. 97
Figure 5.38: Beam with concentrated load, UDL, and linearly varying distributed loads. 98
Figure 5.39: Example (9) three finite elements solution for vertical deflection along beam length at nodal points. 99
Figure 5.40: Example (9) three finite elements solution for rotation along beam length at nodal points. 100

LIST OF ABBREVIATIONS

Abbreviations	Description
PDE	Partial Differential Equation.
FEM	Finite Element Method.
FEDM	Finite Element Displacement Method.
BCs	Boundary Conditions.
DOFs	Degrees of Freedom
FEA	Finite Element Analysis.
EB	Euler-Bernoulli
GUI	Graphical User Interface.
MATLAB	MATrix LABoratory
UDL	Uniformly Distributed Loading
SAP	Structural Analysis Program
STAAD	Structural Analysis and Design Program

Note that: Other abbreviations are defined in text where necessary.

CHAPTER ONE

INTRODUCTION

1.1 Introduction

The analysis of beam structures become of great interest for designing purposes, because beams are most common type of structural component, particularly in civil engineering, and they are widely used in many structures as supporting members for floors in building, decks in bridges, wings in aircraft, or axels for cars [1].

A beam is structural members for which one of the dimensions, the length is significantly greater than the other two. One of the beam theories is Timoshenko beam theory which includes the effect of transverse shear deformation and Euler-Bernoulli beam theory which only considers bending deformation and neglects the effect of transverse shear. These theories are used to model the kinematic behavior of beams (deformation) and to develop governing equation for structural beams. Beam are generally two-dimensional structure subjected to both axial and transverse loads. The axial loading force parallel to reference line (natural axis), which causes axial internal forces just like truss structures, while the transverse loading gives rise to shear forces and bending moment (flexural moment). A beam can deform in two basic modes namely an axial deformation and deflection. For linearly elastic beams, these modes of deformation can be examined independently from one another [1]. The main interest in this research to examine Euler-Bernoulli flexural mode of deformation of thin beams subjected to transverse loads using finite element displacement method (FEDM).

Euler-Bernoulli beam theory makes reasonable assumption that yields equations that quite accurately predict beam behavior for most practical thin beam's problems [2]. Flexural deformation of Euler-Bernoulli beam is class C^1 problem (continuity C^1). The deformation of a beam must have continuous slope $\theta^e = \frac{dw^e}{dx}$ as well as continuous deflection w^eat any neighboring beam elements. This denotes that the Slope $\theta^e$is the first derivative of transverse deflection w^e, both deflection w^e and slope θ^e selected as nodal variable. Therefore, Hermite interpolation function is used to satisfy this class of continuity problem [3].

Most problems do not yield to analytical solution or it would require a disproportionate amount of effort due to the complex nature of governing differential equations (mathematical model) that arise from complex geometry, multiple materials complexities, complex boundary and initial conditions for which we cannot obtain exact solution (the exact behavior at any point within the system). Therefore, numerical simulation provides alternative means of finding numerical approximated solution of the physical system implemented on digital computer to evaluate the solution of the governing equation of a process and estimate its characteristics [4].

The finite element method is a numerical procedure for solution of differential equations to obtain approximate numerical solution of the problem to simulate the response of the physical system. The method is ideally suited for implementation on a digital computer. The finite element method has become the method of choice for solving many engineering problems quickly and efficiently [4, 5]. The method has become one of the leading methods in computer-oriented mechanics for the development of many scientific and engineering branches over the last decades [6].

The finite element model is created by dividing the structure into finite number of elements, the element is inter connected by nodes only. The section of elements for modeling the structure depends upon the behavior and geometry of the structure being analyzed, also the structure can be modeled by combining different types of elements to approximate aspects of structural behavior. The modeling pattern, which is generally called mesh for the finite element method, The accuracy of the result obtained from the analysis depends upon the selection of the finite element type and the number of elements of the mesh. The equilibrium equations can easily be solved using digital computers without having to solve large number of partial differential equations by hand. The displacement at each node of the finite element model is obtained, Then the stresses and strain can be obtained for each element [5].

Due to the large number of equations, computer use is an essential part of the finite element analysis [7], it helps us to program FEM and solve FEM problem quickly and efficiently, and present the computer-generated results in attractive graphics, and also help readers visualize the numerical finite element results in two- and three-dimensional plots [8].

The developed Finite element Program code is written in MATLAB program, in batch-oriented job mode using (m-files), because it provides a structured organization of the functions, easily execution of changes and reruns. MATLAB allow us to focus in the FEM rather than on the programming details [9].

MATLAB is high-level programming language and also an interactive environment for numerical computation, visualization, and application development. MATLAB is convenient to write and understand FEA program, it is designed for dealing with matrices and vectors with ease, these algebraic equations constitute major parts of the FEA program and the majority of engineering systems, this advantage make it particularly suited for programming the finite element method [3,8], and also provides an extensive library of predefined function to make technical programming tasks easier and more efficient [10].

The significance of this research is emphasized by the importance of studying linear static analysis of thin beams subjected to transverse loads. The element used is two-node beam bending element, the finite element formulation adopts the generalized coordinate approach and the principle of potential energy to derive the element equations.

Recently, MATLAB had been used by many researches in developing finite element program. This research is an attempt to extend these programs to cater for linear static analysis of thin beams. A comparison is made between results obtained by two-node beam element and known published analytical exact solution results and exact solution.

1.2 Problem Statement

One of the reasons for FEM's popularity is that the methods result in computer programs versatile in nature that can solve many practical problems with a small effort [5]. Nowadays, structural engineers will encounter advanced commercial finite element software whose capabilities, and the theories behind its development, are far superior [8].

Using finite element computer programs without proper understanding of the theory behind them is very dangerous [5]. Some of the available commercial finite element packages do not provide an insight into the formulation and solution methods, and do not provide deeper insight into the main interface between the customers of FEM program and the codes itself. Therefore, user must realize that the core of the analysis, the user must understand what happens behind the scenes, often referred to as the black box [7].

Developing finite element computer program will be of great help to understand the theory behind FEM, its programming implementation, and its application using the software. The accuracy of numerical approximated solution results depends very much on a deep understanding of the mathematical governing the theory and on the practical application using computer software at the expense of theory, to obtain reliable efficient analysis results [8].

1.3 Research Objective

The objectives of this research are:

1. To review Euler-Bernoulli beam theory assumptions and its modes of deformation.
2. To develop displacement finite elements suitable for the linear static analysis of thin beam to examine their flexural mode of deformation.
3. To develop and implement a finite element formulation into a computer program using MATLAB program.
4. To validate developed program by analyzing a number models of thin beam and comparing the results obtained with known published results.
5. To bridge the gap between the theory of the finite element method and its mathematical formulation, its programming implementation and its application using software, to obtain reliable efficient analysis results.

1.4 Methodology of Study

The methodology of the study is composed of the following Steps:

1. Carrying out an extensive literature review referring to references source of information such as books, journals, research papers, and Internet web site, for beam theories, FEM, computer programming, MATLAB program.
2. Drive the explicit Euler-Bernoulli linear static finite element formulation of thin beams by adopting the principle of minimum potential energy and the generalized coordinate approach.
3. Developing a linear static finite element program to analyze thin beams using MATLAB program.
4. Validate the developed program by selecting nine thin beam examples as a case study with different loading and supporting conditions.

1.5 Outline of Thesis

This research consists of six chapters, the content of which can be summarized as follows:

Chapter One: covers general introduction.

Chapter Two: contains literature review.

Chapter Three: describes the development of linear finite element formulation of thin beam structures.

Chapter Four: describes the development and implementation of linear finite element computer program.

Chapter Five: contain a discussion of the results obtained by program.

Chapter Six: presents the conclusions and recommendations drawn from this study and the suggestion for the future work.

Appendices of this research:

Appendix I: Finite element computer program

Appendix II: Program output results.

CHAPTER TWO

LITERATURE REVIEW

2.1 General Introduction

Examining the flexural mode of deformation of Euler-Bernoulli beam using a finite element displacement method (FEDM), to develop a finite element computer program for the linear static analysis of thin beams using MATLAB is the main objective of this research. To bridge the gap between the theory of the finite element method, its mathematical formulation, its programming implementation and its application using software, to obtain reliable efficient analysis results.

The advent of the digital computer has revolutionized engineering curricula. In this day, the analysis of all but the simplest problem is carried out with the aid of a computer program that not only speed up the calculations but also allows the display of results in fancy graphics.

The current available commercial finite element software is capable of simulating a large variety of complex problems. These commercial software packages come with advanced pre-processing abilities to facilitating the data input and post-processor abilities to presenting the results. Some of the available commercial finite element packages do not provide an insight into the formulation and solution methods, and do not provide deeper insight into the main interface between the customers of FEM program and the codes itself. Therefore, to achieve proficiency in FEA, the user must realize that the core of the analysis, the user must understand what happens behind the scenes, often referred to as the black box, to obtain reliable efficient analysis results [5, 7,11].

2.2 Linear Structural Analysis

In this research, linear static analysis is adopted, therefore the finite element formulation for the analysis of thin beam is under assumption small deflection (small rotation and small displacement) and elastic material properties (linear analysis). The connecting linear relationships is shown in a Tonti diagram Figure 2.1. The important fundamental consequences of the assumption linearity [12], as follows:

1. The geometry of the structural system: is assumed not to change because the displacements are infinitesimal, and equilibrium is expressed in the

un-deformed geometry, consequently, there is no need to distinguish between a material particle and the point in space that it occupies.
2. The definitions of strain and stress are unique.
3. Strains are linearly related to displacements.
4. Stresses are linearly related to forces.
5. The response of the structural system to any applied force or displacement boundary condition is unique.

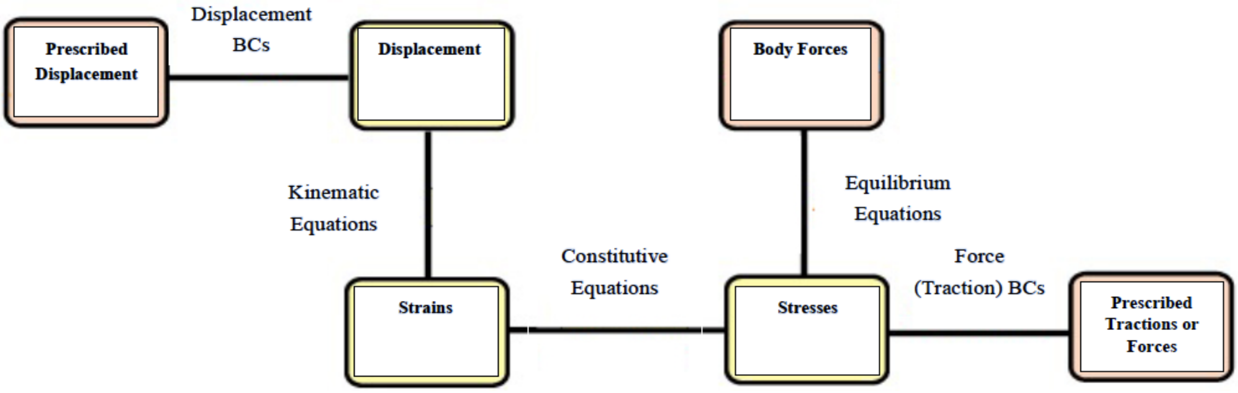

Figure 2.1: Tonti diagram for linear relationships of solid and structure continuum mechanics.

2.3 Beam Introduction

The analysis of these beam structures become of great interest for designing purposes, because beams are most common type of structural component, particularly in Civil Engineering, and they are widely used in many structures as supporting members for floors in building, decks in bridges, wings in aircraft, or axels for cars [7]. Examples of beam structure as shown in Figure 2.2 Figure 2.3, Figure 2.4, and Figure 2.5 below.

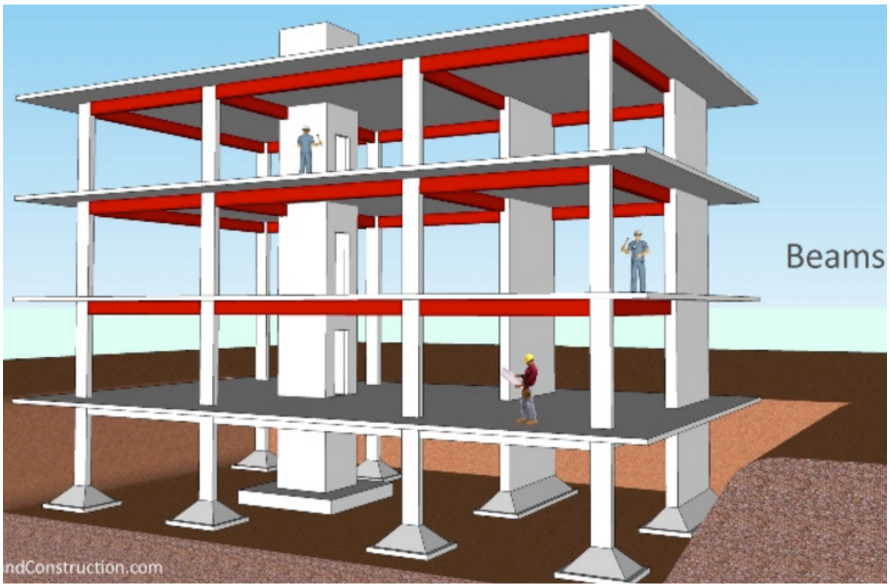

Figure 2.2: Example of beam in building structure.

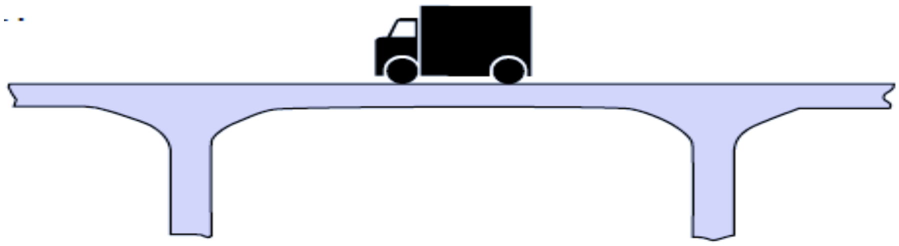

Figure 2.3: Continuous beam bridge under loading.

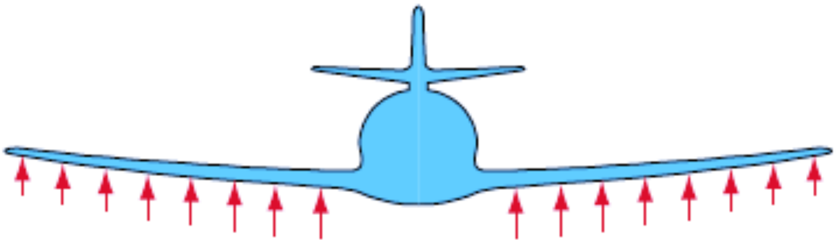

Figure 2.4: Cantilever beam in Airplane wings is clamped at one end and free at the other.

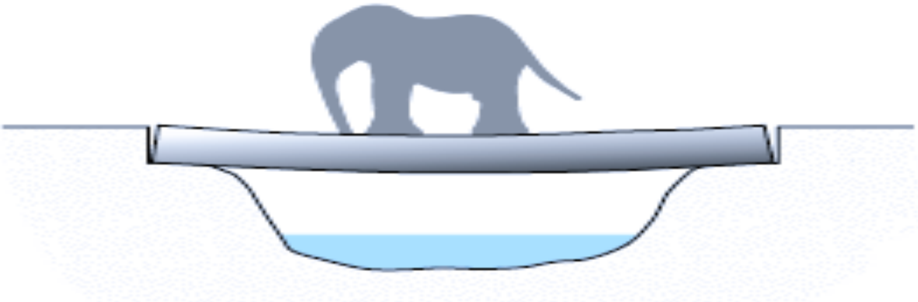

Figure 2.5: A simply supported beam bridge structure.

Beams can be straight or curved depend on their longitudinal axis, as shown in Figure 2.6 and Figure 2.7. Beams can be subjected to a combination of loading actions such as biaxial bending, transverse shears, axial stretching, and possibly torsion. If the internal axial force is compressive, the beam has also to be designed to resist buckling. If the beam is subject primarily to bending and axial forces, designed to be Two-dimensional beams or beam-column (planer frame). Spatial beam support transverse loads that can act on arbitrary direction along the cross section [12].

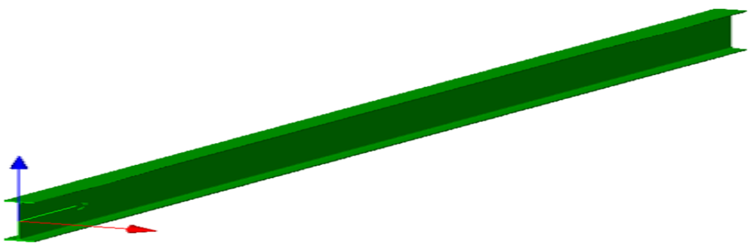

Figure 2.6: Straight beam with straight longitudinal axis.

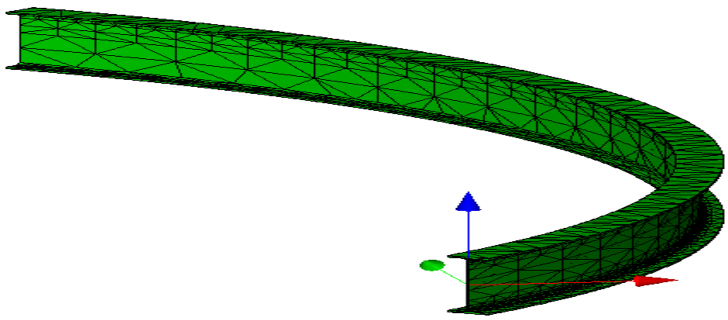

Figure 2.7: Curved beam with curved longitudinal axis.

Beams supporting condition can be cantilever, simple supported, overhanging, continuous, and fixed ended as shown in Figure 2.8 below:

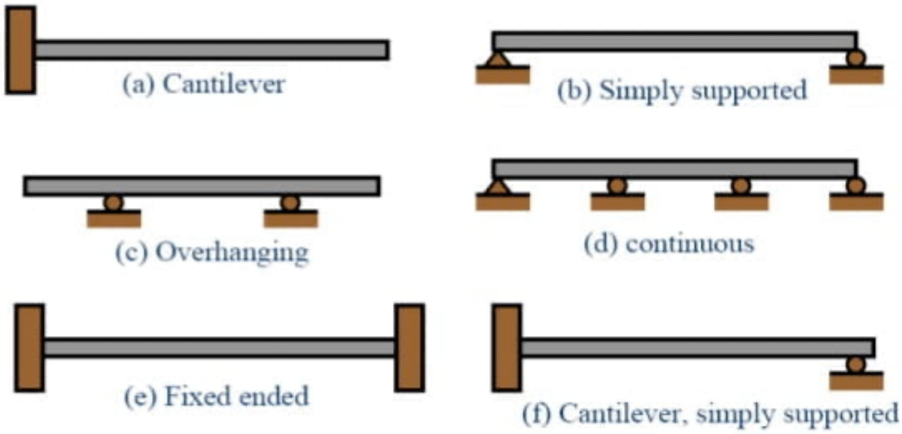

Figure 2.8: Examples of beam supporting conditions.

Beam cross-section can be constant such as in Prismatic beam, also it can vary along beam length [13]. Beam cross-section shape can be of any arbitrary shapes such as (rectangular, square, circular, I-section, T-section, L-section, and U-section) as is shown in Figure 2.9 below. Beam can be made from various types materials such (steel, concrete, timber, aluminum, and composite). Beam can have laminated behave.

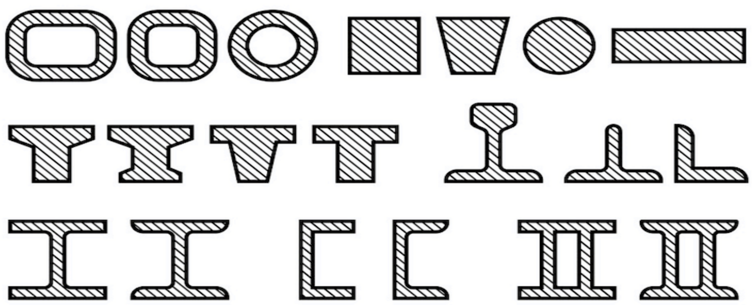

Figure 2.9: Examples of cross-section shapes for a beam member.

Two-dimensional beams are line members (members for which one of the dimensions, the length is significantly greater than the other two). Beam are generally subjected to both axial and transverse loads. The axial loading force parallel to reference line, which causes axial internal forces just like truss structures, while the transverse loading gives rise to shear forces and bending moment. Two-dimensional beams can be described as members on a plane, such as planer frames or beam-column. These beams can be deformed in two basic modes namely an axial deformation and deflection, as shown in Figure 2.10. For linearly elastic beam, these two modes of deformation can be examined independently from one another. This is why many textbooks only examine the flexural mode of deformation for beam structures. The axial mode of deformation gives the same equations as those obtained for one-dimensional elasticity [1].

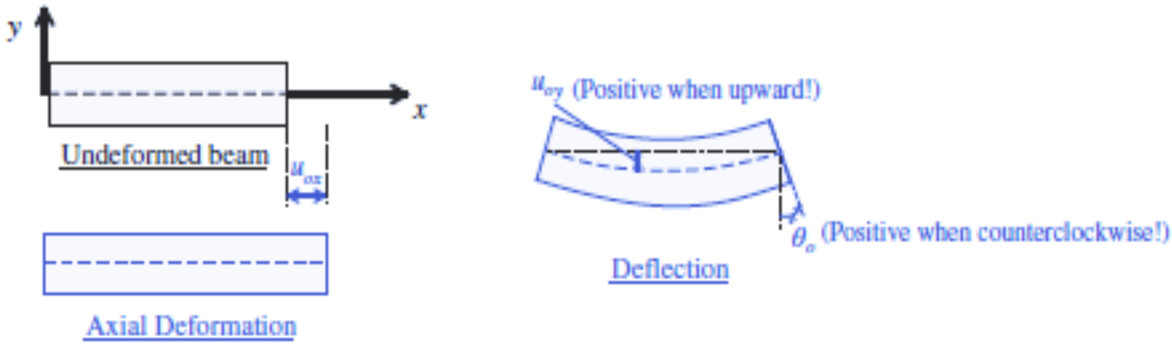

Figure 2.10: beam two-modes of deformation (axial deformation and deflection).

In Euler-Bernoulli beam, the transverse loads are acting perpendicular to the member's principal axis. These, transverse loading can be due to concentrated force F, concentrated moment M, distributed moments $m(x)$, and distributed force $q(x)$ [14], as shown in Figure 2.11.

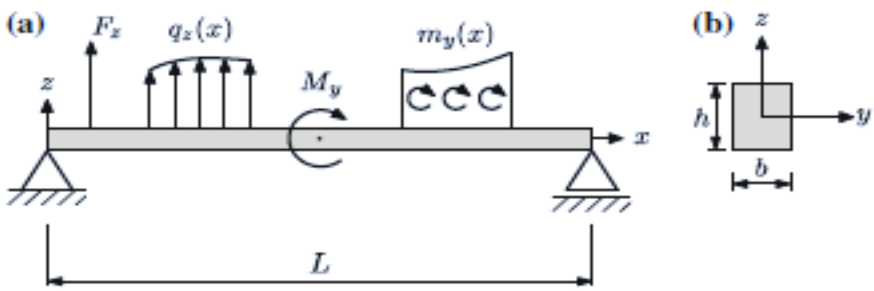

Figure 2.11: (a) A beam with generic transverse loads (b) cross-sectional area.

Beams resist transverse loads and carry it to the supports through bending action is opposed to twisting or axial effects. This bending moment is the primary mechanism that transports loads to the supports and it produces compressive longitudinal stresses in one side of the beam and tensile stresses in the other, these two regions are separated by neutral surface of zero stress. The combination of tensile and compressive stresses produces an internal bending moment [12], as it is shown in Figure 2.12.

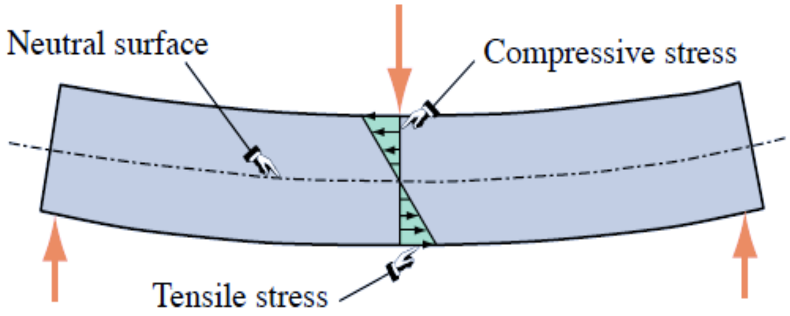

Figure 2.12: Bending action for a beam subjected to transverse loads.

2.4 Classical Beam Theories

Beams are actually three-dimensional bodies, as shown in Figure 2.13. To model such structural member, it is necessarily involving some form of approximation to the underlying physics, by representing the deformed beam by the deformed reference line called elastic curve [1], as shown in Figure 2.14. The two theories used to model the kinematic behavior of beams (deformation), and to develop governing equation for structural beams [1, 14, 15], as follow:

1. Euler-Bernoulli beam theory: is used for thin beams also called (shallow beams or shear-rigid beams).
2. Timoshenko beam theory: is used for deep beams also called (thick beams or shear-flexible beams).

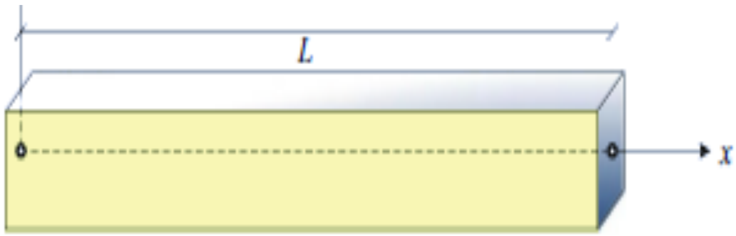

Figure 2.13: Three-dimensional beam geometry in local axis.

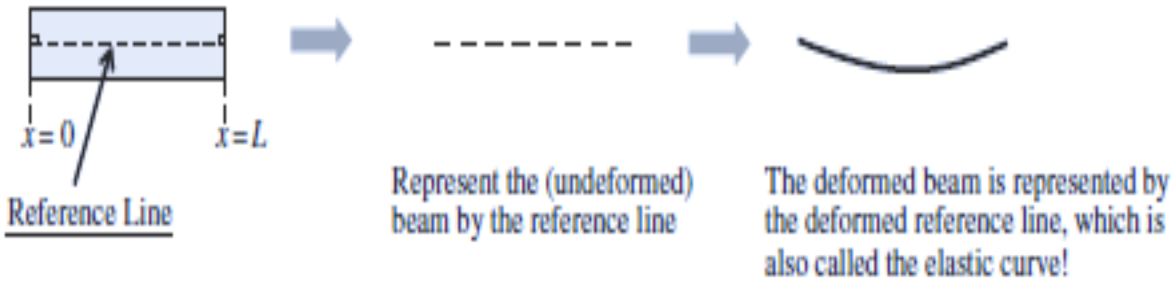

Figure 2.14: Beam deformation is represented by one-dimension reference line.

To determine the suitability of the theory for idealizing the mechanical response of between thin beams and deep beams under transverse loads by using the following relation [15]:

1. For thin beams:

$$\frac{L}{h} > 10 \qquad (2.1)\ a$$

2. For deep beams:

$$\frac{L}{h} \leq 10 \qquad (2.1)\ b$$

Where:
L: Length of beam.
h: Thickness of beam.

2.5 Euler-Bernoulli Beam Theory

2.5.1 Introduction to Euler-Bernoulli Beam Theory

Euler-Bernoulli theory it is also called classical beam theory or engineering beam theory. This theory considers only bending deformation, it neglects the effect of transverse shear deformations [15].

2.5.2 Kinematics of Deformation for Euler-Bernoulli Beam

The kinematic assumptions regarding the displacement field as follows [1]:

1. Plane sections remain plane and perpendicular to the deformed reference line of the beam, as shown in Figure 2.15. The first consequence of this assumption is that the shear strain $\gamma_{xy} = 0$, the way to allow the existence of nonzero shear stresses while the shear strain in zero is to assume that Euler-Bernoulli infinitely rigid against shear strains. This allows a nonzero shear stress, even if the corresponding shear strain is zero. The second consequence of this assumption the rotation of the cross-section θ is equal to the corresponding slope of the deformed reference curve.

$$\theta = \frac{dw}{dx} \qquad (2.2)$$

Where:

θ : Rotation of the cross section.

$\frac{dw}{dx}$: Slope of the deformed reference curve.

2. The normal stress parallel to the cross-sectional plane is equal to zero $\sigma_{yy} = 0$.
3. When determining the axial displacement at a point in a cross-section, we neglect the change of the depth of the cross-section, and we stipulate that the *y*-displacement at every point of the cross section is equal to the *y*-displacement of reference line at the same cross-section.

$$u_y(x, y) = u_{oy}(x) \qquad (2.3)$$

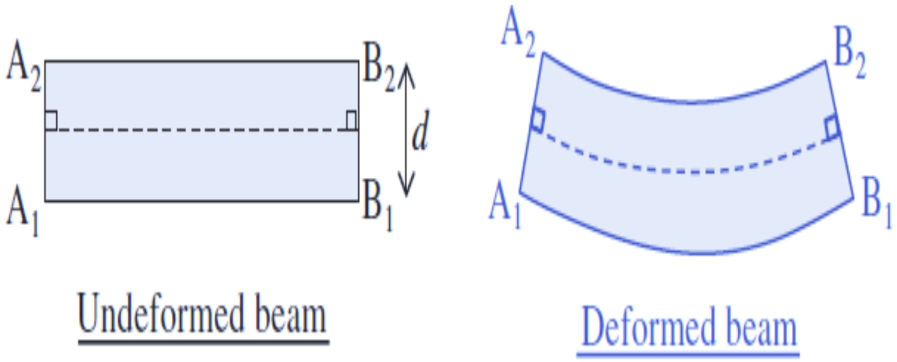

Figure 2.15: Euler-Bernoulli beam segment, plane section remain plane.

2.5.2.1 Axial Displacement of Euler-Bernoulli Beam

To establish the expression the gives the axial displacement, use the geometric considerations obtain from Figure 2.16 below:

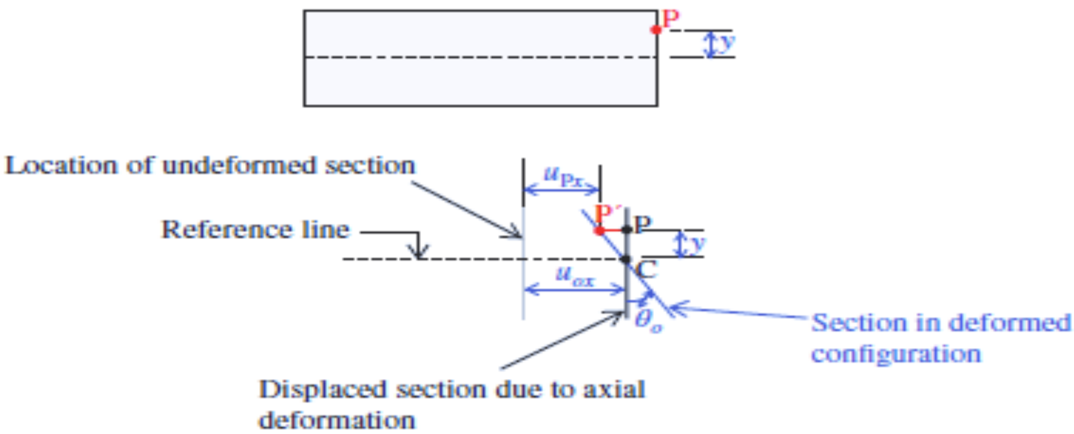

Figure 2.16: Determination of axial displacement at a point P.

The axial displacement for Euler-Bernoulli Beam is given by the following:

$$u_x(x,y) = u_{ox}(x) - y\theta_o(x) \tag{2.4}$$

Where:

u_{ox}: The axial deformation.

$$u_{ox}(x) = u \tag{2.5}$$

y : Distance from the neutral axis.

θ_o : Rotation of cross-section.

2.5.2.2 Axial Strain of Euler-Bernoulli Beam

$$\varepsilon_{xx}(x,y) = \frac{du_x}{dx} = \frac{du_{ox}}{dx} - y\frac{d\theta_o}{dx} = \varepsilon_o - y\varphi = \frac{du}{dx} - y\frac{d^2w}{dx^2} \tag{2.6}$$

Where:

ε_o: The axial strain of the reference line, result from axial deformation due to axial loads.

φ: The curvature of the beam, locally quantifies how curved the geometry of the deformed beam becomes, result from flexural deflection due to transverse loads.

$$\varphi = \frac{d\theta_o}{dx} = \frac{d^2w}{dx^2} \tag{2.7}$$

2.5.2.3 Axial Stress of Euler-Bernoulli Beam

$$\sigma_{xx}(x,y) = E\varepsilon_{xx} = E(\varepsilon_o - y\varphi) = E\frac{du}{dx} - Ey\frac{d^2w}{dx^2} \tag{2.8}$$

Where:

E: The modules of elasticity.

2.5.2.4 Axial Force of Euler-Bernoulli Beam

The axial force N and bending moment M can be obtained as stress resultants, by imagining the cross-section of the beam A, consist of a combination of many points (*x, y, z*), each point having a corresponding infinitesimal sectional area, $dS = (dy).(dz)$, we can "sum" the axial force contributions of each small area dS, as shown in Figure 2.17, to obtain axial force N, at a location *x*, as follows:

$$N = \int\int_A \sigma_{xx}\ dS = EA\varepsilon_o = EA\frac{du}{dx} \tag{2.9}$$

Where:

A : Beam cross-section area

$$A = \int\int_A dS \tag{2.10}$$

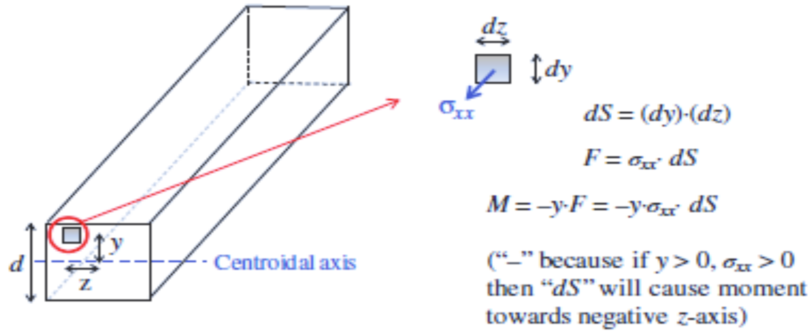

Figure 2.17: Contribution of "small piece" of cross-section to axial force and bending moment.

2.5.2.5 Bending Moment of Euler-Bernoulli Beam

The bending moment about the *y*-axis,

$$M_y = M = \int\int_A -y\,\sigma_{xx}\ dS = EI\varphi = EI\frac{d^2w}{dx^2} \qquad (2.11)$$

Where:

I: The moment of inertia of the cross section of the beam with respect to the *y*-axis.

$$I = \int\int_A y^2\ dS \qquad (2.12)$$

The bending moment about the *z*-axis,

$$M_z = 0 \quad (assumed) \qquad (2.13)$$

2.5.2.6 Generalized Constative Equation of Euler-Bernoulli Beam

The bending moment about the *y*-axis,

$$\{\hat{\sigma}\} = [\hat{D}]\{\hat{\varepsilon}\} \qquad (2.14)\ a$$

$$\begin{Bmatrix} N \\ M \end{Bmatrix} = \begin{bmatrix} EA & 0 \\ 0 & EI \end{bmatrix}\begin{Bmatrix} \varepsilon_o \\ \varphi \end{Bmatrix} = \begin{bmatrix} EA & 0 \\ 0 & EI \end{bmatrix}\begin{Bmatrix} \dfrac{du}{dx} \\ \dfrac{d^2w}{dx^2} \end{Bmatrix} \qquad (2.14)\ b$$

Where:

$\{\hat{\sigma}\}$: The generalized stress.

$[\hat{D}]$: The modules matrix.

$\{\hat{\varepsilon}\}$: The generalized strain.

$$\{\hat{\varepsilon}\} = \begin{Bmatrix} \varepsilon_o \\ \varphi \end{Bmatrix} = \begin{Bmatrix} \dfrac{du}{dx} \\ \dfrac{d^2w}{dx^2} \end{Bmatrix} \quad (2.15)$$

Where:

$\{\varepsilon_o\}$: The axial strain of the reference line, result from axial deformation due to axial loads.

$\{\varphi\}$: The curvature of the beam, locally quantifies how curved the geometry of the deformed beam becomes, result from flexural deflection due to transverse loads.

2.6 The Governing Equilibrium PDE and BCs of Euler-Bernoulli Beam

2.6.1 The Governing Equilibrium PDE

The Strong form of the governing partial differential equations (PDE) (mathematical modal) considering axial deformation and flexural deformation of two-dimensional Euler-Bernoulli beam [14] as follows:

$$\frac{d}{dx}\left(EA\frac{du}{dx}\right) + w_b(x) = 0 \qquad (Axial\ PDE) \qquad (2.16)\ a$$

$$\frac{d^2}{dx^2}\left(EI\frac{d^2w}{dx^2}\right) - w_q(x) = 0 \qquad (Flexural\ PDE) \qquad (2.16)\ b$$

Where:

$w_b(x)$: The intensity of body force.

A : Beam cross-section area.

u : The axial displacement.

EA : The beam axial rigidity.

$w_q(x)$: The distributed load along the length.

EI : The beam flexural rigidity.

w : The transverse deflection due to vertical displacement in *y*-direction.

E : The modulus of elasticity.

I : The moment of inertia of the cross section of the beam with respect to the *y*-axis.

2.6.2 The Essential and Natural BCs

The essential boundary conditions (primary unknown variables) the generalized deflection [1,14], as follows:

$$u \qquad (Essential\ Axial\ BCs) \qquad \textit{(2.17) a}$$

$$w \qquad (\ Essential\ Flexural\ BCs) \qquad \textit{(2.17) b}$$

$$\theta = \frac{dw}{dx} \qquad (Essential\ Flexural\ BCs) \qquad \textit{(2.17) c}$$

Where:

u : The axial deformation.

w: The deflection due to vertical displacement in y-direction.

θ : Rotation of the cross section.

$\frac{dw}{dx}$: slope of the deformed reference curve, is the 1st derivative of deflection w.

The natural boundary conditions (secondary unknown variables) the generalized forces [1,14], as follows

$$N(x) = EA\frac{du}{dx} \qquad (Natural\ Axial\ BCs) \qquad \textit{(2.18) a}$$

$$M(x) = EI\frac{d^2w}{dx^2} \qquad (Natural\ Flexural\ BCs) \qquad \textit{(2.18) b}$$

$$V(x) = \frac{dM}{dx} = EI\frac{d^3w}{dx^3} \qquad (Natural\ Flexural\ BCs) \qquad \textit{(2.18) c}$$

Where:

$N(x)$: The axial force.

$M(x)$: The bending moment.

$V(x)$: Transverse shear force.

These governing equilibrium PDE (mathematical models) are used to simulate the response of the physical system and predict the behavior of a system in all physical scales, they can be solved using analytical methods and numerical methods [16], as shown in Figure (2.18) below.

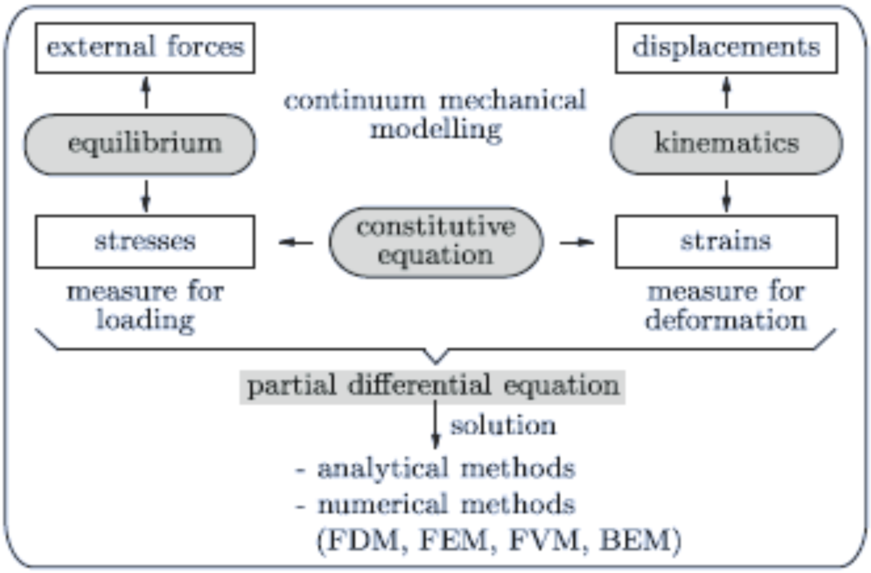

Figure 2.18: PDE solution methods.

2.7 Analytical Methods

Analytical solution is called "closed form solution or exact solution" are more intellectually satisfying for the solution of partial differential equations. Unfortunately, they tend to be restricted to regular geometrics and simple boundary conditions. Because in the exact methods of analysis, the deviation of the governing equations and their solution is often difficult due to the complex nature of governing differential equations that arise from complex geometry, multiple materials complexities, complex boundary and initial conditions for which difficult to obtain exact solution (the exact behavior of a system at any point within the system). Most problems faced by the engineer either do not yield to analytical treatment or doing so would require a disproportionate amount of effort. Therefore, the practical way out is use numerical methods that provide alternative means of finding solution, through numerical approximated solution of the physical system [12,16].

2.8 Computational Methods

Computational mechanics is a body of knowledge connected with the development of mathematical models and use of numerical simulation of physical systems. It is used to solve specific problems by model-based simulation through numerical methods implemented on digital computers [4,12].

2.8.1 Mathematical Models

To analyze an engineering system, a mathematical model is developed to describe the behavior of the system [3]. Developing a mathematical model that describe the physical system, it is the most important step in engineering practice, because it must be done by a human not by a computer [12].

The mathematical model of a process (governing equations), is the analytical investigation of any physical process by a set of mathematical equations that expresses the essential features of a physical system in terms of variables that describe the system, and allowing the determination of state variables and in turn. These mathematical models typically correspond to differential equations [1]. These governing equations are often difficult to solve using exact analytical solution methods [4].

2.8.2 Numerical Simulation

Numerical simulation is the use of numerical methods and computer to evaluate the solution of the governing equation of a process and estimate its characteristics [4]. The use computer helps us to solve and programming a FEM problem that involve large number of equations quickly and efficiently with the results presented in attractive graphics [4,8].

2.8.3 Numerical Methods

Numerical methods typically transform the governing differential equations to set of algebraic equations of a discrete model of the continuum that are to be solved using computers. These procedures in essence reduce the continuous-system mathematical model to discrete idealization [4]. Numerical solutions are approximation exact solution only at discrete points, called nodes. The first step of any numerical procedure is discretization. This discretization process divides the medium of interest into a number of small subregions called elements and connected through nodes [17]. The classification of numerical methods based on the discretization techniques by which the continuum mathematical model is discretized in space (i.e., converted to a discrete model of finite number of degrees of freedom) [12] are:

1. Finite Element Method (FEM).
2. Finite Difference Method (FDM).
3. Finite Volume Methods (FVM).
4. Boundary Element Method (BEM).
5. Spectral Method.
6. Mesh-Free Method.

In this research, the numerical methods adopted for the analysis of Euler-Bernoulli flexural mode of deformation is finite element displacement method (FEDM).

2.9 Finite Element Methods (FEM)

The finite element method is a numerical procedure for solution of differential equations to obtain approximate numerical solution of the problem to simulate the response of the physical system. The method is ideally suited for implementation on a digital computer. The FEM has become the method of choice for solving many engineering problems quickly and efficiently [4,5], and, one of the leading methods in computer-oriented mechanics for the development of many scientific and engineering branches over the last decades [6].

2.9.1 General Description of FEM

In the FEM, the actual continuum or body of matter as a solid, liquid, or gas, that has infinite degrees of freedom is replaced by a computing model (finite element model) that has only finite degrees of freedom and represented as an assemblage of subdivisions called elements [16,18]. The finite element model is created by dividing the structure into finite number of elements, the element is inter connected by nodes only [5]. Since the actual variation of the field variable inside the continuum is not known, we assume that variation of the field variables inside a finite element can be approximated by a simple function. These approximating functions (also called interpolation models) are defined in terms of the values of the field variables at the nodes. When field equations (like equilibrium equations) for the whole continuum are written, the new unknowns will be the nodal values of the field variable. By solving the finite element equations (simultaneous algebraic equations), which are generally in the form of matrix equations, the nodal values of the field variable will be known. Once these are known, the approximating functions define the field variables throughout the assemblage of element [16]. The FEM is characterized by three features [4], as follows:

1. The domain is represented by a collection of elements and the collection of finite elements is called mesh.
2. Over each finite element, the physical process is approximated by function of the desired type (polynomial or otherwise), and algebraic equation relating physical quantities at selective points, called nodes, of the elements developed.
3. The elements equations are assembled using continuity and/or balance of physical quantities.

The section of elements for modeling the structure depends upon the behavior and geometry of the structure being analyzed, the structure can also be modeled by combining different types of elements to approximate aspects of structural behavior. The modeling pattern, which is generally called mesh for the finite element method, The accuracy of the result obtained from the analysis depends upon the selection of the finite element type and the number of elements of the mesh. The equilibrium equations can easily be solved using digital computers without having to solve large number of partial differential equations by hand. The displacement at each node of the finite element model is obtained, Then the stresses and strain can be obtained for each element [5].

The finite element formulation of the problem results in a system of simultaneous algebraic equations rather than the solution of differential equations [1], as shown in Figure 2.19 below:

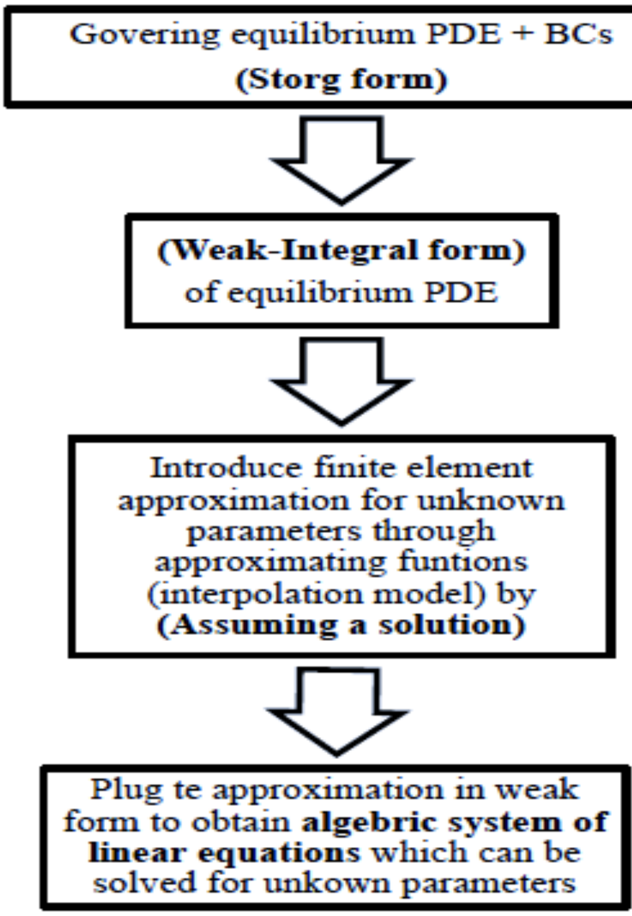

Figure 2.19: Typical conceptual steps to obtain the finite element solution of a problem.

2.9.2 Advantages of FEM

The advantages of FEM which has the ability to perform the following [2]:

1. Model irregularity shaped bodies quite easily.
2. Handle general load conditions without difficulty.
3. Model bodies composed of several different materials because the element equations are evaluated individually.
4. Handle unlimited numbers and kinds of boundary conditions.
5. Vary the size of the elements to make it possible to use small element where necessary.
6. Alter the finite element model relatively easily and cheaply.
7. Include dynamic effects to conduct dynamic analysis.
8. Handle nonlinear behavior such as geometric and materials nonlinearities.

2.9.3 The Application of the FEM

The FEM is extensively used in the field of structural mechanics, also applied to solve other types of engineering problems, such as heat conduction, fluid dynamics, seepage flow, and electric and magnetic fields. The general nature of its theory, make it applicable to wide variety of boundary and initial in engineering. A boundary value problem is the one in which the solution is sought in the domain of a body subjected to the satisfaction of prescribed boundary (edge) condition on the dependent variables or their derivatives. The engineering applications of the FEM in the three major categories of boundary value problems, namely (Equilibrium or steady state or time-independent problems, Eigenvalue problems, and Propagation or Transient problems), it is used in these areas of study such as (civil engineering structures, Aircraft structures, Heat conduction, Geomechanics, Nuclear engineering, biomedical engineering, Mechanical design, Electrical machines and electromagnetics, hydraulic and water resources engineering, hydrodynamics) [16].

In structural engineering used in the simulation of civil and aerospace structures. In automotive industry the method replaces expensive experimental crash tests. In manufacturing processes involve metal forming. In micromechanics it helps investigating the finer-scale characteristics of a material that need to be taken into account such as damage of materials to projectile penetration, the formation of microscopic cracks of inhomogeneous, composite materials subjected to specific types of loading [1].

2.9.4 General FEM Steps for Structural Mechanics Problem

The general steps of the FEM for structural problems [2], as follows:

1. Discretize and select the element types.
2. Select a displacement function.
3. Define the strain/displacement and stress/strain relationships.
4. Derive element stiffness matrix and equations.
5. Assemble the element equations to obtain the global or total equations and introduce boundary conditions.
6. Solve for the unknown degrees of freedom (or generalized displacement or primary variables). Using an elimination method (such as Gauss's method) or an iterative method (such as the Gauss-Seidel method).
7. Solve for the element strains and stresses (unknown secondary variables)
8. Interpret the results.

2.9.5 The Weak-Integral Form of Equilibrium PDE

The equations of weak form are usually in integral form and require a weaker continuity on the field variables. Due the weaker requirement on the filed variables and the integral form of governing expression, a formulation based on a weak form is expected to lead to set of equations for the discretized system that give much more accurate results, especially for problems of complex geometry. Hence, the weak form is preferred by many for obtaining an approximate solution. Thus, the FEM based on a weak form type of formulation yields a set of well-behaved algebraic system equations. The advantages of weak formulations are relaxed continuity requirement can be used in the weak formulation not like strong form which is usually require that the assumed approximated solution to be higher derivative than the weak form [16]. A weak from of the system equations is usually derived using one of the following widely used methods [19]:

1. Energy principles: can be categorized as a special from of the variational principle which is particularly suited for problems of the mechanics of solids and structures.
2. Weighted residual methods: is a more general mathematical tool application, in principle, for solving all kind of partial differential equations.

2.9.6 The Approaches for the Derivation of Finite Element Equations

2.9.6.1 Direct Approach

Direct equilibrium methods are based on the physical reasoning to establish the element equations in terms of pertinent variables, the approach uses the basic principles of engineering science. The method is an extension of matrix displacement approach and its applicable only for simple problems such as deriving element stiffness matrices for one-dimensional elements involving springs, uniaxial bars, trusses and beams. The two general direct approaches traditionally associated with the FEM as applied to structural mechanics problem are [16]:

a. Force or flexibility method: uses internal forces as the unknowns of the problem. To obtain the governing equations, first the equilibrium equations are used. Then necessary additional equations are found by introducing compatibility equations. The result is a set of algebraic equations for determining the redundant or unknown forces.
b. Displacement or stiffness methods: it is more desirable method because its formulation is simpler for most structural analysis problems. assumes the displacement of the nodes as the unknowns of the problem. For instance, compatibility conditions requiring that elements connected at a common node, along a common edge, or on a common surface before loading remain connected at that node, edge, or surface after deformation takes place are initially satisfied. Then the governing equations are expressed in terms of nodal displacements using the equations of equilibrium and an applicable law relating forces to displacements.

2.9.6.2 Weight Residual Approach

In these methods, the finite element equations are derived directly from the governing differential equations of the problem without reliance on the variational statement of the problem, it is used when the corresponding functional is difficult to find or does not exist. The method replaces the differential equations by approximate algebraic equation. The method offers the most general procedure for deriving finite element equation and can be applied to most all practical problems of science and engineering, it can be used to obtain approximate solution to linear and nonlinear differential equations. The methods of weighted functions based on the choice of the weighting function are (Collocation method, Least squares method, Moment method, Subdomain method, and Galerkin’s Method). The most popular method is Galerkin's method. Galerkin method used by the analysts of solid mechanics, fluid mechanics, heat flow, elastically engineering. In mechanics of solid it turns out to be virtual work method [16].

2.9.6.3 Variational Approach

The variational methods consisting of among the subset's energy methods and the principle of virtual work. Variational methods are the widely used numerical approximations approach in elasticity and structural mechanics. The method is based on the application of variational calculus, which deals with the extremization of functionals in the form of integrals and it is a method for studying the maxima and minima of functionals. In fact, in the energy methods of structural mechanics, the functional usually the potential energy (physical concept), and it is used as basis for developing the governing differential equations [16].

The major limitation of the method is that it requires the physical or engineering problem to be stated in variational form, which may not be possible in all cases. The reasons for using variational methods as follows, if the variational integral from is known, no need to derive the corresponding differential equation. Also, most of the important variational statement for problems in engineering and physics have been the functional form is well known for over 200 years. Another important feature of variational methods is that often dual principles exist that allow one to establish both an upper bound estimate and lower bound estimate for an approximate solution, these features can be very helpful in establishing accurate error estimates for adopting solutions. The variational method is much easier to use for two and three-dimensional elements [20].

In these methods, the FEA is interpreted as an approximate means for solving variational problems. Since most physical and engineering problems can be formulated in variational form, the FEM can be readily applied for finding their approximate solution. The FEM as an approximate method of solving variational problems [20].

Variational method has made finite element analysis a versatile method when using this method stiffness matrices and consistent load vector can be assembled easily [21]. The energy methods can be used to drive elements equations are [2]:

a. The principles of minimum potential energy: applies to linear-elastic materials, which states (For conservative systems, of all the kinematically admissible displacement fields, those corresponding to equilibrium extremize the total potential energy. If the extremum condition is a minimum, the equilibrium state is stable).

b. The principles of virtual work: applies to linear and nonlinear analysis.
c. Castigliano's theorem.

2.9.7 The Two Selection Attributes for the FEM Choices

The two key selection attributes based on unknown variables and solution choice [18], as follows:

1. For primary unknown variable(s) choice:
   a. Displacement unknown
   b. Stress unknown
   c. Mixed energy Principle (hybrid element): both displacement and stress are unknowns, for hybrid element, we may suppose a stress (or displacement) field within the element and another displacement (or stress) field on the element boundary, the application of mixed energy principle necessitate a simultaneous assumption of stress field and displacement filed within the element.
2. For the solution choice:
   a. Stiffness.
   b. Flexibility.
   c. Combined.

2.9.8 Finite Element Displacement Methods (FEDM)

In this research, for examining flexural mode of deformation for Euler-Bernoulli beam, finite element displacement methods are used. The approaches used for derivation of finite element equation is the variational of minimum potential energy. The primary unknowns' variables are nodal deflection (transverse displacement and rotation), the secondary unknown variables are nodal force (shear force and bending moment). The equation used for the solution are equilibrium equation in a form of simultaneous algebraic equations.

2.10 Interpolation Displacement Function

The finite element model is created by dividing the structure into finite number of elements, the element is inter connected by nodes only [5]. Since the actual variation of the field variable inside the continuum is not known, we assume that variation of the field variables inside a finite element can be approximated by approximating functions (also called interpolation models) are defined in terms of the values of the field variables at the nodes [16].

In the FEA aim to find the field variables at nodal points by rigorous analysis, assuming at any point inside the element basic variables is a function of values nodal points of the element. this functions which relates the field variables at any points within the element to the field variables of nodal points is called shape function or interpolations function or approximating function. When the FEDM is used to find the solution, we need to assume polynomial displacement function [16].

2.10.1 Guidelines for Selecting Polynomial Form of Displacement Function

The consideration has to be taken into account when choosing the order of the polynomial displacement function [16],as follows:

1. The interpolation polynomial should satisfy, as far as possible, the convergence requirement [16].
2. The pattern of variation of the filed variables resulting from the polynomial model should be independent of the local coordinate system [16]. This property is knowns as geometric Isotropy and spatial isotropy. The guidelines to construct polynomial series with the desired property of isotropy, that a complete polynomial has geometric isotopy, and a not completed polynomial but contain appropriate terms to preserve symmetry have geometric isotropy, by dropping only terms that occur in. symmetric pairs [21].
3. The number of generalized coordinates (number of terms) should be equal to the number of nodal degrees of freedom of the elements [16].

2.11 Convergence Requirement for Polynomial Displacement

The difference between accuracy and convergence term. Accuracy refers to the difference between the exact solution and the finite element solution. Convergence refers to the accuracy as the number of elements in the mesh is increased [4].

The FEM is a numerical technique, we obtain a sequence of approximate solution as the element size is reduced successively, this sequence will converge to the exact solution if the interpolation polynomial satisfies the convergence requirements [16,18], these requirement as follows:

2.11.1 Completeness Requirement

The completeness requirement is directly applicable to the interpolation functions. The completeness conditions consist of two main requirements (rigid body motion, and constant state of strain) [22]. First for rigid body requirement is satisfied by selecting a constant term in the polynomial displacement function, so that the displacement function could translate with a zero value of strain at all points within the element, i.e., that element does not generate strain (zero-strain field). Second for constant state of strain is satisfied by including a constant strain term in the polynomial displacement function, so that the displacement function should be able to represent constant strain fields, so that the derivatives of the displacement function can be constant on the element to reflect the constant strain of the element [16].

The completeness principle states that when the size of the element shrinks to zero (mesh refinement), the assumed trial function must be able to represent the following [8]:

Table 2.1: The representation of trail function and its derivatives for each class of problem.

Class of Problem	Representation of Trial Function and its Derivatives
For a class C^0 problem (continuity C^0)	A constant value of the exact function as well constant vales of its first-order derivatives.
For a class C^1 problem (continuity C^1)	A constant value of the exact function as well constant vales of its first-order and second-order derivatives.
For a class C^n problem (continuity C^n)	A constant value of the exact function as well constant vales of its derivatives up to the *n*th order.

2.11.2 Continuity or Compatibility Requirement

The compatibility requirement means that displacement model must continuous within the element and compatibles between the adjacent elements (element boundaries) to ensures that no gaps or overlaps can appear between the elements. Discontinuities of displacement between adjacent Element will produce infinite strain (infinite strain energy) on the contact surface of adjacent elements. Therefore, the displacement function should assure that the strain on the contact surface of adjacent element is finite [21,23]. Therefore, the assumed trial function must be able to represent the following [8]:

Table 2.2: The continuity of trail function and its derivatives for each class of problem.

Class of Problem	Continuity of Trial Function and its Derivatives
For a class C^0 problem (continuity C^0)	The trial solution must be continuous across the boundary of the element but not necessarily its derivatives.
For a class C^1 problem (continuity C^1)	Both The trial solution and its first-order derivatives must be continuous across the boundary of the element but not necessarily its second-order derivatives.
For a class C^n problem (continuity C^n)	The trial solution and its *(n-1)* th order derivatives must be continuous across the boundary of the element but not necessarily its *n*th order derivatives.

Using Gauss numerical integration can solve this problem, using Isoperimetric elements, the calculation is conducted on Gaussian integral points, and the strain energy on the contact surface in not included [18].

In flexure beam problems, bending element such as the Bernoulli beam model, the strain energy terms include second derivatives of displacement due to moment curvature relation $M = EI\frac{d^2w}{dx^2}$. Hence, to satisfy this compatibility requirement, not only displacement continuity but slope continuity should be satisfied. this means that the first derivative of displacement (slope) across interelement boundaries also must be continuous. Hence, in such flexure problems displacement and their first derivatives (slope) are selected as nodal field variables. This problem known as C^1 continuity problem. Therefore, Hermite shape function used for C^n continuity problems [21].

2.12 Conforming and Non-Conforming Elements

2.12.1 Conforming Elements

If the interpolations' polynomial function satisfies all convergence requirement both (complete and compatible), the approximate solution converges to the correct solution and do not display strange or pathologic behaviors even for a relatively course discretization of the structure when refining the mesh to increase number of elements, the resulting element called conforming element, which will lead to monotonic convergence [16,23]. The increase FEA accuracy and the Convergence to the exact solution for displacement as the number of elements of a finite element solution is increased [2], as shown in Figure 2.20.

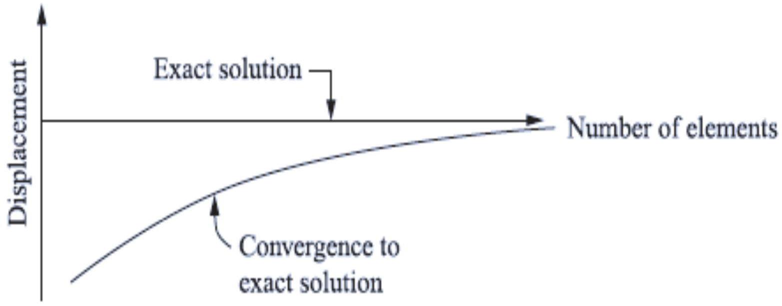

Figure 2.20: The increase FEA accuracy and the convergence to the exact solution for displacement as the number of elements of a finite element solution is increased.

Monotonic convergence toward an exact solution is the process in which successive approximation solution (finite element solution) approach the exact solution consistently without changing sign or direction [2].

2.12.2 Non-Conforming Elements

Non-conforming elements are elements that violate the compatibility requirement, and when it assembled it might be incapable of reproducing the constant strain condition even though the individual elements are complete. Therefore, required to pass the patch test, in which a constant strain filed is applied to an assembly of arbitrarily-oriented element [23].

Several interpolation polynomials that do not meet all the requirement have been used. In some cases, acceptable convergence has been obtained [16]. The good convergence towards the exact mathematical solution due incompatibilities disappears with increasing mesh refinement [24]. Mesh refinement methods solve non-monotonic convergence problems, when the size of element infinitely shrinks, the stains of elements will approach the state of constant strain in the neighborhood of each point [16].

The main disadvantage of using non-conforming elements is that we no longer know in advance that correct solution is reached [21]. In structural problems, interpolation polynomial satisfying all the convergence requirement always led to the convergence of the displacement solution from below, while nonconforming element may converge either from below or from above [16].

Non-conforming elements are very often deliberately adopted, because they tend to soften the behavior of the model and counteract the natural over stiffness of the displacement functions [24].

2.13 Finite Element Solution

2.13.1 The Error Concept in Finite Element Analysis

The use of computational simulation is integrally connected to the concept of approximation, in which that the term approximation implies that the computational representation of a physical process is governed by modified versions of governing mathematical equations, this modification leads to the fact that computational simulation provides approximate values for quantities. The accuracy of the approximated values depend on how closely the computational modification matches the original mathematical expressions. Therefore, due to this approximation the error obtained from the following [1]:

$$(Error) = (Approximated\ Value) - (Exact\ Value) \qquad (2.19)$$

The three sources of errors in the finite element solution are (error due to the approximation of the domain, error due to the approximation of the solution, and error due to numerical computation such as numerical integration and round-off errors in a computer) [4].

2.13.2 The Lowest Limit Solution of Minimum Potential Energy

When using principle of minimum potential energy, the finite element displacement solution will be less than the exact solution, which it called the lowest limit solution. The element taken form the body as a part of the continuum should have infinite degrees of freedom, which becomes finite degrees of freedom after applying displacement function, which means the elements stiffness has been increased. So, the calculation approximate solution of displacement is less than the exact solution [18]. Therefore, the FEM gives lower bound values. Hence it is desirable that as the finite element analysis mesh is refined, the solution approaches the exact values. This is one of the requirements of shape function to ensure convergence criteria [21].

2.13.3 Methods to Improve Convergence of Finite Element Results

The construction of an efficient finite element model involves (representing the geometry of the problem accurately, developing a finite element mesh to reduce the bandwidth, choosing a proper interpolation model to obtain the desired accuracy in the solution). Unfortunately, there is no priori method of creating a reasonably efficient finite element model that can ensure a specified degree of accuracy to assessing the convergence of finite element model. Some adoptive finite element methods have been developed to employ the results from previous meshes to estimate the magnitude and distribution of solution errors and to adoptively improve the finite element model [16].

The four basic approaches to adoptively improve a finite element model are (Subdivided selected elements called h-method, increase the order of the polynomial of selected element called p-refinement, move node points in fixed element topology called r-refinement, and define a new mesh having a better distribution of elements.). Various combinations of these approached are also possible. Determining which of these approaches is the best for a particular class of problems is a complex problem that must consider the cost of the entire solution process [16].

The mesh refinement conditions are (all previous (coarse) meshes must be contained in the refined meshes, the elements must be smaller in such a way that every point of the solution region can be always be within an element, The form of the interpolation polynomial must remain unchanged during the process of the mesh refinement) [16].

2.13.4 Checking FEA Results

Choosing a suitable model and checking the accuracy and validity of the answers is really novel problem [8]. Hence, a comparison should be made between computed results from finite element program with results from other available techniques for instance approximate mechanics of materials formulas, experimental data, and numerical analysis of simpler but similar problems may be used for comparison, particularly if you have no real idea of the magnitude of the answers [2].

In addition, the analyst must ensure that the results agree with engineering intuition and behavior for the problem, and verifying whether the solution satisfies the specified boundary and symmetry conditions. If necessary, the problem needs to be solved by changing the boundary conditions, loads or materials to find whether the resulting FEA solution behave as per engineering intuition and expectations [16].

2.14 The Use of Computer for Numerical Computation

2.14.1 Role of Computer in the Development of FEM

The FEA result in a very large number of algebraic equations which made the method extremely difficult and impractical to use. However, with the advent of computer, the solution of thousands of equations in a matter of minutes became possible [2]. Therefore, the computer is the basic need for the application of the method to analyze the effects of various parameters of the system on its response (computer-generated results) to gain a better understanding of the system [4].

The fast improvement in computer hardware technology and slashing of cost of computers have boosted the FEM and resulted in the writing of computational programs to handle various complicated structural and nonstructural problems [2,21]. There are many available commercial finite element packages fulfill these needs. The commercial packages provide user-friendly data input platforms and elegant and easy-to-follow display formats. However, the packages do not provide an insight into the formulation and solution methods [7].

2.14.2 Commercial Finite Element Packages

There are number of computer program packages available for the solution of a variety of structural and solid mechanics problems. Some of the programs have been developed in such a general manner that the same program can be used for the solution of problems belonging to different branches of engineering with little or no modification [21].

The popularity of FEM has motivated the development of multiple finite element programs commercial software and research-oriented software (Open-source finite element codes) [1].

Commercial software is typically marketed by private companies and accompanied by appropriate graphical user inter-faces (GUI) to facilitate use. Some examples of commercial programs [1], as follows:

1. NASTRAN: Developed by MacNeal-Schwendler Corporation and stemmed from an effort by NASA to create a computer code for the analysis of aerospace structures.
2. ANSYS: Developed by E. Wilson at UC Berkeley is one of the first programs to allow nonlinear analysis.
3. SAP 2000, ETABS and Perform3D: used by civil structural engineering community and emphasize seismic design.
4. ADINA: Developed by K.-J. Bathe a student of Wilson and currently a faculty member at MIT.
5. ABAQUS: is one of the popular commercial programs, created by David Hibbitt and marketed by Simulia. this program has a very wide range of capabilities with high-performance, parallel computing hardware.

6. LS-DYNA: created by John Hallquist and marketed by Liver-more Software and Technology Corporation (LSTC), this program has a very wide range of capabilities with high-performance, parallel computing hardware.
7. STAAD-PRO
8. SOLIDWORKS
9. GT-STEUDEL
10. NISA
11. COMSOL

Open-source finite element codes is primarily aimed for researchers and educators, it is used for understanding the method and the associated numerical analysis algorithms. User is granted a complete access to the entire analysis program, which is not possible in some of commercial software, The two interesting and popular open-source programs for civil structural engineering community are (FEAP, and Opensees) [1].

2.14.3 General and Special Purpose Finite Element Program

There are two general computer methods of approach to the solution of a problem by finite element methods [2], as follows:

1. Large commercial programs are general-purpose finite element programs are designed to solve many types of problems.
2. Small commercial program are special-purpose finite element programs are designed to solve specific problems.

Table 2.3: Comparison between (general and special) purpose finite element program.

General-Purpose Programs	
Advantages	**Disadvantages**
1. The input is well organized and is developed with user ease in mind. Users do not need special knowledge of computer software or hardware. Preprocessors are readily available to help create the finite element model. 2. The programs are large systems that often can solve many types of	1. The initial cost of developing general-purpose programs is high. 2. General-purpose programs are less efficient than special-purpose programs because the computer must make many checks for each problem, some of which would not be necessary if a special-purpose program were used.

problems of large or small size with the same input format. 3. Many of the programs can be expanded by adding new modules for new kinds of problems or new technology. Thus, they may be kept current with a minimum of effort. 4. With the increased storage capacity and computational efficiency of PCs, many general-purpose programs can now be run on PCs. 5. Many of the commercially available programs have become very attractive in price and can solve a wide range of problems	3. Many of the programs are proprietary. Hence the user has little access to the logic of the program. If a revision must be made, it often has to be done by the developers.
Special -Purpose Finite Element Programs	
Advantages	**Disadvantages**
1. The programs are usually relatively short, with low development costs. 2. Small computers are able to run the programs. 3. Additions can be made to the program quickly and at a low cost. 4. The programs are efficient in solving the problems they were designed to solve.	1. The inability to solve different classes of problems. This one must have many programs as there are different classes of problems to be solved.

2.14.4 The Standard Capabilities of General-Purpose Finite Element Program

The complete capabilities of the programs and their cost are best obtained through program reference manuals and websites, some of these capabilities [2],as follows:

1. Element types available, such as beam, plane stress, and three-dimensional solid.

2. Type of analysis available, such as static and dynamic.
3. Material behavior, such as linear-elastic and nonlinear.
4. Load types, such as concentrated, distributed, thermal, and displacement (settlement).
5. Data generation, such as automatic generation of nodes, elements and restraints (most programs have preprocessors to generate the mesh for the model).
6. Plotting, such as original and deformed geometry and stress and temperature contours (most programs have postprocessors to aid in interpreting results in graphical form).
7. Displacement behavior, such as small and large displacement and buckling.
8. Selective output, such as at selected nodes, elements, and maximum or minimum values.
9. All programs include at least the bar, beam, plane stress, plate -bending, and three-dimensional solid elements and most now include heat-transfer analysis capabilities.

2.14.5 The Basic Structure of a Finite Element Program

Understanding basic program structure of the FEA is an important part for better comprehension of the finite element method [3]. The objective of FEA is to determine the unknowns (degrees of freedom) at the nodes and the resulting support reaction in any structure. There are three basic phases involved in FEA program structure for structural mechanic problem as follows below and in Figure 2.21 [25]:

1. Pre-processing phase: Build analysis model for computer, this phase is made by user to define the problem then processing the given data and printing out required result data, such discretizing the domain geometry into specific finite elements defined by number of nodes, number of degrees of freedom, coordinate of each node, element type, nodal connectivity, specifying the materials properties, the magnitude and point of application for loads and boundary conditions, number of Gauss points and weight if numerical integration is used. Type of analysis (linear or nonlinear).
2. Processing phase: conducting analysis, this phase is made by computer to developing a set of linear or nonlinear algebraic equation simultaneously to obtain nodal results (the unknown solution of the primary variables)

3. Post-processing phase: Analyzing output results, this phase is made by user by viewing the results in graphical form, and obtaining results on other desirable quantities or variables of interest from nodal variables such as stress, strain and moment (the unknown solution of secondary variables).

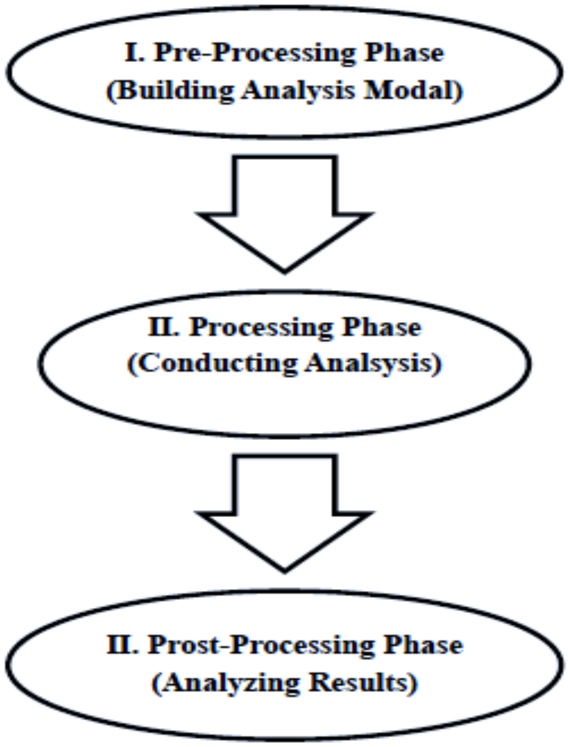

Figure 2.21: Basic structural phases of a finite element computer program for structural mechanics problem.

2.15 Computer Programming

A computer does what you ask, not what you wish. If a computer delivers an unsatisfactory result, it is most likely because you did not properly translate your wish. This is made by proper developing algorithm process, which is defining the finite sequence of well-defined intrastation that to be followed in computer calculations (logic of implementation) [26].

Programming is to generate a list of instructions suitable for execution by a computer, through splitting the problem into a list of elementary operations, then using a programming language to put these operations into coding [26].

2.15.1 Programming Languages

There are hundreds of programming languages. Unfortunately, none of them is the ‘‘ silver bullet’’ language that is good for every situation. Some of the programming languages are better for speed, some for conciseness, some for controlling hardware, some for reducing mistakes, some for graphics and sound output, some for numerical calculations. There are several ways to categorize programming languages based on the eye of the programmer to low-level programming language such as (assembler, C, C++, Forth, and LISP), and high-level programming language such as (Fortran, Tcl, Java, JavaScript, PHP, Per, Python, and MATLAB), and also based on the compliance with other programming languages [26].

MATLAB is a high-level programming language for numerical computation, visualization, application development and programming. It provides an interactive environment for iterative exploration, design and problem solving. It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations. it provides built-in graphics for visualizing data and tools for creating custom plots. MATLAB's programming interface gives development tools for improving code quality maintainability and maximizing performance. It provides tools for building applications with custom graphical user interfaces (GUI). It provides function for integrating MATLAB based algorithms with external applications and languages such a (C, Java, .NET, and Microsoft Excel) [26].

2.16 MATALB Program

2.16.1 Introduction to MATLAB Program

MATLAB (short for MATrix LABoratory) is a special-purpose computer program optimized to perform engineering and scientific calculations. Started as a program designed to perform matrix mathematics, but over the years it has grown into a flexible computing system capable of solving essentially any technical problem. The software product is developed by the Math Works corporation [10].

The MATLAB program implements the MATLAB language and provides an extensive library of predefined extremely wide variety of functions to make technical programming tasks easier and more efficient and functions makes it much easier to solve technical problems in MATLAB than in other languages. These functions often solve very complex problems in a single step, saving large amount of time. Doing the same thing in another computer language usually involves writing complex programs yourself or buying a third-party software package [10].

The MATLAB language is a combination of a procedural programming language, an integrated development environment (IDE) that includes an editor and debugger, and an extremely rich set of functions that perform many types of technical calculations [10].

The MATLAB language is a procedural programming language, meaning that the engineer writes programming procedures, which are effectively mathematical recipes for solving a problem. This makes MATLAB very similar to other procedural languages such as Fortran or C. however, the extremely rich list of predefined functions and platting tools makes it superior to these other languages for many engineering analysis applications [10].

2.16.2 MATLAB Desktop Layout

MATLAB desktop layout, as shown in Figure 2.22 shows the default MATLAB desktop layout. Figure 2.23 shows the command window after entering commands. Figure (2.24) shows the toolstrip, which allows you to select from a wide variety of MATLAB tools and commands.

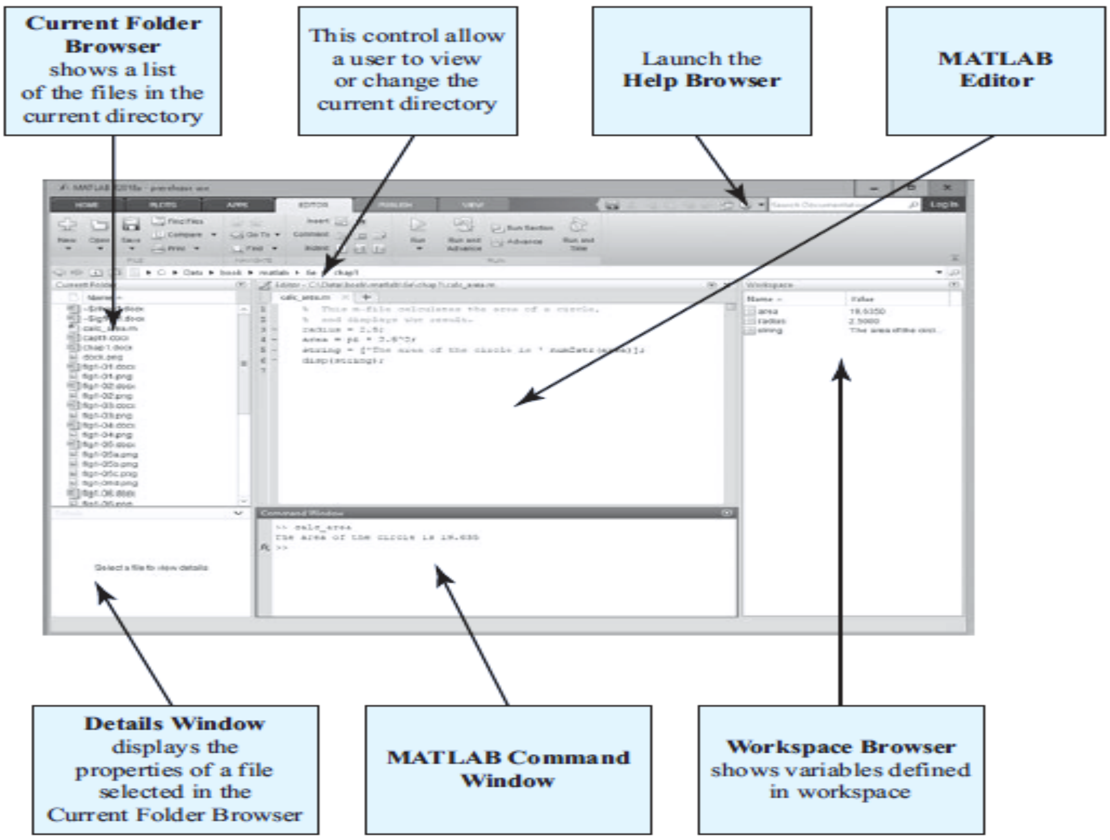

Figure 2.22: The default MATAB desktop layout.

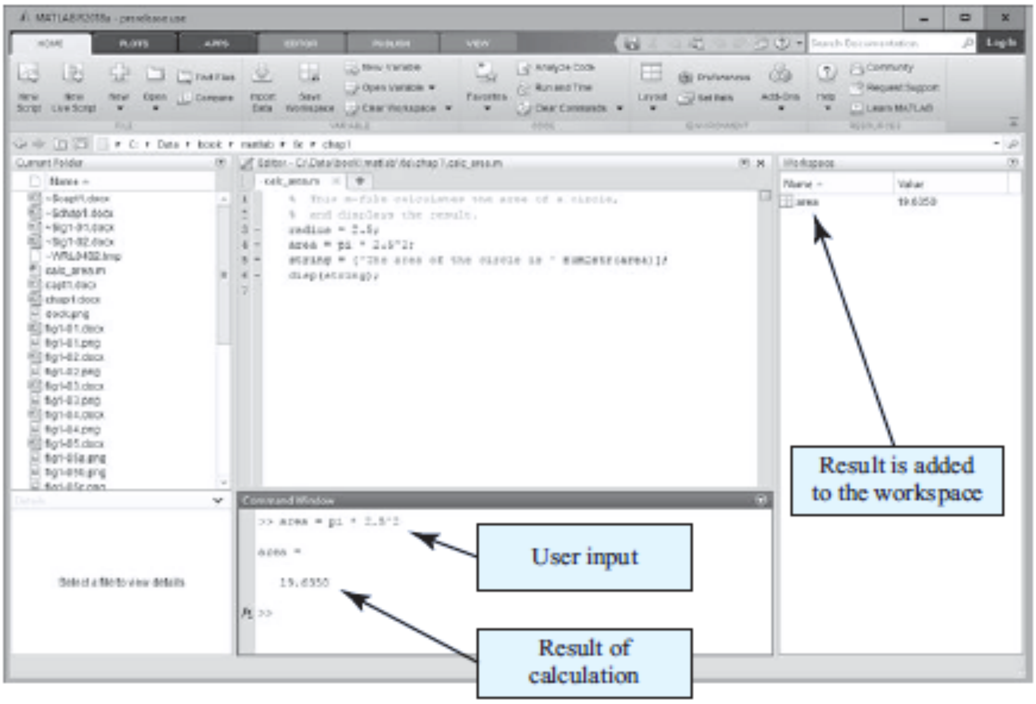

Figure 2.23: The Command Window, after entering commands and see responses here.

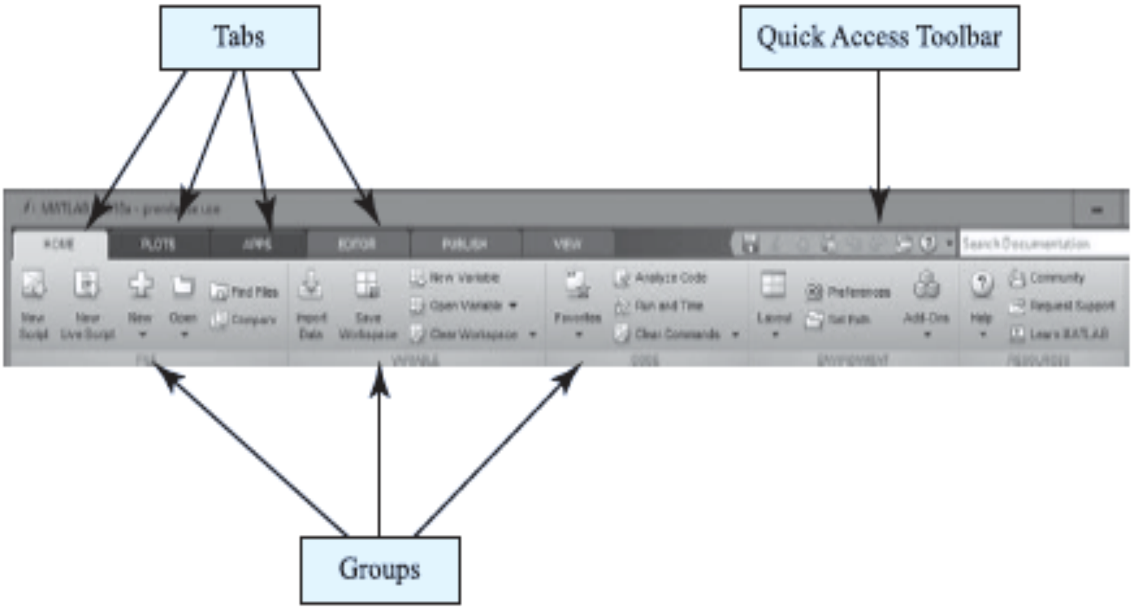

Figure 2.24: MATLAB toolstrip, which allows you to select from a wide variety of tools and commands.

2.17 Programming the FEM using MATLAB

MATLAB is high-level programming language, is convenient to write and understand FEA program, because is specially designed for dealing with matrices and vectors with ease, these algebraic operations constitute major parts of the FEA program and the majority of engineering systems. Therefore, this advantage make it particularly suited for programming the finite element method [3,6,8,27].

MATLAB has built-in graphics features to help readers visualize the numerical results in two and three-dimensional plots. Graphical presentation of numerical data is important to interpret the finite element results. The power of MATLAB is represented by the length and simplicity of code, one page of MATLAB code may be equivalent to many pages of other computer language source codes [3,27]. Performing FEA using MATLAB can be carried out and run under either the following [9,27]:

1. Interactive mode session: in this mode the functions are evaluated one by one in the MATLAB command window.
2. Batch oriented job mode: in this mode a sequence of functions is written in a file named (m-file), and evaluated by writing the file name in the command window. This mode is more flexible way of performing FEA because the (m-file) can be written in an ordinary editor, this mode recommended because it gives a structured organization of the functions and easily execution of changes and reruns.

CHAPTER THREE
LINEAR FINITE ELEMENT FORMULATION OF THIN BEAMS FOR FLEXURAL DEFORMATION

3.1 Introduction

Beams are generally subjected to both, axial loads causes (axial deformation) and transverse loads causes (transverse deformation). The latter is resulted from both shear forces (shear deformation) and bending moment (flexural deformation). For linearly elastic beams, these modes of deformation can be examined independently from one another. The main interest in this research is to analyze the bending deformation of thin straight beams [1]. Therefore, the finite element formulation developed based on Euler-Bernoulli beam assumptions is used to predict bending behavior of thin straight beams subjected to transverse loading [15].

Euler-Bernoulli Beam assumptions neglects the effect of transverse shear deformations from the shear force and only considers bending deformation [15]. The theory makes reasonable assumption that yields equations that quite accurately predict beam behavior for most practical beam's problems [2].

The transverse deformation of thin beams is only in direction perpendicular to its axis, which produces bending effects as opposed to twisting or axial effects [19], This bending deformation is measured as a transverse (deflection) w^e and a rotation θ^e[2].

Euler-Bernoulli beam is classified as class C^1 problem (continuity C^1). The slope at any point along the beams equals the first derivative of the deflection curve $\theta^e = \frac{dw^e}{dx}$, such elements is named slope conforming elements [21]. This mean that the deformation of a beam must have continuous slope $\theta^e = \frac{dw^e}{dx}$ as well as continuous deflection w^eat any neighboring beam elements, and both deflection w^e and slope $\theta^e = \frac{dw^e}{dx}$ selected as nodal variable. In other words, the continuity of the derivatives of the main variables is required. Hermite interpolation function is used to satisfy this type of problem (class C^1 problem (continuity C^1)), it satisfies the continuity of both the transverse deflection $w^e$and the first derivatives of transverse deflection (slope) $\theta^e = \frac{dw^e}{dx}$ between any neighboring beam elements [3,21].

Thin beam is modeled geometrically as a straight bar of an arbitrary cross-section, with two degrees of freedom at each node, which are transverse (deflection) $w^e$and a rotation θ^e in a local coordinate system [19].

In the FEA of thin beam bending element, bending moment is not obtained using bending stress formula from basic solid mechanics $\sigma_x^e = -\frac{My}{I}$, is found using the stress resultant curvature relation $M = EI\frac{d^2w}{dx^2}$, by taking second derivations on the transverse displacement function, this relation relates the bending moment to transverse deflection same as in elementary beam theory. Shear force is derived by taking third derivations on the transverse displacement function $= EI\frac{d^3w}{dx^3}$. For the uniformly loaded beam, the resulting shear force is a constant throughout the single-element model [2,21,28].

Internal hinge results in a discontinuity in the slope or rotation of the deflection curve at the hinge. Therefore, the element stiffness matrix must be modified to include hinge desired effect, it should be accounted only once on either the right end or left end of the element, but not on both [2,8].

The stiffness matrix for a beam element was developed for loading applied only at its nodes. Therefore, distributed loading and concentrated loads applied other than at the natural intersection of two elements must be converted to nodal loads, instead of creating a node at a point of concentrated load application. The work-equivalence method is used to replace loads by a set of discrete loads (Statically equivalent nodal loads) these equivalent loads tend to have the same effect on the beam as the actual original distributed load, and they are always of opposite sign from the fixed-end reactions [2,8].

The analysis of thin straight beam bending with local axis are colinear with global axes there is no need for coordinate transformation. For each element set up the local stiffness matrix and directly assemble it into the global stiffness matrix [8].

The displacement computed using statically equivalent nodal loads are exact in a finite elements sense at nodes. However, the internal reactions computed in individual elements are not. This require same adjustment to obtain the correct nodal reactions which is can be verified by simple static equilibrium equations [2,8].

3.2 The Governing PDE and BCs of (EB) Beam for Flexural Deformation

The Tonti diagram for the governing equations of the Bernoulli-Euler model under transverse loading [12], is shown in Figure 3.1 as below:

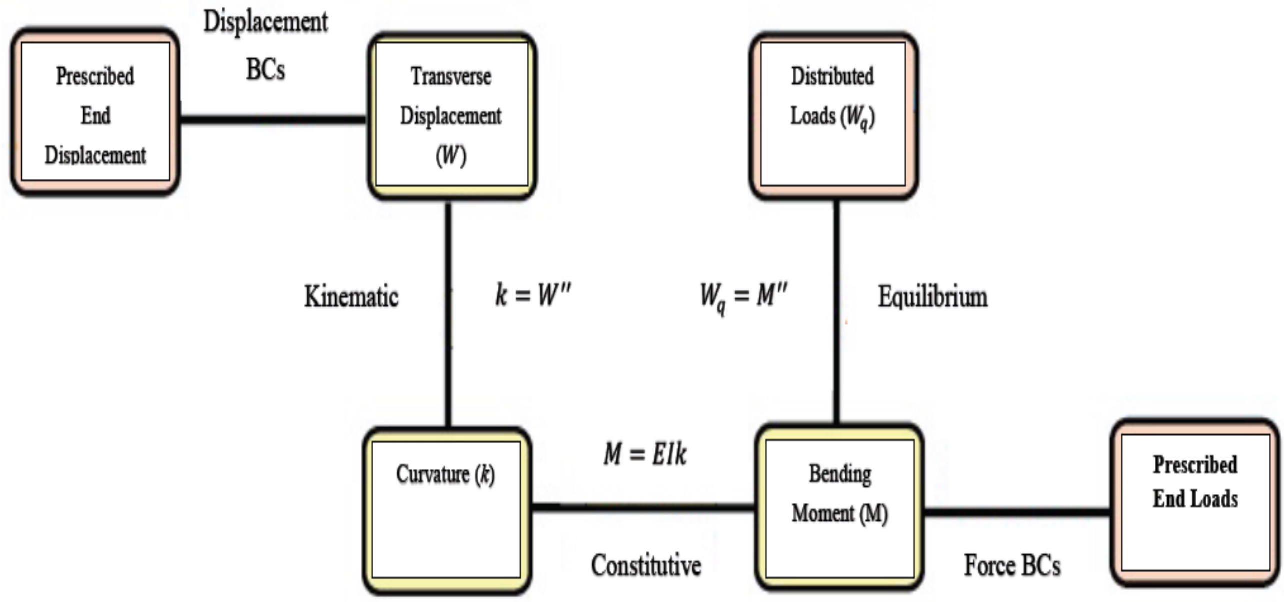

Figure 3.1: The Tonti diagram for the governing equations of the Bernoulli-Euler model.

3.2.1 The Governing Equilibrium PDE OF EB Beam for Flexural Deformation

The strong form of the governing equilibrium PDE of Euler-Bernoulli beam is a 4th order differential equation that governs the elementary linear-elastic beam behavior and it is used for the static analysis of beam bending [14], it is given by:

$$\frac{d^2}{dx^2}\left(EI\frac{d^2w}{dx^2}\right) - w_q(x) = 0 \qquad (Flexural\ PDE) \qquad (3.1)$$

Where:

$w_q(x)$: The distributed load along the length.

EI : The beam flexural rigidity.

w : The transverse deflection due to vertical displacement in y-direction.

E : The modulus of elasticity.

I : The moment of inertia of the cross section of the beam with respect to the y-axis.

3.2.2 The Essential and Natural BCs of EB Beam for Flexural Deformation

1. The essential boundary conditions (primary unknown variables) the generalized deflection of Euler-Bernoulli Beam [1,14], as follows:

$$w \qquad (Essential\ Flexural\ BCs) \qquad (3.2)\ a$$

$$\theta = \frac{dw}{dx} \qquad (Essential\ Flexural\ BCs) \qquad (3.2)\ b$$

Where:

w: The deflection due to vertical displacement in y-direction.

θ : Rotation of the cross section.

$\frac{dw}{dx}$: slope of the deformed reference curve, is the 1st derivative of deflection w.

2. The natural boundary conditions (secondary unknown variables) the generalized forces of Euler-Bernoulli Beam, as follows:

$$M(x) = EI\frac{d^2w}{dx^2} \qquad (Natural\ Flexural\ BCs) \qquad (3.3)\ a$$

$$V(x) = \frac{dM}{dx} = EI\frac{d^3w}{dx^3} \qquad (Natural\ Flexural\ BCs) \qquad (3.3)\ b$$

Where:

$M(x)$: The bending moment.

$V(x)$: Transverse shear force.

3.3 Formulation of EB Beam for Flexural Deformation

Adopting the generalized coordinate approach and the principle of potential energy to derive the element equations for Euler-Bernoulli beam element subjected transverse loads, using two-nodes thin beam bending element with two degrees of freedom per each node.

3.3.1 Geometric Definition of Beam Element

Using two-nodes beam element with two degrees of freedom per each node, are transverse displacement $w^e$and rotation θ^e, as shown in Figure 3.2 below:

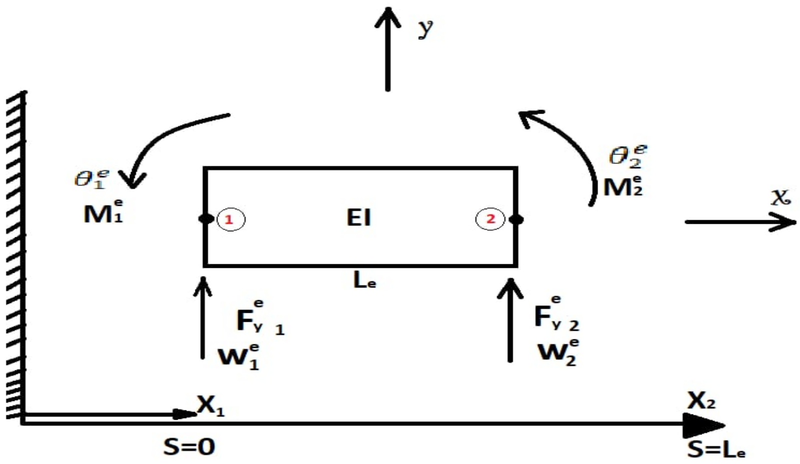

Figure 3.2: Two-nodes beam element in local coordinate systems, nodal transverse displacement w_i^e, rotation θ_i^e, force F_{yi}^e , and moment M_i^e.

3.3.2 Sign Convention of Beam Element

The sign convention used for nodal beam element. Positive direction for nodal (transverse displacements w_i^e, rotations θ_i^e, forces F_{yi}^e, and moments M_i^e) are shown in Figure 3.3. Positive direction for nodal (rection forces and reaction moments M_i^R) are shown in Figure 3.4 below [2]:

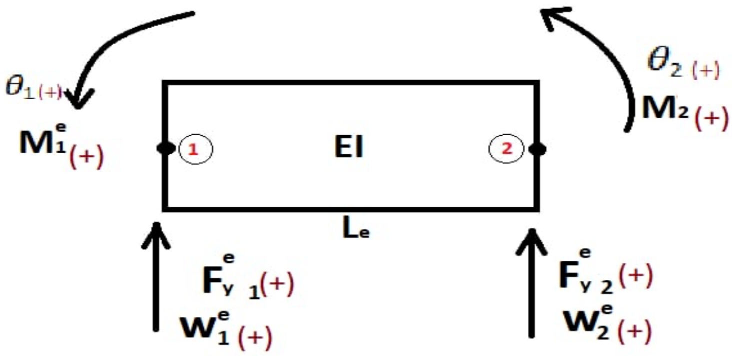

Figure 3.3: Sign convention for positive nodal transverse displacements w_i^e and rotations θ_i^e nodal forces F_{yi}^e and moments M_i^e.

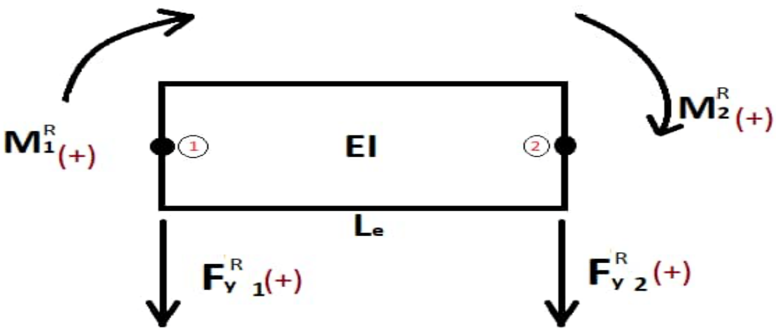

Figure 3.4: Sign convention for positive nodal reaction forces F_{yi}^R and reaction moments M_i^R.

3.3.3 Hermite Interpolation Function

Analyzing Euler-Bernoulli beam for bending is a class C^1 problem (continuity C^1) [2,20]. This mean that the deformation of a beam must have continuous slope $\theta^e = \frac{dw^e}{dx}$ as well as continuous transverse displacement w^eat any neighboring beam elements, and both transverse displacement w^e and slope $\theta^e = \frac{dw^e}{dx}$ are selected as nodal variable. This denotes that the Slope $\theta^e = \frac{dw^e}{dx}$ is the first derivative of transverse deflection w^e. Hermite interpolation function is used to satisfy this class C^1 problem (continuity C^1), it satisfies the continuity of both the transverse deflection $w^e$and the fist derivatives of transverse deflection (slope) $\theta^e = \frac{dw^e}{dx}$. Assume cubic Hermite polynomial function with four-parameter for the four degrees of freedom per element. To define the main variables for beam bending deformation due to transverse displacement w^e and rotation θ^e. The polynomial form of assumed interpolation function as shown below [3,21]:

$$w^e(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^3 \qquad \textit{(3.4) a}$$

From the assumption for the Euler-Bernoulli beam, the slope is first-derivatives of Transverse displacement w^e of Equations *(2.23) a*, as follows:

$$\theta^e(x) = \frac{dw^e}{dx} = \alpha_2 + \alpha_3 x + \alpha_4 x^2 \qquad \textit{(3.4.) b}$$

Where:

$w^e(x)$: The local element transverse displacement in y-direction.

$\theta^e(x)$: The rotation in the *x-y* plane with respect to the *z*-axis.

$\alpha_1, \alpha_2, \alpha_3, \alpha_4$: The four unknown constants found using the end conditions.

Now both transverse deflection w^e and it is first derivative slope $\theta^e = \frac{dw^e}{dx}$ are continuous between two neighboring elements, this satisfies continuity (compatibility) for class C^1 problem (continuity C^1. The convergence requirement for Hermite interpolation function for class C^1 problem (continuity C^1) as follows [8]:

1. Compatibility requirement for class C^1 problem (continuity C^1): Both the assumed trial solution for (vertical displacement (deflection)) and its first-order derivatives (slope) must be continuous across the

boundary of the element but not necessarily its second-order derivatives.

2. Completeness requirement (rigid body displacement and constant stain state) for class C^1 problem (continuity C^1): The assumed trial function for (vertical displacement (deflection)) must be able to represent A constant value of the exact function as well constant vales of its first-order derivatives and second-order derivatives.

For The Integral form of the governing partial differential equation require that the assumed trial function of an element to be continuous with nonzero derivatives up to second-order derivatives (twice differentiable) due to term $\frac{d^2w}{dx^2}$ and satisfies the essential boundary condition which automatically satisfies the continuity conditions [4,19]. Therefore, the use of Hermite interpolation function will meet all the convergence requirement. Therefore, both completeness condition (rigid body motion and constant strain condition) and compatibility condition (continuity condition) are met. Therefore, the resulting element is conforming element and the convergence will be monotonic convergence, which mean that the approximate solution will converges to the exact solution when refining the mesh [4].

Transverse flexural deformation of Euler-Bernoulli beam element based on cubic Hermite interpolation function is given by:

$$w^e(x) = N_{1b}^e w_1^e + N_{2b}^e \theta_1^e + N_{3b}^e w_2^e + N_{4b}^e \theta_2^e \qquad (3.5)$$

Where:

w_1^e , w_2^e: The nodal transverse displacement (deflection) in y-direction at nodes.

θ_1^e , θ_2^e: The nodal slope in the *x-y* plane with respect to the *z*-axis at nodes.

N_{1b}^e , N_{3b}^e: Hermite shape functions for transverse displacement w^e at nodes.

N_{2b}^e, N_{4b}^e: Hermite shape functions for rotation of cross-section (slope) θ^e at nodes.

L_e : Length of element.

The vector of local Hermite shape functions of flexural deformation at the *i*th node, is given by:

$$[N_{ib}^e] = [N_{1b}^e \quad N_{2b}^e \quad N_{3b}^e \quad N_{4b}^e] \qquad (3.6)\ a$$

$$[N_{ib}^e] = \left[\left(1 - \frac{3x^2}{L_e^2} + \frac{2x^3}{L_e^3}\right)\left(x - \frac{2x^2}{L_e} + \frac{2x^3}{L_e^2}\right)\left(\frac{3x^2}{L_e^2} - \frac{2x^3}{L_e^3}\right)\left(-\frac{x^2}{L_e} + \frac{x^3}{L_e^2}\right)\right] \quad (3.6)\ b$$

The specialty of Hermite polynomials is their values and the values of their derivatives up to nth order are (unity or zero) at the end points of the interval (0 to 1), substituting $at\ node\ (1)\ x = 0$ and $at\ node\ (2)\ x = L_e$, we found the properties of Hermit cubic interpolation function as it shown in Table 3.1, and in Figure 3.5 [4,21].

$$Shape\ function\ and\ its\ dertivative = 1\ at\ node\ i\ and\ zero\ at\ all\ other\ nodes \quad (3.7)$$

Table 3.1: The properties of Hermit cubic interpolation function [23].

Type of Function	**Translation Shape Function**				**Rotational Shape Function**			
Node No.	N_{1b}^e	N_{2b}^e	N_{3b}^e	N_{4b}^e	$\frac{dN_{1b}^e}{dx}$	$\frac{dN_{2b}^e}{dx}$	$\frac{dN_{3b}^e}{dx}$	$\frac{dN_{4b}^e}{dx}$
$at\ node\ (1)\ (x = 0)$	1	0	0	0	0	1	0	0
$at\ node\ (2)\ (x = L_e)$	0	0	1	0	0	0	0	1

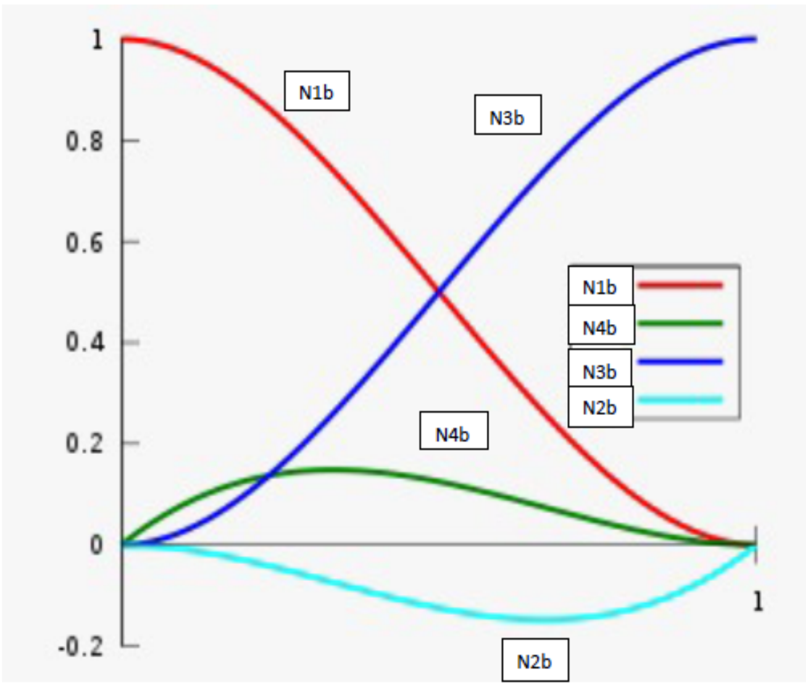

Figure 3.5: Plots of properties of Hermite cubic translational shape function.

3.3.4 Element Displacement Field for Flexural Deformation

In order to relate forces to deformation, a fundamental assumption in beam bending is that a plane section before bending remains plane and normal to the neutral axis after bending. Hence, the axial displacement *u* due to the transverse displacement *w* at a point *y* above the neutral axis can be expresses [1,2,15], as follows:

$$w^e(x) = \frac{dw}{dx} = \begin{bmatrix} N_{1b}^e & N_{2b}^e & N_{3b}^e & N_{4b}^e \end{bmatrix} \begin{Bmatrix} w_1^e \\ \theta_1^e \\ w_2^e \\ \theta_2^e \end{Bmatrix} \quad (3.8)\ a$$

$$w^e(x) = \frac{dw}{dx} = [N_{ib}^e]\{d_i^e\} \quad (3.8)\ b$$

Where:

$[N_{ib}^e]$: The vector of local Hermite shape functions of flexural deformation at the *i*th node.

$\{d_i^e\}$: The vector of local nodal element degrees of freedom at the *i*th node.

3.3.5 Element Strain Field for Flexural Deformation

The axial strain ε_{xx}^e is given by [1,2,15]:

$$\varepsilon_{xx}^e = \frac{d^2w}{dx^2} = \begin{bmatrix} \frac{dN_{1b}^e}{dx} & \frac{dN_{2b}^e}{dx} & \frac{dN_{3b}^e}{dx} & \frac{dN_{4b}^e}{dx} \end{bmatrix} \begin{Bmatrix} w_1^e \\ \theta_1^e \\ w_2^e \\ \theta_2^e \end{Bmatrix} \quad (3.9)\ a$$

$$\varepsilon_{xx}^e = \frac{d^2w}{dx^2} = [B_{ib}^e]\{d_i^e\} \quad (3.9)\ b$$

Where:

$[B_{ib}^e]$: Strain-displacement matrix for flexural deformation $(w\ \&\ \theta)$ of the first-derivative of shape function at *i*th node.

$$[B_{ib}^e] = \begin{bmatrix} \frac{dN_{1b}^e}{dx} & \frac{dN_{2b}^e}{dx} & \frac{dN_{3b}^e}{dx} & \frac{dN_{4b}^e}{dx} \end{bmatrix} \quad (3.10)\ a$$

$$[B_{ib}^e] = \left[\left(-\frac{6}{L_e^2} + \frac{12x}{L_e^3}\right) \left(-\frac{4}{L_e} + \frac{6x}{L_e^2}\right) \left(\frac{6}{L_e^2} - \frac{12x}{L_e^3}\right) \left(-\frac{2}{L_e} + \frac{6x}{L_e^2}\right) \right] \quad (3.10)\ b$$

3.3.6 Element Bending Moment or Stress Resultant Curvature Matrix

For a linear elastic material, the axial stress σ_x^e are related to the axial strain by Young's Modulus E (Hooke's law) is given by [2,14,28]:

$$\sigma_x^e = -E\,y\frac{d^2w}{dx^2} \qquad (3.11)$$

From basic solid mechanics the moment curvature relation is 2nd order PDE obtained from flexural deformation of Euler-Bernoulli beam, and it is given by [28]:

$$M = EI\frac{d^2w}{dx^2} \qquad (3.12)\ a$$

$$\frac{d^2w}{dx^2} = \frac{M}{EI} \qquad (3.12)\ b$$

The beam flexure or bending stress formula, which is can relate moment and axial stress is obtained by substituting Equation *(3.12) b* in Equation *(3.11),* we obtain the following [28]:

$$\sigma_x^e = -\frac{My}{I} \qquad (3.13)$$

Form elementary beam theory, the bending moment are related to the transverse displacement w function. Because we will use these relationships in the derivation of the beam element stiffness matrix, we now present them as in (the stress resultant curvature) as shown below in Equation *(3.14) a,* which relates the bending moment to the transverse displacement function [28]. This equation is used to represent general stress of a beam [2,14,15,21,28], as follows:

$$M = EI\frac{d^2w}{dx^2} = EI\,[B_{ib}^e]\{d_i^e\} \qquad (3.14)\ a$$

$$M = EI\begin{bmatrix}\frac{dN_{1b}^e}{dx} & \frac{dN_{2b}^e}{dx} & \frac{dN_{3b}^e}{dx} & \frac{dN_{4b}^e}{dx}\end{bmatrix}\begin{Bmatrix}w_1^e\\ \theta_1^e\\ w_2^e\\ \theta_2^e\end{Bmatrix} \qquad (3.14)\ b$$

$$M = EI\left[\left(-\frac{6}{L_e^2}+\frac{12x}{L_e^3}\right)\left(-\frac{4}{L_e}+\frac{6x}{L_e^2}\right)\left(\frac{6}{L_e^2}-\frac{12x}{L_e^3}\right)\left(-\frac{2}{L_e}+\frac{6x}{L_e^2}\right)\right]\begin{Bmatrix}w_1^e\\ \theta_1^e\\ w_2^e\\ \theta_2^e\end{Bmatrix} \qquad (3.14)\ c$$

Where:

M: The bending moment

EI: The beam of flexural rigidity.

3.3.7 Element Transverse Shear Force

The shear force is third order PDE from the elementary basic equations for flexural deformation (bending) of Euler-Bernoulli beam. For uniformly loaded beam, the resulting shear force is a constant within each beam element used in the model [2,28], as shown below:

$$V = \frac{dM}{dx} = EI\frac{d^3w}{dx^3} = EI\,[B_{ib}^e]'\{d_i^e\} \qquad (3.15)\ a$$

$$V = EI\begin{bmatrix}\frac{d^2N_{1b}^e}{dx^2} & \frac{d^2N_{2b}^e}{dx^2} & \frac{d^2N_{3b}^e}{dx^2} & \frac{d^2N_{4b}^e}{dx^2}\end{bmatrix}\begin{Bmatrix}w_1^e\\ \theta_1^e\\ w_2^e\\ \theta_2^e\end{Bmatrix} \qquad (3.15)\ b$$

$$V = EI\begin{bmatrix}\left(\frac{12}{L_e^3}\right) & \left(\frac{6}{L_e^2}\right) & \left(-\frac{12}{L_e^3}\right) & \left(\frac{6}{L_e^2}\right)\end{bmatrix}\begin{Bmatrix}w_1^e\\ \theta_1^e\\ w_2^e\\ \theta_2^e\end{Bmatrix} \qquad (3.15)\ c$$

Where:

$[B_i^e]'$: Strain-displacement matrix for flexural deformation $(w\ \&\ \theta)$ of second-derivative of shape at the *i*th node.

$$[B_i^e]' = \begin{bmatrix}\left(\frac{12}{L_e^3}\right) & \left(\frac{6}{L_e^2}\right) & \left(-\frac{12}{L_e^3}\right) & \left(\frac{6}{L_e^2}\right)\end{bmatrix} \qquad (3.16)$$

3.3.8 Element Stiffness Matrix for Flexural Deformation

The stiffness matrix that relates the nodal displacement to the nodal forces. A beam may contain an internal hinge as shown in Figure 3.6, which results in a discontinuity in the slope or rotation of the deflection curve at the hinge. Therefore, stiffness matrix is modified to include hinge desired effect. The element stiffness matrix for bending is given by following [2,8]:

$$[K_b^e] = EI\int_0^L [B_{ib}^e]^T[B_{ib}^e]dx \qquad (3.17)\ a$$

Where:

$[K_b^e]$: The local element stiffness matrix for bending.

$[B_{ib}^e]$: Strain-displacement matrix for flexural deformation $(w\ \&\ \theta)$ of the first-derivative of shape function at *i*th node.

EI: The beam flexural rigidity.

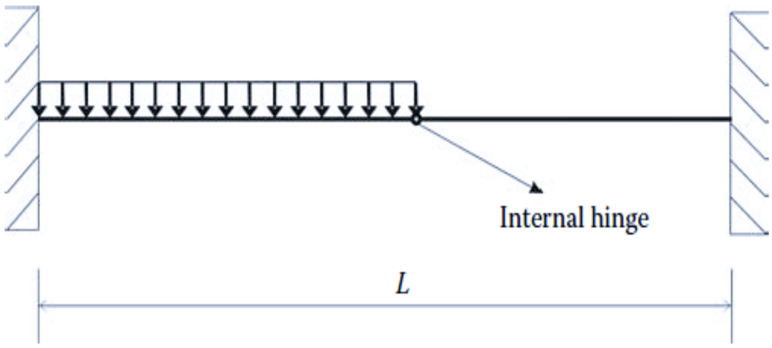

Figure 3.6: Beam with Internal hinge.

3.3.8.1 Element Stiffness Matrix without Internal Hinge Considered

$$[K_b^e]_{No.H} = \begin{bmatrix} \frac{12EI}{L_e^3} & \frac{6EI}{L_e^2} & -\frac{12EI}{L_e^3} & \frac{6EI}{L_e^2} \\ \frac{6EI}{L_e^2} & \frac{4EI}{L_e} & -\frac{6EI}{L_e^2} & \frac{2EI}{L_e} \\ -\frac{12EI}{L_e^3} & -\frac{6EI}{L_e^2} & \frac{12EI}{L_e^3} & -\frac{6EI}{L_e^2} \\ \frac{6EI}{L_e^2} & \frac{2EI}{L_e} & -\frac{6EI}{L_e^2} & \frac{4EI}{L_e} \end{bmatrix} \qquad (3.17)\ b$$

Where:

$[K_b^e]_{No.H}$: The local element stiffness matrix for bending without internal hinge.

EI: The beam flexural rigidity.

L_e: Element Length.

3.3.8.2 Element Stiffness Matrix with Internal Hinge Considered

At the hinge node the bending moment is equal to zero, but not the rotation. To model the hinge, we consider the hinge to be accounted and placed only once on either the right end or left end of an element, as shown in Figure 3.7, but not on both elements, because this will result in singular stiffness matrix. Therefore, the modified stiffness matrix considering internal hinge at its right or left end given as following [2,8]:

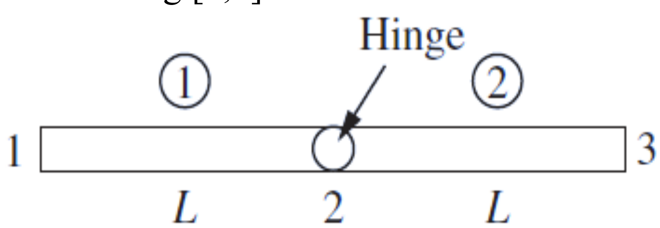

Figure 3.7: Beam with internal hinge considered either at right or left end of the element.

1. Modified element local element stiffness matrix considering a hinge at its left end

$$[K_b^e]_{H.R} = \begin{bmatrix} \frac{3EI}{L_e^3} & \frac{3EI}{L_e^2} & -\frac{3EI}{L_e^3} & 0 \\ \frac{3EI}{L_e^2} & \frac{3EI}{L_e} & -\frac{3EI}{L_e^2} & 0 \\ -\frac{3EI}{L_e^3} & -\frac{3EI}{L^2} & \frac{3EI}{L_e^3} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.18}$$

Where:

$[K_b^e]_{H.R}$: The modified local element stiffness matrix for first element with a nodal hinge at its right end.

2. Modified element local element stiffness matrix considering a hinge at its left end

$$[K_b^e]_{H.L} = \begin{bmatrix} \frac{3EI}{L_e^3} & 0 & -\frac{3EI}{L_e^3} & \frac{3EI}{L_e^2} \\ 0 & 0 & 0 & 0 \\ -\frac{3EI}{L_e^3} & 0 & \frac{3EI}{L_e^3} & -\frac{3EI}{L_e^2} \\ \frac{3EI}{L_e^2} & 0 & -\frac{3EI}{L_e^2} & \frac{3EI}{L_e} \end{bmatrix} \tag{3.19}$$

Where:

$[K_b^e]_{H.L}$: The modified local element stiffness matrix for second element with a nodal hinge at its left end.

3.3.9 Element Nodal Load Vector

Using the principle of minimum potential energy, the total work done by transverse loading for Euler-Bernoulli beam bending as follows [2]:

1. Distributed loading (Elements loads).
2. Concentrated loading (Joint loads).

3.3.9.1 The Work-Equivalence Method

The stiffness matrix for a beam element was developed for loading applied only at its nodes. To be compatible with the developed stiffness matrix, the distributed loading must be converted to nodal loads only, and also if the concentrated loads acting on beam applied other than at the natural intersection of two elements must be converted instead of creating a node or place a node at it a point of application. Therefore, the work-equivalence method is used. The concept of work-equivalence methods used replace a distributed load and concentrated loads applied at location other than at the natural intersection of two elements by a set of discrete loads (Statically equivalent nodal loads), these equivalent loads tend to have the same effect on the beam as the actual original distributed load, and they are always of opposite sign from the fixed-end reactions, as shown below [2]:

$$\{F^e\}_{distributed} = \{F_i^e\}_{discrete} \tag{3.20}$$

Where:

$\{F^e\}_{distributed}$: The local element distributed loading.

$\{F_i^e\}_{discrete}$: The vector of local element discrete loads that replace distributed loads by statically equivalent nodal loads at the *i*th node.

The work due to distributed loading is given by:

$$\{F^e\}_{distributed} = \int_0^L w_q(x)\, w(x)\, dx \tag{3.21}$$

Where:

$w_q(x)$: The distributed load along the length.

$w(x)$: The local element transverse displacement (deflection) in y-direction.

The work due to the replaced discrete loads is given by:

$$\{F_i^e\}_{discrete} = w_1 F_{y1}^0 + \theta_1 M_1^0 + w_2 F_{y2}^0 + \theta_2 M_2^0 \qquad (3.22)\ a$$

$$\{F_i^e\}_{discrete} = \{F_{y1}^0 \quad M_1^0 \quad F_{y2}^0 \quad M_2^0\} \begin{Bmatrix} w_1 \\ \theta_1 \\ w_2 \\ \theta_2 \end{Bmatrix} = \{F_i^0\}\{d_i^e\} \qquad (3.22)\ b$$

Where:

$\{d_i^e\}$: The vector of local nodal element deflection due to discrete loads at the *i*th node.

$\{F_i^0\}$: The vector of local element statically equivalent nodal loads (Element loads) at the *i*th node

$$\{F_i^0\} = \begin{Bmatrix} F_{y1}^0 \\ M_1^0 \\ F_{y1}^0 \\ M_1^0 \end{Bmatrix} \qquad (3.23)\ a$$

Where:

F_{y1}^0 , F_{y2}^0: Equivalent nodal forces applied at nodes (Element loads).

M_1^0 , M_2^0: Equivalent nodal moments applied at nodes (Element loads).

To represent the local element nodal force vector $\{F_i^e\}$ for beam element as the sum of those nodal forces resulting from statically equivalent nodal loads and concentrated nodal loading:

$$\{F_i^e\} = \{F_i^0\} + \{F_i^c\} \qquad (3.24)$$

Where:

$\{F_i^e\}$: The vector of local element nodal loads due to statically equivalent loads and concentrated nodal loads at the *i*th node.

$\{F_i^0\}$: Equivalent nodal element loads due to distributed loads (Element loads) at the *i*th node.

$\{F_i^c\}$: Concentrated nodal load applied at nodes (Joint loads) at the *i*th node.

3.3.9.2 Distributed Loading

In the appendix there is a list of statically equivalent nodal loads for most common types loading presented. For example, for the uniformly distributed load the generalized local statically equivalent nodal loads that replace uniformly distributed load at the *i*th node, are given by [2,4]:

$$\{F_i^0\} = \begin{Bmatrix} F_{y1}^0 \\ M_1^0 \\ F_{y2}^0 \\ M_2^0 \end{Bmatrix} = \begin{Bmatrix} -\dfrac{w_q L_e}{2} \\ -\dfrac{w_q L_e^2}{12} \\ -\dfrac{w_q L_e}{2} \\ \dfrac{w_q L_e^2}{12} \end{Bmatrix} \qquad (3.23)\ b$$

Where:

w_q: The distributed load along the element length.

L_e: Length of element.

3.3.9.3 Concentrated Loading

Concentrated nodal loading $\{F_i^c\}$ (Joint loads) is applied directly at the nodes. Hence, it can be concentrated nodal force and concentrated nodal moment, as follows:

$$\{F_i^c\} = \begin{Bmatrix} F_{y1}^C \\ M_1^C \\ F_{y2}^C \\ M_2^C \end{Bmatrix} \qquad (3.25)$$

Where:

$\{F_i^c\}$: The vector of concentrated nodal loads applied directly at the *i*th node (Joint loads).

F_{y1}^C , F_{y2}^C: Concentrated nodal forces applied at nodes (Joint loads).

M_1^C , M_2^C: Concentrated nodal moments applied at nodes (Joint loads).

3.3.10 Coordinate Transformation

Coordinate transformation is used to transform the beam element matrices from the local coordinate system into the global coordinate system. However, the transformation is necessary when there is more than beam element in the beam structure with different orientations such case in curved beam and in frame structures as shown in Figure 3.8 and Figure 3.9 below which require coordinate transformation [8,19]. In this research, the developed finite element computer program is used for the analysis of thin straight beam. Therefore, there is no need to transform the element matrixes from the local to global coordinates since both sets of axes are colinear. For each element set up the local stiffness matrix and directly assemble it into the global stiffness matrix.

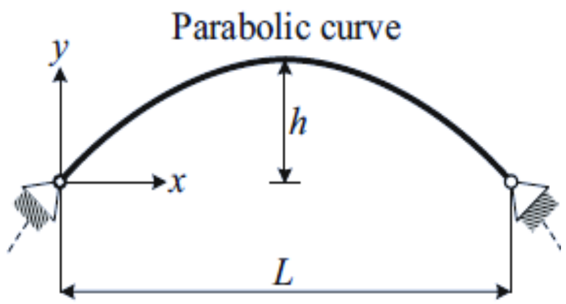

Figure 3.8: Curved beam with local axes not colinear with global axes [29].

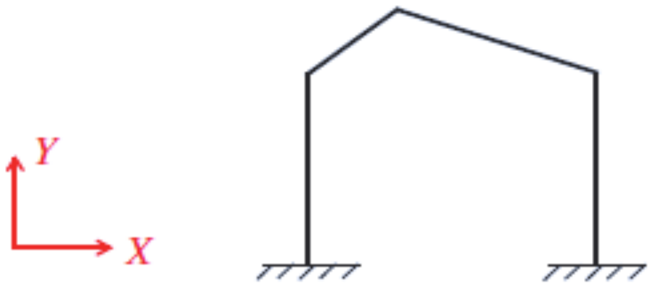

Figure 3.9: Frame structure with different members orientation in global coordinate system [1].

3.3.11 Global System of Equations

For a system of connected planar beams, the global system of equation formed using Euler-Bernoulli Beam Element [2]:

$$[K_b]^s\{d_i\}^s = \{F_i\}^s \tag{3.26) a$$

$$[K_b]^s\{d_i\}^s = \left\{F_i^0\right\}^s + \{P_i^c\}^s \tag{3.26) b$$

Where:

$[K_b]^s$: The global stiffness matrix for thin beam bending.

$\{d_i\}^s$: The vector of global structure nodal degrees of freedom at the *i*th node.

$\{F_i\}^s$: The vector of global nodal loads at the *i*th node.

$$\{F_i\}^s = \left\{F_i^0\right\}^s + \{P_i^c\}^s \tag{3.27}$$

Where:

$\left\{F_i^0\right\}^s$: The vector of global equivalent nodal loads at the *i*th node.

$\{P_i^c\}^s$: The vector of global concentrated nodal loads at the *i*th node.

3.3.12 Support Nodal Reactions

The reactions at support are noting but end equilibrium forces. The displacement computed using statically equivalent nodal loads are exact in a finite elements sense at nodes. However, to account for distributed loads or concentrated loads acting on beam element to calculated the reaction this requires same adjustment, to obtain the correct nodal reactions as can be verified by simple static equilibrium equations, the following steps must be used [2]:

1. Replace the distributed load by it statically equivalent nodal loads to identify the nodal force and moment used in the solution $\{F^e\}_{distributed} = \left\{F_i^0\right\}$.
2. Assemble global equations $[K_b]^s\{d_i\}^s = \{F_i\}^s$

   Apply the boundary conditions to reduce the set of equations

$$[[K^R]^s]\{d_i^R\}^s = \{F_i\}^s \tag{3.28) a$$

3. Solve this reduced set of equation to obtain the unknown nodal degrees of freedom (displacement and rotation)

$$\{d_i^R\}^s = [[K^R]^s]^{-1}\{F_i\}^s \tag{3.28) b$$

4. Obtain the vector of effective global nodal forces.

$$\left\{F_i^{(e)}\right\}^s = [K]^s\{d_i^R\}^s \tag{3.29}$$

Where:

$[K]^s$: The global stiffness matrix.

$\{d_i^R\}^s$: The vector all global nodal displacement obtained from global reduced matrix equation which combines (the determined displacement with displacement boundary condition).

5. Obtain the correct final global nodal reactions (forces and moments) at the supports

$$\{F_i^R\}^s = \left\{F_i^{(e)}\right\}^s - \{F_i^0\}^s \qquad \textit{(3.30) a}$$

Where:

$\left\{F_i^{(e)}\right\}^s$: The vector of effective global nodal forces at the *i*th node.

$\{F_i^0\}^s$: The vector of global equivalent nodal loads at the *i*th node.

$\{F_i^R\}^s$: The vector of correct global nodal support reactions at the *i*th node.

$$\{F_i^R\}^s = \begin{Bmatrix} F_{y1}^R \\ M_1^R \\ \vdots \\ F_{yn}^R \\ M_n^R \end{Bmatrix} \qquad \textit{(3.30) b}$$

Where:

F_{yi}^R : The vector of vertical global nodal force reaction at *i*th node.

M_i^R: The vector of global nodal moment reaction at *i*th node.

CHAPTER FOUR

DISCRIPTION OF LINEAR FINITE ELEMENT COMPUTER PROGRAM

4.1 Introduction

In this chapter finite element computer programs are developed to include the theory presented in the previous chapters. The finite element formulation is used to derive the element equations which is used in the finite element computer program. First, it is important to review the steps of FEM for structural mechanics problem. Hence, to understand the three phases of finite element program structure (pre-processing, processing, and post-processing) for structural mechanics problems.

Important thing to be kept in mind, computer do what you ask, not what you wish. If a computer delivers an unsatisfactory result, it is most likely because you did not properly translate your wish"[26] . Programming is to generate a list of instructions suitable for execution by a computer, through splitting the problem into a list of elementary operations, then using a programming language to put these operations into coding [3]. Therefore, once logic of implementation of the FEA variables is understood programming can be carried out using any programming language [4].

MATLAB is a high-level language specially designed for dealing with matrices, this makes it particularly suited for programing the FEM. It allows new finite element programmers to focus on the FEM rather on the programming details [3,6,8,27].

The computer program is named LSATSBB (Linear Static Analysis of Thin Straight Beam Bending), The program was coded in MATLAB R2019b. The program can be used to analyze thin straight beam bending under static transverse loads. The program described here include the linear formulation for 2-node thin beam bending element. The program developed have the facility to use different material types. The mesh generation is done manually. The sign conventions adopted are as follows:

1. Nodal transverse displacements w_i^e and nodal force F_{yi}^e are positive in the positive y-direction.
2. Nodal rotations θ_i^e and nodal moment M_i^e are positive in the counter-clockwise direction.

3. Nodal reaction force F_{yi}^{R} are positive in the negative y-direction.
4. Nodal reaction moment M_{i}^{R} are positive in the clockwise direction.

The main program flow chart shown in Figure 4.1
Appendix (A) shows the main program code.
Appendix (B) shows two sample of output file.
Brief descriptions will be presented for the main program, sub-programs and functions.

4.2 The Structure of Finite Element Program

The structure of main program organizes the three basic phases of FEA program for structural mechanic problem such as pre-processing, processing, and post-processing as it explained in Figure 4.1. In the pre-processing phase, the mesh generation is done manually, and also the all-input data used to build analysis modal are entered manually in the input sub-program. In the processing phase, the program conducts serval analysis steps to get unknown nodal deflection (primary unknown variables). In the post-processing phase, the program calculates the nodal member actions (secondary unknow variables) and print finite element solution result.

The main program control various tasks, three sub-programs and six functions. Some of these functions is called by other functions. The name and description of each supplementary subprograms is presented in Table 4.1. Hence, the name and description of each function is presented in Table 4.2, as follows:

Table 4.1: The name and description of each supplementary subprograms.

Sub-Program	Description
Sub_Input_Modal_Data.m	Sub-program to enter modal data for thin beam to build analysis modal
Sub_Print_Display_Modal_Data.m	Sub-program to print a display of input data for review.
Sub_Print_Solution_Results.m	Sub-program to print FEA solution results.

Table 4.2: The name and description of each function.

Functions	Description
$[\mathbf{g}] = \mathbf{beam\_g}\ (\mathbf{i})$	Function to form element steering vector form element nodal DOFs
$[\mathbf{F}] = \mathbf{form\_beam\_F}\ (\mathbf{F})$	Function to from the global loads vector.
$[\mathbf{KK}] = \mathbf{form\_KK}\ (\mathbf{KK}, \mathbf{kg}, \mathbf{g})$	Functions to form the global stiffness matrix.
$[\mathbf{kl}] = \mathbf{beam\_k}\ (\mathbf{i})$	Function to calculate local element stiffness matrix.
$[\mathbf{F}] = \mathbf{Assem\_Elem\_loads}\ (\mathbf{F}, \mathbf{fg}, \mathbf{g})$	Function to Assemble element loads to global force vector
$[\mathbf{F}] = \mathbf{Assem\_Joint\_loads}\ (\mathbf{F})$	Function to Assemble Joint loads to global force vector

4.3 The Main Program

The flow chart of main program (**Main_FEA_Program.m**) is presented Figure 4.1. and also presented the flow chart of the steps to obtain global loads vector Figure 4.2, and flow charts of the steps to obtain global stiffness matrix Figure 4.3. The program can be extended for the linear static analysis of two-dimensional plane frame structures based on Euler-Bernoulli theory. Also, the program may include other types of analysis e.g., nonlinear and dynamic response.

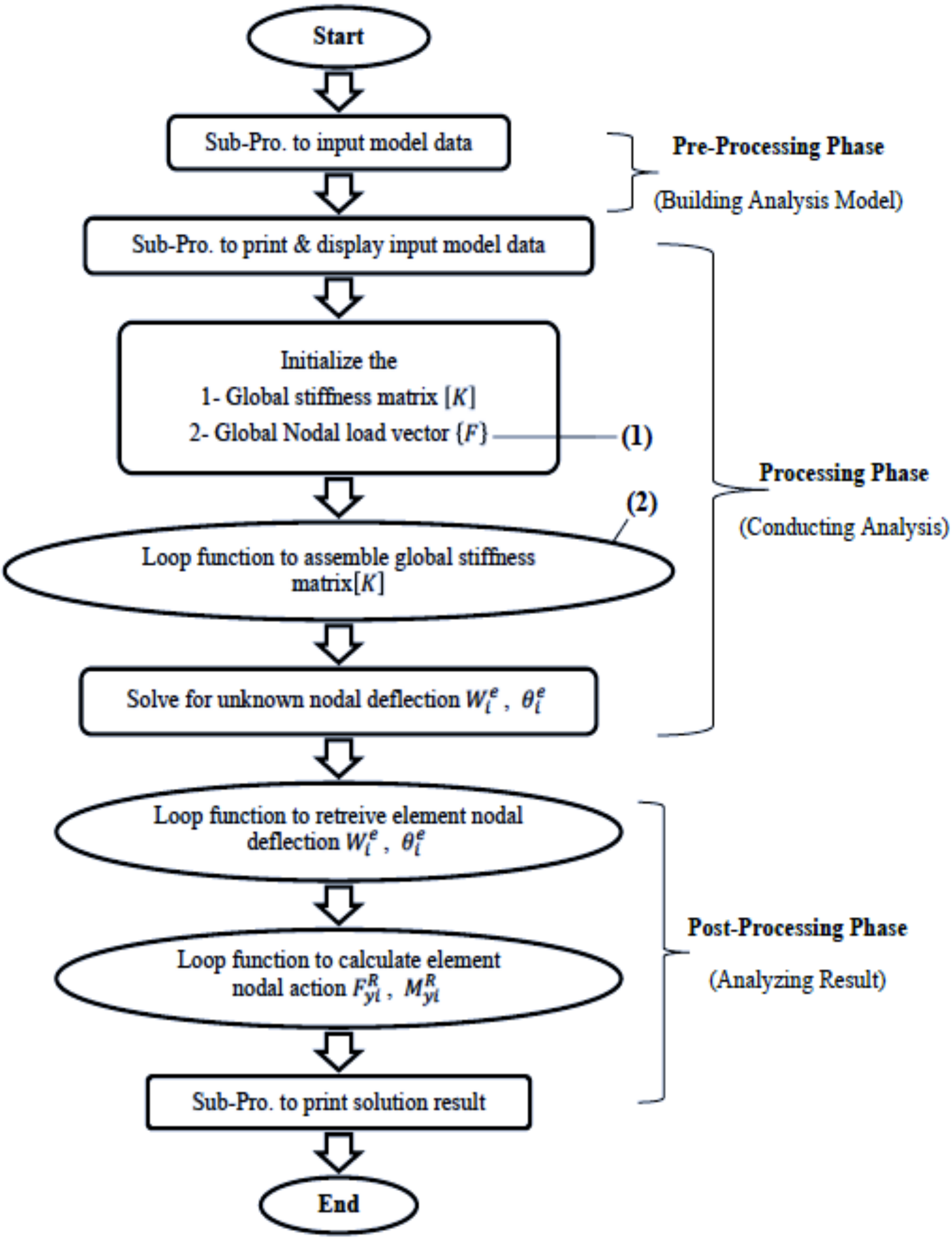

Figure 4.1: Flow chart of main program.

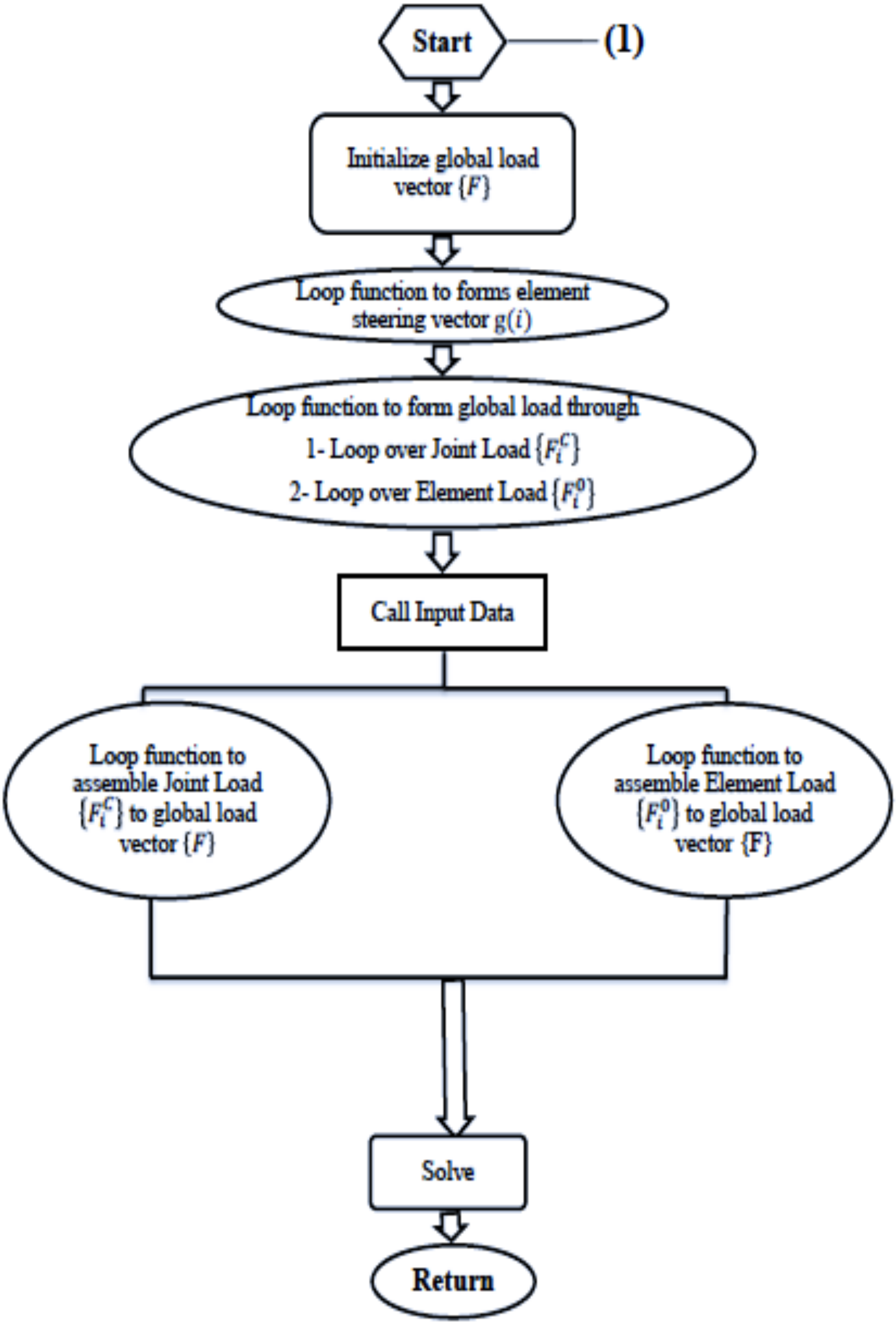

Figure 4.2: Flow chart of the steps to obtain global loads vector $\{F\}$.

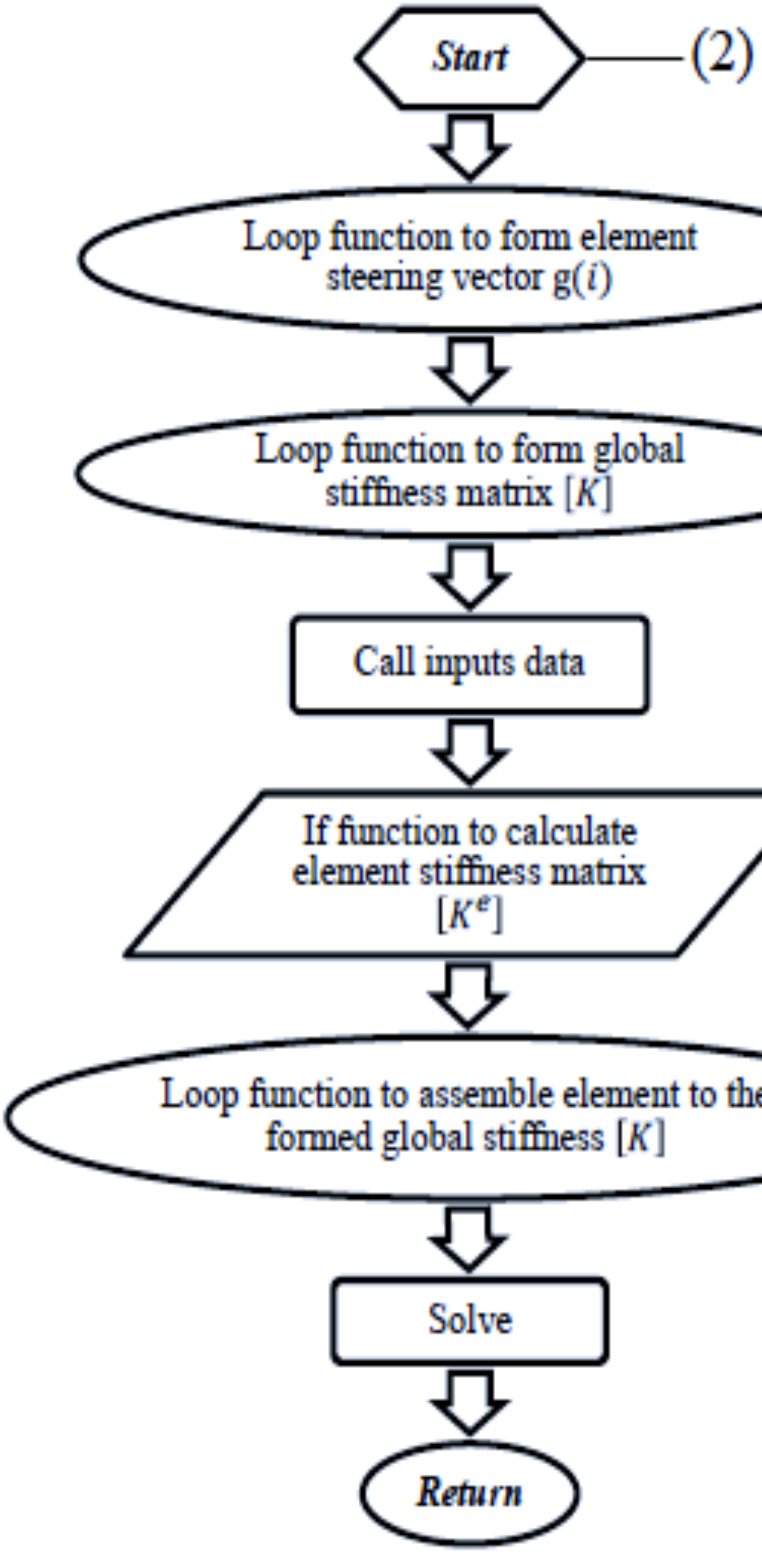

Figure 4.3: Flow chart of the steps to obtain global stiffness matrix $[K]$.

4.3.1 Sub-Program **Sub_Input_Modal_Data.m**

The main program reads the necessary data that defines the structures from the sub-program (**Sub_Input_Modal_Data.m**), these data are (geometrical data, nodal coordinates, nodal connectivity, geomatic properties for each element, boundary conditions, internal hinge, concentrated joint nodal loads, and statically equivalent nodal loads). The reading is free format and separated by commas (;). The user is free to adopt any consistent set of units for lengths and forces. Since the basic building block in MATLAB is a matrix, the data is prepared in the form of tables whenever possible as they are very easily translated into matrices. The details of writing modal data in the sub-program (**Sub_Input_Modal_Data.m**) are as follows:

4.3.1.1 Geometrical Data

Table 4.3: Input modal for geometrical data.

Variable	Description
nnd	Number of nodes.
nel	Number of elements.
nne	Number of nodes per elements.
nodof	Number of degrees of freedom per node.
eldof	Number of degrees of freedom per element.

4.3.1.2 Nodal Coordinate Data

Table 4.4: Input modal for nodal coordinate data.

Variable	Description
geom = [coordinate value of node (1) = 0; coordinate value of node (2); … …. coordinate value of node (n)] ;	The vector form of nodal x-coordinate.

4.3.1.3 Element Connectivity Data

Table 4.5: Input modal for element connectivity data.

Variable	Description
connec = [1st and 2nd node of element (1); 3rd and 4th node of element (2); …. nth and mth node of element (nel)] ;	The matrix form of element connectivity.

4.3.1.4 Element Materials and Geometrical Properties Data

Table 4.6: Input modal for element materials and geometrical properties data.

Variable	Description
prop = [(E)for element (1) (I) for element (1); ….; (E)for element (nel) (I) for element (nel)];	The matrix form of element materials (Young's modules) and geometrical properties (second moment of intertia of the cross-section).

4.3.1.5 Boundary Condition Data

Table 4.7: Input modal for boundary conditions data.

Each node has two degrees of freedom transverse deflection w_i^e and rotations θ_i^e. For restrained degrees of freedom take (value = 0), and for free degrees of freedom take (value = 1).	
Variable	**Description**
nf(nnd, 1) = value restrained or free ;	Prescribed nodal degrees of freedom at node (i) for deflection w_i^e.
nf(nnd, 2) = value restrained or free ;	Prescribed nodal degrees of freedom at node (i) for rotations θ_i^e.

4.3.1.6 Internal Hinges Data

Table 4.8: Input modal for internal hinges data.

If internal hinges considered assign (value = 0), if not assign (value = 1). A hinge must be considered for one element only at its left end or its right end, not on both elements.	
Variable	**Description**
hinge(nel, 1) = value if considered or not ;	The hinge is considered at its left end for element (i).
hinge(nel, 2) = value if considered or not ;	The hinge is considered at its right end for element (i).

4.3.1.7 Joint Nodal Load Data

Table 4.9: Input modal for joint nodal loads data.

Variable	Description
$\text{Joint\_loads}(nnd,:) = [\text{value } F_{yi}^{C} \quad \text{value } M_{yi}^{C}]$;	Prescribed value for nodal force and nodal moment

4.3.1.8 Element Nodal Loads Data

Table 4.10: Input modal for element nodal loads data.

Variable	Description
$\text{Element\_loads}(nnd,:) = [\text{value } F_{y1i}^{0} \quad \text{value } M_{y1i}^{0} \; \text{value } F_{y2i}^{0} \quad \text{value } M_{y2i}^{0}]$;	Prescribed value for nodal force and nodal moment at the 1st and 2nd node of element

4.3.2 Sub-Program **Sub_Print_Display_Modal_Data.m**

After entering all input data and selecting the required input file to be analyzed. When pressing a run button to conduct the analysis, the sub-program prints a display review in the MATLAB command window for input modal data of the selected file.

4.3.3 Sub-program **Sub_Print_Solution_Results.m**

After conducting all computation required in processing and post-processing phase, the sub-program print FEA solution results.

4.3.4 Function $[\mathbf{g}] = \mathbf{beam\_g}\ (\mathbf{i})$

Since element loads (the statically equivalent nodal loads) are element based (for all four nodal element degrees of freedom). Therefore, there is an important need for steering vector function [g] that contain the number of degrees of freedom of the nodes of the element. This function is formed by creating a loop over the element degrees of freedom by retrieving element nodal connectivity and element BCs. If a degree of freedom is not restrained it is susceptible of carrying the element loads.

The steering vector function one of most important functions in the program. The function is called three times in the main program, and in two of the sub-programs and most functions such as (forming the global loads vector, assembly of element loads into global loads vector, forming global stiffness matrix, the assembly of element stiffness matrix into global stiffness matrix, and the calculation of local member action).

4.3.5 Function $[\mathbf{F}] = \mathbf{form\_beam\_F}\ (\mathbf{F})$

This function forms the global load vector. This function is formed by crating one loop for joints loads and one loop for element loads. If a degree of freedom is not restrained it is susceptible of carrying a load.

4.3.6 Function $[\mathbf{KK}] = \mathbf{form\_KK}\ (\mathbf{KK}, \mathbf{kg}, \mathbf{g})$

This function forms the global stiffness matrix. This function is formed by creating a loop over element degrees of freedom using steering vector function.

4.3.7 Function $[\mathbf{kl}] = \mathbf{beam\_k}\ (\mathbf{i})$

This function calculates the element stiffness in local coordinates. Since the element axes are colinear with global axes there is no need to transform the element stiffness matrix from local to global coordinates. Therefore, for each element from (one to number of element), we set up the local stiffness matrix and directly assemble it into the formed global stiffness matrix. This function is formed by loop over number of elements and if statement for the element with or without internal hinge considered at its left or right end but only once.

4.3.8 Function $[\mathbf{F}] = \mathbf{Assem\_Elem\_loads}\ (\mathbf{F}, \mathbf{fg}, \mathbf{g})$

This function assembles the element loads (statically equivalent nodal loads) to the formed global load vector. This function formed by creating a loop over the elements degrees of freedom using steering vector function.

4.3.9 Function $[\mathbf{F}] = \mathbf{Assem\_Joint\_loads}\ (\mathbf{F})$

This function assembles the joint nodal loads to the formed global loads vector. This function is formed by creating a loop over a number of nodes and the number of degrees of freedom of per node.

4.4 Program Implementation

All the program described in the preceding sections were coded in Program. Necessary checks for debugging the code were carried out. The programs were then applied to verify the performance of the element as shown in the next chapter.

CHAPTER FIVE
VERIFICATION OF PROGRAM RESULTS

5.1 Introduction

In this chapter numerical examples discussed will be compared to the developed code output. nine different beams will be conducted for this purpose as follows:

1. Cantilever beam subjected to UDL.
2. Fixed-fixed beam subjected to concentrated loads
3. Overhanging beam subjected to UDL.
4. Simple supported beam with varying cross-section subjected to concentrated loads.
5. Cantilever beam subjected to UDL and concentrated loads.
6. Continuous beam subjected to UDL and concentrated loads.
7. Beam with internal hinge subjected to UDL.
8. Cantilever beam subjected to linearly varying distributed load.
9. Beam subjected to concentrated load, UDL, and linearly varying distributed load.

For each examples the basic structural data is displayed with sketches. The analysis results are compared with published results is given in tabulated and graphical form, and also the location of maximum nodal values is presented.

5.2 Application to Numerical Examples

5.2.1 Example (1): Cantilever Beam Subjected to UDL.

Cantilever beam subjected to uniformly distributed load, as shown in Figure 5.1. the modules of elasticity of the beam $E = 30 \times 10^6$ psi, the second moment of area of cross-section I =100 in^4, beam length $L = 100$ in, and uniform load $w = 20$ Ib/in. compare the finite element solution to the exact classical beam theory solution (Double-integration method) using one and two element finite element solution. Example 4.5 Daryl L. Logon [2].

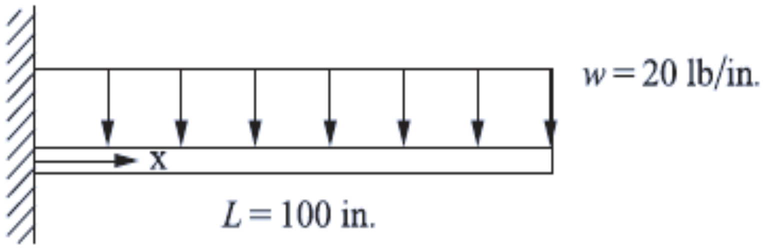

Figure 5.1: Cantilever beam subjected to UDL.

5.2.1.1 Example (1): For One Element Solution at Nodal points

For one finite element solution for deflection at nodal points, the program results obtained are in good agreement with those obtained by Daryl L. Logon [2], as shown in (Table 5.1, Table 5.2, Figure 5.2, and Figure 5.3).

Table 5.1: Example (1) one element solution for vertical deflection at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Published Exact Result**	**Present Difference**
		w_i^e (in)	w_i^e (in)	w_i^e (in)	w_i^e (%)
Node (1)	0	0	0	0	0
Node (2)	100	-0.0833	-0.0833	-0.0833	0

Table 5.2: Example (1) one element solution for rotation at nodal points.

Nodal Number	**Nodal coordinate (m)**	**Program Result**	**Published FEA Result**	**Published Exact Result**	**Present Difference**
		θ_i^e (rad)	θ_i^e (rad)	θ_i^e (rad)	θ_i^e (%)
Node (1)	0	0	0	0	0
Node (2)	100	-0.00111	-0.00111	-0.00111	0

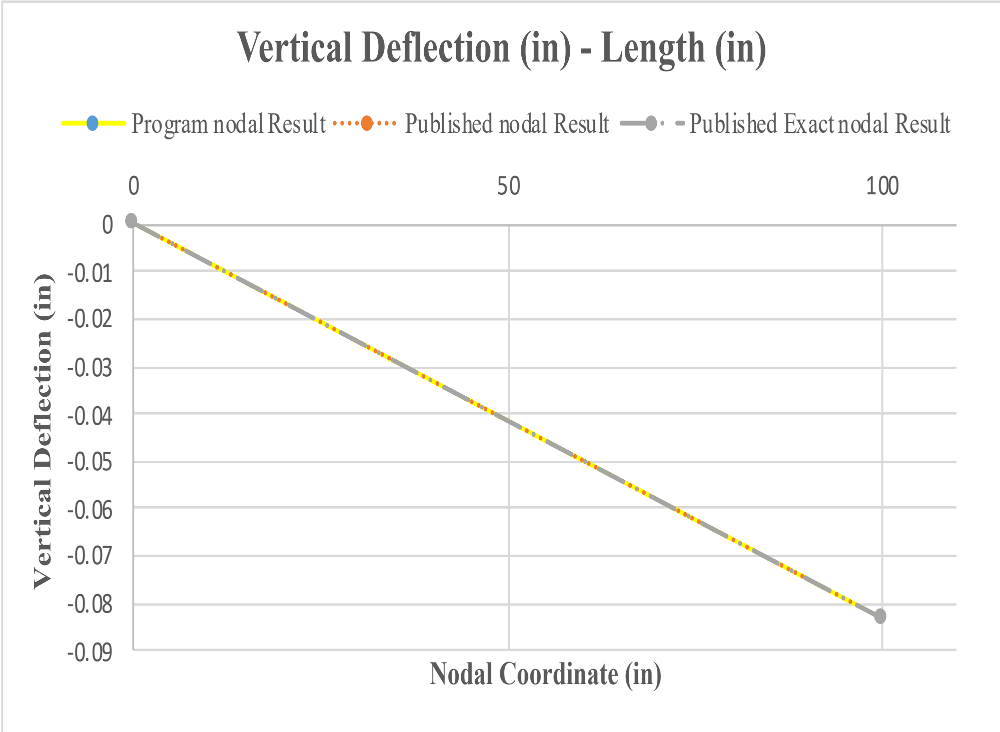

Figure 5.2: Example (1) one finite elements solution for vertical deflection along beam length at nodal points.

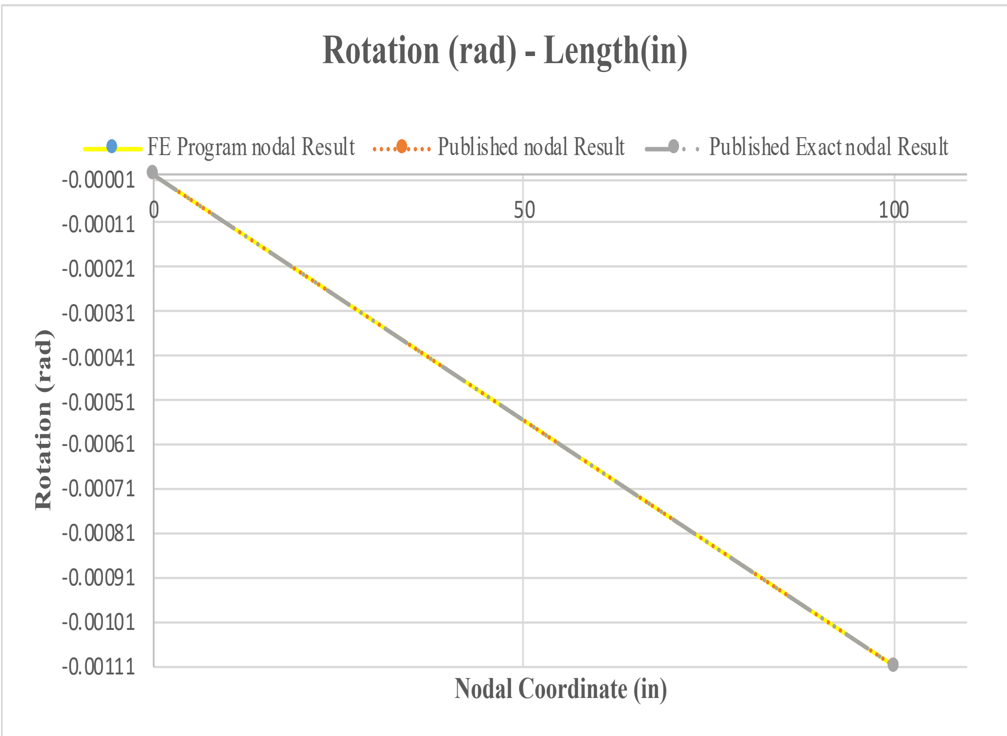

Figure 5.3: Example (1) one finite elements solution for rotation along beam length at nodal points.

5.2.1.2 Example (1): For One Element Solution at Nodal points and Mid-Length

For one finite element solution for displacement at nodal points and at mid-length, the program results obtained are in good agreement with those obtained by Daryl L. Logon [2], as shown in (Table 5.3, and Figure 5.4).

Table 5.3: Example (1) one element solution for vertical deflection at nodal points and mid-length of a beam at distance ($x = 50$ in).

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Published Exact Result**	**Present Difference**
		w_i^e (in)	w_i^e (in)	w_i^e (in)	w_i^e (%)
Node (1)	0	0	0	0	0
$at\ x = 50$ in	50	-0.0278	-0.0278	-0.0295	5.76 %
Node (2)	100	-0.08333	-0.08333	-0.08333	0

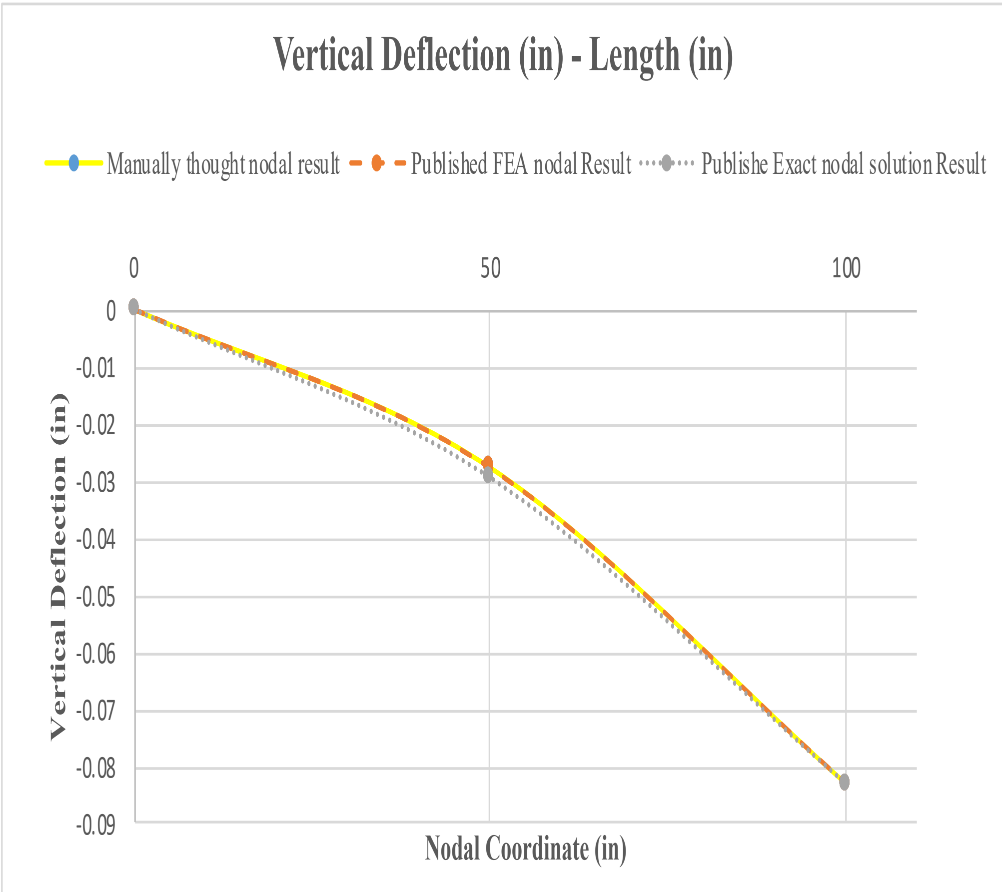

Figure 5.4: Example (1) one finite elements solution for vertical deflection along beam length at nodal points and mid-length of a beam ($x = 50$ in).

5.2.1.3 Example (1): For Two Elements Solution at Nodal Points

For two finite elements solution for deflection at nodal points, the program results obtained are in good agreement with those obtained by Daryl L. Logon [2], as shown in (Table 5.4, Table 5.5, Figure 5.5, and Figure 5.6).

Table 5.4: Example (1) two elements solution for vertical deflection at nodal points.

Nodal Number	Nodal Coordinate (m)	Program Result	Published FEA Result	Published Exact Result	Present Difference
		w_i^e (in)	w_i^e (in)	w_i^e (in)	w_i^e (%)
Node (1)	0	0	0	0	0
Node (2)	50	-0.02951	-0.02951	-0.02951	0
Node (3)	100	-0.08333	-0.0833	-0.0833	0.0360 %

Table 5.5: Example (1) two elements solution for rotation at nodal points.

Nodal Number	Nodal Coordinate (m)	Program Result	Published FEA Result	Published Exact Result	Present Difference
		θ_i^e (rad)	θ_i^e (rad)	θ_i^e (rad)	θ_i^e (%)
Node (1)	0	0	0	0	0
Node (2)	50	-0.00097	-0.000972	-0.000972	0.2057 %
Node (3)	100	-0.00111	-0.00111	-0.00111	0

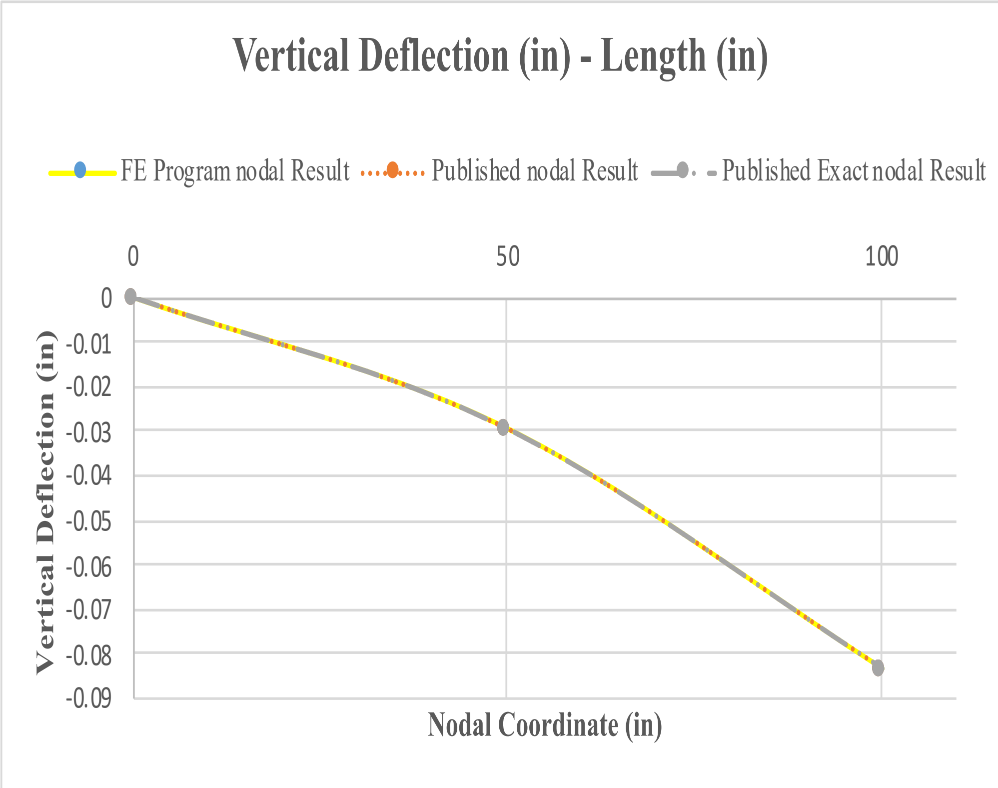

Figure 5.5: Example (1) two finite elements solution for vertical deflection along beam length at nodal points.

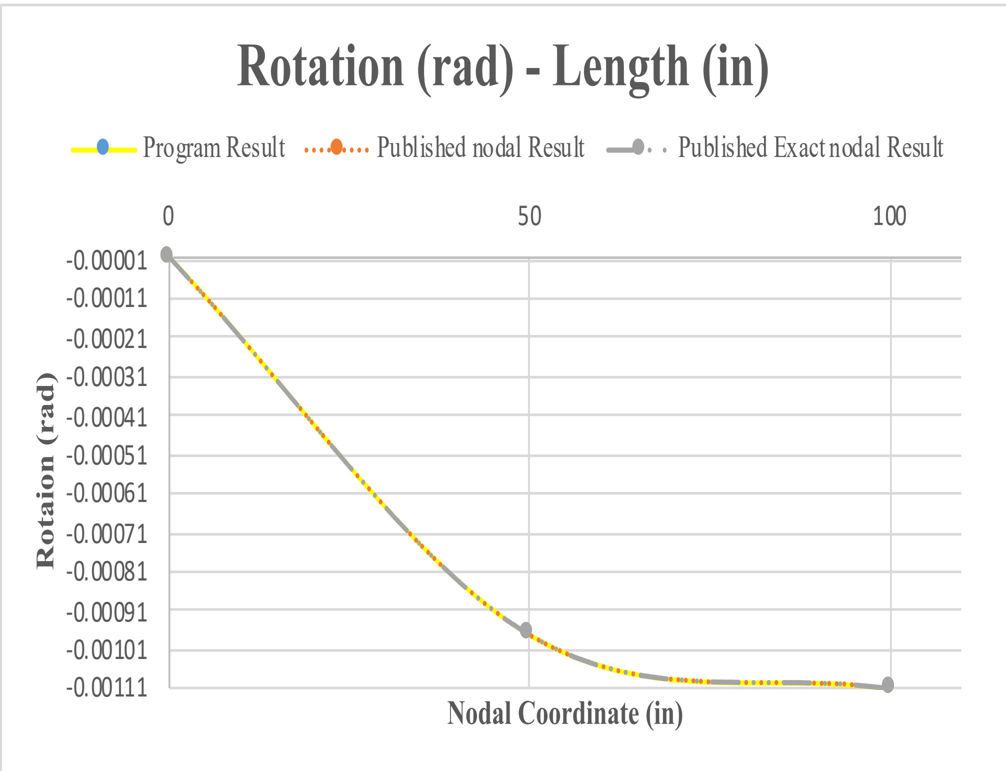

Figure 5.6: Example (1) two finite elements solution for rotation along beam length at nodal points.

5.2.2 Example (2) Fixed-Fixed Beam Subjected to Concentrated Loads

Determine the displacement and rotation under the force and moment located at the center of the beam shown in Figure 5.7. The beam has been discretized into two elements. the beam is fixed at each end. A downward force of 10 kN and an applied moment 20 kN.m act at the center of beam. The modules of elasticity of the beam $E = 210$ GPa and the second moment of area of cross-section $I = 4 \times 10^{-4}$ m^4 throughout the beam length. Example 4.4 Daryl L. Logon [2].

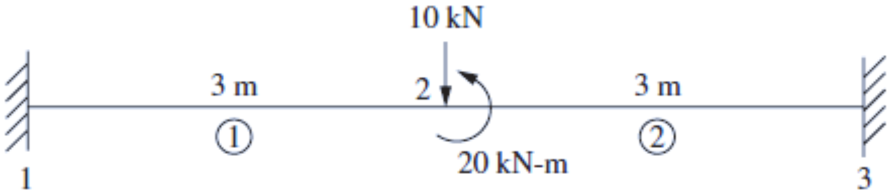

Figure 5.7: Fixed-fixed beam subjected to concentrated loads.

For two finite elements solution for nodal variables at nodal points, the program results obtained are in good agreement with those obtained by Daryl L. Logon, as shown in (Table 5.6, Tables 5.7, Table 5.8, Table 5.9, Figure 5.9, Figure 5.10, Figure 5.11, and Figure 5.12). Figure 5.8 shows how nodal loads is assembled the Table 5.8 and Table 5.9.

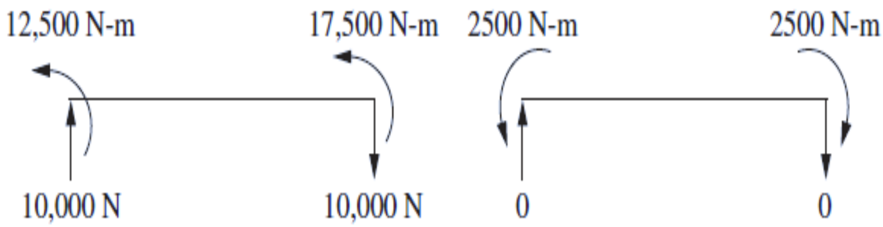

Figure 5.8: Nodal forces and moments acting on each element.

Table 5.6: Example (2) two elements solution for vertical deflection at nodal points.

Nodal Number	Nodal Coordinate (m)	Program Result	Published FEA Result	Present Difference
		w_i^e(m)	w_i^e(m)	w_i^e (%)
Node (1)	0	0	0	0
Node (2)	3	-0.00013	-0.0001339	2.912 %
Node (3)	6	0	0	0

Table 5.7: Example (2) two elements solution for rotation at nodal points.

Nodal Number	Nodal Coordinate (m)	Program Result	Published FEA Result	Present Difference
		θ_i^e(rad)	θ_i^e(rad)	θ_i^e (%)
Node (1)	0	0	0	0
Node (2)	3	0.00009	0.00008928	0.8064 %
Node (3)	6	0	0	0

Table 5.8: Example (2) two elements solution for shear force at nodal points.

Nodal Number	Nodal Coordinate (m)	Program Result	Published FEA Result	Present Difference
		F_i^e(N)	F_i^e(N)	F_i^e (%)
Node (1)	0	10,000	10,000	0
Node (2)	3	-10,000	-10,000	0
Node (3)	6	0	0	0

Table 5.9: Example (2) two elements solution for bending at nodal points.

Nodal Number	Nodal coordinate (m)	Program nodal result	Published FEA nodal result	Present Difference
		M_i^e(N.m)	M_i^e(N.m)	M_i^e (%)
Node (1)	0	12,500	12,500	0
Node (2)	3	20,000	20,000	0
Node (3)	6	-2,500	-2,500	0

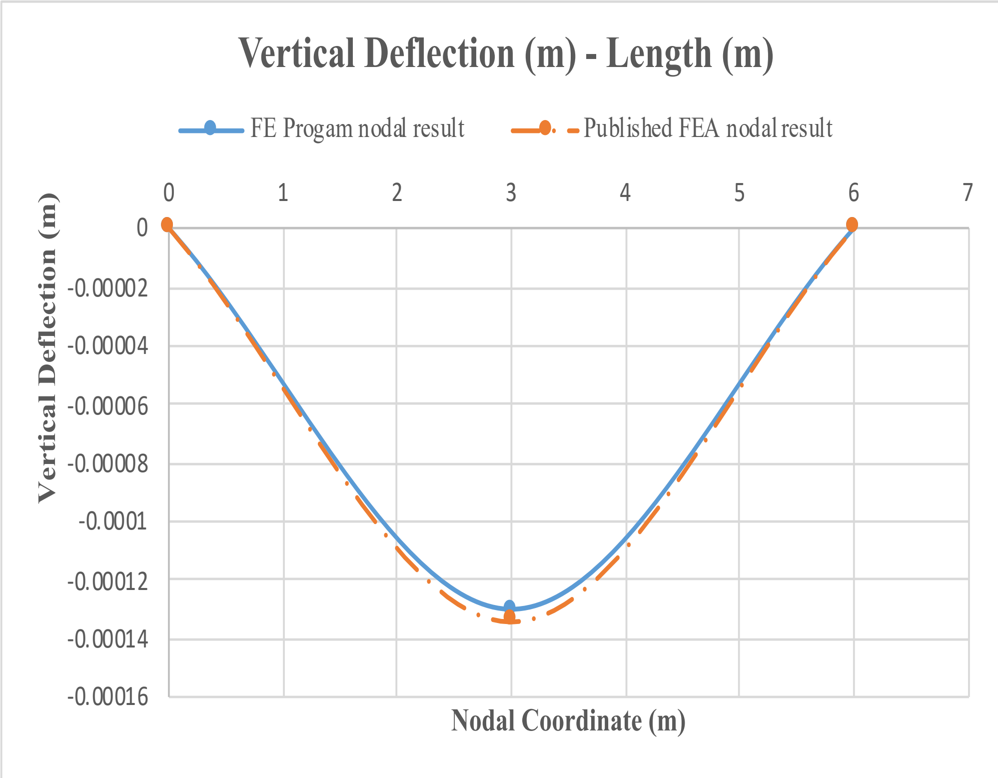

Figure 5.9: Example (2) two finite elements solution for vertical deflection along beam length at nodal points.

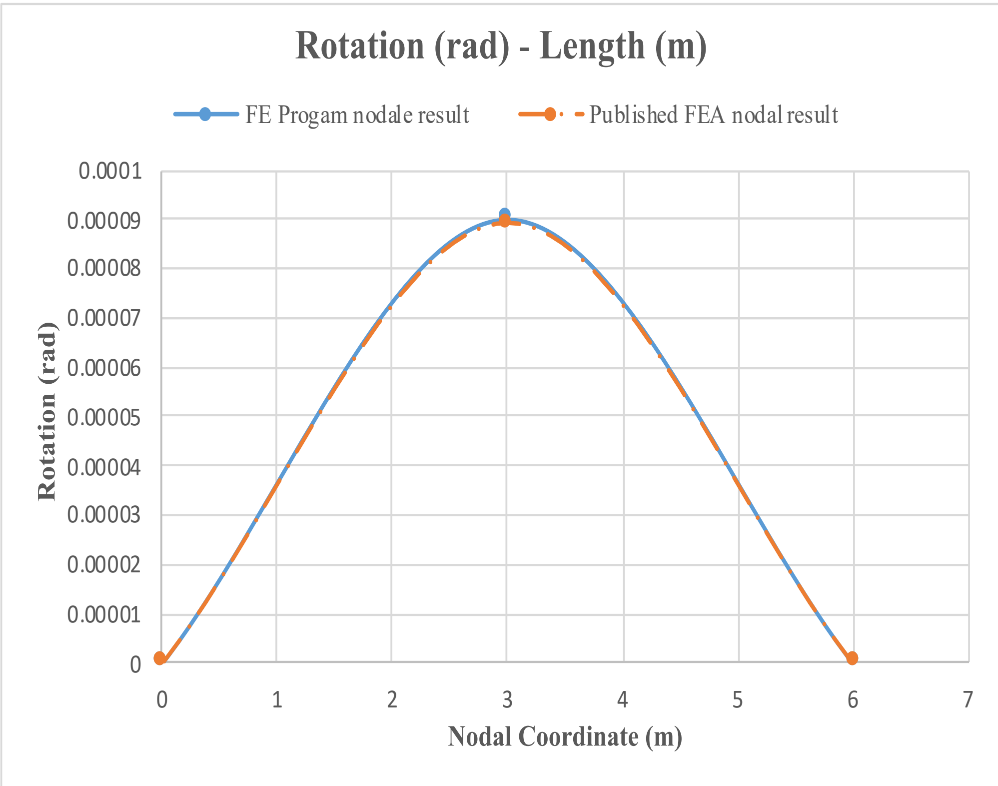

Figure 5.10: Example (2) two finite elements solution for rotation along beam length at nodal points.

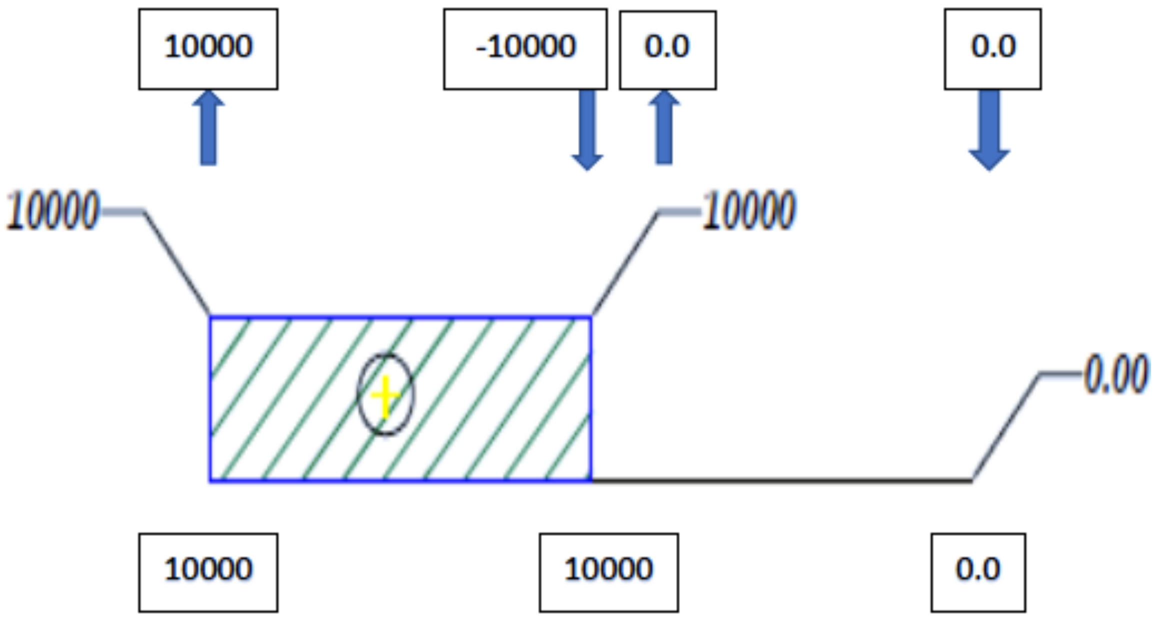

Figure 5.11: Example (2) two finite elements solution for shear force along beam length at nodal points.

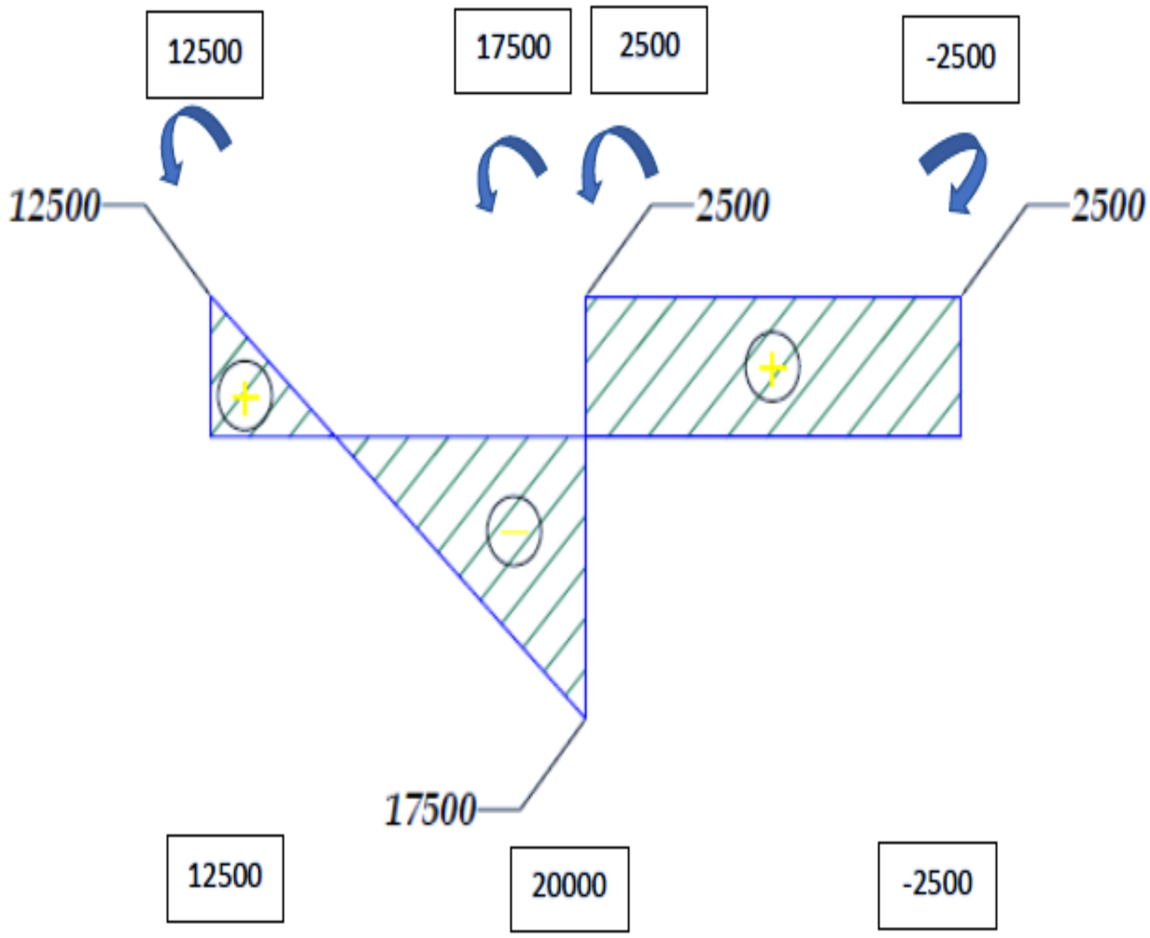

Figure 5.12: Example (2) two finite elements solution for bending moment along beam length at nodal points.

5.2.3 Example (3): Overhanging Beam Subjected to UDL.

A beam shown in Figure 5.13 is wide-flange $W310 \times 52$ with a cross sectional area of 6650 mm^2 and depth of 317 mm. the second moment of area of cross-section $I = 118.6 \times 10^6$ mm^4 the beam is subjected to a uniformly distributed loads 25,000 N/m. the modules of elasticity of the beam $E = 200$ GPa. Determine the vertical displacement at node (3) and the rotation at node (2) and (3). Also compute the reaction forces and moment at node (1) and (2). Example 4.4 Saeed Moaveni [17].

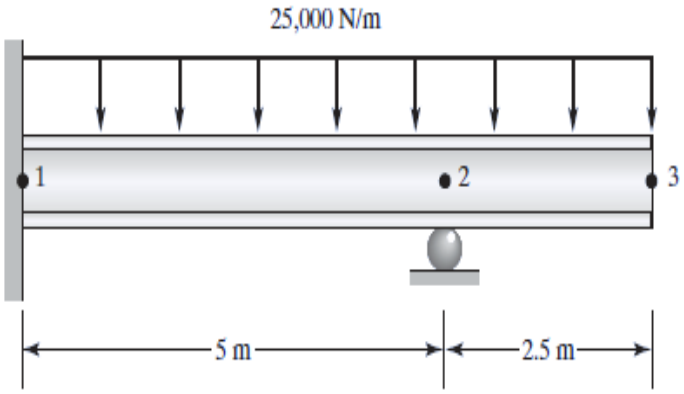

Figure 5.13: Overhanging beam subjected to UDL.

For two finite elements solution for nodal variables at nodal points, the program results obtained are in good agreement with those obtained by Saeed Moaveni, as shown in (Table 5.10, Table 5.11, Table 5.12, Table 5.13, Figure 5.14, Figure 5.15, Figure 5.16, and Figure 5.17).

Table 5.10: Example (3) two elements solution for vertical deflection at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Present Difference**
		w_i^e(m)	w_i^e(m)	w_i^e (%)
Node (1)	0	0	0	0
Node (2)	5	0	0	0
Node (3)	7.5	-0.00858	-0.0085772	0.03265 %

Table 5.11: Example (3) two elements solution for rotation at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Present Difference**
		θ_i^e(rad)	θ_i^e(rad)	θ_i^e (%)
Node (1)	0	0	0	0
Node (2)	5	-0.00137	-0.001372	0.14757 %
Node (3)	7.5	-0.00412	-0.004117	0.07286 %

Table 5.12: Example (3) two elements solution for shear force at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result** F_i^e(N)	**Published FEA Result** F_i^e(N)	**Present Difference** F_i^e (%)
Node (1)	0	54,687	54,687	0
Node (2)	5	132,812.6	132,814	0.00105 %
Node (3)	7.5	0	0	0

Table 5.13: Example (3) two elements solution for bending at nodal points.

Nodal Number	**Nodal coordinate (m)**	**Program nodal result** M_i^e(N.m)	**Published FEA nodal result** M_i^e(N.m)	**Present Difference** M_i^e (%)
Node (1)	0	39,062	39,062	0
Node (2)	5	0	0	0
Node (3)	7.5	0	0	0

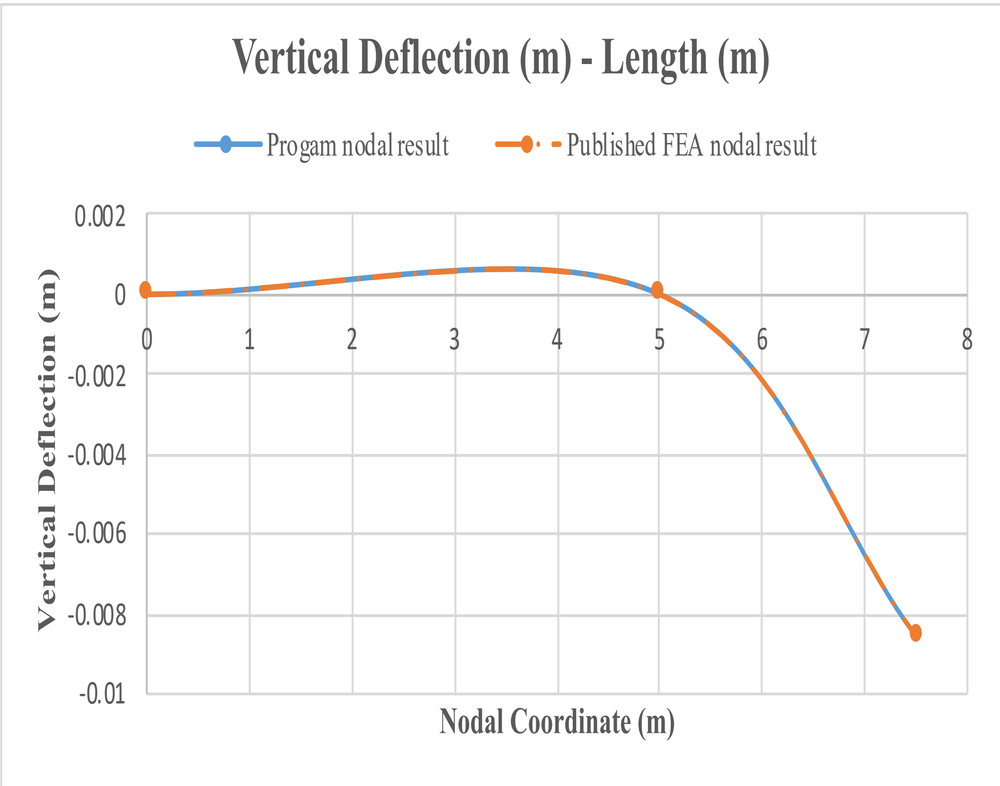

Figure 5.14: Example (3) two finite elements solution for vertical deflection along beam length at nodal points.

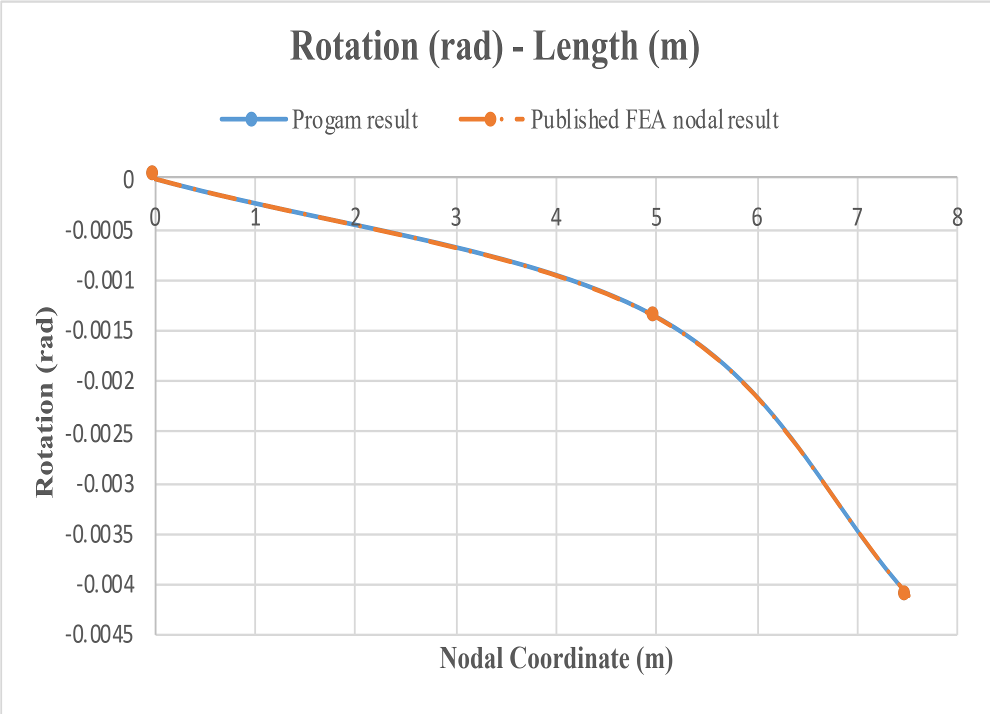

Figure 5.15: Example (3) two finite elements solution for rotation along beam length at nodal points.

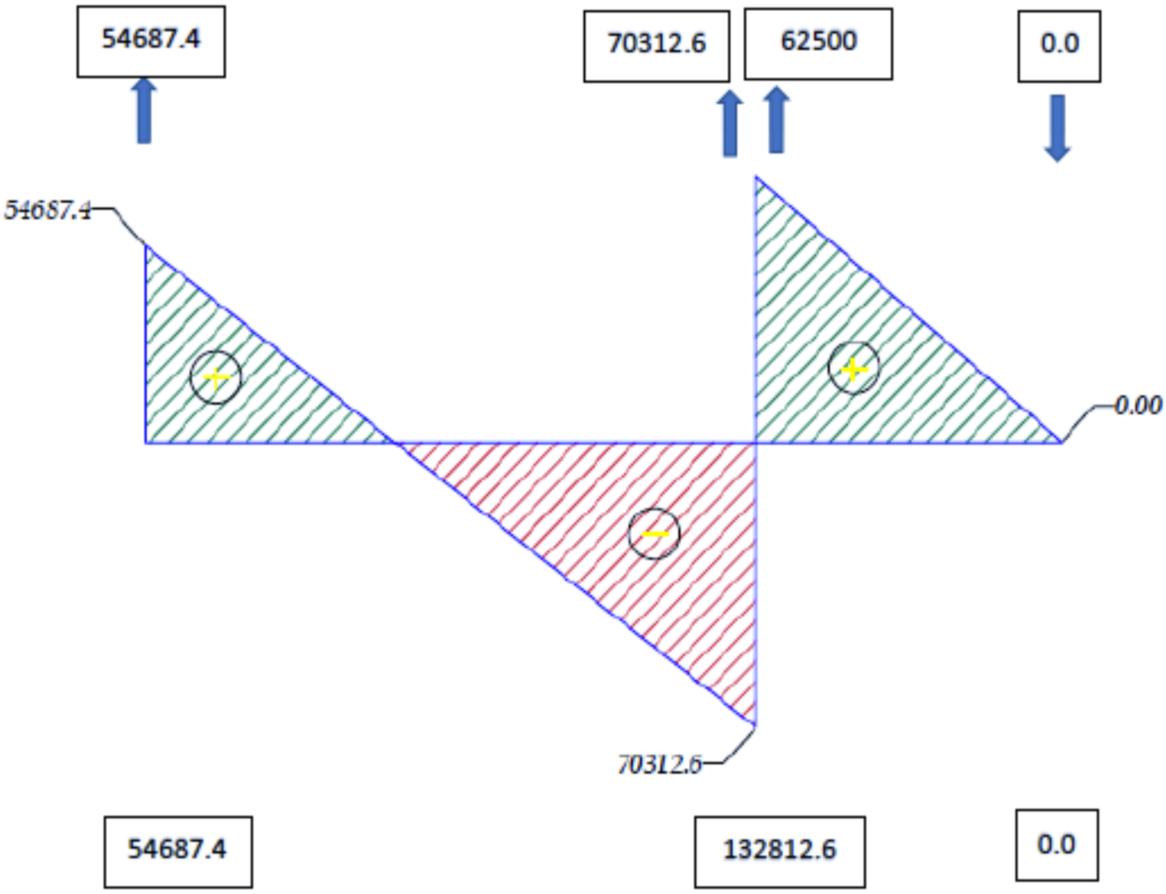

Figure 5.16: Example (3) two finite elements solution for shear force along beam length at nodal points.

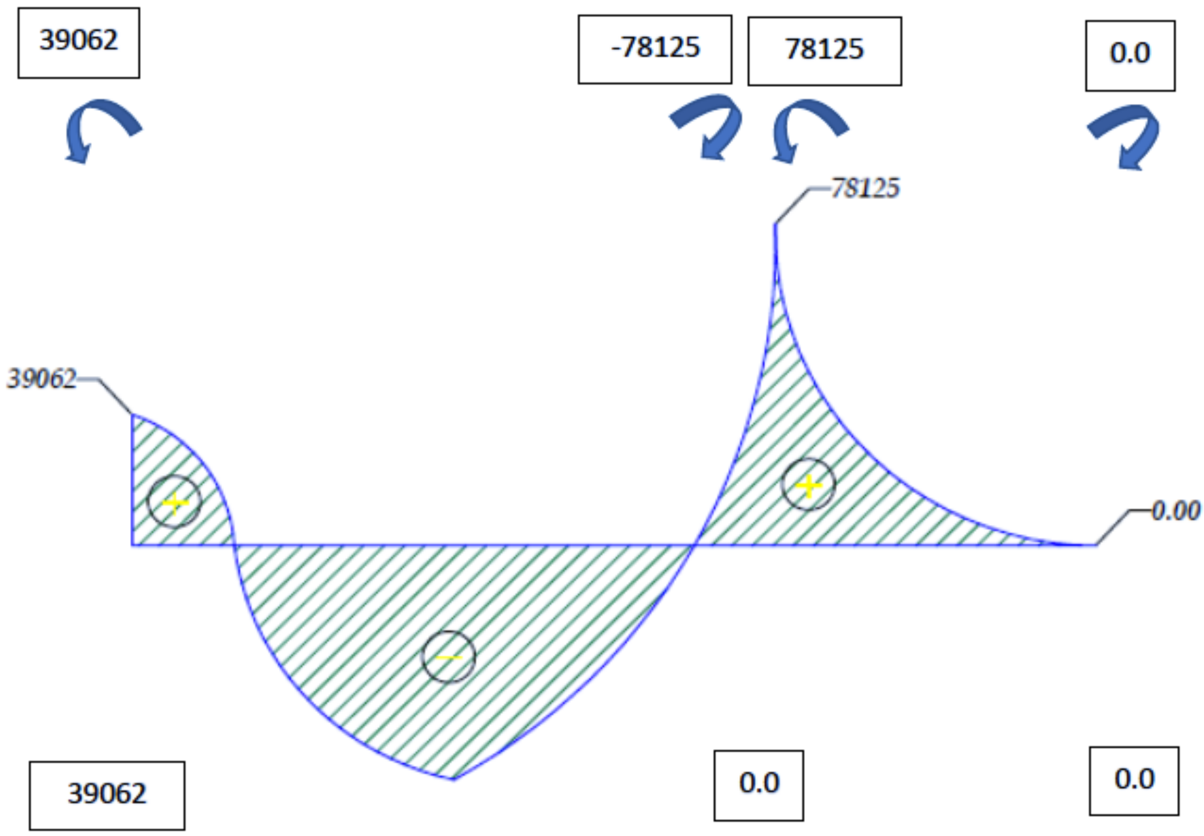

Figure 5.17: Example (3) two finite elements solution for bending moment along beam length at nodal points.

5.2.4 Example (4): Simple Supported Beam with varying Cross-Section Subjected to Concentrated Loads.

For the three-segment beam depicted in Figure 5.18 with varying cross-section and points loads. Determine the transverse nodal displacement at points A, B, C, and D and the reactions at end points. the modules of elasticity of the beam $E = 200$ GPa and the second moment of area of cross-sections $I_{AB} = 5 \times 10^{-6}$ m^4, $I_{BC} = 2.5 \times 10^{-6}$ m^4, $I_{CD} = 6.25 \times 10^{-7}$ m^4. Example 6.1 Khameel Bayo Mustapha [15].

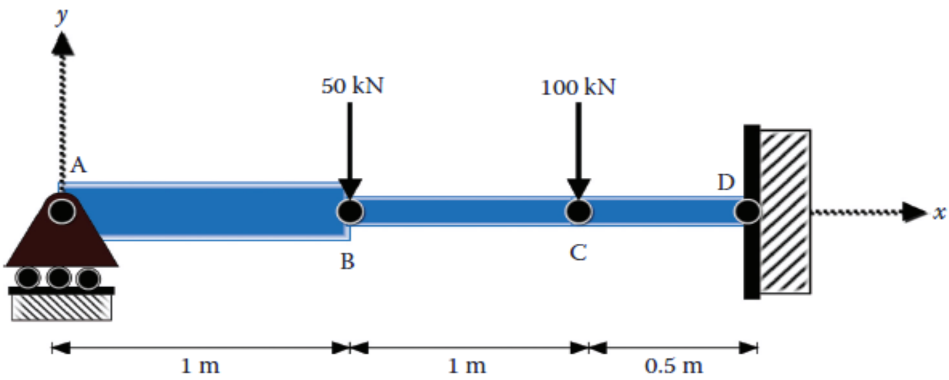

Figure 5.18: Simple supported beam with varying cross-section subjected to concentrated loads.

For three finite elements solution for nodal variables at nodal points, the program results obtained are in good agreement with those obtained by Khameel Bayo Mustapha, as shown in (Table 5.14, Table 5.15, Table 5.16, Table 5.17, Figure 5.19, Figure 5.20, Figure 5.21, and Figure 5.22).

Table 5.14: Example (4) three elements solution for vertical deflection at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Present Difference**
		w_i^e(m)	w_i^e(m)	w_i^e (%)
Node (1)	0	0	0	0
Node (2)	1	-0.03004	-0.03004	0
Node (3)	2	-0.01864	-0.01864	0
Node (4)	2.5	0	0	0

Table 5.15: Example (4) three elements solution for rotation at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Present Difference**
		θ_i^e(rad)	θ_i^e(rad)	θ_i^e (%)
Node (1)	0	-0.03586	-0.03586	0
Node (2)	1	-0.01842	-0.01842	0
Node (3)	2	0.03618	0.03618	0
Node (4)	2.5	0	0	0

Table 5.16: Example (4) three elements solution for shear force at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Present Difference**
		F_i^e(N)	F_i^e(N)	F_i^e (%)
Node (1)	0	34,868.42	34,868	0.0012 %
Node (2)	1	-50,000	-50,000	0
Node (3)	2	-100,000	-100,000	0
Node (4)	2.5	115,131.58	115,131.58	0

Table 5.17: Example (4) three elements solution for bending at nodal points.

Nodal Number	**Nodal coordinate (m)**	**Program nodal result**	**Published FEA nodal result**	**Present Difference**
		M_i^e(N.m)	M_i^e(N.m)	M_i^e (%)
Node (1)	0	0	0	0
Node (2)	1	0	0	0
Node (3)	2	0	0	0
Node (4)	2.5	-37,828.95	-37,829	0.00013 %

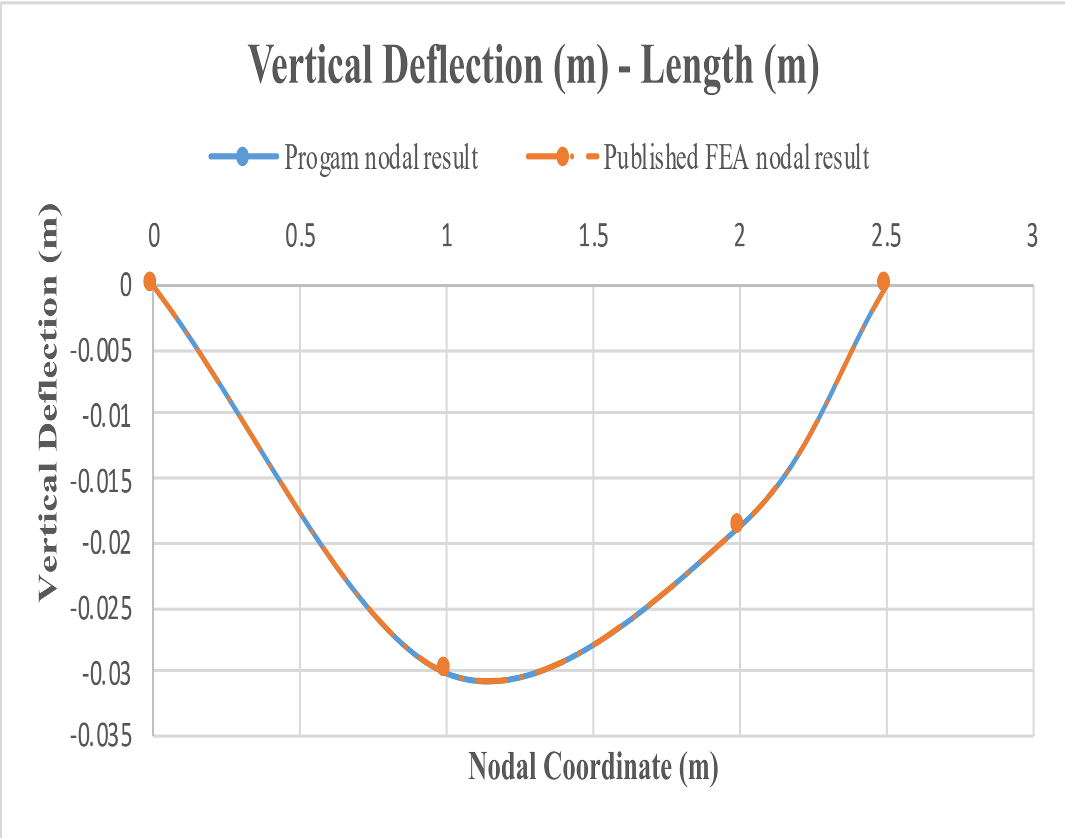

Figure 5.19: Example (4) three finite elements solution for vertical deflection along beam length at nodal points.

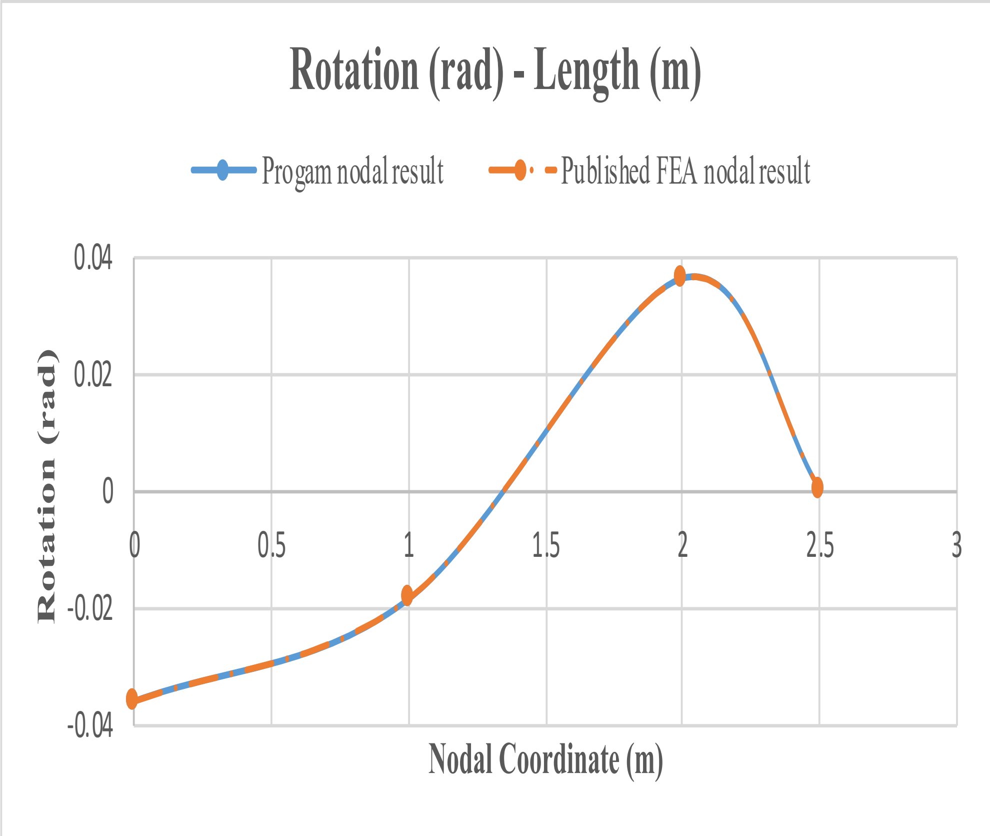

Figure 5.20: Example (4) three finite elements solution for rotation along beam length at nodal points.

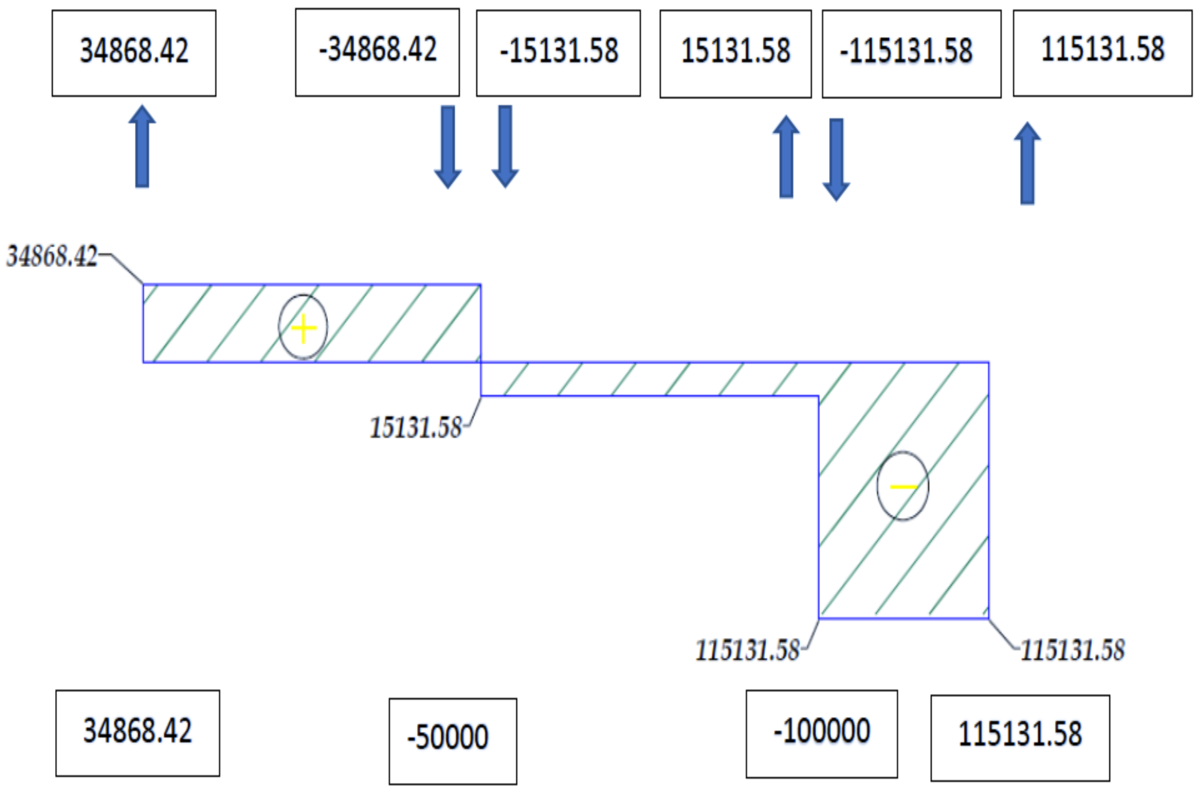

Figure 5.21: Example (4) three finite elements solution for shear force along beam length at nodal points.

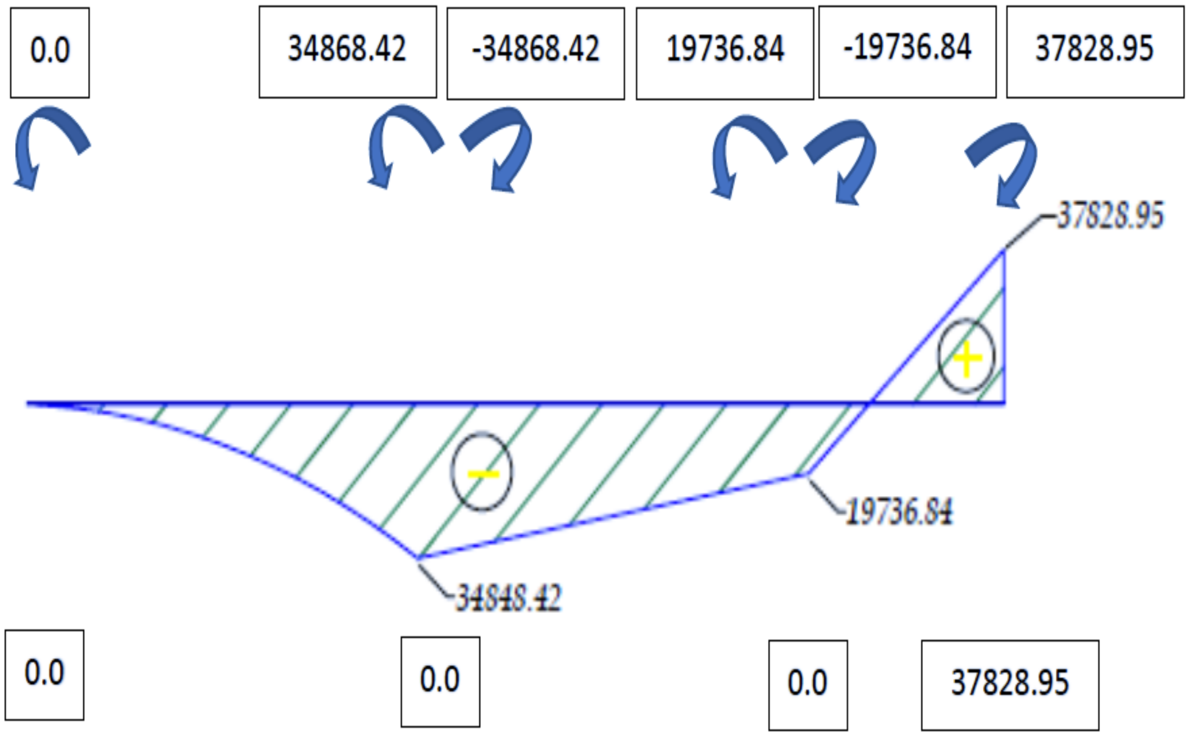

Figure 5.22: Example (4) three finite elements solution for bending moment along beam length at nodal points.

5.2.5 Example (5): Cantilever Beam Subjected to UDL and Concentrated Loads.

A wooden beam is loaded as shown in Figure 5.23. The wood material has the modules of elasticity of the beam $E = 12$ GPa and the second moment of area of cross-section $= 10.67 \times 10^{-4}$ m^4. Determine the deflection and slope at points, also determine the reaction force and bending moment at points A, B, C, and D. Example 6.2 Khameel Bayo Mustapha [15].

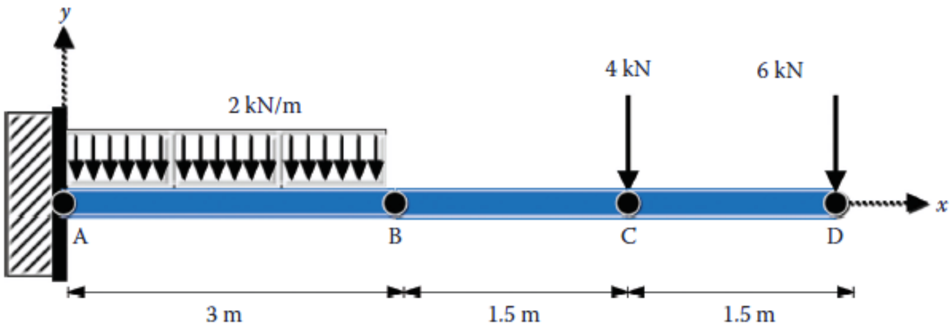

Figure 5.23: Cantilever beam subjected to UDL and concentrated loads.

For three finite elements solution for nodal variables at nodal points, the program results obtained are in good agreement with those obtained by

Khameel Bayo, as shown in (Table 5.18, Table 5.19, Table 5.20, Table 21, Figure 5.24, Figure 5.25, Figure 5.26, and Figure 5.27).

Table 5.18: Example (5) three elements solution for vertical deflection at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Present Difference**
		w_i^e(m)	w_i^e(m)	w_i^e (%)
Node (1)	0	0	0	0
Node (2)	3	-0.01705	-0.017	0.2941 %
Node (3)	4.5	-0.03348	-0.0335	0.0597 %
Node (4)	6	-0.05166	-0.0517	0.0773 %

Table 5.19: Example (5) three elements solution for rotation at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Present Difference**
		θ_i^e(rad)	θ_i^e(rad)	θ_i^e (%)
Node (1)	0	0	0	0
Node (2)	3	-0.00984	-0.0098	0.4081 %
Node (3)	4.5	-0.01177	-0.0118	0.2542 %
Node (4)	6	-0.0123	-0.0123	0

Table 5.20: Example (5) three elements solution for shear force at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Present Difference**
		F_i^e(N)	F_i^e(N)	F_i^e (%)
Node (1)	0	16,000	16,000	0
Node (2)	3	0	0	0
Node (3)	4.5	-4,000	-4,000	0
Node (4)	6	-6,000	-6,000	0

Table 5.21: Example (5) three elements solution for bending at nodal points.

Nodal Number	Nodal coordinate (m)	Program nodal result	Published FEA nodal result	Present Difference
		M_i^e(N.m)	M_i^e(N.m)	M_i^e (%)
Node (1)	0	63,000	63,000	0
Node (2)	3	0	0	0
Node (3)	4.5	0	0	0
Node (4)	6	0	0	0

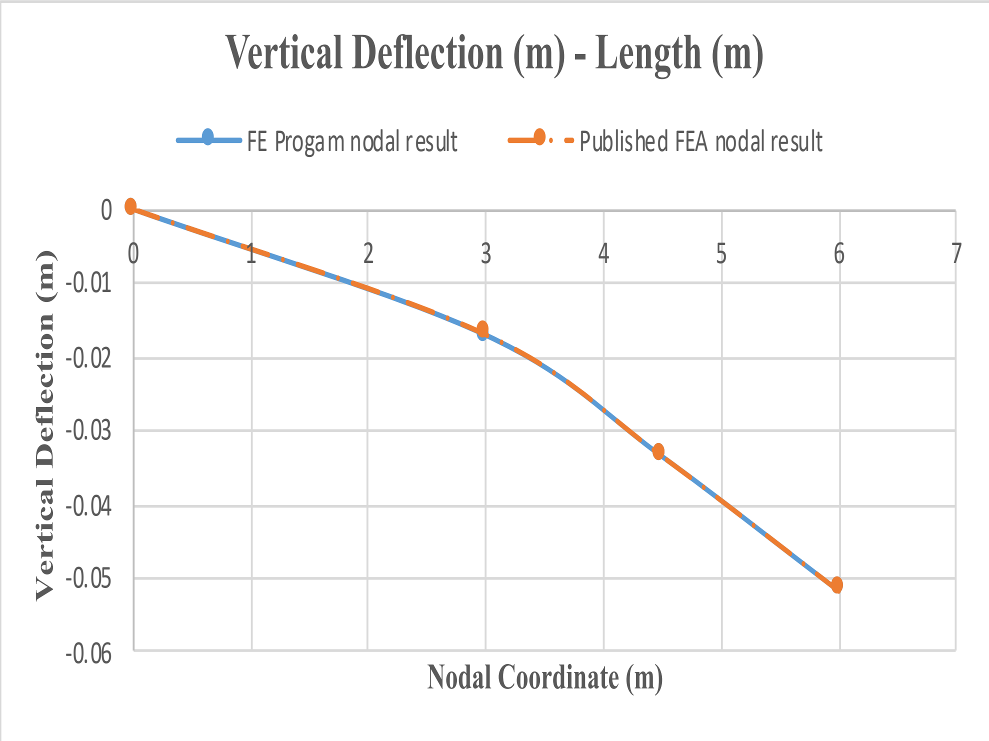

Figure 5.24**:** Example (5) three finite elements solution for vertical deflection along beam length at nodal points.

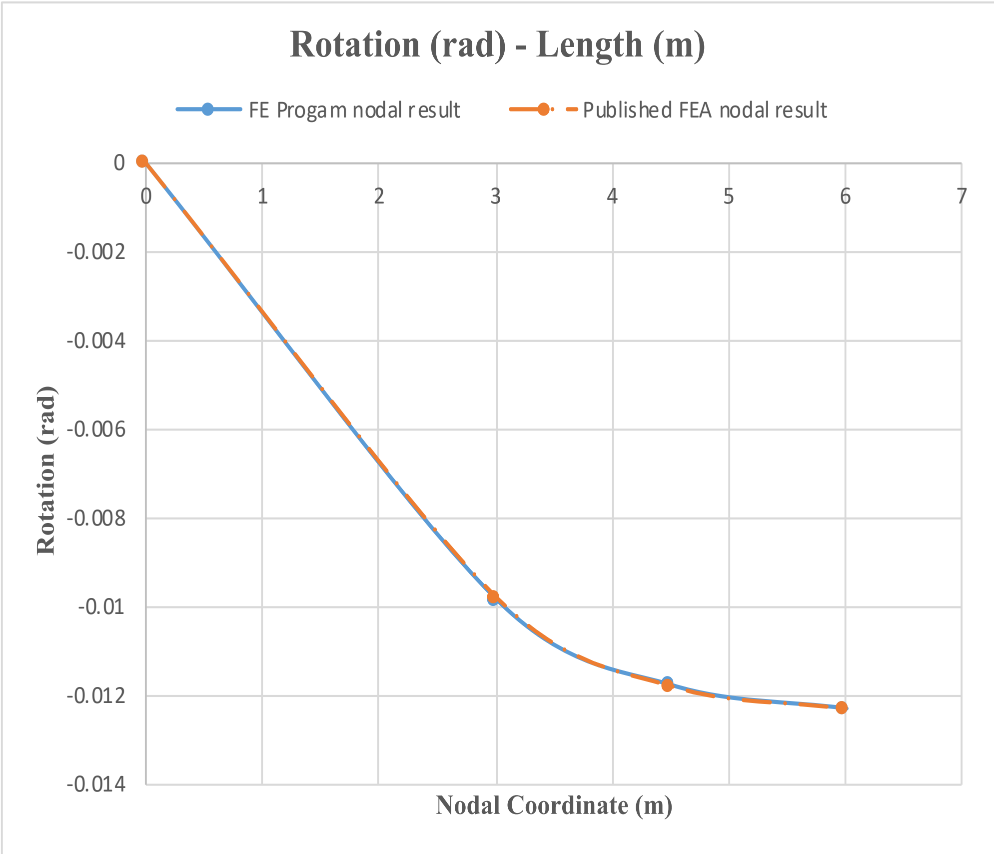

Figure 5.25: Example (5) three finite elements solution for rotation along beam length at nodal points.

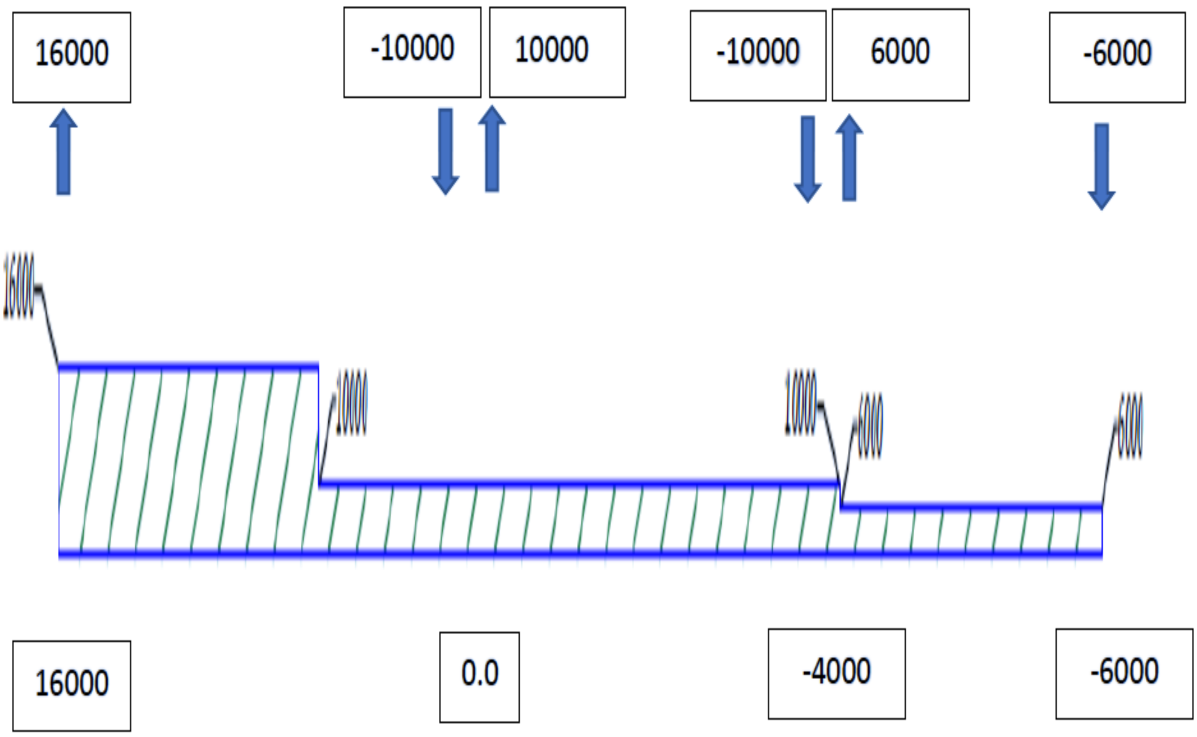

Figure 5.26: Example (5) three finite elements solution for shear force along beam length at nodal points.

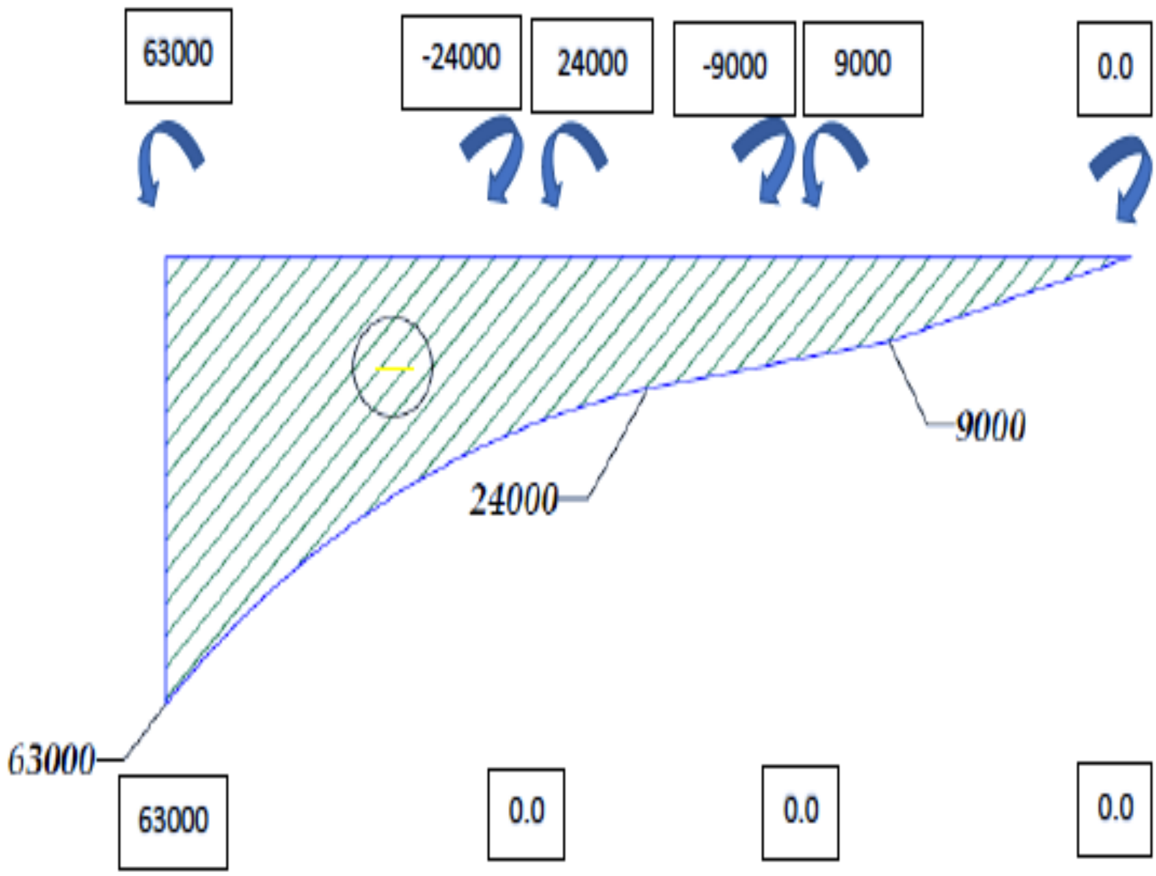

Figure 5.27: Example (5) three finite elements solution for bending moment along beam length at nodal points.

5.2.6 Example (6): Continuous Beam Subjected to UDL and Concentrated Loads.

A continuous beam shown in Figure 5.28. Obtain the deflection of the beam using the beam element just described in Figure 5.29. Assume the beam flexural rigidity $EI = 1$. Example 4.7 P. Seshu [30].

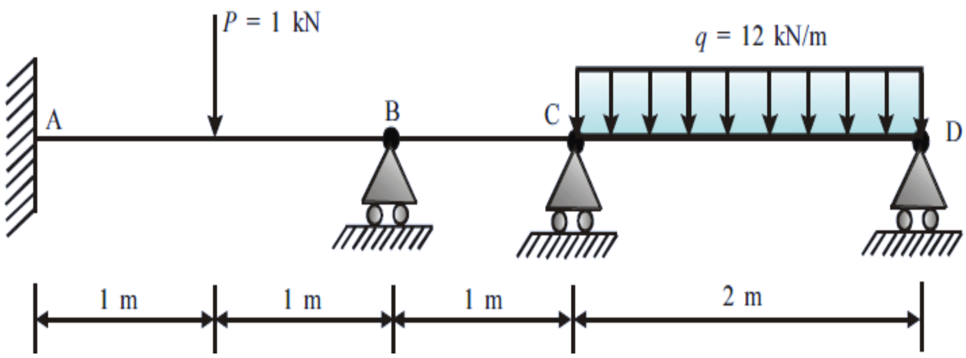

Figure 5.28: Continuous beam subjected to UDL and concentrated loads.

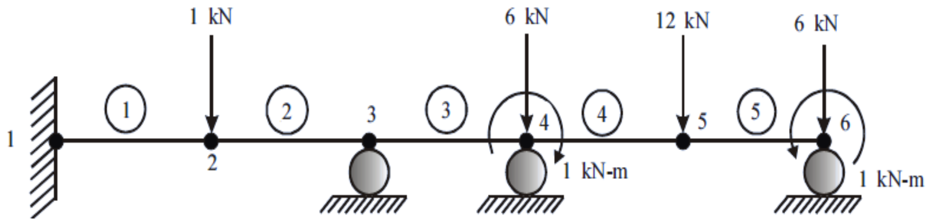

Figure 5.29: Finite element modal for Example (6)

For five finite elements solution for deflection at nodal points, the program results obtained are in good agreement with those obtained by P. Seshu, as shown in (Table 5.22. Table 5.23, Figure 5.30, and Figure 5.31).

Table 5.22: Example (6) five elements solution for vertical deflection at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Present Difference**
		w_i^e(m)	w_i^e(m)	w_i^e (%)
Node (1)	0	0	0	0
Node (2)	1	-0.15697	-0.157	0.0191 %
Node (3)	2	0	0	0
Node (4)	3	0	0	0
Node (5)	4	-1.47198	-1.472	0.0013 %
Node (6)	5	0	0	0

Table 5.23: Example (6) five elements solution for rotation at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Present Difference**
		θ_i^e(rad)	θ_i^e(rad)	θ_i^e (%)
Node (1)	0	0	0	0
Node (2)	1	-0.1153	-0.1153	0
Node (3)	2	0.46121	0.4612	0.0021 %
Node (4)	3	-1.25862	-1.2586	0.0015 %
Node (5)	4	-0.34267	-0.3427	0.0087 %
Node (6)	5	2.62931	2.62931	0

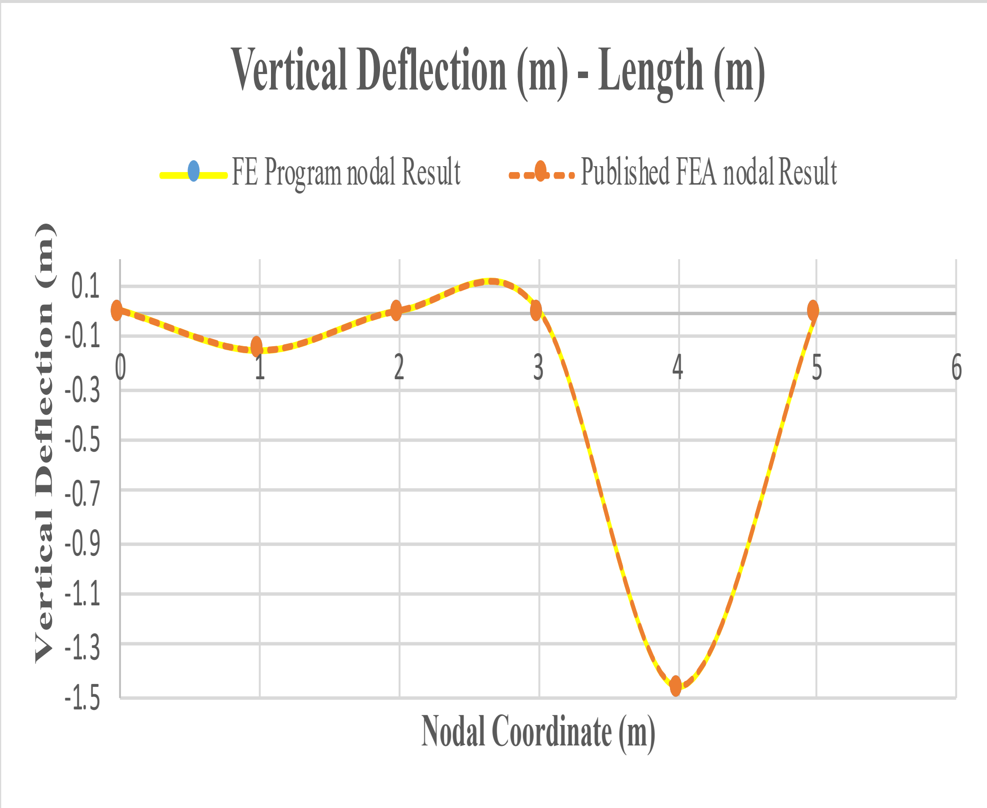

Figure 5.30: Example (6) five finite elements solution for vertical deflection along beam length at nodal points.

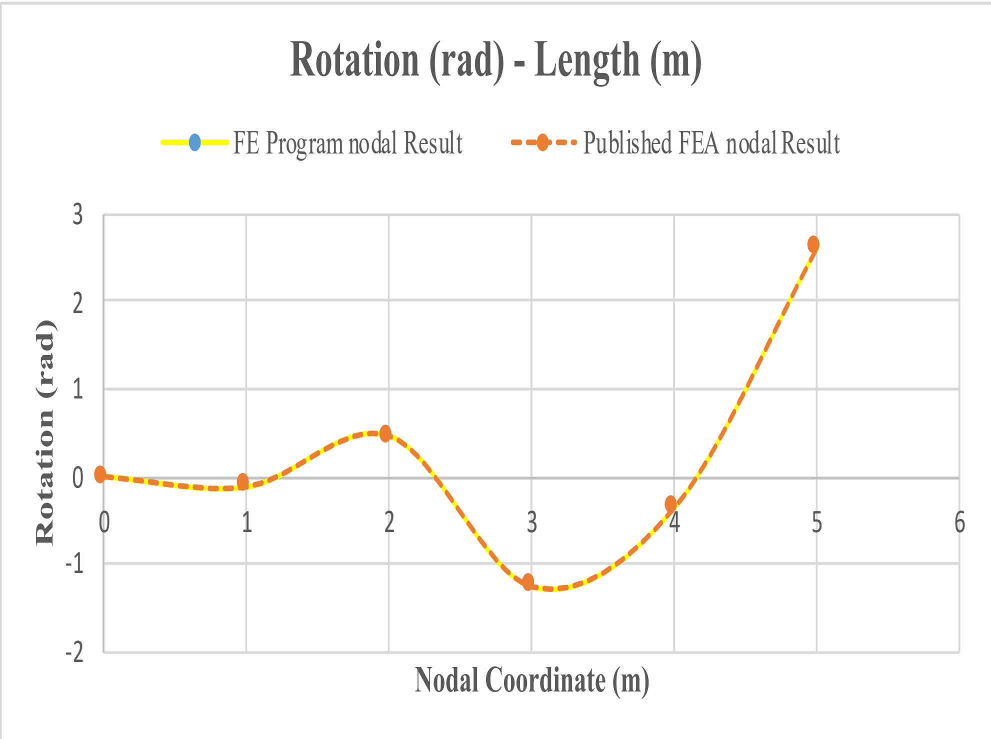

Figure 5.31: Example (6) five finite elements solution for rotation along beam length at nodal points.

5.2.7 Example (7): Beam with Internal Hinge Subjected to UDL.

Determine the slope at node (2) and the deflections and slope at node (3) for the beam with internal hinge located at node (3), the beam is loaded as shown in Figure 5.32. Node (1) and (4) are fixed and there is a knife-edge support at node (2). the modules of elasticity of the beam $E = 210$ GPa and the second moment of area of cross-section $I = 2 \times 10^{-4}$ m^4. Example 4.11 Daryl L. Logon [2].

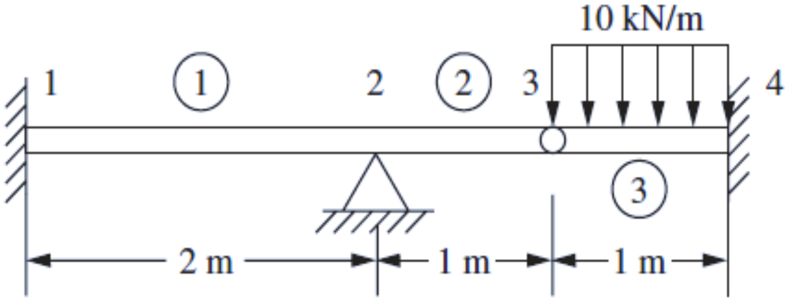

Figure 5.32: Beam with internal hinge and uniformly distributed loading.

For three finite elements solution for deflection at nodal points, the program results obtained are in good agreement with those obtained by Daryl L. Logon, as shown in (Table 5.24, Table 5.25, Figure 5.33, and Figure 5.34).

Table 5.24: Example (7) three elements solution for vertical deflection at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Present Difference**
		w_i^e(m)	w_i^e(m)	w_i^e (%)
Node (1)	0	0	0	0
Node (2)	2	0	0	0
Node (3)	3	-2.126 × 10^{-5}	-2.126 × 10^{-5}	0
Node (4)	4	0	0	0

Table 5.25: Example (7) three elements solution for rotation at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Present Difference**
		θ_i^e(rad)	θ_i^e(rad)	θ_i^e (%)
Node (1)	0	0	0	0
Node (2)	2	-1.276× 10^{-5}	-1.276× 10^{-5}	0
Node (3)	3	-2.693× 10^{-5}	-2.693× 10^{-5}	0
Node (4)	4	0	0	0

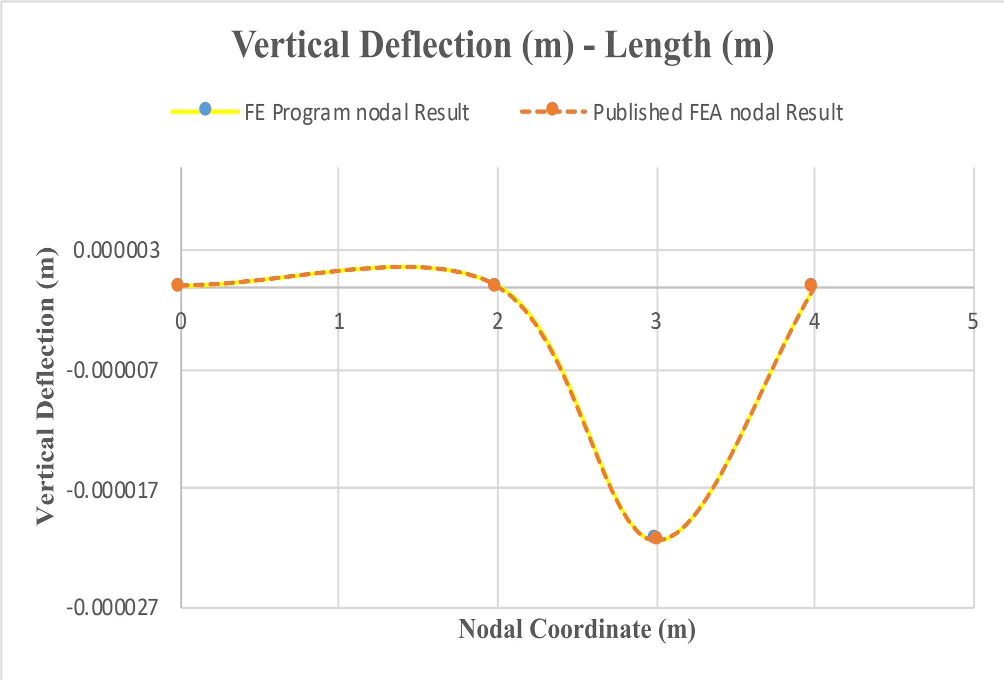

Figure 5.33: Example (7) three finite elements solution for vertical deflection along beam length at nodal points.

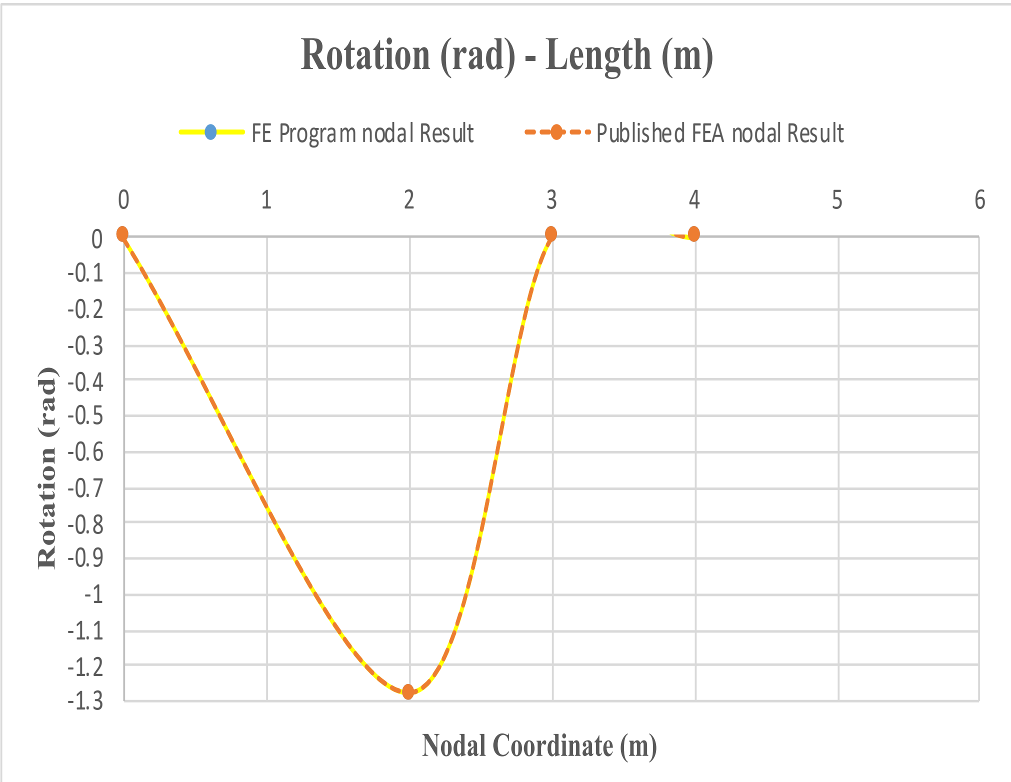

Figure 5.34: Example (7) three finite elements solution for rotation along beam length at nodal points.

5.2.8 Example (8): Cantilever Beam Subjected to Linearly varying Distributed Load.

Cantilever beam subjected to linearly varying distributed load $q_0 = 24$ kN/m and point load $F_0 = 60$ kN as shown in Figure 5.35. The module of elasticity of a beam $E = 200 \times 10^6$ kN/m^2 and the second moment of area of cross-section $= 29 \times 10^6$ mm^4. Using two elements for solution the length of beam $L = 3$m. Determine deflection field and bending moment. Example 5.2.1 J.N. Reddy [4].

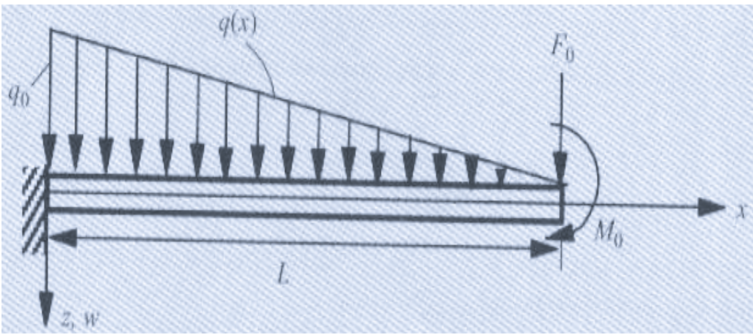

Figure 5.35: Cantilever beam with linearly varying distributed loads.

For two finite elements solution for deflection at nodal points, the program results obtained are in good agreement with those obtained by J.N. Reddy, as shown in (Table 5.26, Table 5.27, Figure 5.36, and Figure 5.37).

Table 5.26: Example (8) two elements solution for vertical deflection at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Present Difference**
		w_i^e(m)	w_i^e(m)	w_i^e (%)
Node (1)	0	0	0	0
Node (2)	1.5	-0.03337	-0.0333	0.2102 %
Node (3)	3	-0.010428	-0.01043	0.0191 %

Table 5.27: Example (8) two elements solution for rotation at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Present Difference**
		θ_i^e(rad)	θ_i^e(rad)	θ_i^e (%)
Node (1)	0	0	0	0
Node (2)	1.5	0.03929	0.0393	0.0254 %
Node (3)	3	-0.05121	-0.0512	0.0195 %

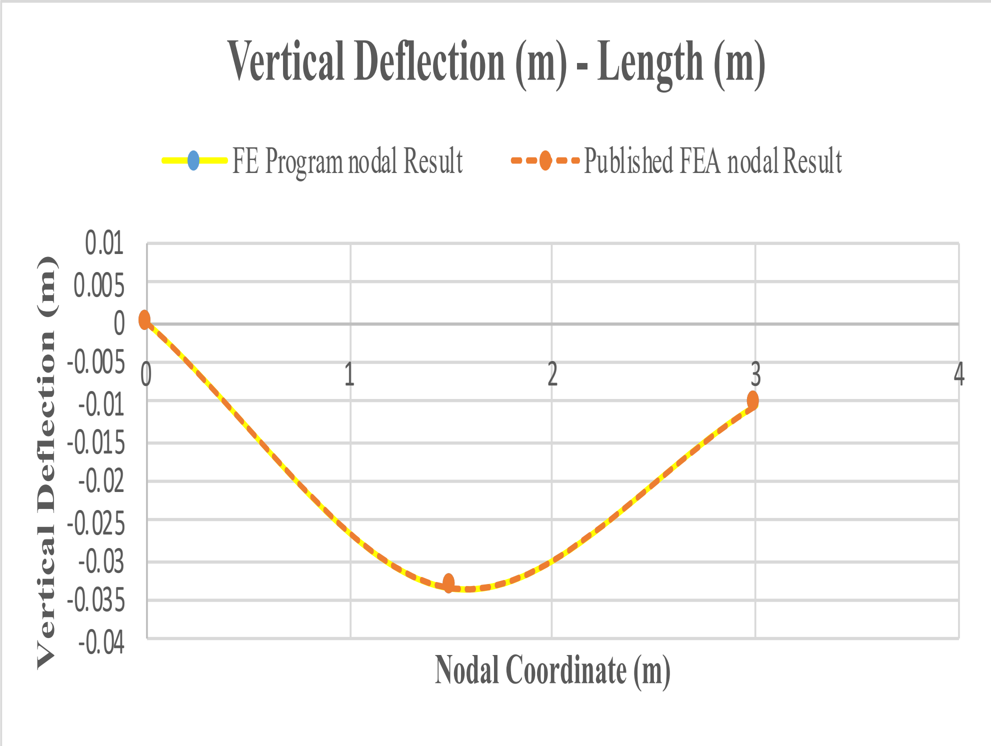

Figure 5.36: Example (8) two finite elements solution for vertical deflection along beam length at nodal points.

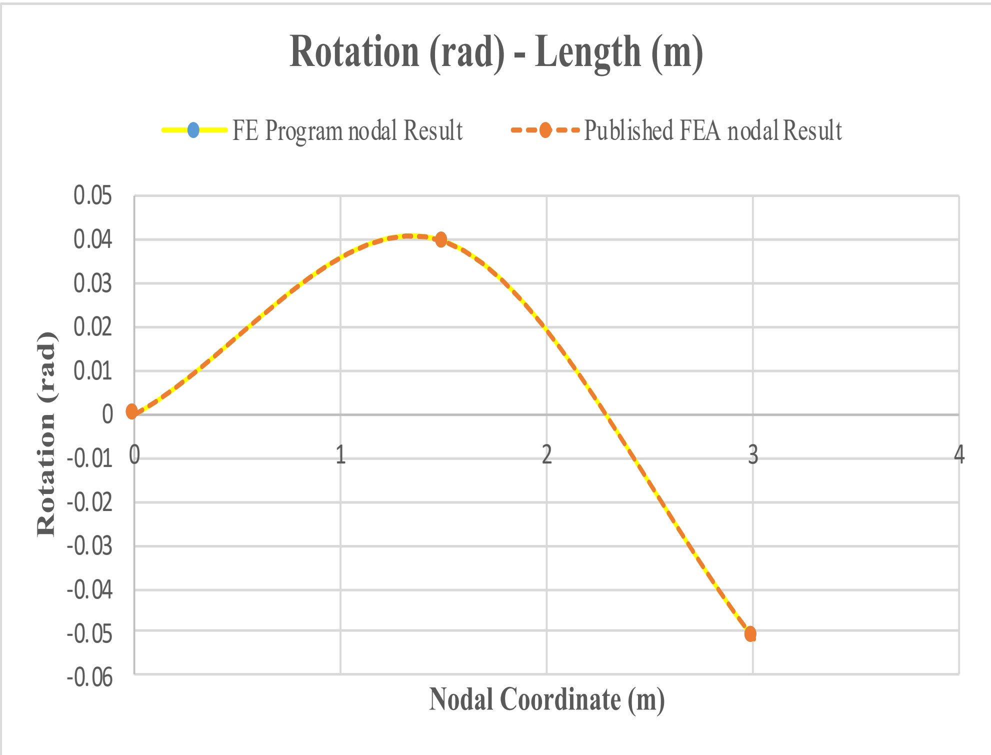

Figure 5.37: Example (8) two finite elements solution for rotation along beam length at nodal points.

5.2.9 Example (9): Beam with Concentrated Load, UDL and Linearly Varying Distributed Load.

A beam as shown in Figure 5.38, the beam is made of steel with modules of elasticity $E = 30 \times 10^{-6}$ psi, and the second moment of area of cross-section $= 4.5 \text{ in}^4$. Find the transverse deflection using Euler-Bernoulli beam finite element. Example 5.2.3 J.N. Reddy [4].

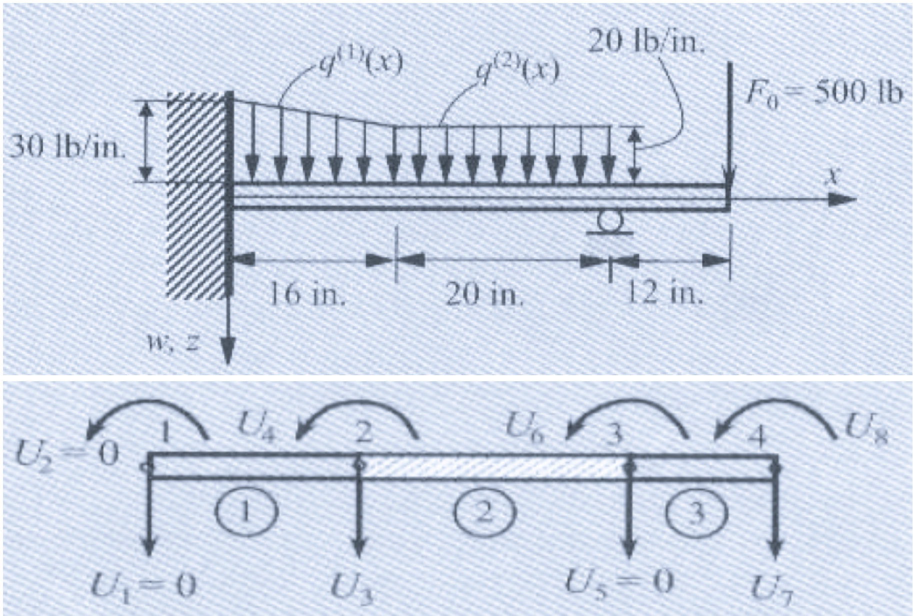

Figure 5.38: Beam with concentrated load, UDL, and linearly varying distributed loads.

For three finite elements solution for deflection at nodal points, the program results obtained are in good agreement with those obtained by J.N. Reddy, as shown in (Table 5.28, Table 5.29 Figure 5.39, and Figure 5.40).

Table 5.28: Example (9) three elements solution for vertical deflection at nodal points.

Nodal Number	Nodal Coordinate (in)	Program Result	Published FEA Result	Present Difference
		w_i^e(in)	w_i^e(in)	w_i^e (%)
Node (1)	0	0	0	0
Node (2)	16	0.00036	-0.000322	11.8012 %
Node (3)	36	0	0	0
Node (4)	48	-0.00519	0.00515	0.7767 %

Table 5.29: Example (9) three elements solution for rotation at nodal points.

Nodal Number	**Nodal Coordinate (m)**	**Program Result**	**Published FEA Result**	**Present Difference**
		θ_i^e(rad)	θ_i^e(rad)	θ_i^e (%)
Node (1)	0	0	0	0
Node (2)	16	0.00006	0.0000593	2.9159 %
Node (3)	36	-0.00025	-0.0002513	0.5173 %
Node (4)	48	-0.00052	-0.000518	0.3861 %

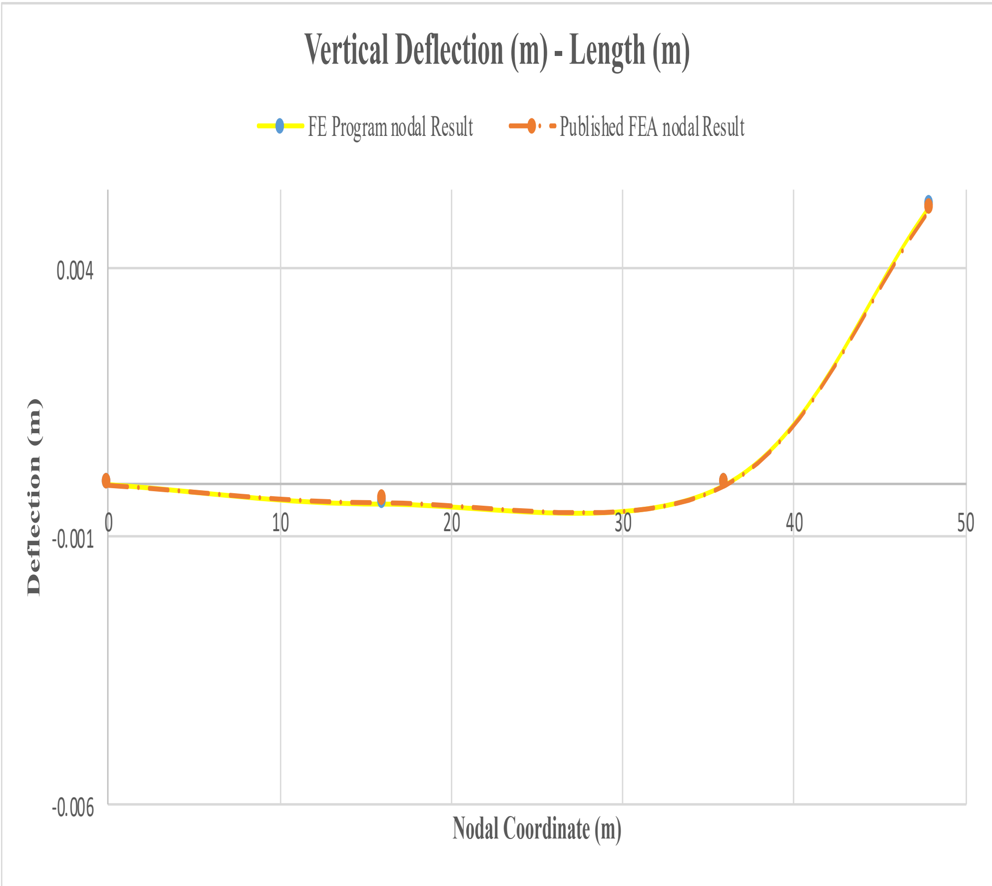

Figure 5.39: Example (9) three finite elements solution for vertical deflection along beam length at nodal points.

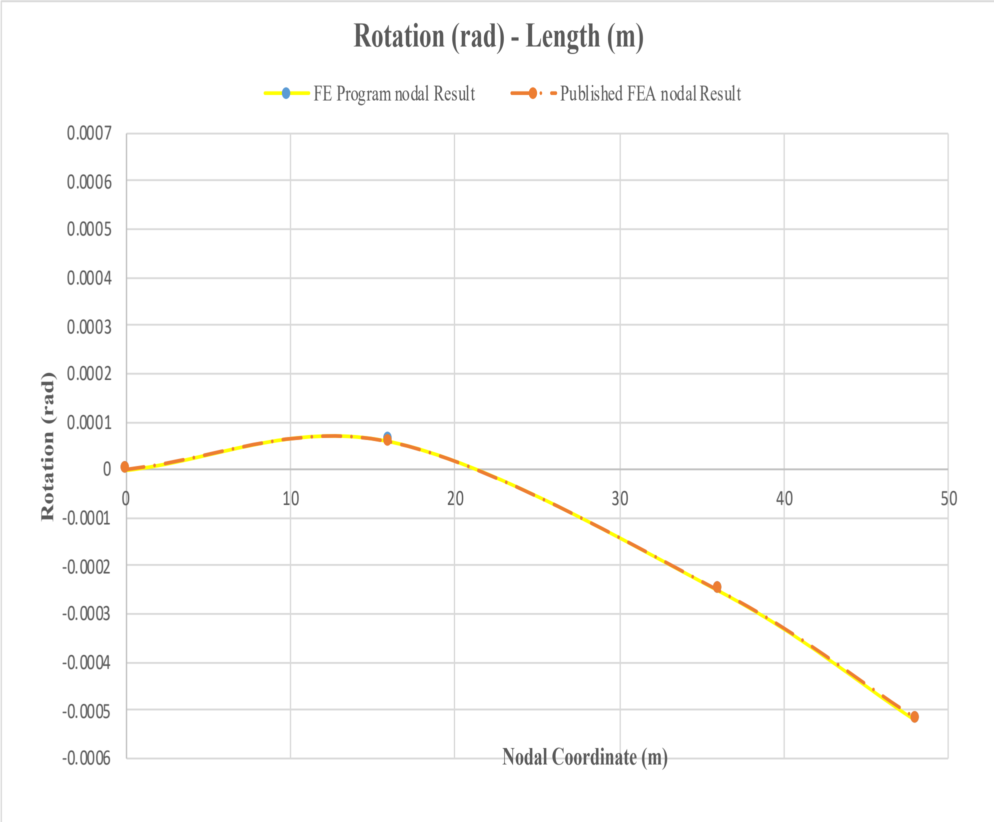

Figure 5.40: Example (9) three finite elements solution for rotation along beam length at nodal points.

5.3 Discussion of The Results

In this research a computer program coded using MATLAB R2019b, it was developed for the linear static finite element analysis of thin straight beams of two-nodes The approach adopted to drive the element equations based on the principle of minimum potential energy and the generalized coordinate approach.

In order to compared the developed code outputs, nine different beams with different loading and boundary conditions will be conducted for this purpose include a cantilever beam, fixed beam, overhanging beam, simple supported beam, continuous beam, and beam with internal hinge

In Example (1): Cantilever Beam Subjected to UDL.
Example (1): For One Element Solution at Nodal Points
For one finite element solution for deflection at nodal points using the same two-nodes beam element. The program results obtained are in good agreement with the exact solution result and FEA result obtained by Daryl L. Logon [2]. Hence, the location of maximum nodal deflection values is presented in Table 5.30, as follows:

Table 5.30: Discussion on the location of maximum values of Example (1).

No.	Variable Name	Node Number	Nodal Coordinate (in)	Comment on Location
1	Vertical deflection w_i^e	Node (2)	$x = 100$ in	At free end.
2	Rotation θ_i^e	Node (2)	$x = 100$ in	At free end.

The reason that finite element solution for nodal deflection values is correct and identically match the exact beam solution values, because the element nodal forces were calculated on the basis on the assumed cubic (third order) displacement polynomial field within each beam element [2].

Example (1): For One Element Solution at Nodal Points and at Mid-Length. For one finite element solution for deflection at nodal points using the same two-nodes beam element. The program results obtained are in good agreement with the exact solution result and FEA result obtained by Daryl L. Logon [2]. Hence, the location of maximum nodal deflection values is presented in Table 5.31, as follows:

Table 5.31: Discussion on the location of maximum values of Example (1).

No.	Variable Name	Node Number	Nodal Coordinate (in)	Comment on Location
1	Vertical deflection w_i^e	Node (2)	$x = 100$ in	At free end.
2	Rotation θ_i^e	Node (2)	$x = 100$ in	At free end.

The finite element solution values of the deflection at other locations along the beam, are lower than the exact beam theory solution. this is always true for beams subjected to some form of distributed load that are modeled using the cubic displacement function. The exception of this results is at nodes, where the beam theory and finite element results are identical because of the work-equivalence concept used to replace the distributed load by statically equivalent nodal loads [2].

In the exact beam theory solution, the displacement evaluated using quartic (fourth order), while the finite element solution assumes a cubic displacement behavior in each beam under all load's conditions. However, the more element used in the model, the finite element solution converges to the actual one [2].

Example (1): For Two Elements Solution at Nodal Points
For two finite element solutions for deflection at nodal points using the same two-nodes beam element. the program results obtained are in good agreement with the exact solution result and FEA result obtained by Daryl L. Logon [2]. Hence, the location of maximum nodal deflection values is presented in Table 5.32, as follows:

Table 5.32: Discussion on the location of maximum values of Example (1).

No.	Variable Name	Node Number	Nodal Coordinate (in)	Comment on Location
1	Vertical deflection w_i^e	Node (2)	$x = 100$ in	At free end.
2	Rotation θ_i^e	Node (2)	$x = 100$ in	At free end.

In Example (2): The fixed-fixed beam subjected to concentrated loads.
The two finite elements solution for nodal variables at nodal points using the same two-nodes beam element. The program results obtained are in good agreement with those obtained by Daryl L. Logon [2]. Hence, the location of maximum nodal variables values is presented in Table 5.33, as follows:

Table 5.33: Discussion on the location of maximum values of Example (2).

No.	Variable Name	Node Number	Nodal Coordinate (m)	Comment on Location
1	Vertical deflection w_i^e, rotation θ_i^e, and bending moment M_i^e	Node (2)	$x = 3$ m	At the mid-span.
2	Shear force F_i^e	Node (1) Node (2)	$x = 0$ m $x = 3$ m	At the support and at mid-span

In Example (3): Overhanging beam subjected to UDL.

The two finite elements solution for nodal variables at nodal points using the same two-nodes beam element. The program results obtained are in good agreement with those obtained by Saeed Moaveni [17]. Hence, the location of maximum nodal variables values is presented in Table 5.34, as follows:

Table 5.34: Discussion on the location of maximum values of Example (3).

No.	Variable Name	Node Number	Nodal Coordinate (m)	Comment on Location
1	Vertical deflection w_i^e, and rotation θ_i^e	Node (3)	$x = 7.5$ m	At free end.
2	Shear force F_i^e	Node (2)	$x = 5$ m	At the support.
3	Bending moment M_i^e	Node (1)	$x = 0$ m	At the fixed support.

In Example (4): Simple supported beam with varying cross-section subjected to concentrated loads.

The three finite elements solution for nodal variables at nodal points using the same two-nodes beam element. The program results obtained are in good agreement with those obtained by Khameel Bayo Mustapha [15]. Hence, the location of maximum nodal variables values is presented in Table 5.35, as follows:

Table 5.35: Discussion on the location of maximum values of Example (4).

No.	Variable Name	Node Number	Nodal Coordinate (m)	Comment on Location
1	Vertical deflection w_i^e	Node (2)	$x = 1$ m	At mid-span.
2	Rotation θ_i^e	Node (3)	$x = 2$ m	At mid-span.
3	Shear force F_i^e	Node (4)	$x = 5$ m	At the fixed support.
4	Bending moment M_i^e	Node (4)	$x = 0$ m	At the fixed support.

In Example (5): Cantilever beam subjected to UDL and concentrated loads.

The three finite elements solution for nodal variables at nodal points using the same two-nodes beam element. The program results obtained are in good agreement with those obtained by Khameel Bayo Mustapha [15] . Hence, the location of maximum nodal variable values is presented in Table 5.36, as follows:

Table 5.36: Discussion on the location of maximum values of Example (5).

No.	Variable Name	Node Number	Nodal Coordinate (m)	Comment on Location
1	Vertical deflection w_i^e, and rotation θ_i^e	Node (4)	$x = 6$ m	At the free end.
2	Shear force F_i^e, and Bending moment M_i^e	Node (1)	$x = 0$ m	At the fixed support.

In Example (6): Continuous beam subjected to UDL and concentrated loads. The five finite elements solution for deflection at nodal points using the same two-nodes beam element. The program results obtained are in good agreement with those obtained by P. Seshu [30]. Hence, the location of maximum nodal deflection values is presented in Table 5.37, as follows:

Table 5.37: Discussion on the location of maximum values of Example (6).

No.	Variable Name	Node Number	Nodal Coordinate (m)	Comment on Location
1	Vertical deflection w_i^e	Node (5)	$x = 4$ m	At mid-span.
2	Rotation θ_i^e	Node (6)	$x = 5$ m	At the right end support.

In Example (7): Beam with Internal Hinge subjected to UDL.
The three finite elements solution for deflection at nodal points using the same two-nodes beam element. The program results obtained are in good agreement with those obtained by Daryl L. Logon [2]. Hence, the location of maximum nodal deflection values is presented in Table 5.39, as follows"

Table 5.38: Discussion on the location of maximum values of Example (7).

No.	Variable Name	Node Number	Nodal Coordinate (m)	Comment on Location
1	Vertical deflection w_i^e	Node (3)	$x = 3$ m	At the internal hinge.
2	Rotation θ_i^e	Node (2)	$x = 2$ m	The maximum value in the deflection curve. The internal results in a discontinuity in the slope or rotation of the deflection curve at the hinge as it appears in Figure 5.34 above.

In Example (8): Cantilever beam subjected to linear varying distributes loads. The two finite elements solution for deflection at nodal points using the same two-nodes beam element. The program results obtained are in good agreement with those obtained by J.N. Reddy [4]. Hence, the location of maximum nodal deflection values is presented in Table 5.39, as follows:

Table 5.39: Discussion on the location of maximum values of Example (8).

No.	Variable Name	Node Number	Nodal Coordinate (m)	Comment on Location
1	Vertical deflection w_i^e	Node (2)	$x = 1.5$ m	At mid-span.
2	Rotation θ_i^e	Node (3)	$x = 3$ m	At the free end.

In Example (9): Subjected to a linear varying distributed load, UDL, and concentrated loads.
The three finite elements solution for deflection at nodal points using the same two-nodes beam element. The program results obtained are in good agreement with those obtained by J.N. Reddy [4]. Hence, the location of maximum nodal deflection values is presented in Table 5.40, as follows:

Table 5.40: Discussion on the location of maximum values of Example (9).

No.	Variable Name	Node Number	Nodal Coordinate (in)	Comment on Location
1	Vertical deflection w_i^e	Node (4)	$x = 48$ in	At the free end.
2	Rotation θ_i^e	Node (4)	$x = 48$ in	At the free end.

CHAPTER SIX

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusion

In this research, Euler-Bernoulli beam flexural mode of deformation has been examined using a finite element displacement method (FEDM). One of the main objectives of this thesis is to develop a finite element computer program for the linear static analysis of thin beams to examine bending deformation, which is classified as class C^1 problem (continuity C^1).

In this research, the program results obtained from the analysis of different loading and support configuration of thin straight beams are compared with known published results and hence the following conclusions are drawn:

1. First conclusion, for the convergence using two nodes Euler-Bernoulli beam element shows good performance when used to analyze thin straight beams for bending. This is conformed when comparing results obtained with known published exact analytical solution.
2. Second conclusion, there is no significance difference between the results.

6.2 Recommendations for Further Studies

From the work presented in this research, the following recommendations for future work directions are suggested:

1. Expending the program to include transformation from local to global and the modification of the program to include axial effect for the linear analysis of two-dimensional planer frame based on the theory adopted.
2. Expending the program to include other types of analysis such as nonlinear and dynamic response.
3. Developing a graphical user interface (GUI) using MATLAB to ease the pre-processing and post-processing phase.

REFERENCES

[1] Loannis Koutromanos, James McClure, and Christopher Roy, Fundamentals of Finite Element Analysis: Linear Finite Element Analysis, 1st Edition, John Wiley & Sons Ltd, 2018.

[2] Daryl L. Logan, A First Course in the Finite Element Method, 6th Edition, Cengage Learning, 2016.

[3] Kwon Young W., The Finite Element Method Using MATLAB, 2nd Edition, CRC Press, 2000.

[4] Reddy J.N., An Introduction to the Finite Element Method, 3rd Edition, McGraw Hill Education, 2006.

[5] Kim Nam H., Sankar Bhavani V., and Kumer Ashok V., Introduction to Finite Element Analysis and Design, 2nd Edition, John Wiley & Sons Ltd, 2018.

[6] Baaser Herbert, Development and Application of the Finite Element Method based on MATLAB, 1th Edition, Springer, 2010.

[7] Chandruoathla Tirupathi R., and Belegundu Ashok D., Introduction to Finite Elements in Engineering, 4th Edition, Pearson Education Limited, 2011.

[8] Khennane Amar, Introduction to Finite Element Analysis Using MATLAB and Abaqus, 1st Edition, CRC Press, 2013.

[9] Austrell P.E., Dohlblom O., Lindemann J., Olsson A., Olsson K.G., and Persson K., CALFEM: A Finite Element Toolbox, Version 3.4, Structural Mechanics Lund University-Sweden, 2004.

[10] Chapman Stephen J., MATLAB Programming for Engineers, 6th Edition, Cengage Learning lnc, 2020.

[11] Smith I.M., Griffiths D.V., and Margetts L., Programming the Finite Element Method, 4th Edition, John Wiley & Sons Ltd, 2014.

[12] Felippa Carlos A, Introduction to Finite Element Methods (Textbook), 1st Edition, Aerospace Engineering Science Department of the University of Colorado at Boulder, 2004.

[13] Segerlind Larry J., Applied Finite Element Analysis, 2nd Edition, Jon Wiley & Sons lnc, 1984.

[14] Ochsner Andreas, Computational Statics and Dynamics: An introduction based on the Finite Element Method, 1st Edition, Springer, 2016.

[15] Mustapha Khameel, Finite Element Computations in Mechanics with R, 1st Edition, Taylor & Francis Group, 2018.

[16] Reo Singiresu S., The Finite Element Method in Engineering, 6th Edition, Elsevier lnc, 2018.
[17] Moaveni Saeed, Finite Element Analysis: Theory and Application with ANSYS, 4th Edition, Pearson Education Limited, 2015.
[18] Zhu Bofang, The Finite Element Method: Fundamentals and Applications in Civil, Hydraoulic, Mechanical and Aeronautical Engineering, 1st Edition, John Wiley & Sons Singapore Ltd, 2018.
[19] Liu G. R., and Quek S.S., The Finite Element Method: A Practical Course, 1st Edition, Elsevier Science Ltd, 2003.
[20] Akin J.E., Finite Element Analysis with Error Estimators, 1st Edition, Elsevier Butterworth-Heinemann, 2005.
[21] Bhavikatti S.S., Finite Element Analysis, 1st Edition, New Age International (P) Limited Publishers, 2005.
[22] Hutton David V., Fundamentals of Finite Element Analysis, 1st Edition, The McGraw-Hill Companies, 2004.
[23] Moatamedi M., and Khawaja H., Finite Element Analysis, 1st Edition, CRC Press, 2018.
[24] Guaranine Federico, and Walker Alastair C., Energy Methods in Structural Mechanics: A Comprehensive Introduction to Matrix and Finite Element Methods of Analysis, 1st Edition, Thomas Telford Publishing, 1999.
[25] Pidaparti Ramana M., Engineering Finite Element Analysis, 1st Edition, Morgan & Claypool Publishers LLC, 2017.
[26] Mikhailov Eugeniy E., Programming with MATLAB for Scientists: A Beginner's Introduction, 1st Edition, CRC Press, 2017.
[27] Ferreira Antonio J. M., and Fantuzzi Nicholas, MATLAB Codes for Finite Element Analysis: Solids and Structures, 2nd Edition, Springer, 2020.
[28] Bhatti M. Asghar, Fundamental Finite Element Analysis and Applications: with Mathematica and MATLAB Computations, 1st Edition, John Wiley & Sons lnc, 2005.
[29] Gan Buntaraas S., An Isogeometric Approach to Beam Structures: Bridging the Classical to Modern Technique, 1st Edition, Springer, 2018.
[30] Seshu P., Textbook of Finite Element Analysis, 1st Edition, PHI Learning Private Limited, 2006.

APPENDIX (I)

FINITE ELEMENT COMPUTER PROGRAM

```
%=========================================================================
=%
%                 FINITE ELEMENT COMPUTER PROGRAM
%
%
%
%          FOR LINEAR STATIC ANALYSIS OF THIN STAIGHT BEAMS (BENDING)
%                            (LSATSBB)
%
%
%
%                      USING MATLAB R2019b
%
%=========================================================================
=%
%                        THE MAIN PROGRAM
%
%=========================================================================
=%
%                   (1) PRE-PROCESSING PHASE
%
%=========================================================================
=%
%          (1.1)  CLEAR ALL VARIABLES AND SCREEN MEMORY
%
%-------------------------------------------------------------------------
-%
clc % CLEAR ALL COMMANDS IN THE COMMAND WINDOW
clear % CLEAR ALL VARIABLES IN WORKSPACE
%-------------------------------------------------------------------------
-%
%     (1.2) THE GLOBAL VARIABLES USED IN MAIN PROGRAM FROM FUNCTIONS
%
%-------------------------------------------------------------------------
-%
prop nf Element_loads Joint_loads force Hinge
global nnd nel nne nodof eldof n geom connec F ...
%-------------------------------------------------------------------------
-%
%          (1.3)   EXECUT AND DISPLAY MAIN PROGRAM
%
%-------------------------------------------------------------------------
-%
display('Executing Main_FEA_Program'); % EXECUTING AND DISPLAY FORCES
RESULTS
%                                      IN COMMAND WINDOW
%-------------------------------------------------------------------------
-%
% (1.4) AFTER COMPUTATION DISPLAY THE OUTPUT TEXT FILE FOR FEA SOLUTION
%
%       RESULTS
%
%-------------------------------------------------------------------------
-%
```

```
display('Results printed to file : FEA_Results.txt '); % DISPLAY IN TEXT
FILE
fid=fopen('FEA_Results.txt','w');
%-----------------------------------------------------------------------
-%
% (1.5) SELECT THE INPUT SUB-PROGRAM FILE FOR BEAM MODEL TO BE ANALYZED
%
%                      (BUILDING ANALYSIS MODAL)
%
%-----------------------------------------------------------------------
-%
Sub_Input_Modal_Data ; % SUB-PROGRAM FILE NAME FOR BEAM MODEL
%=======================================================================
=%
%                       (2) PRE-PROCESSING PHASE
%
%                          (CONDUCTING ANALYSIS)
%
%=======================================================================
=%
%        (2.1) PRINT AND DISPLAY INPUT MODAL DATA FOR REVIEW
%
%-----------------------------------------------------------------------
-%
Sub_Print_Display_Model_Data % SUB-PROGRAM TO PRINT & DISPLAY OF MODAL
DATA
%-----------------------------------------------------------------------
-%
%          (2.2) INITIALIZE GLOBAL STRUCTURE EQUATIONS TO ZERO
%
%-----------------------------------------------------------------------
-%
KK =zeros(n) ; % INITIALIZE GLOBAL STRIFFNESS MATRIX TO ZERO
F=zeros(n,1); % INITIALIZE GLOBAL FORCE VECTOR TO ZERO
F = form_beam_F(F); % FUNCTION TO FORMING GLOABAL FORCE VECTOR
%      (2.3) LOOP FUNCTION TO ASSEMBLE GLOBAL STIFFNESS MATRIX
%
%-----------------------------------------------------------------------
-%
for i=1:nel   % LOOP FOR NUMBER OF ELEMENT
kl=beam_k(i); % USING FUNCTION THAT FORM ELEMENT MATRIX
g=beam_g(i) ; % USING FUNCTION THAT RETRIEVE ELEMENT STEERING VECTOR
KK =form_KK(KK, kl, g); % USING FUNCTION THAT FORM GLOBAL STIFFNESS
MATRIX
end
%-----------------------------------------------------------------------
-%
%      (2.4) THE SOLUTION OF GLOBAL DEFLECTION VECTOR
%
%-----------------------------------------------------------------------
-%
delta = KK\F ; % SOLUTION VECTOR FOR NODAL UNKNOWNS
%=======================================================================
=%
%                      (3) POST-PROCESSING PHASE
%
%                          (ANALYSIS RESULT)
%
%=======================================================================
=%
% (3.1) LOOP FUNCTION TO RETRIVE ELEMENT NODAL DEFLECTION FOR RECTION
%
```

```
%       CORRECTION
%
%------------------------------------------------------------------------
-%
for i=1:nnd   % LOOP FOR NUMBER OF NODES
for j=1:nodof % LOOP FOR NUMBER OF DOFs
node_disp(i,j) = 0;
if nf(i,j)~= 0 % IF STATEMENT TO CONTROL NODAL DEFLECTION FROM SOLUTION
    %            BASED ON BCs
node_disp(i,j) = delta(nf(i,j)) ;
end
end
end
%------------------------------------------------------------------------
-%
%  (3.2) LOOP FUNCTION TO CALCULATE CORRECT LOCAL MEMBERS NODAL REACTION
%
%        LOADS
%
%------------------------------------------------------------------------
-%
for i=1:nel   % LOOP FOR NUMBER OF ELEMENT
kl=beam_k(i); % USING FUNCTION THAT FORM ELEMENT MATRIX
g=beam_g(i) ; % USING FUNCTION THAT RETRIEVE ELEMENT STEERING VECTOR
for j=1:eldof % LOOP FOR NUMBER OF DOFs
if g(j)== 0
ed(j)=0.; % FOR RESTRAINED DOFs THE DEFECTION IS ZERO
else
ed(j) = delta(g(j));% SOLUTION VECTOR FOR NODAL UNKNOWNS AFTER APPLYING
BCs
end
end
fl = kl*ed' ; % ELEMENT JOINT NODAL LOADS
f0 = Element_loads(i,:) % ELEMENT STATICALLY EQUIVALENT NODAL LOADS
force(i,:) = fl-f0' % THE CORRECT LOCAL NODAL REACTION
end
%------------------------------------------------------------------------
-%
% (3.3) SUB-PROGRAM TO PRINT FINITE ELEMENT ANALYSIS SOLUTION RESULT
%
%------------------------------------------------------------------------
-%
Sub_Print_Solution_Results; % SUB-PROGRAM FOR PRINTING FEA RESULTS
fclose(fid);
%========================================================================
=%
%                        END OF MAIN PROGRAM
%
%========================================================================
```

APPENDIX (II)

PROGRAM OUTPUT RESULTS

EXAMPLE (1) ONE FINITE ELEMENT SOLUTION PROGRAM OUTPUT

```
******* Sub_Print_Display_Model_Data **************
----------------------------------------------------
Number of nodes: 2
Number of elements: 1
Number of nodes per element: 2
Number of degrees of freedom per node: 2
Number of degrees of freedom per element: 4
----------------------------------------------------
Node coordinate X direction
 1, 0000.00
 2, 0100.00
----------------------------------------------------
Element connectivity Node_1 Node_2
 1, 1, 2
----------------------------------------------------
Element properties E I
 1, 3e+07, 100
----------------------------------------------------
-------------Nodal freedom---------------------------
Node displacement & rotation
 1, 0, 0
 2, 1, 2
----------------------------------------------------
-----------------Applied Nodal Loads-------------------
Node load_& moment in Y direction
 1, 0000.00, 0000.00
 2, -1000.00, 16666.66
----------------------------------------------------
Total number of active degrees of freedom, n = 2
------------------------------------------------------
 ******* PRINTING ANALYSIS RESULTS **************
----------------------------------------------------
Global force vector F
 -1000
 16666.7

----------------------------------------------------
Displacement solution vector: delta
 -0.08333
 -0.00111
```

```
------------------------------------------------------
Nodal displacements
Node displacement & rotation
 1,  0.00000,  0.00000
 2, -0.08333, -0.00111
------------------------------------------------------
Members actions
element fy1 M1 Fy2 M2
 1,   2000.00, 100000.00,      0.00,      0.00
```

EXAMPLE (1) TWO FINITE ELEMENT SOLUTION PROGRAM OUTPUT

```
******* Sub_Print_Display_Model_Data **************
------------------------------------------------------
Number of nodes: 3
Number of elements: 2
Number of nodes per element: 2
Number of degrees of freedom per node: 2
Number of degrees of freedom per element: 4
------------------------------------------------------
Node coordinate X direction
 1, 0000.00
 2, 0050.00
 3, 0100.00
------------------------------------------------------
Element connectivity Node_1 Node_2
 1, 1, 2
 2, 2, 3
-----------------------------------------------------
Element properties E I
 1, 3e+07, 100
 2, 3e+07, 100
------------------------------------------------------
-------------Nodal freedom----------------------------
Node displacement & rotation
 1, 0, 0
 2, 1, 2
 3, 3, 4
------------------------------------------------------
-----------------Applied Nodal Loads-------------------
Node load_& Moment in Y direction
 1, 0000.00, 0000.00
 2, -1000.00, 0000.00
 3, -500.00, 4166.67
------------------------------------------------------
Total number of active degrees of freedom, n = 4
--------------------------------------------------------
```

```
 ******* PRINTING ANALYSIS RESULTS **************
------------------------------------------------------
Global force vector F
 -1000
 0
 -500
 4166.67
------------------------------------------------------
Displacement solution vector: delta
 -0.02951
 -0.00097
 -0.08333
 -0.00111
------------------------------------------------------
Nodal displacements
Node displacement & rotation
 1,  0.00000,  0.00000
 2, -0.02951, -0.00097
 3, -0.08333, -0.00111
------------------------------------------------------
Members actions
element fy1 M1 Fy2 M2
 1,   2000.00, 100000.00,  -1000.00, -25000.00
 2,   1000.00,  25000.00,     -0.00,      0.00
```