



**HAL**  
open science

# Geometric Nonlinear Analysis of Timoshenko Beams

S. Amir Mousavi Lajimi

► **To cite this version:**

S. Amir Mousavi Lajimi. Geometric Nonlinear Analysis of Timoshenko Beams. [Technical Report] University of Waterloo. 2009. hal-02918610

**HAL Id: hal-02918610**

**<https://hal.science/hal-02918610>**

Submitted on 20 Aug 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Geometric Nonlinear Analysis Of Timoshenko Beams

with a comprehensive study of linear and nonlinear computational  
mechanics of solids and structures

S.Amir Mousavi Lajimi

Department of Systems Design Engineering

University Of Waterloo

September 2009

### **Abstract**

The linear finite element analysis of solids and structures are discussed in the first part of the report. The finite element formulations for a three dimensional problem is derived and significant issues are addressed. Three different types of elements, the bilinear iso-parametric quadrilaterals, the quadratic triangles and the linear tetrahedrons are used to solve a linear plate problem and the results are compared.

The analysis of a common geometric nonlinearity encountered in structural mechanics is dealt with in the second part of the report. A brief introduction to nonlinear analysis is provided, while the geometric nonlinearity is discussed in detail. Timoshenko beam analysis is considered as the one dimensional version of Reissner-Mindlin plate theory and the nonlinear strain-displacement relation are treated in an appropriate way to avoid unrealistic simplifications. The force vector and tangent stiffness matrix are derived and the formulation is extended to implement the trigonometric basis functions. A major issue in geometric nonlinear analysis, namely locking, is addressed and reduced order integrations are implemented to avoid the consequences. The convergence of the model is checked with available analytical solutions. An Euler method in combination with Newton-Raphson method is used to fully analyze the geometric nonlinearity.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	The Finite Element Method . . . . .	1
1.3	Report Organization . . . . .	2
<b>I</b>	<b>The Linear Computational Mechanics</b>	<b>3</b>
<b>2</b>	<b>Development of the Finite Element Method</b>	<b>4</b>
2.1	Introduction . . . . .	4
2.2	Basic Equations . . . . .	4
2.2.1	Equilibrium Equations . . . . .	4
2.2.2	Constitutive Relations . . . . .	5
2.2.3	Strain-Displacement Relations . . . . .	6
2.2.4	Boundary Conditions . . . . .	6
2.2.5	Compatibility Equations . . . . .	7
2.3	Formulations . . . . .	7
2.3.1	Differential Equation Methods . . . . .	7
2.3.2	Variational Methods . . . . .	7
2.3.3	Formulations of Finite Element Equations . . . . .	9
<b>3</b>	<b>Elements and Interpolation Functions</b>	<b>12</b>
3.1	Introduction . . . . .	12
3.2	The Bilinear Quadrilateral Lagrangian Element . . . . .	12
3.2.1	Element Shape Functions . . . . .	12
3.2.2	Implementation . . . . .	13
3.3	The Quadratic Triangular Element . . . . .	14
3.3.1	Geometry . . . . .	14
3.3.2	Element Shape Functions . . . . .	14
3.3.3	Implementation . . . . .	15
3.3.4	Gauss Quadrature for Triangular Elements . . . . .	16
3.4	The Linear Tetrahedral (Solid) Element . . . . .	17
3.4.1	Geometry . . . . .	17
3.4.2	Element Shape Functions . . . . .	18
3.4.3	Implementation . . . . .	18
<b>4</b>	<b>Results</b>	<b>20</b>
4.1	The Bilinear Quadrilateral Element . . . . .	20
4.2	The Quadratic Triangular Element . . . . .	21
4.3	The Linear Tetrahedral Element . . . . .	22
4.4	Discussion . . . . .	23

<b>II</b>	<b>The Nonlinear Computational Mechanics</b>	<b>24</b>
<b>5</b>	<b>Nonlinear Finite Element Method</b>	<b>25</b>
5.1	Introduction . . . . .	25
5.2	Nonlinearities in Solid and Structural Mechanics . . . . .	25
5.3	Solution Procedures . . . . .	25
5.3.1	Geometric Nonlinearity . . . . .	26
5.3.2	Material Nonlinearity . . . . .	26
<b>6</b>	<b>Finite Element Analysis of Nonlinear Timoshenko Beams</b>	<b>27</b>
6.1	Introduction . . . . .	27
6.2	Background and Basic Equations . . . . .	27
6.2.1	Straight Beams . . . . .	27
6.2.2	Curved Beams . . . . .	29
6.3	Beam Element and Interpolation Functions . . . . .	31
6.4	Constitutive Relations . . . . .	33
6.5	Finite Element Formulation of Equilibrium Equations . . . . .	33
6.5.1	Tangent Stiffness Matrix . . . . .	34
6.6	Reduced Numerical Integration . . . . .	37
<b>7</b>	<b>Results</b>	<b>39</b>
7.1	Cantilever with vertical tip load . . . . .	39
7.2	Quarter Circle Beam with Radial Tip Load . . . . .	41
7.3	Discussion . . . . .	45
<b>8</b>	<b>Conclusions</b>	<b>46</b>
8.1	Conclusions . . . . .	46
8.2	Future Works . . . . .	46
<b>A</b>	<b>Source codes</b>	<b>47</b>
A.1	Quadrilateral Bilinear Element . . . . .	47
A.2	Quadratic Triangular Element . . . . .	49
A.3	Linear Tetrahedral Element . . . . .	50
A.4	Nonlinear Timoshenko Beam Analysis . . . . .	51
<b>B</b>	<b>Locking</b>	<b>58</b>

# List of Figures

3.1	Quadrilateral Lagrangian Element . . . . .	12
3.2	Quadratic Triangular Element . . . . .	14
3.3	Triangular Coordinate . . . . .	14
3.4	Linear Tetrahedron Element . . . . .	17
4.1	Thin plate under uniformly distributed load. . . . .	20
4.2	Thin plate mesh using quadrilateral elements. . . . .	21
4.3	Thin plate mesh using quadratic triangular elements. . . . .	21
4.4	Thin plate mesh using linear tetrahedral elements . . . . .	22
6.1	The straight beam element . . . . .	28
6.2	The curved beam element . . . . .	30
6.3	An automated code to produce Lagrangian basis functions. . . . .	32
7.1	The cantilever straight beam. . . . .	39
7.2	The cantilever straight beam results, $L/h = 4$ . . . . .	40
7.3	The cantilever straight beam results, $L/h = 10$ . . . . .	41
7.4	The cantilever straight beam results, $L/h = 50$ . . . . .	42
7.5	The quarter circle curved cantilever beam. . . . .	43
7.6	The cantilever curved beam results . . . . .	43
7.7	The p-convergence of the results . . . . .	44
7.8	The nonlinear force-displacement curve . . . . .	44
7.9	The difference between nonlinear and linear solution . . . . .	45
7.10	The force-displacement curve using Newton-Raphson method . . . . .	45

# List of Tables

- 4.1 Global force and displacement vectors using Quadrilateral element . . . . . 21
- 4.2 Global force and displacement vectors for Triangular element . . . . . 22
- 4.3 Global force and displacement vectors for Tetrahedral Element . . . . . 23
- 4.4 Convergence of the result for quadrilateral element . . . . . 23
  
- 7.1 The material and geometric data of straight beams. . . . . 40

# Chapter 1

## Introduction

### 1.1 Motivation

The computational mechanics is associated with the implementations of the computational methods in solving the problems of the mechanics including solid, structural, and fluid mechanics. Working in computational mechanics is concerned with developing discrete models, numerical methods, and solution procedures for a wide range of problems arising from studying continuum mechanics.

The mathematical descriptions or differential governing equations, are customarily referred to as the strong form of a problem. The conclusive objective can be summarized as to have an algebraic equation or system of equations rather than differential governing equations which is ideal for implementation on the digital computer. These equations may be linear or nonlinear depending on the nature of the problem <sup>1</sup>.

The most conventionally used solution technique is the finite difference method, which makes use of the calculus of finite differences. The method is simply formulated and put into use by replacing differential operators with difference operators. However, the solutions are approximated for certain numbers of nodes. Therefore, other methods such as the method of weighted residuals and the finite element method are implemented to approximate the solutions of a physical problem where a continuous solution is mostly desired.

While implementing some methods such as the method of weighted residuals is quite straightforward for simple domains, for complex domains such as curved boundaries it might not be convenient and adaptable. The finite element method is introduced to overcome those impediments, and provides an accessible method for many engineering problems. On the other hand, in the analysis of real structures dealing with the existing nonlinearities is not avoidable. Thus, it is necessary to include different types of nonlinearities to simulate the true behavior of the structure. The nonlinear finite element analysis provides a powerful tool to deal with different types of nonlinear problems.

### 1.2 The Finite Element Method

The domain is discretized to some smaller domains, where the error concerned with the discretization of the domain can be reduced by reducing the size of the elements, the shape of the elements, and the number of the elements. The global system of equations is realized through a process called assembly, and the geometric boundary conditions are applied at the end before solving the system of algebraic

---

<sup>1</sup>The governing differential equations display what is expected regarding linearity or nonlinearity of the final form.



equations. Indeed, the procedure is unambiguous, except a nonlinearity appears ahead in the governing equations of the system.

The genuine systems are always nonlinear, however linear solutions are easier to compute at lower costs. The linear solutions do satisfy the actual requirements to certain degrees, however to gain a better understanding of the behavior of the continuum and physical phenomena a nonlinear analysis is imperative. A nonlinear differential equation results in a nonlinear algebraic system, where the solution should be sought using some iterative procedures. Therefore, the computational cost will increase and some new issues such as stability and optimality of the result might arise.

The main classes of nonlinearities are geometric nonlinearities and material nonlinearities. While, the latter occur when the stress-strain or force-displacement law is not linear, or when material properties change with the applied loads, the former involves nonlinearities in kinematic quantities such as the strain-displacement relations in solids. Such nonlinearities can occur due to large displacements, large strains, large rotations, and so on. Contact can also be classified as a geometric nonlinearity because the area of contact is a function of the deformation <sup>1</sup>. The main objective of this work is to provide the nonlinear finite element equations encountered when analyzing a Timoshenko beam. The nonlinearity appears when considering a nonlinear strain-displacement relation. Hence, the geometric nonlinearity is considered here.

### 1.3 Report Organization

The report is presented in two parts: the linear finite element analysis and the nonlinear finite element analysis. Development of the finite element method for a linear analysis is described in chapter 2. Chapter 3 includes a review of several types of elements and interpolation functions. In the final chapter of the first part, chapter 4, the results of the linear finite element analysis of a problem is presented. Second part, starts with an introduction to the nonlinear finite element method in chapter 5. Chapter 6 is devoted to the derivation of the nonlinear finite element equations for Timoshenko beam. It includes the required formulations for analyzing the straight as well as curved beams, an explanation about the beam element, several types of shape functions and integration schemes. In chapter 7 the results are provided for two cases where a comparison is made between the linear and nonlinear solutions. Finally, the conclusions and future works are briefly mentioned in chapter 8.

---

<sup>1</sup>Contact can be considered in another class called nonlinear boundary conditions.

## Part I

# The Linear Computational Mechanics

# Chapter 2

## Development of the Finite Element Method

### 2.1 Introduction

One appreciates miscellaneous types of problems solved by the finite element method in the field of applied mechanics including the linear analysis of solids and structures under small deformation, and the nonlinear analysis of solids and structures. In this chapter, we are dealing with the finite element elastic analysis of one-, two-, and three-dimensional problems.

### 2.2 Basic Equations

In solid and structural mechanics, the problem is primarily to find the displacement distributions and stresses under the external loads and boundary conditions. Therefore, the initial step would be to satisfy the equilibrium equations, that is considering an element of the material inside the body, it must be in equilibrium due to the internal forces (stresses) developed as a result of the external provocations.

#### 2.2.1 Equilibrium Equations

When a body is in equilibrium, the force and moment equilibrium equations for the overall body have to be satisfied. This leads to the following equations for each point of a body,

$$\begin{aligned}\frac{\partial\sigma_{xx}}{\partial x} + \frac{\partial\sigma_{xy}}{\partial y} + \frac{\partial\sigma_{zx}}{\partial z} + b_x &= 0 \\ \frac{\partial\sigma_{xy}}{\partial x} + \frac{\partial\sigma_{yy}}{\partial y} + \frac{\partial\sigma_{yz}}{\partial z} + b_y &= 0 \\ \frac{\partial\sigma_{zx}}{\partial x} + \frac{\partial\sigma_{yz}}{\partial y} + \frac{\partial\sigma_{zz}}{\partial z} + b_z &= 0\end{aligned}\tag{2.1}$$

where  $\sigma_{xx}, \sigma_{yy}$  and  $\sigma_{zz}$  are the normal stresses,  $\sigma_{xy}, \sigma_{yz}$  and  $\sigma_{zx}$  are the shear stresses, and  $b_x, b_y$  and  $b_z$  are the body forces per unit volume acting along the  $x, y$  and  $z$  respectively.

## 2.2.2 Constitutive Relations

The secondary dependent variables are correlated to the primary dependent variables through appropriate relations involving derivatives of the primary dependent variables. In addition, the constitution of the material must be taken into account. This is accomplished through the constitutive relations. The constitutive relations describe the response of a certain system or body to an applied loading. In the case of linearly elastic materials, in a three-dimensional analysis, the stress-strain relations, or constitutive relations, are given by Hook's law as

$$\underline{\epsilon} = \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \epsilon_{xy} \\ \epsilon_{yz} \\ \epsilon_{zx} \end{Bmatrix} = [C]\underline{\sigma} \equiv [C] \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{Bmatrix} \quad (2.2)$$

where  $[C]$  is a matrix of elastic constants given by,

$$[C] = \frac{1}{E} \begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{bmatrix} \quad (2.3)$$

$E$  is Young's modulus and  $\nu$  is Poisson's ratio of the material. Stresses are expressed in terms of strains as,

$$\underline{\sigma} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{Bmatrix} = [D]\underline{\epsilon} \equiv [D] \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \epsilon_{xy} \\ \epsilon_{yz} \\ \epsilon_{zx} \end{Bmatrix} \quad (2.4)$$

$$[D] = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1-\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \quad (2.5)$$

Two states of stress distribution are possible in two-dimensional problems, namely, plane stress and plain strain. The plane strain assumption is valid when the body is very long in one direction and its geometry and loading do not vary considerably in the longitudinal direction. For example, the analysis of dams and cylinder can be made using the plane strain assumption. The dependent variables are considered to be functions of two independent directions, say  $x$  and  $y$ , provided a section far from the ends is considered. In this case the stress-strain relation is given by

$$\underline{\sigma} = [D]\underline{\epsilon} \quad (2.6)$$

with,

$$[D] = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \quad (2.7)$$

The  $z$ -component of the stress will be nonzero and is given by

$$\sigma_{zz} = \nu (\sigma_{xx} + \sigma_{yy}) \quad (2.8)$$

and  $\sigma_{yz} = \sigma_{zx} = 0$ .

The premise of the plane stress is valid for bodies with one small dimension in one of the coordinate directions. For instance, in the analysis of thin plates when loaded in the plane the plane stress assumption is justified to be used. In plane stress distribution, it is assumed that

$$\sigma_{zz} = \sigma_{yz} = \sigma_{zx} = 0 \quad (2.9)$$

where  $z$  represents the perpendicular direction to the plane of the plate. In this case, the stress-strain relation is given by

$$\underline{\sigma} = [D]\underline{\epsilon} \quad (2.10)$$

with,

$$[D] = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad (2.11)$$

In the case of plane stress, the component of strain in the  $z$ - direction will be nonzero and is given by

$$\epsilon_{zz} = -\frac{\nu}{E} (\sigma_{xx} + \sigma_{yy}) \quad (2.12)$$

and  $\epsilon_{yz} = \epsilon_{zx} = 0$ .

### 2.2.3 Strain-Displacement Relations

Strains are induced in a body during the change of its shape as a result of an imposed set of loads. These strains vary through the volume of the body and can be related to the displacements at each point of the body. Defining  $u, v,$  and  $w$  as three components of displacements parallel to the  $x, y$  and  $z$  directions, the normal strains in  $x, y$  and  $z$  directions are computed by

$$\epsilon_{xx} = \frac{\partial u}{\partial x}, \quad \epsilon_{yy} = \frac{\partial v}{\partial y}, \quad \epsilon_{zz} = \frac{\partial w}{\partial z} \quad (2.13)$$

The shear strain is defined as the decrease in the right angle between two fibers which were at right angles to each other before deformation. Therefore, shear strains in the  $xy, yz$  and  $zx$  planes are

$$\epsilon_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}, \quad \epsilon_{yz} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}, \quad \epsilon_{zx} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \quad (2.14)$$

### 2.2.4 Boundary Conditions

Boundary conditions are divided to two major categories: the forced, geometric or Dirichlet boundary conditions which are defined as displacement boundary conditions, and natural, physical, or Neumann boundary conditions which are restrictions on the surface traction or stresses. The boundary conditions on displacements require the body or structure to take on predefined displacements or deflections at certain points, while the natural boundary conditions require that the stresses induced must be in equilibrium with the external forces applied at certain points on the boundary of the body.

## 2.2.5 Compatibility Equations

The displacement field inside a body or structure must be continuous as well as single valued. This is known as the condition of the compatibility. The condition of compatibility can be described from a different point of view. It can be observed from Eqs. (2.13) and (2.14) that the three strains  $\epsilon_{xx}$ ,  $\epsilon_{yy}$  and  $\epsilon_{xy}$  are derived from only two displacements  $u$  and  $v$ . This implies that there should be a relation between  $\epsilon_{xx}$ ,  $\epsilon_{yy}$  and  $\epsilon_{xy}$  if these strains correspond to a compatible deformation. This relation is called the “compatibility equation”. In three-dimensional elasticity problems, there are totally six compatibility equations to be satisfied. In the case of two-dimensional plane strain problems, a single equation must be met as

$$\frac{\partial^2 \epsilon_{xx}}{\partial y^2} + \frac{\partial^2 \epsilon_{yy}}{\partial x^2} = \frac{\partial^2 \epsilon_{xy}}{\partial x \partial y} \quad (2.15)$$

For plane stress problems, the equations are expressed as

$$\frac{\partial^2 \epsilon_{xx}}{\partial y^2} + \frac{\partial^2 \epsilon_{yy}}{\partial x^2} = \frac{\partial^2 \epsilon_{xy}}{\partial x \partial y}, \quad \frac{\partial^2 \epsilon_{zz}}{\partial y^2} = \frac{\partial^2 \epsilon_{zz}}{\partial x^2} = \frac{\partial^2 \epsilon_{zz}}{\partial x \partial y} \quad (2.16)$$

In the case of one-dimensional problems the condition of compatibility is automatically satisfied.

## 2.3 Formulations

Most of the solid and structural mechanics problems can be formulated starting from either the governing differential equation of the problem using, for example, the principle of virtual work, or a variational principle such as the principle of minimum potential energy. In the following a brief description of both approaches are provided.

### 2.3.1 Differential Equation Methods

In this type of formulations, a method such as Galerkin method is used for finding an approximate solution to a differential equation. Therefore, the approach is concerned with the direct use of the differential equation; it does not require the existence of a functional. The Galerkin method is probably the most popular one in this category. Application of the Galerkin method requires that the following conditions be satisfied [3] (as cited in [7]):

- The weighting or test functions are chosen from the same set as the trial functions.
- The trial and test functions must be linearly independent.
- The trial and test functions should be chosen from the first  $P$  functions of a complete set of functions.
- The trial functions should exactly satisfy the boundary conditions and, if applicable, the initial conditions.

### 2.3.2 Variational Methods

Using variational methods requires having a correct functional which gives the corresponding finite element equations when finding the stationary point of it. There are several functionals to be used in mechanics such as the principle of minimum potential energy or Hamilton’s principle which are succinctly explained here.

## Principle of Minimum Potential Energy

The potential energy of an elastic body,  $\Pi$ , is expressed as

$$\Pi = U - W \quad (2.17)$$

where  $U$  is the strain energy and  $W$  is the work done on the body by the external forces. The principle of minimum potential energy states that of all displacements satisfying the given compatibility, kinematic or boundary conditions those which satisfy the equilibrium equations make the potential energy assume a minimum value. Therefore, at equilibrium

$$\delta\Pi = \delta U - \delta W = 0 \quad (2.18)$$

where  $\delta$  shows the first variation taken with respect to the displacements. The strain energy of a linear elastic body is defined as

$$U = \frac{1}{2} \iiint_V \underline{\epsilon}^T \underline{\sigma} dV \quad (2.19)$$

where  $V$  is the volume of the body. Using the stress-strain relation, Eq.(2.4), the strain energy can be expressed as

$$U = \frac{1}{2} \iiint_V \underline{\epsilon}^T [D] \underline{\epsilon} dV \quad (2.20)$$

The work done by the external forces is computed using

$$W = \iiint_V \underline{b}^T \underline{d} dV + \iint_{S_t} \underline{t}^T \underline{d} dS_t \quad (2.21)$$

where

$$\underline{b} = \begin{Bmatrix} b_x \\ b_y \\ b_z \end{Bmatrix} \quad (2.22)$$

is known as the body force vector,

$$\underline{t} = \begin{Bmatrix} t_x \\ t_y \\ t_z \end{Bmatrix} \quad (2.23)$$

is the vector of prescribed surface traction defined on  $S_t$ , and

$$\underline{d} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \quad (2.24)$$

is the vector of displacements. When the principle of minimum potential energy is used to derive the finite element equations, we assume a simple form of variation for the displacement field within each element and derive conditions which will minimize the corresponding functional,  $\Pi$ . The resulting equations are the approximate equilibrium equations while the compatibility conditions are identically satisfied.

## Hamilton's Principle

Hamilton's principle states that of all admissible configurations that a body can take as it moves from configuration  $a$  at time  $t_1$  to configuration  $b$  at time  $t_2$ , the path that satisfies the Newton's law at each instant during the interval (and is thus the actual locus of configurations) is the path that extremizes the time integral of the Lagrangian during the interval. Therefore, Hamilton's principle can be stated as

$$\delta \int_{t_1}^{t_2} L dt = 0 \quad (2.25)$$

where the Lagrangian ( $L$ ) is defined as

$$L = T - U \quad (2.26)$$

The kinetic energy ( $T$ ) of a body is given by

$$T = \frac{1}{2} \iiint_V \rho \underline{\dot{d}}^T \underline{\dot{d}} dV \quad (2.27)$$

where  $\rho$  is the density of the material and  $\underline{\dot{d}} = \{\dot{u}, \dot{v}, \dot{w}\}^T$  is the vector of velocity components at any point inside the body. The potential energy can be found from Eqs. (2.17) - (2.21). This variational principle is used for dynamics problems.

## Other Variational Principles

There are several other variational principles such as the principle of minimum complementary energy and the minimum of Reissner energy. To derive the finite element equations using the principle of stationary Reissner energy, a form of the variation for both displacement and stress fields within the element is assumed. This leads to the mixed method of finite element analysis.

### 2.3.3 Formulations of Finite Element Equations

The principle of minimum potential energy is used for deriving the equilibrium equations for a three-dimensional problem. The nodal degrees of freedom are treated as unknowns in this formulation. The final form of equilibrium equations can be obtained by setting the first partial derivatives of  $\Pi$  with respect to each nodal degrees of freedom equal to zero. The various steps associated the derivation of the equilibrium equations are summarized below.

- The body or structure is divided into  $ne$  finite elements.
- The displacement model within an element  $e$  is assumed as

$$\underline{d} = \begin{Bmatrix} u(x, y, z) \\ v(x, y, z) \\ w(x, y, z) \end{Bmatrix} = [N] \underline{d}^e \quad (2.28)$$

where  $\underline{d}^e$  is the vector of nodal displacement degrees of freedom of the element and  $[N]$  is the matrix of shape functions.

- The element stiffness matrix and force vectors are to be derived. The potential energy of an element  $e$  is expressed as

$$\Pi^e = \frac{1}{2} \iiint_{V^e} \underline{\epsilon}^T [D] \underline{\epsilon} dV - \iint_{S_t^e} \underline{d}^T \underline{t} dS_t - \iiint_{V^e} \underline{d}^T \underline{b} dV \quad (2.29)$$



where  $V^e$  is the volume of the element and  $S_t^e$  is the portion of the surface of the element over which surface traction is applied.<sup>1</sup> The strain vector can be expressed in terms of nodal unknowns as

$$\underline{\epsilon} = \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \epsilon_{xy} \\ \epsilon_{yz} \\ \epsilon_{zx} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial z} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \end{Bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = [B]\underline{d}^e \quad (2.30)$$

where

$$[B] = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} [N] \quad (2.31)$$

Then, the stresses are obtained using (2.4). If  $\underline{P}$  denotes the vector of nodal forces, the total potential energy of the structure or body can be expressed as

$$\Pi = \sum_{e=1}^{ne} \Pi^e - \underline{d}^T \underline{P} \quad (2.32)$$

where

$$\underline{d} = \begin{Bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{Bmatrix} \quad (2.33)$$

is the vector of nodal displacements of the entire structure or body and  $m$  is the total number of degrees of freedom. The summation in (2.32) implies the expansion of element matrices to “structure or body size” followed by summation of overlapping terms. Therefore, the total potential energy of the body or structure takes the following form

$$\begin{aligned} \Pi &= \frac{1}{2} \underline{d}^T \left[ \sum_{e=1}^{ne} \iiint_{V^e} [B]^T [D] [B] dV \right] \underline{d} \\ &\quad - \underline{d}^T \sum_{e=1}^{ne} \left( \iint_{S_t^e} [N]^T \underline{t} dS_t + \iiint_{V^e} [N]^T \underline{b} dV \right) - \underline{d}^T \underline{P} \end{aligned} \quad (2.34)$$

The system of algebraic equations of the structure can be realized setting the first variation of (2.34) to zero. Thus, (2.34) is modified to

$$\left( \sum_{e=1}^{ne} [K^e] \right) \underline{d} = \underline{P} + \sum_{e=1}^{ne} (\underline{P}_t^e + \underline{P}_b^e) \quad (2.35)$$

---

<sup>1</sup>The superscript  $e$  have been removed from  $\underline{d}^T$ ,  $\underline{b}$  and  $\underline{t}$  for clarity of notation.

where

$$[K^e] = \iiint_{V^e} [B]^T [D] [B] dV \quad (2.36)$$

is the element stiffness matrix,

$$\underline{P}_t^e = \iint_{S_t^e} [N]^T \underline{t} dS_t \quad (2.37)$$

is the element load vector due to the surface traction and

$$\underline{P}_b^e = \iiint_{V^e} [N]^T \underline{b} dV \quad (2.38)$$

represents the element load vector due to the body forces.

- The global system of equations can now be expressed as

$$[K] \underline{d} = \underline{F} \quad (2.39)$$

where

$$[K] = \sum_{e=1}^{ne} [K^e] \quad \text{assembled stiffness matrix} \quad (2.40)$$

and

$$\underline{F} = \underline{P} + \sum_{e=1}^{ne} \underline{P}_b^e + \sum_{e=1}^{ne} \underline{P}_t^e \quad \text{assembled nodal load vector} \quad (2.41)$$

- The solution for the nodal displacements can be obtained after solving (2.39). However, this should be done after applying the boundary conditions.
- The outputs from last step may further be post-processed to obtain stresses, strains, element deformations and etc.

# Chapter 3

## Elements and Interpolation Functions

### 3.1 Introduction

This chapter continues with the computer implementation of the two- and three-dimensional finite elements. It covers the derivation of the finite element equations for bilinear quadrilateral Lagrangian element, quadratic triangular element and linear tetrahedral (solid) element. The area and tetrahedral coordinates are introduced, numerical integration for the triangular geometry is defined, and stiffness matrices are computed.

### 3.2 The Bilinear Quadrilateral Lagrangian Element

#### 3.2.1 Element Shape Functions

The schematic of a quadrilateral element is shown in Fig. 3.1. The bilinear Lagrangian element is considered as a combination of two linear elements in two perpendicular directions,  $r$  and  $s$ , in element coordinate system.

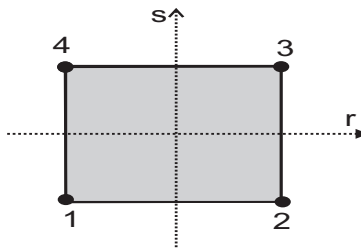


Figure 3.1: Quadrilateral Lagrangian Element

Considering a linear approximation of trial function, the element shape function are expressed as

$$N_1(r, s) = \left[ \frac{1}{2}(1-r) \right] \left[ \frac{1}{2}(1-s) \right] \quad (3.1)$$

$$N_2(r, s) = \left[ \frac{1}{2}(1+r) \right] \left[ \frac{1}{2}(1-s) \right] \quad (3.2)$$

$$N_3(r, s) = \left[ \frac{1}{2}(1+r) \right] \left[ \frac{1}{2}(1+s) \right] \quad (3.3)$$

$$N_4(r, s) = \left[ \frac{1}{2}(1-r) \right] \left[ \frac{1}{2}(1+s) \right] \quad (3.4)$$

where  $r$  and  $s$  denote the natural coordinates of the element. For an iso-parametric element, each Cartesian coordinate is interpolated using

$$x = N_1(r, s)x_1 + N_2(r, s)x_2 + N_3(r, s)x_3 + N_4(r, s)x_4 \quad (3.5)$$

$$y = N_1(r, s)y_1 + N_2(r, s)y_2 + N_3(r, s)y_3 + N_4(r, s)y_4 \quad (3.6)$$

### 3.2.2 Implementation

The partial derivatives with respect to the global coordinates is mapped to the partial derivatives with respect to the natural coordinates through

$$\begin{Bmatrix} \frac{\partial}{\partial r} \\ \frac{\partial}{\partial s} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{Bmatrix} \quad (3.7)$$

where

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} \quad (3.8)$$

Considering the strain-displacement relations, (2.30) and (2.31), the  $\mathbf{B}$  matrix can be determined from

$$\mathbf{B} = \begin{bmatrix} J_{11} \frac{\partial N_1}{\partial r} + J_{12} \frac{\partial N_1}{\partial s} & 0 \\ 0 & J_{12} \frac{\partial N_1}{\partial r} + J_{22} \frac{\partial N_1}{\partial s} \\ J_{12} \frac{\partial N_1}{\partial r} + J_{22} \frac{\partial N_1}{\partial s} & J_{11} \frac{\partial N_1}{\partial r} + J_{12} \frac{\partial N_1}{\partial s} \end{bmatrix} \quad (3.9)$$

where  $J_{11}$ ,  $J_{12}$ ,  $J_{21}$ , and  $J_{22}$  are components of  $\mathbf{J}^{-1}$ . Finally, the element stiffness matrix is computed by

$$\mathbf{K}^e = \int_{-1}^{+1} \int_{-1}^{+1} \mathbf{B}^T \mathbf{D} \mathbf{B} \det(\mathbf{J}) \, dr \, ds \quad (3.10)$$

which is calculated using Gauss quadrature rule. The simplest two-dimensional Gauss rules are called product rules. They are obtained by applying the one-dimensional rules to each independent variable in turn. Therefore,

$$\int_{-1}^{+1} \int_{-1}^{+1} f(r, s) \, dr \, ds = \int_{-1}^{+1} dr \int_{-1}^{+1} f(r, s) \, ds \approx \sum_{i=1}^{ngp_i} \sum_{j=1}^{ngp_j} w_i w_j f(r_i, s_j) \quad (3.11)$$

where  $ngp_i$  and  $ngp_j$  are the number of Gauss points in the  $r$  and  $s$  directions,  $w_i$  and  $w_j$  are the corresponding weights, and  $f(r, s)$  is a generic function. Usually the same number  $ngp_i = ngp_j$  is chosen if the shape functions are taken to be the same in the  $r$  and  $s$  directions, which is the case here.

### 3.3 The Quadratic Triangular Element

#### 3.3.1 Geometry

The geometry of the six-node triangle shown in Fig. 3.2 is specified by the location of its three corner nodes on the  $x, y$  plane. The corner nodes are labeled 1, 2, 3 while traversing the sides in counterclockwise fashion. Then, the middle nodes are numbered in the same fashion providing that node 4 is placed between 1 and 2.

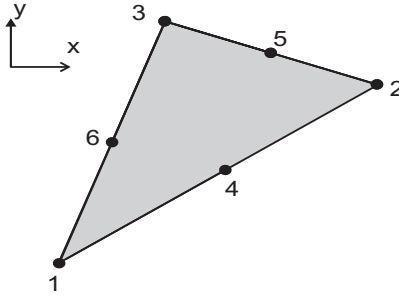


Figure 3.2: Quadratic Triangular Element

#### 3.3.2 Element Shape Functions

Consider the parent three-node triangular element shown in Fig. 3.3 with an arbitrary point  $p$  inside the element.

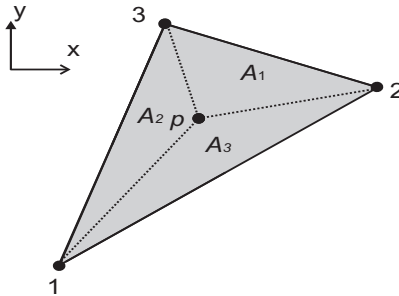


Figure 3.3: Triangular Coordinate

The area of the triangle can be expressed as

$$A = A_1 + A_2 + A_3$$

Dividing both sides by  $A$  results in

$$1 = \frac{A_1}{A} + \frac{A_2}{A} + \frac{A_3}{A}$$

or

$$1 = L_1 + L_2 + L_3$$

where

$$L_1 = \frac{A_1}{A} \quad L_2 = \frac{A_2}{A} \quad L_3 = \frac{A_3}{A}$$

Then, the natural coordinates of  $p$  is defined with any of  $L_1$  and  $L_2$  or  $L_1$  and  $L_3$  or  $L_2$  and  $L_3$ . Defining

$$N_I = L_I, \quad I = 1, 2, 3 \quad (3.12)$$

$N$  has the required properties of shape functions namely

$$N_1 + N_2 + N_3 = 1$$

and

$$N_I(L_J) = \delta_{IJ}$$

Now, if we add middle nodes we require that each shape function corresponding to the corner nodes to be zero in the middle and vice-versa. For example, for node 4 it is not hard to see that

$$L_1 = L_2 = \frac{L}{2}$$

and

$$L_3 = 0$$

Hence, we take

$$N_4 = 4L_1L_2$$

For node 1,  $N_1$  should be zero at 4 and 6, thus

$$N_1 = L_1(2L_1 - 1)$$

Therefore, element shape functions for six-node triangular element are given by

$$N_I = L_I (2L_I - 1), \quad I = 1, 2, 3 \quad (3.13)$$

$$N_4 = 4L_1L_2, \quad N_5 = 4L_3L_2, \quad N_6 = 4L_1L_3 \quad (3.14)$$

### 3.3.3 Implementation

The shape functions and their natural derivatives are

$$\underline{N}^T = \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \end{bmatrix} = \begin{bmatrix} L_1 (2L_1 - 1) \\ L_2 (2L_2 - 1) \\ L_3 (2L_3 - 1) \\ 4L_1L_2 \\ 4L_3L_2 \\ 4L_3L_1 \end{bmatrix} \quad (3.15)$$

$$\frac{\partial \underline{N}^T}{\partial L_1} = \begin{bmatrix} 4L_1 - 1 \\ 0 \\ 0 \\ 4L_2 \\ 0 \\ 4L_3 \end{bmatrix} \quad \frac{\partial \underline{N}^T}{\partial L_2} = \begin{bmatrix} 0 \\ 4L_2 - 1 \\ 0 \\ 4L_1 \\ 4L_3 \\ 0 \end{bmatrix} \quad \frac{\partial \underline{N}^T}{\partial L_3} = \begin{bmatrix} 0 \\ 0 \\ 4L_3 - 1 \\ 0 \\ 4L_2 \\ 4L_1 \end{bmatrix} \quad (3.16)$$

The physical coordinates  $x$  and  $y$  are interpolated using  $\underline{N}$  as

$$\begin{bmatrix} 1 \\ x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ y_1 & y_2 & y_3 & y_4 & y_5 & y_6 \end{bmatrix} \underline{N}^T \quad (3.17)$$

where  $(x_I, y_I)$  with  $I = 1, \dots, 6$  denote the global or physical coordinates of nodes. Additionally, we can write

$$\frac{\partial}{\partial x} = \frac{\partial L_i}{\partial x} \frac{\partial}{\partial L_i}, \quad \frac{\partial}{\partial y} = \frac{\partial L_i}{\partial y} \frac{\partial}{\partial L_i} \quad (3.18)$$

and

$$\frac{\partial x}{\partial L_i} = x_I \frac{\partial N_I}{\partial L_i}, \quad \frac{\partial y}{\partial L_i} = y_I \frac{\partial N_I}{\partial L_i} \quad (3.19)$$

where repeated indices show the summation convention. Combining, (3.17) and (3.19) and because

$$\frac{\partial x}{\partial x} = \frac{\partial y}{\partial y} = 1, \quad \frac{\partial 1}{\partial x} = \frac{\partial 1}{\partial y} = \frac{\partial x}{\partial y} = \frac{\partial y}{\partial x} = 0$$

we obtain

$$\begin{bmatrix} 1 & 1 & 1 \\ x_I \frac{\partial N_I}{\partial L_1} & x_I \frac{\partial N_I}{\partial L_2} & x_I \frac{\partial N_I}{\partial L_3} \\ y_I \frac{\partial N_I}{\partial L_1} & y_I \frac{\partial N_I}{\partial L_2} & y_I \frac{\partial N_I}{\partial L_3} \end{bmatrix} \begin{bmatrix} \frac{\partial L_1}{\partial x} & \frac{\partial L_1}{\partial y} \\ \frac{\partial L_2}{\partial x} & \frac{\partial L_2}{\partial y} \\ \frac{\partial L_3}{\partial x} & \frac{\partial L_3}{\partial y} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3.20)$$

The coefficient matrix of (3.20) will be *Jacobian matrix* and denoted by  $\mathbf{J}$ , and the Jacobian is

$$J = \frac{1}{2} \det \mathbf{J}$$

Introducing

$$\mathbf{JP} = \begin{bmatrix} 1 & 1 & 1 \\ x_I \frac{\partial N_I}{\partial L_1} & x_I \frac{\partial N_I}{\partial L_2} & x_I \frac{\partial N_I}{\partial L_3} \\ y_I \frac{\partial N_I}{\partial L_1} & y_I \frac{\partial N_I}{\partial L_2} & y_I \frac{\partial N_I}{\partial L_3} \end{bmatrix} \begin{bmatrix} \frac{\partial L_1}{\partial x} & \frac{\partial L_1}{\partial y} \\ \frac{\partial L_2}{\partial x} & \frac{\partial L_2}{\partial y} \\ \frac{\partial L_3}{\partial x} & \frac{\partial L_3}{\partial y} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3.21)$$

If  $J \neq 0$ , solving this system gives

$$\mathbf{P} = \begin{bmatrix} \frac{\partial L_1}{\partial x} & \frac{\partial L_1}{\partial y} \\ \frac{\partial L_2}{\partial x} & \frac{\partial L_2}{\partial y} \\ \frac{\partial L_3}{\partial x} & \frac{\partial L_3}{\partial y} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3.22)$$

Considering the definition of  $\mathbf{P}$ , the derivatives of the shape functions

$$\frac{\partial N_I}{\partial x} = \frac{\partial N_I}{\partial L_i} \frac{\partial L_i}{\partial x} \quad (3.23)$$

$$\frac{\partial N_I}{\partial y} = \frac{\partial N_I}{\partial L_i} \frac{\partial L_i}{\partial y} \quad (3.24)$$

yields finally the compact form

$$\begin{bmatrix} \frac{\partial N_I}{\partial x} & \frac{\partial N_I}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial N_I}{\partial L_1} & \frac{\partial N_I}{\partial L_2} & \frac{\partial N_I}{\partial L_3} \end{bmatrix} \mathbf{P} \quad (3.25)$$

### 3.3.4 Gauss Quadrature for Triangular Elements

Gauss quadrature rules for triangles must be symmetric, that is if the quadrature point  $(L_1, L_2, L_3)$  is present in the Gauss integration rule with weight  $w$ , then all other points obtainable by permuting the three triangular coordinates arbitrarily must appear in that rule, and have the same weight. The simplest Gauss rule for a triangle has one sample point located at the centroid. For a straight sided triangle

$$\frac{1}{A} \int_{A^e} f(L_1, L_2, L_3) dA \approx f\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right) \quad (3.26)$$

The next rule includes three sample points

$$\frac{1}{A} \int_{A^e} f(L_1, L_2, L_3) dA \approx \frac{1}{3} f\left(\frac{2}{3}, \frac{1}{6}, \frac{1}{6}\right) + \frac{1}{3} f\left(\frac{1}{6}, \frac{2}{3}, \frac{1}{6}\right) + \frac{1}{3} f\left(\frac{1}{6}, \frac{1}{6}, \frac{2}{3}\right) \quad (3.27)$$

For example, the numerically integrated stiffness matrix is

$$\mathbf{K}^e = \int_{A^e} \mathbf{B}^T \mathbf{D} \mathbf{B} dA \approx \sum_{gp_i=1}^{ng} w_i f(L_{1i}, L_{2i}, L_{3i}) \quad (3.28)$$

where  $f(L_1, L_2, L_3) = \mathbf{B}^T \mathbf{D} \mathbf{B}$  and  $L_{Ii}$  denotes the shape function  $L_I$  evaluated at Gauss point  $gp_i$ .

## 3.4 The Linear Tetrahedral (Solid) Element

### 3.4.1 Geometry

The linear tetrahedron, Fig. 3.4, is not usually used for stress analysis because of its poor performance. On the other hand, when objective is to compute primary variables, as in thermal analysis, the linear tetrahedron is acceptable. However, the element is very useful in introducing the primary steps of formulation of 3D solid elements.

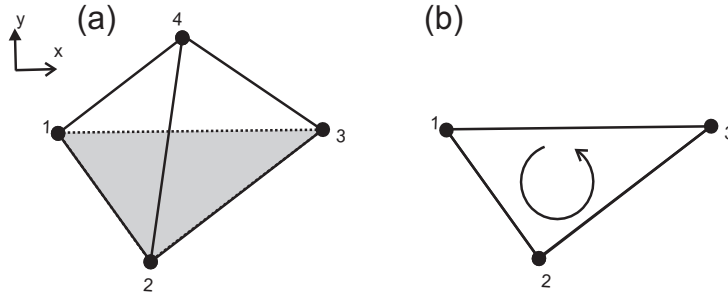


Figure 3.4: (a) The linear tetrahedron element: also called the 4-node tetrahedron; (b) Node numbering convention.

The geometry of the element is fully defined by the position of the four corner nodes:

$$(x_I, y_I, z_I) \quad I = 1, 2, 3, 4$$

The volume measure of the tetrahedron is denoted by  $\Gamma$  and is given by

$$\Gamma = \frac{1}{6} \det \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix} \quad (3.29)$$

With this definition it is seen that the volume is a signed quantity. A numbering rule that grants the positivity of this quantity is as follows:

1. Pick a face and number the nodes counterclockwise as for triangular elements.
2. The excluded corner will be numbered as the last one, 4 in this case. See Fig. 3.4.



### 3.4.2 Element Shape Functions

The set of tetrahedral coordinates  $(L_1, L_2, L_3, L_4,)$  is the three-dimensional analog of the triangular coordinate. Here, the coordinates are defined in terms of volume instead of area, however the process of deriving shape functions is the same as triangular elements except it should be interpreted in terms of volumes rather than areas.

### 3.4.3 Implementation

The sum of the four coordinates is identically one. The global coordinates  $x, y$  and  $z$  are mapped to the tetrahedral coordinates through

$$\begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_2 \\ L_4 \end{bmatrix} \quad (3.30)$$

Inverting this relation gives

$$\begin{bmatrix} L_1 \\ L_2 \\ L_2 \\ L_4 \end{bmatrix} = \frac{1}{6\Gamma} \begin{bmatrix} 6\Gamma_1 & a_1 & b_1 & c_1 \\ 6\Gamma_2 & a_2 & b_2 & c_2 \\ 6\Gamma_3 & a_3 & b_3 & c_3 \\ 6\Gamma_4 & a_4 & b_4 & c_4 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} \quad (3.31)$$

The values of  $a_I, b_I$  and  $c_I$  are computed using

$$\begin{aligned} a_1 &= y_2 z_{43} - y_3 z_{42} + y_4 z_{32} & a_2 &= -y_1 z_{43} + y_3 z_{41} - y_4 z_{31} \\ a_3 &= y_1 z_{42} - y_2 z_{41} + y_4 z_{21} & a_4 &= -y_1 z_{32} + y_2 z_{31} - y_3 z_{21} \\ b_1 &= -x_2 z_{43} + x_3 z_{42} - x_4 z_{32} & b_2 &= x_1 z_{43} - x_3 z_{41} + x_4 z_{31} \\ b_3 &= -x_1 z_{42} + x_2 z_{41} - x_4 z_{21} & b_4 &= x_1 z_{32} - x_2 z_{31} + x_3 z_{21} \\ c_1 &= x_2 y_{43} - x_3 y_{42} + x_4 y_{32} & c_2 &= -x_1 y_{43} + x_3 y_{41} - x_4 y_{31} \\ c_3 &= x_1 y_{42} - x_2 y_{41} + x_4 y_{21} & c_4 &= -x_1 y_{32} + x_2 y_{31} - x_3 y_{21} \end{aligned}$$

where the abbreviations  $x_{ij} = x_i - x_j$ ,  $y_{ij} = y_i - y_j$  and  $z_{ij} = z_i - z_j$  are used. From (3.30) and (3.31) it can readily be seen that

$$\frac{\partial x}{\partial L_I} = x_I, \quad \frac{\partial y}{\partial L_I} = y_I, \quad \frac{\partial z}{\partial L_I} = z_I \quad (3.32)$$

$$6\Gamma \frac{\partial L_I}{\partial x} = a_I, \quad 6\Gamma \frac{\partial L_I}{\partial y} = b_I, \quad 6\Gamma \frac{\partial L_I}{\partial z} = c_I \quad (3.33)$$

Therefore, the derivatives of the shape functions with respect to the global coordinates are expressed as

$$\begin{aligned} \frac{\partial N}{\partial x} &= \frac{\partial N}{\partial L_I} \frac{\partial L_I}{\partial x} = \frac{1}{6\Gamma} \frac{\partial N}{\partial L_I} a_I \\ \frac{\partial N}{\partial y} &= \frac{\partial N}{\partial L_I} \frac{\partial L_I}{\partial y} = \frac{1}{6\Gamma} \frac{\partial N}{\partial L_I} b_I \\ \frac{\partial N}{\partial z} &= \frac{\partial N}{\partial L_I} \frac{\partial L_I}{\partial z} = \frac{1}{6\Gamma} \frac{\partial N}{\partial L_I} c_I \end{aligned} \quad (3.34)$$

The displacement field over the tetrahedron is defined by  $u, v$  and  $w$ . These are linearly interpolated over the element from their nodal values

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \\ v_1 & v_2 & v_3 & v_4 \\ w_1 & w_2 & w_3 & w_4 \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{bmatrix} \quad (3.35)$$

Then, the strain and stress fields can be obtained using (2.30) and (2.31). Assuming elastic moduli is constant inside the element, the stiffness matrix can be realized using (2.36) as

$$\mathbf{K}^e = \Gamma \mathbf{B}^T \mathbf{D} \mathbf{B} \quad (3.36)$$

The element stiffness matrix is  $12 \times 12$ . For linear tetrahedral element it can be directly evaluated in closed form or by a one-point (centroid) integration rule.

# Chapter 4

## Results

The bilinear quadrilateral, quadratic triangular, and linear tetrahedral elements are used to solve a similar problem. A thin plate is considered loaded uniformly at one end shown in Fig. 4.1. The thickness is  $0.03\text{ m}$  and other dimensions can be seen in Fig. 4.1. Material properties are Young's modulus  $E = 10^8\text{ Pa}$  and Poisson's ratio  $\nu = 0.3$ . The external load is  $q = 2000\text{ kN/m}^2$ .

### 4.1 The Bilinear Quadrilateral Element

The domain is subdivided into two elements only for illustration purposes. Finite element mesh and nodal coordinates in meters are shown in Fig. 4.2. The total force due to the distributed load is divided between nodes 3 and 6 equally, i.e.  $7.5\text{ kN}$  per node. Because the plate is thin a plane stress assumption is valid. The constitutive matrix is obtained from (2.11). The source code is presented in Appendix A.1.

The global nodal displacement vector and the global nodal force vector are given in Table 4.1. The real computed displacements are obtained by multiplying each value with  $1e - 5$ . The number of elements must be increased to get reliable results, although for this simple problem a few elements suffice. The horizontal and vertical forces at node 1 are forces of  $7.5000\text{ kN}$  (directed to the left) and  $1.5793\text{ kN}$  (directed downwards). The horizontal and vertical forces at node 4 are forces of  $7.5000\text{ kN}$  (directed to the left) and  $1.5793\text{ kN}$  (directed upwards). Clearly, the force equilibrium is satisfied with these results.

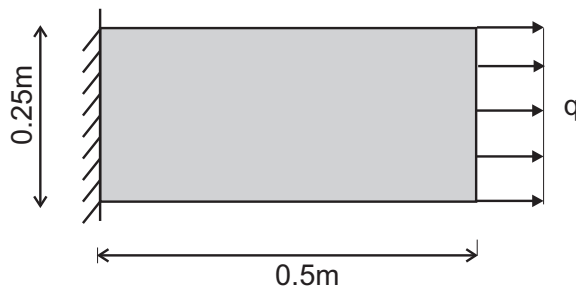


Figure 4.1: Thin plate under uniformly distributed load.

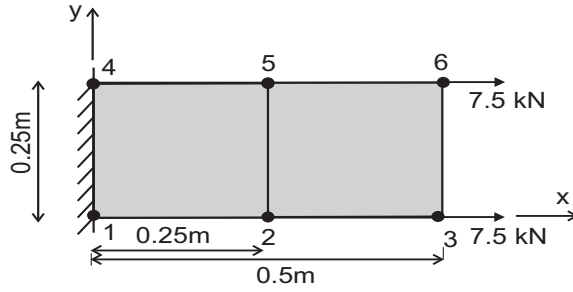


Figure 4.2: Thin plate mesh using quadrilateral elements.

Table 4.1: Global force and displacement vectors

Node	Direction	Force	Displacement
1	x	-7.5000	0
1	y	-1.5793	0
2	x	-0.0000	0.4815
2	y	-0.00000	0.0885
3	x	7.5000	0.9842
3	y	0.0000	0.0704
4	x	-7.5000	0
4	y	1.5793	0
5	x	0.0000	0.4815
5	y	0.0000	-0.0885
6	x	7.5000	0.9842
6	y	-0.0000	-0.0704

## 4.2 The Quadratic Triangular Element

Initially, the domain is discretized to two elements. Finite element mesh and nodal coordinates in meters are shown in Fig. 4.3. The total force due to the distributed load is divided between side nodes equally, i.e.  $5\text{ kN}$  per node. Because the plate is thin a plane stress assumption is valid. The constitutive matrix is obtained from (2.11). The source code is presented in Appendix A.2.

The global nodal displacement vector and the global nodal force vector are given in Table 4.2. The actual computed displacements are obtained by multiplying each value with  $1e-4$ . Obviously, force equilibrium is satisfied for this problem. However, the number of elements must be increased to get reliable results.

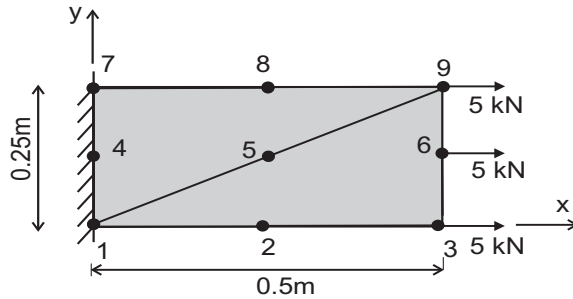


Figure 4.3: Thin plate mesh using quadratic triangular elements.

Table 4.2: Global force and displacement vectors

Node	Direction	Force	Displacement
1	x	-2.7886	0
1	y	-1.3052	0
2	x	0.0000	0.0509
2	y	-0.0000	0.0077
3	x	5.0000	0.1087
3	y	0.0000	0.0137
4	x	-9.4228	0
4	y	-0.3339	0
5	x	-0.0000	0.0478
5	y	0.0000	0.0016
6	x	5.0000	0.0936
6	y	-0.0000	0.0017
7	x	-2.7886	0
7	y	1.6391	0
8	x	-0.0000	0.0493
8	y	-0.0000	-0.0061
9	x	5.0000	0.1082
9	y	-0.0000	-0.0065

### 4.3 The Linear Tetrahedral Element

As said before, the linear tetrahedral (solid) element is a three-dimensional finite element with both local, tetrahedral coordinates, and global coordinates. Each linear tetrahedral has four nodes with three degrees of freedom at each node. The same domain as before is considered, Fig. 4.1, but five tetrahedral are fit to the domain Fig. 4.4. Notice the global system of coordinates where y-axis is along horizontal direction. The distributed load is divided by four to give an equal contribution,  $3.75 \text{ kN}$ , to each side node, 4, 3, 7 and 8. The material and geometric properties are kept constant. The corresponding finite element code is given in Appendix A.3.

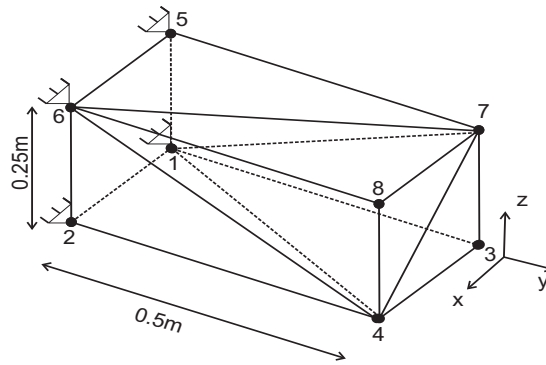


Figure 4.4: Thin plate mesh using linear tetrahedral elements. The thickness has been magnified for illustration purpose.

The global nodal displacement vector and the global nodal force vector are given in Table 4.3. Multiplying each value of the displacements in the table with  $1e - 5$  gives the actual computed displacements. It can be easily verified that force equilibrium is satisfied for this problem. However, the number of elements

Table 4.3: Global force and displacement vectors

Node	Direction	Force	Displacement
1	x	-21.0547	0
1	y	-4.2779	0
1	z	-7.2612	0
2	x	20.4792	0
2	y	-3.2221	0
2	z	-2.4652	0
3	x	-0.0000	-0.0005
3	y	3.7500	0.8565
3	z	0.0000	0.0173
4	x	0.0000	-0.0211
4	y	3.7500	0.8511
4	z	-0.0000	0.0106
5	x	-20.4792	0
5	y	-3.2221	0
5	z	2.4652	0
6	x	21.0547	0
6	y	-4.2779	0
6	z	7.2612	0
7	x	0.0000	0.0211
7	y	3.7500	0.8511
7	z	-0.0000	-0.0106
8	x	-0.0000	0.0005
8	y	3.7500	0.8565
8	z	0	-0.0173

must be increased to get reliable results.

## 4.4 Discussion

Using bilinear quadrilateral element the horizontal displacement at node 3 or 6 is  $0.09842 \times 10^{-4}$ . At nodes 3, 6 and 9 in second analysis using quadratic triangular element the displacements are  $0.1087 \times 10^{-4}$ ,  $0.0936 \times 10^{-4}$  and  $0.1082 \times 10^{-4}$  respectively. Implementing three- dimensional linear solid element, gives  $0.08565 \times 10^{-4}$ ,  $0.08511 \times 10^{-4}$ ,  $0.08511 \times 10^{-4}$  and  $0.08565 \times 10^{-4}$  at nodes 3, 4, 7 and 8 respectively. As it was expected using different elements give close results because the analysis is not complicated and the primary dependent variable, displacement, is compared. However, for better results increasing the number of elements is necessary.

Some additional analysis are down using quadrilateral elements increasing the number of elements to 8. The displacement results for one of the side nodes (the other one is the same) are summarized in Table 4.4. The convergence of the result is as expected. It could be seen that after increasing the number of elements the displacements approach the results using the quadratic triangular element as the latter is a higher order element. The actual computed values are obtained by multiplying the numbers by  $10^{-4}$ .

Table 4.4: Convergence of the result for quadrilateral element

No. of elements	1	2	4	8
Displacement in side nodes	0.09744	0.09842	0.09895	0.09913

## Part II

# The Nonlinear Computational Mechanics

# Chapter 5

## Nonlinear Finite Element Method

### 5.1 Introduction

Commonly, mechanics problems contain nonlinearities, and except for simple cases, exact analytical solutions cannot be obtained. The nonlinear finite element method is a powerful technique in solving diverse physical and engineering problems. In the following sections, different types of nonlinearities and solution procedures are explained in short, a very common nonlinearity encountered in analysis of plates and beams will be discussed in detail and the results of finite element analysis of Timoshenko beams are provided for two cases.

### 5.2 Nonlinearities in Solid and Structural Mechanics

Naturally, nonlinearities are present in any structure. However, under some assumptions such as infinitesimal deformation the nonlinear structure is approximated by the linear one, which is valid for many practical purposes. Structural nonlinearities are classified into two categories: geometric and material, though some other classifications may separate kinematic or boundary nonlinearities. Nonlinearities caused by several factors are grouped as [4]:

1. Large displacements where the original equilibrium equations should be updated, that is the geometry should be updated and stiffness matrix should be re-calculated at each step.
2. Large rotations which results in a nonlinear force- displacement relationship, and incremental effect is computed as the geometric stiffness matrix.
3. Nonlinear constitutive law in some materials such as rubber-like materials and composites.
4. Large strain in plastics, some metals and rubbers.

The first two items are associated with large deformation category, involving geometrical nonlinearity, whereas the last two items belong to the material nonlinearity.

### 5.3 Solution Procedures

For linear problems the stiffness matrix is constant, that is if the load is doubled, displacement is doubled too. However, an application of the finite element method to the nonlinear problems leads to a set of nonlinear algebraic equations, that is the stiffness matrix (or tangent stiffness matrix) will change at



each step. Therefore, using a method for computing the tangent stiffness matrix at each increment is necessary.

### 5.3.1 Geometric Nonlinearity

In inspecting the geometric nonlinearity, the Newton-Raphson method is probably the most popular one. Assuming  $\underline{d}^i$ , which is displacement at  $i$ -th iteration, is known we are looking for  $\underline{d}^{i+1}$ . The Newton-Raphson method is the fastest solution method, but there is no guarantee for convergence if the initial guess is far from the solution. Additionally, the tangent stiffness matrix should be calculated at each step which is costly in terms of computation. The matrix equation solves for incremental displacement  $\Delta\underline{d}^i$ , and the result is used to update the displacement until residual, the difference between internal and external force vectors, gets close enough to zero (or becomes less than a prescribed tolerance).

An obvious modification to this solution procedure is to keep the original tangent stiffness. The tangent stiffness is updated at the first step of each increment and is maintained constant up to convergence. There are some other variations of the method such as the initial stress method of solution which takes the procedure one stage further and only uses the stiffness matrix from the very first incremental solution.

### 5.3.2 Material Nonlinearity

Several methods are implemented to deal with material nonlinearity. For example, the Prandtl-Russ equation for the plastic strain increments is combined with the von Mises yield criteria for material characterization. An iterative procedure is then employed for the solution of the associated static problem [4].

## Chapter 6

# Finite Element Analysis of Nonlinear Timoshenko Beams

### 6.1 Introduction

The Euler-Bernoulli theory of beams is based on the assumption that a material plane that is normal to the neutral axis before deformation remains normal to the neutral axis after deformation. Furthermore, normals remain straight (they do not bend), and keep the same length (they do not extend)<sup>1</sup>. Therefore, the effects of shear deformation are neglected, while in many situations such as for stubby beams this contribution cannot be overlooked. Accordingly, the Timoshenko's theory of beams has been introduced as a means of accounting for the effects of shear in a simple manner.

The aim here is to provide a geometric nonlinear analysis of Timoshenko beams under the assumption of small rotations and implement the finite element formulations using Lagrangian and trigonometric basis functions. Analysis are carried out for clamped straight and a quarter curved beams. Where appropriate, answers are compared to exact solutions.

### 6.2 Background and Basic Equations

The Timoshenko beam theory is associated with the Reissner-Mindlin plate theory, in which the effects of transverse shear are taken into account while formulating the behavior of rectangular plates. The theory was given for statics by Reissner and extended to dynamics by Mindlin. A significant result of using Reissner-Mindlin plate theory is that shape functions require  $C^0$  continuity, however,  $C^0$  shear-flexible continuous elements are susceptible to shear locking which can decrease the performance of these elements. The discretized equations are derived for straight and curved beams separately, the final form of the tangent stiffness matrices are presented and reduced integration schemes are introduced in order to avoid shear and membrane locking.

#### 6.2.1 Straight Beams

Analysis of Timoshenko beam can be considered as a particular case of plate analysis using Reissner-Mindlin plate theory, where the dimension is reduced to one. It should be noted that a proper statement

---

<sup>1</sup>Normals are lines perpendicular to the beam's neutral plane and are thus embedded in the beam's cross sections.

for shear stress in the beam is expressed as

$$\sigma_{xz} = G\gamma(x, z) \quad (6.1)$$

where  $G$  is the shear modulus and  $\gamma(x, z)$  gives the shear angle at any point inside the beam. Notice that, the shear angle is different from the total slop of the beam. The total slop  $dw/dx$  of the midline emanating from shear deformation and bending deformation can then be given as the sum of two parts as follows

$$\frac{dw}{dx} = \varphi(x) + \gamma(x) \quad (6.2)$$

where  $\varphi(x)$  is the rotation of line elements along the midline due to bending only. Therefore, the  $z$ -dependence of the shear angle is surpassed for the sake of simplicity in a one-dimensional beam approach. In order to include the nonuniform shear stress distribution at a section without losing the simplicity, the corresponding shear stress-strain relation, (6.1), is modified to

$$\sigma_{xz} = kG\gamma(x) \quad (6.3)$$

where  $k$  is the shear correction factor. Cowper [1] gives the best explanation of the shear correction factor. Suffice to say here that  $k$  is a function of the cross section and, may also be a function of the Poisson's ratio. The curvilinear coordinates are used with the following nonlinear form of the strain-displacement relations

$$\epsilon_{xx} = \frac{\partial u}{\partial x} + \frac{1}{2} \left( \frac{\partial w}{\partial x} \right)^2 \quad (6.4)$$

$$\gamma_{xz} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \quad (6.5)$$

where  $x$  and  $z$  are perpendicular directions in an orthogonal coordinate system such that  $x$  lies along the neutral axis, or any other reference line such as the centroidal axis of the beam, and  $z$  is normal to the  $x$  at any cross-section. The tangential displacement  $u$  is in the  $x$ -direction, and normal deflection is denoted by  $w$  in the  $z$  direction. As previously explained, the  $z$ -dependence of transverse deflection,  $\gamma$ , will be ignored. Thus, the displacements are given by

$$u(x, z) = \bar{u}(x) + z\phi_x(x) \quad (6.6)$$

$$w(x, z) = \bar{w}(x) \quad (6.7)$$

where  $\bar{u}$  indicates a tangential displacement to the midline measured at any point on this line,  $\bar{w}$  is the measure of normal deflection and  $\phi_x$  is the measure of rotation, Fig. 6.1. Upon substitution of (6.7) in (6.3) and (6.5), the strain-displacement relations are taking the following form

$$\epsilon_{xx} = \frac{\partial \bar{u}}{\partial x} + z \frac{\partial \phi_x}{\partial x} + \frac{1}{2} \left( \frac{\partial \bar{w}}{\partial x} \right)^2 \quad (6.8)$$

$$\gamma_{xz} = \frac{\partial \bar{w}}{\partial x} + \phi_x \quad (6.9)$$

or

$$\epsilon = \begin{bmatrix} \frac{\partial \bar{u}}{\partial x} + z \frac{\partial \phi_x}{\partial x} + \frac{1}{2} \left( \frac{\partial \bar{w}}{\partial x} \right)^2 \\ \frac{\partial \bar{w}}{\partial x} + \phi_x \end{bmatrix} \quad (6.10)$$

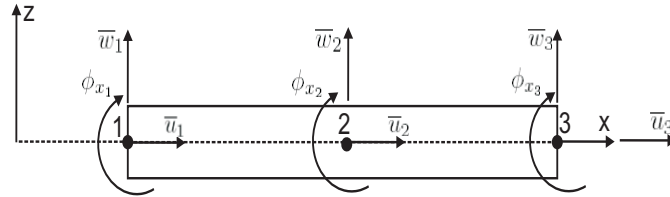


Figure 6.1: The straight beam element

where  $\underline{\epsilon} = \{\epsilon_{xx}, \gamma_{xz}\}^T$ . Then, (6.10) can be decomposed to get

$$\underline{\epsilon} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial x} & 1 \end{bmatrix} \underline{d} + z \begin{bmatrix} 0 & 0 & \frac{\partial}{\partial x} \\ 0 & 0 & 0 \end{bmatrix} \underline{d} + \frac{1}{2} \begin{bmatrix} \frac{\partial \bar{w}}{\partial x} \\ 0 \end{bmatrix} \begin{bmatrix} 0 & \frac{\partial \bar{w}}{\partial x} & 0 \end{bmatrix} \quad (6.11)$$

where  $\underline{d} = \{\bar{u}, \bar{w}, \phi\}^T$ . Therefore, the strain-displacement relations can be divided to linear and nonlinear parts. The nonlinear part of strain-displacement relation can be rearranged as

$$\underline{\epsilon}_N = \frac{1}{2} \begin{bmatrix} \frac{\partial \bar{w}}{\partial x} \\ 0 \end{bmatrix} \begin{bmatrix} 0 & \frac{\partial}{\partial x} & 0 \end{bmatrix} \underline{d} \quad (6.12)$$

This procedure results in the following form of the strain-displacement relation where the linear part is separated from the nonlinear part for further manipulations

$$\underline{\epsilon} = \left( \mathbf{B}'_{L_0} + \mathbf{B}'_{L_1} z + \frac{1}{2} \mathbf{B}'_N \Theta \right) \underline{d} \quad (6.13)$$

$$= \left[ \sum_{i=0}^1 \mathbf{B}'_{L_i} z^i \right] \underline{d} + \frac{1}{2} \left[ \mathbf{B}'_N(\underline{d}) \Theta \right] \underline{d} \quad (6.14)$$

where

$$\mathbf{B}'_{L_0} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial x} & 1 \end{bmatrix} \quad (6.15)$$

$$\mathbf{B}'_{L_1} = \begin{bmatrix} 0 & 0 & \frac{\partial}{\partial x} \\ 0 & 0 & 0 \end{bmatrix} \quad (6.16)$$

$$\mathbf{B}'_N = \begin{bmatrix} \frac{\partial \bar{w}}{\partial x} \\ 0 \end{bmatrix} \quad (6.17)$$

$$\Theta = \begin{bmatrix} 0 & \frac{\partial}{\partial x} & 0 \end{bmatrix} \quad (6.18)$$

## 6.2.2 Curved Beams

Straight beams can be considered as a special case of the curved beams with zero curvature. The following form of strain-displacement relations are to be considered for curved beams [6]

$$\epsilon_{xx} = \frac{1}{\kappa} \frac{\partial u}{\partial x} + \frac{w}{\kappa} \frac{\partial \kappa}{\partial z} + \frac{1}{2} \left( \frac{1}{\kappa} \frac{\partial w}{\partial x} \right)^2 \quad (6.19)$$

$$\gamma_{xz} = \frac{1}{\kappa} \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} - \frac{u}{\kappa} \frac{\partial \kappa}{\partial z} \quad (6.20)$$

where all variables have been introduced before, see Sec. 6.2.1, except  $\kappa$ . The  $\kappa$  parameter which is named Lamé coefficient will be defined as

$$\kappa = \alpha(1 + \rho z) \quad (6.21)$$

where curvature  $\rho$  is defined to be  $1/R$  with  $R$  is the radius of the curved beam element, Fig. 6.2. Notice that  $\rho = 0$  and  $\alpha = 1$  for straight beams. Making use of (6.7), results in strain expressions to become

$$\epsilon_{xx} = \frac{1}{\kappa} \frac{\partial \bar{u}}{\partial x} + \frac{\bar{w}}{\kappa} \frac{\partial \kappa}{\partial z} + \frac{z}{\kappa} \frac{\partial \phi}{\partial x} + \frac{1}{2} \left( \frac{1}{\kappa} \frac{\partial \bar{w}}{\partial x} \right)^2 \quad (6.22)$$

$$\gamma_{xz} = \frac{1}{\kappa} \frac{\partial \bar{w}}{\partial x} + \frac{\bar{u}}{\partial z} - \frac{\bar{u}}{\kappa} \frac{\partial \kappa}{\partial z} + \phi - \frac{z}{\kappa} \left( \phi \frac{\partial \kappa}{\partial z} \right) \quad (6.23)$$

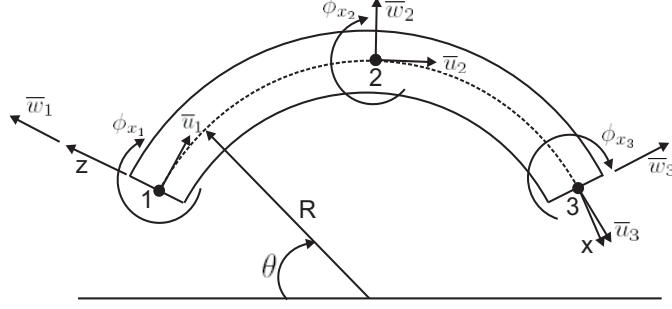


Figure 6.2: The curved beam element

To consider thick as well as thin beams a  $1/\kappa$  is factored out from strain-displacement relations and replaced by it's binomial expansion as [6]

$$\frac{1}{\kappa} = \frac{1}{\alpha} (1 - \rho z + \rho^2 z^2 - \dots) \quad (6.24)$$

which is subsequently truncated to terms of  $\mathbf{O}(z^2)$ . Then, the strain-displacements can be again re-expressed the same as (6.14) with some slight modifications. Therefore, the final relation is given by

$$\underline{\epsilon} = \left[ \sum_{i=0}^2 B'_{L_i} z^i \right] \underline{d} + \frac{1}{2} \left[ \sum_{i=0}^2 B'_{N_i} z^i(d) \right] \Theta \underline{d} \quad (6.25)$$

where

$$B'_{L_0} = \begin{bmatrix} \frac{1}{\alpha} \frac{\partial}{\partial x} & \rho & 0 \\ -\rho & \frac{1}{\alpha} \frac{\partial}{\partial x} & 1 \end{bmatrix} \quad (6.26)$$

$$B'_{L_1} = \begin{bmatrix} -\frac{\rho}{\alpha} \frac{\partial}{\partial x} & -\rho^2 & \frac{1}{\alpha} \frac{\partial}{\partial x} \\ \rho^2 & -\frac{\rho}{\alpha} \frac{\partial}{\partial x} & -\rho \end{bmatrix} \quad (6.27)$$

$$B'_{L_2} = \begin{bmatrix} \frac{\rho^2}{\alpha} \frac{\partial}{\partial x} & \rho^3 & -\frac{\rho}{\alpha} \frac{\partial}{\partial x} \\ -\rho^3 & \frac{\rho^2}{\alpha} \frac{\partial}{\partial x} & \rho^2 \end{bmatrix} \quad (6.28)$$

$$B'_{N_0} = \begin{bmatrix} \frac{1}{\alpha^2} \frac{\partial \bar{w}}{\partial x} \\ 0 \end{bmatrix} \quad (6.29)$$

$$B'_{N_1} = \begin{bmatrix} \frac{-2\rho}{\alpha^2} \frac{\partial \bar{w}}{\partial x} \\ 0 \end{bmatrix} \quad (6.30)$$

$$B'_{N_2} = \begin{bmatrix} \frac{3\rho^2}{\alpha^2} \frac{\partial \bar{w}}{\partial x} \\ 0 \end{bmatrix} \quad (6.31)$$

$$\Theta = \begin{bmatrix} 0 & \frac{\partial}{\partial x} & 0 \end{bmatrix} \quad (6.32)$$

### 6.3 Beam Element and Interpolation Functions

As in standard finite element analysis, the continuous displacement vector,  $\underline{d}$ , is replaced by a discrete approximation as

$$\underline{d} = \mathbf{N}\underline{d}_e \quad (6.33)$$

where

$$\mathbf{N} = [\mathbf{N}_1, \mathbf{N}_2, \dots, \mathbf{N}_{nne}] \quad (6.34)$$

where  $nne$  denotes the number of nodes per element and  $\underline{d}_e$  is the displacement vector for one element. Each block of the matrix of shape functions is given by

$$\mathbf{N}_I = \begin{bmatrix} N_I & 0 & 0 \\ 0 & N_I & 0 \\ 0 & 0 & N_I \end{bmatrix} \quad (6.35)$$

and  $I$  varies between one and the number of nodes per element. In order to treat curved beams as well as straight beams all degrees of freedom are to be modeled with the same order of basis functions. Trigonometric basis functions and Lagrange polynomial basis functions of order one through three are used. A typical Lagrange polynomial is given by

$$\Lambda_I^m(\xi) = \frac{\prod_{q=1, q \neq I}^{m+1} (\xi - \xi_q)}{\prod_{q=1, q \neq I}^{m+1} (\xi_I - \xi_q)} \quad (6.36)$$

where  $m$  denotes the order (degree) of the polynomial,  $I$  represents the local (element) node number, and  $\xi$  is the natural coordinate ranging from  $-1$  to  $+1$ . For a one-dimensional Lagrangian element containing  $n_{en}$  nodes, the interpolation function associated with node  $I$  will be the Lagrange polynomial of degree  $(n_{en} - 1)$  that takes on the value of one at node  $I$  and the value of zero at the remaining nodes. This is written as

$$N_I = \Lambda_I^{(n_{en}-1)} \quad I = 1, 2, \dots, n_{en} \quad (6.37)$$

This domain is easily mapped to the actual domain. A code is developed which is able to produce an interpolation function of any order, Fig. 6.3.

The trigonometric basis functions for an element with three nodes are given as [5]

$$\begin{aligned} N_1(\theta) &= \frac{\sin(\theta - \theta_2) - \sin(\theta - \theta_3) + \sin(\theta_2 - \theta_3)}{\sin(\theta_1 - \theta_2) - \sin(\theta_1 - \theta_3) + \sin(\theta_2 - \theta_3)} \\ N_2(\theta) &= \frac{\sin(\theta - \theta_3) - \sin(\theta - \theta_1) + \sin(\theta_3 - \theta_1)}{\sin(\theta_1 - \theta_2) - \sin(\theta_1 - \theta_3) + \sin(\theta_2 - \theta_3)} \\ N_3(\theta) &= \frac{\sin(\theta - \theta_1) - \sin(\theta - \theta_2) + \sin(\theta_1 - \theta_2)}{\sin(\theta_3 - \theta_1) - \sin(\theta_3 - \theta_2) + \sin(\theta_1 - \theta_2)} \end{aligned} \quad (6.38)$$

where  $\theta$  identifies the angular position of the nodes, Fig. 6.2. However, the trigonometric shape functions can be adapted to be used along with the curvilinear coordinates. Indeed, this is necessary when treating straight beams with the trigonometric interpolation functions. Therefore, the following forms of the

```

1   $\xi_1 = -1$ 
2   $\xi_{en} = 1$ 
3  if  $n_{en} > 2$ 
4      for  $I = 2 : (n_{en} - 1)$ 
5           $\xi_I = \xi_1 + (I - 1) \times (2 / (n_{en} - 1))$ 
6      end
7  end
8  for  $I = 1 : n_{en}$ 
9       $M = 1$           an arbitrary variable
10     for  $J = 1 : n_{en}$ 
11         if  $J \neq I$ 
12              $N_I = ((\xi - \xi_J) / (\xi_I - \xi_J)) \times M$ 
13              $M = N_I$ 
14         end
15     end
16 end

```

Figure 6.3: An automated code to produce Lagrangian basis functions.

trigonometric shape functions have been proposed to be used with the curvilinear distance,  $x$  [6]:

$$\begin{aligned}
 N_1(x) &= \frac{\sin \left[ \frac{2\pi n}{L}(x - x_2) \right] - \sin \left[ \frac{2\pi n}{L}(x - x_3) \right] + \sin \left[ \frac{2\pi n}{L}(x_2 - x_3) \right]}{\sin \left[ \frac{2\pi n}{L}(x_1 - x_2) \right] - \sin \left[ \frac{2\pi n}{L}(x_1 - x_3) \right] + \sin \left[ \frac{2\pi n}{L}(x_2 - x_3) \right]} \\
 N_2(x) &= \frac{\sin \left[ \frac{2\pi n}{L}(x - x_3) \right] - \sin \left[ \frac{2\pi n}{L}(x - x_1) \right] + \sin \left[ \frac{2\pi n}{L}(x_3 - x_1) \right]}{\sin \left[ \frac{2\pi n}{L}(x_1 - x_2) \right] - \sin \left[ \frac{2\pi n}{L}(x_1 - x_3) \right] + \sin \left[ \frac{2\pi n}{L}(x_2 - x_3) \right]} \\
 N_3(x) &= \frac{\sin \left[ \frac{2\pi n}{L}(x - x_2) \right] - \sin \left[ \frac{2\pi n}{L}(x - x_1) \right] + \sin \left[ \frac{2\pi n}{L}(x_1 - x_2) \right]}{\sin \left[ \frac{2\pi n}{L}(x_1 - x_2) \right] - \sin \left[ \frac{2\pi n}{L}(x_1 - x_3) \right] + \sin \left[ \frac{2\pi n}{L}(x_2 - x_3) \right]}
 \end{aligned} \tag{6.39}$$

where  $L$  is the total length of the finite element model of the beam and  $n$  is a parameter which governs the wave content of the trial solution. It can be shown that for a cantilever beam  $n = 0.25$  is a very good choice.

Replacing  $\underline{d}$  from (6.33) in (6.25) gives the discrete strain-displacement relation as

$$\underline{\epsilon} = \left( \left[ \sum_{i=0}^2 \mathbf{B}'_{L_i} z^i \right] + \frac{1}{2} \left[ \sum_{i=0}^2 \mathbf{B}'_{N_i} z^i(d) \right] \Theta \right) \underline{d} \tag{6.40}$$

$$= \left( \mathbf{B}'_L + \frac{1}{2} \mathbf{B}'_N \Theta \right) \underline{d} \tag{6.41}$$

$$= \left( \mathbf{B}'_L \mathbf{N} + \frac{1}{2} \mathbf{B}'_N \Theta \mathbf{N} \right) \underline{d}_e \tag{6.42}$$

$$= \left( \mathbf{B}'_L \mathbf{N} + \frac{1}{2} \mathbf{B}'_N \mathbf{N}' \right) \underline{d}_e \tag{6.43}$$

$$= \left( \mathbf{B}_L + \frac{1}{2} \mathbf{B}_N \right) \underline{d}_e \tag{6.44}$$

In case of straight beams  $\mathbf{B}'_{L_2}$ ,  $\mathbf{B}'_{N_1}$  and  $\mathbf{B}'_{N_2}$  will vanish, see (6.13). It can be readily seen that

$$\Theta \mathbf{N}_I = \begin{bmatrix} 0 & \frac{\partial}{\partial x} & 0 \end{bmatrix} \begin{bmatrix} N_I & 0 & 0 \\ 0 & N_I & 0 \\ 0 & 0 & N_I \end{bmatrix} \tag{6.45}$$

$$= \begin{bmatrix} 0 & \frac{\partial N_I}{\partial x} & 0 \end{bmatrix} \tag{6.46}$$

Therefore,

$$\Theta \mathbf{N} = \mathbf{N}' \quad (6.47)$$

Accordingly, the final form of the strain-displacement relation is given by

$$\underline{\epsilon} = \mathbf{B}_T \underline{d}_e \quad (6.48)$$

where

$$\mathbf{B}_T = \mathbf{B}_L + \frac{1}{2} \mathbf{B}_N \quad (6.49)$$

This approach has been introduced by Wood and Zienkiewicz [13] and further developed by Pica et al. [9] in nonlinear finite element analysis of different types of geometric nonlinearities seen in plates and shells.

## 6.4 Constitutive Relations

For a homogeneous isotropic material under the assumptions of Timoshenko beam theory the constitutive operator can be shown to be

$$\mathbf{D} = \begin{bmatrix} E & 0 \\ 0 & kG \end{bmatrix} \quad (6.50)$$

where  $E$  is elastic modulus,  $G$  is shear modulus, and  $k$  is the shear correction factor to allow for nonuniform shear stress distribution which has been introduced before in Sec. 6.2.

## 6.5 Finite Element Formulation of Equilibrium Equations

We start with strain energy of a straight beam, develop the stiffness matrices, and eventually generalize the method to treat curved beams as well. Replacing strain-displacement relations, (6.48), in the strain energy expression, (2.20), gives

$$U_e = \frac{1}{2} \int_{V_e} \underline{d}_e^T \mathbf{B}_T^T \mathbf{D} \mathbf{B}_T \underline{d}_e \, dV \quad (6.51)$$

Taking first variation of the strain energy and noting  $\mathbf{B}_N$  is a function of displacements leads to the following expression

$$\delta U_e = \int_{V_e} \delta \left( \underline{d}_e^T \mathbf{B}_T^T \right) \mathbf{D} \mathbf{B}_T \underline{d}_e \, dV \quad (6.52)$$

$$= \int_{V_e} \delta \left( \mathbf{B}_L \underline{d}_e + \frac{1}{2} \mathbf{B}_N \underline{d}_e \right)^T \mathbf{D} \mathbf{B}_T \underline{d}_e \, dV \quad (6.53)$$

$$= \int_{V_e} \delta \underline{d}_e^T \left( \mathbf{B}_L^T + \mathbf{B}_N^T \right) \mathbf{D} \mathbf{B}_T \underline{d}_e \, dV \quad (6.54)$$

$$= \int_{V_e} \delta \underline{d}_e^T \left( \mathbf{B}_L^T + \mathbf{B}_N^T \right) \underline{\sigma} \, dV \quad (6.55)$$

Therefore, the final form of the stress resultant force vector is given by

$$\underline{f}^{int}(\underline{d}_e) = \int_{V_e} \left( \mathbf{B}_L^T + \mathbf{B}_N^T \right) \underline{\sigma} \, dV \quad (6.56)$$

$$= \int_{V_e} \left( \overline{\mathbf{B}}_T^T \right) \underline{\sigma} \, dV \quad (6.57)$$



and the finite element equilibrium equation takes the following form

$$\underline{f}^{int}(\underline{d}_e) = \underline{f}^{ext} \quad (6.58)$$

where the equivalent nodal load vector  $\underline{f}^{ext}$  includes all nodal point forces, body forces and surface traction. Equation (6.58) can have the dual role of representing either the element, or in an assembled form the total equilibrium equation. It is a nonlinear equation in  $\underline{d}_e$  or  $\underline{d}$  since  $\mathbf{B}_N$  and  $\underline{\sigma}$  are functions of  $\underline{d}_e$  or  $\underline{d}$ . The nonlinear equations can be solved using an appropriate solution procedure such as the modified Newton-Raphson method.

### 6.5.1 Tangent Stiffness Matrix

The tangent stiffness matrix is defined as

$$\mathbf{K}_T = \frac{\partial \underline{R}}{\partial \underline{d}} = \frac{\partial}{\partial \underline{d}} \left( \underline{f}^{int}(\underline{d}) - \underline{f}^{ext} \right) \quad (6.59)$$

or

$$\mathbf{K}_T = \frac{\partial \underline{R}}{\partial \underline{d}} = \frac{\partial}{\partial \underline{d}} \left( \underline{f}^{int}(\underline{d}) \right) \quad (6.60)$$

where

$$\underline{R} = \underline{f}^{int}(\underline{d}_e) - \underline{f}^{ext} \quad (6.61)$$

and noting that  $\underline{f}^{ext}$  is constant in this case. The tangent stiffness matrix can be written as

$$\mathbf{K}_T = \begin{bmatrix} \frac{\partial R_1}{\partial d_1} & \frac{\partial R_1}{\partial d_2} & \cdots & \frac{\partial R_1}{\partial d_m} \\ \frac{\partial R_2}{\partial d_1} & \frac{\partial R_2}{\partial d_2} & \cdots & \frac{\partial R_2}{\partial d_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial R_m}{\partial d_1} & \frac{\partial R_m}{\partial d_2} & \cdots & \frac{\partial R_m}{\partial d_m} \end{bmatrix} \quad (6.62)$$

where  $m$  is the number of degrees of freedom per element ( $e$  subscript is removed for notational clarity). For convenience the stress resultants are written as

$$\underline{\sigma} = \{\sigma_1, \sigma_2\}^T \quad (6.63)$$

where  $\sigma_1 \equiv \sigma_{xx}$  and  $\sigma_2 \equiv \sigma_{xz}$ . Then, the equilibrium equation, (6.61) will be expanded as

$$\underline{R}(\underline{d}) = \int_{V^e} \left\{ \begin{array}{c} \overline{\mathbf{B}}_{T1k}^T \sigma_k \\ \overline{\mathbf{B}}_{T2k}^T \sigma_k \end{array} \right\} dV - \underline{f}^{ext} = 0 \quad (6.64)$$

or

$$\underline{R}(\underline{d}) = \int_{V^e} \left\{ \begin{array}{c} \overline{\mathbf{B}}_{Tk1} \sigma_k \\ \overline{\mathbf{B}}_{Tk2} \sigma_k \end{array} \right\} dV - \underline{f}^{ext} = 0 \quad (6.65)$$

where repeated indices means summation on the corresponding indices. Differentiating  $\underline{R}(\underline{d})$  with respect to the displacement vector gives the tangent stiffness matrix as

$$K_{Tij} = \int_{V^e} \frac{\partial \overline{\mathbf{B}}_{Tk1}}{\partial d_j} \sigma_k + \overline{\mathbf{B}}_{Tk1} \frac{\partial \sigma_{kk}}{\partial d_j} dV \quad (6.66)$$

or

$$\mathbf{K}_T = \int_{V^e} \overline{\mathbf{B}}_T^T \frac{\partial \underline{\sigma}}{\partial \underline{d}} + \underline{\sigma} \frac{\partial \overline{\mathbf{B}}_T^T}{\partial \underline{d}} dV \quad (6.67)$$

Therefore, the tangent stiffness matrix can be cast as

$$\mathbf{K}_T = \int_{V^e} \overline{\mathbf{B}}_T^T \nabla(\underline{\sigma}) + \Xi \overline{\mathbf{B}}_T dV \quad (6.68)$$

where

$$\nabla(\underline{\sigma}) = \begin{bmatrix} \frac{\partial \sigma_1}{\partial d_1} & \frac{\partial \sigma_1}{\partial d_2} & \cdots & \frac{\partial \sigma_1}{\partial d_m} \\ \frac{\partial \sigma_2}{\partial d_1} & \frac{\partial \sigma_2}{\partial d_2} & \cdots & \frac{\partial \sigma_2}{\partial d_m} \end{bmatrix} \quad (6.69)$$

and

$$\Xi = \begin{bmatrix} \sigma_1 \frac{\partial}{\partial d_1} & \sigma_2 \frac{\partial}{\partial d_1} \\ \sigma_1 \frac{\partial}{\partial d_2} & \sigma_2 \frac{\partial}{\partial d_2} \\ \vdots & \vdots \\ \sigma_1 \frac{\partial}{\partial d_m} & \sigma_2 \frac{\partial}{\partial d_m} \end{bmatrix} \quad (6.70)$$

Presenting stress-strain relation as

$$\underline{\sigma} = \mathbf{D} \underline{\epsilon} \quad (6.71)$$

or

$$\underline{\sigma} = \mathbf{D} \mathbf{B}_T \underline{d} \quad (6.72)$$

It could be readily seen that

$$\nabla(\underline{\sigma}) = \mathbf{D} \overline{\mathbf{B}}_T \quad (6.73)$$

Substituting this result into the first term inside the integral, (6.68), and making use of (6.44) yields

$$\mathbf{K}_T = \int_{V^e} \overline{\mathbf{B}}_T^T \mathbf{D} \overline{\mathbf{B}}_T dV \quad (6.74)$$

$$= \int_{V^e} (\mathbf{B}_L + \mathbf{B}_N)^T \mathbf{D} (\mathbf{B}_L + \mathbf{B}_N) dV \quad (6.75)$$

$$= \int_{V^e} \mathbf{B}_L^T \mathbf{D} \mathbf{B}_L dV + \int_{V^e} \mathbf{B}_L^T \mathbf{D} \mathbf{B}_N + \mathbf{B}_N^T \mathbf{D} \mathbf{B}_L + \frac{1}{2} \mathbf{B}_N^T \mathbf{D} \mathbf{B}_N dV \quad (6.76)$$

$$= \mathbf{K}_L + \mathbf{K}_N \quad (6.77)$$

As a result, the linear part of the stiffness matrix,  $\mathbf{K}_L$ , has been separated from the nonlinear part,  $\mathbf{K}_N$  which makes further manipulations easier. There is one term left in (6.68), to be evaluated. The initial stress matrix,  $\mathbf{K}_\sigma$ , is defined as

$$\mathbf{K}_S = \int_{V^e} \Xi \overline{\mathbf{B}}_T dV \quad (6.78)$$

Considering the fact that  $\mathbf{B}_L$  is not a function of  $\underline{d}_e$ , it follows that

$$\Xi \mathbf{B}_L = 0 \quad (6.79)$$

and

$$\mathbf{K}_S = \int_{V^e} \Xi \mathbf{B}_N dV \quad (6.80)$$

$$= \int_{V^e} \Xi \mathbf{B}'_N \mathbf{N}' dV \quad (6.81)$$

$$= \int_{V^e} \begin{bmatrix} \sigma_1 \frac{\partial}{\partial d_1} & \sigma_2 \frac{\partial}{\partial d_1} \\ \sigma_1 \frac{\partial}{\partial d_2} & \sigma_2 \frac{\partial}{\partial d_2} \\ \vdots & \vdots \\ \sigma_1 \frac{\partial}{\partial d_m} & \sigma_2 \frac{\partial}{\partial d_m} \end{bmatrix} \begin{bmatrix} \frac{1}{\kappa^2} \frac{\partial \bar{w}}{\partial x} \\ 0 \end{bmatrix} \mathbf{N}' dV \quad (6.82)$$

$$= \int_{V^e} \begin{bmatrix} \frac{\sigma_1}{\kappa^2} \frac{\partial}{\partial d_1} \frac{\partial \bar{w}}{\partial d_1} \\ \frac{\sigma_1}{\kappa^2} \frac{\partial}{\partial d_2} \frac{\partial \bar{w}}{\partial d_1} \\ \vdots \\ \frac{\sigma_1}{\kappa^2} \frac{\partial}{\partial d_m} \frac{\partial \bar{w}}{\partial d_1} \end{bmatrix} \mathbf{N}' dV \quad (6.83)$$

Note that for straight beams  $\kappa = 1$ . From discretization

$$\bar{w} = N_I \bar{w}_I \quad (6.84)$$

then,

$$\frac{\partial \bar{w}}{\partial x} = \bar{w}_I \frac{\partial N_I}{\partial x} \quad (6.85)$$

so,

$$\begin{aligned} \frac{\partial}{\partial d_J} \frac{\partial \bar{w}}{\partial x} &= \delta_{JI} \frac{\partial N_I}{\partial x} \\ &= \frac{\partial N_I}{\partial x} \end{aligned} \quad (6.86)$$

This leads the initial stress matrix to take the following form

$$\mathbf{K}_\sigma = \int_{V^e} \begin{bmatrix} 0 \\ \frac{\sigma_1}{\kappa^2} \frac{\partial N_I}{\partial x} \\ 0 \\ \vdots \\ 0 \\ \frac{\sigma_1}{\kappa^2} \frac{\partial N_m}{\partial x} \\ 0 \end{bmatrix} \mathbf{N}' dV \quad (6.87)$$

$$= \int_{V^e} \mathbf{N}'^T \frac{\sigma_1}{\kappa^2} \mathbf{N}' dV \quad (6.88)$$

$$(6.89)$$

Notice that  $dV = \kappa dz dy dx$ , where  $\kappa$  is the Lamé coefficient. The corresponding equation for straight beams is recovered replacing  $\alpha = 1$ ,  $\rho = 0$  and thus  $\kappa = 1$ . The stress vector is expressed as

$$\begin{aligned} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \end{Bmatrix} &= \mathbf{D} \left( \mathbf{B}_L + \frac{1}{2} \mathbf{B}_N \right) \underline{d}_e \\ &= \mathbf{D} \left[ \sum_{i=0}^2 \left( \mathbf{B}'_{L_i} \mathbf{N} + \frac{1}{2} \mathbf{B}'_{N_i} \mathbf{N}' \right) z^i \right] \underline{d}_e \end{aligned} \quad (6.90)$$

Substituting (6.90) in (6.89), and considering the appropriate form of the constitutive matrix in order to calculate the axial stress and replacing  $dV$  by  $\kappa dz dy dx$  and taking the following relations into account the initial stress matrix can be integrated using an appropriate numerical scheme.

$$A_{yz} = \int \int dy dz \quad (6.91)$$

$$\bar{z}A_{yz} = \int \int z dy dz \quad (6.92)$$

$$I_{yy} = \int \int z^2 dy dz \quad (6.93)$$

where  $A_{yz}$  is the area of the cross-section of the beam,  $\bar{z}$  is the distance between centroid and the reference line which is zero in this case, and  $I_{yy}$  is the second moment of inertia of the cross-section. It must be noted that the higher order  $z$  terms are truncated in order to be consistent with the truncation of the binomial expansion of the Lamé coefficient.

The linear and nonlinear stiffness matrices are computed in the same way as the initial stress matrix. Here, the linear stiffness matrix is calculated. Recall the linear stiffness matrix from (6.77), and substitute 6.90 in the matrix gives

$$\begin{aligned} \mathbf{K}_L &= \int \int \int \left( \sum_{i=0}^2 \mathbf{B}'_{L_i} \mathbf{N} z^i \right)^T \mathbf{D} \left( \sum_{i=0}^2 \mathbf{B}'_{L_i} \mathbf{N} z^i \right) \kappa dz dy dx \\ &= \int \int \int \left( \mathbf{B}'_{L_0} \mathbf{N} + \mathbf{B}'_{L_1} \mathbf{N} z + \mathbf{B}'_{L_2} \mathbf{N} z^2 \right)^T \mathbf{D} \\ &\quad \left( \mathbf{B}'_{L_0} \mathbf{N} + \mathbf{B}'_{L_1} \mathbf{N} z + \mathbf{B}'_{L_2} \mathbf{N} z^2 \right) \alpha (1 + \rho z) dz dy dx \end{aligned} \quad (6.94)$$

Upon expansion and neglecting higher order  $z$  terms, the linear stiffness matrix becomes

$$\begin{aligned} \mathbf{K}_L &= \alpha \int \left[ A_{yz} \left( \mathbf{B}'_{L_0} \mathbf{N} \right)^T \mathbf{D} \left( \mathbf{B}'_{L_0} \mathbf{N} \right) + I_{yy} \left( \mathbf{B}'_{L_0} \mathbf{N} \right)^T \mathbf{D} \left( \mathbf{B}'_{L_2} \mathbf{N} \right) \right. \\ &\quad \left. + I_{yy} \left( \mathbf{B}'_{L_1} \mathbf{N} \right)^T \mathbf{D} \left( \mathbf{B}'_{L_1} \mathbf{N} \right) + I_{yy} \left( \mathbf{B}'_{L_2} \mathbf{N} \right)^T \mathbf{D} \left( \mathbf{B}'_{L_0} \mathbf{N} \right) \right. \\ &\quad \left. + \rho \left( I_{yy} \left( \mathbf{B}'_{L_0} \mathbf{N} \right)^T \mathbf{D} \left( \mathbf{B}'_{L_1} \mathbf{N} \right) + I_{yy} \left( \mathbf{B}'_{L_1} \mathbf{N} \right)^T \mathbf{D} \left( \mathbf{B}'_{L_0} \mathbf{N} \right) \right) \right] dx \end{aligned} \quad (6.95)$$

The nonlinear stiffness matrix can be computed in exactly the same way as the linear one.

## 6.6 Reduced Numerical Integration

As previously mentioned, Sec. 6.2, using Lagrange polynomials of the same orders makes the element susceptible to shear or membrane locking, see Appendix B. In order to avoid this problem several reduced order integration schemes are used namely reduced shear integration, reduced membrane integration and fully reduced integration. If  $n$  is the degree of complete polynomials in an element's basis functions, Gauss integration with  $n + 1$  or more points in each direction is full and Gauss integration with  $n$  points or less in each direction is reduced.

It is worthy to note that different terms in the general expression for the stiffness matrix  $\mathbf{K}$  can be integrated using different integration formulae. Performing integration with different schemes for different terms in the stiffness matrix is called selective reduced integration (SRI). An advantage of the SRI is that it retains the correct rank of  $\mathbf{K}$ .

When both the membrane and shear contributions are integrated with a reduced formula, the practice is referred to as fully reduced integration. The benefit here is the reduction in computational efforts required for calculating  $\mathbf{K}$  since the number of quadrature points is directly changing the computational costs. The disadvantage of uniform reduced integration is that the rank of  $\mathbf{K}$  may be reduced, resulting in the singularity or near singularity of the global stiffness matrix.

For integrating trigonometric basis functions several methods could be employed. Here, a composite trapezoidal rule with a very large number of finite intervals has been chosen. The composite trapezoidal rule is given by [2]

$$\int_a^b f(x) dx = T(h) + R_T, \quad T(h) = \frac{h}{2}(f_0 + f_n) + h \sum_{i=1}^{n-1} f_i \quad (6.96)$$

The global truncation error is

$$R_T = -\frac{h^3}{12} \sum_{i=0}^{n-1} f''(\zeta_i) = -\frac{1}{12}(b-a)h^2 f''(\xi), \quad \xi \in [a, b] \quad (6.97)$$

Note that

$$x_0 = a, \quad x_i = x_0 + ih, \quad x_n = b \quad (6.98)$$

where  $h = (b - a)/n$  is the step length.

# Chapter 7

## Results

Two cantilever beams are considered with tip loads. The beams are loaded by a point load at the free end of the beam. The displacement vectors are computed and compared with the available analytical solutions. Additionally, a comparison between different Lagrange polynomials with trigonometric basis functions is made. The effects of reduced integration is also investigated. Finally, a comparison is made between the linear and nonlinear finite element analysis. The main files of the corresponding source code can be found in A.4.

### 7.1 Cantilever with vertical tip load

The vertical tip deflection of a cantilever of length  $L$  under a vertical tip load  $P$  is

$$w_{max} = \frac{PL^3}{3EI} \left( 1 + \frac{3EI}{kGAL^2} \right) \quad (7.1)$$

where  $w$  is the transverse displacement,  $E$  and  $G$  are the Young's and shear modulus,  $I$  is the second moment of inertia of the cross-section,  $A$  is the cross-section area and  $k$  is the shear correction factor, Fig. 7.1. A series of cantilever beams with different length,  $L$ , to depth,  $h$ , ratios have been considered,  $L/h \in \{4, 10, 50\}$ . The material and geometric input data are summarized in Table 7.1.

The results of finite element analysis of straight cantilever beams under Timoshenko assumptions for  $L/h = 4$  using different types of Lagrangian basis functions along with trigonometric basis functions is illustrated in Fig. 7.2. Both full and reduced integration schemes have been examined. As it was expected, cubic polynomial basis functions results in zero error, for any number of elements, regardless of the order of integration used, because the analytical solution for the displacement is a cubic polynomial in  $x$ . Using quadratic basis functions with the reduced shear (RS) or fully reduced (FR) integration



Figure 7.1: The cantilever straight beam.

Table 7.1: The material and geometric data of straight beams.

Parameter	Value	Dimension
Cross-sectional area $A_{yz}$	1	$in^2$
Second moment of inertia $I$	1/12	$in^4$
Length $L$	4, 10, 50	$in$
Shear correction factor $k$	5/6	-
Young's modulus $E$	$10^3$	$psi$
Poisson's ratio $\nu$	1/3	-

yields exact result. While, reduced membrane integration (RM) has no such effect, but improves the convergence rate.

The trigonometric basis functions demonstrate higher performance than most of the Lagrangian basis functions. The results for trigonometric basis functions have been obtained using  $n = 0.25$ . A simple comparison between several  $n$  values shows that  $n = 0.25$  gives the best result in this case. All these results agree with the results reported in [10] and [6], except our findings show that using a composite trapezoidal rule for integrating trigonometric functions results in a slower convergence in compare with [6].

It should be mentioned that the relative error is defined as

$$\text{Relative Error}\% = 100 \times \left| \frac{\tilde{d} - d}{d} \right| \quad (7.2)$$

where  $\tilde{d}$  is an approximate value whose exact value is  $d$ . A converged result in this case means the computed internal energy has reached to the true internal energy.

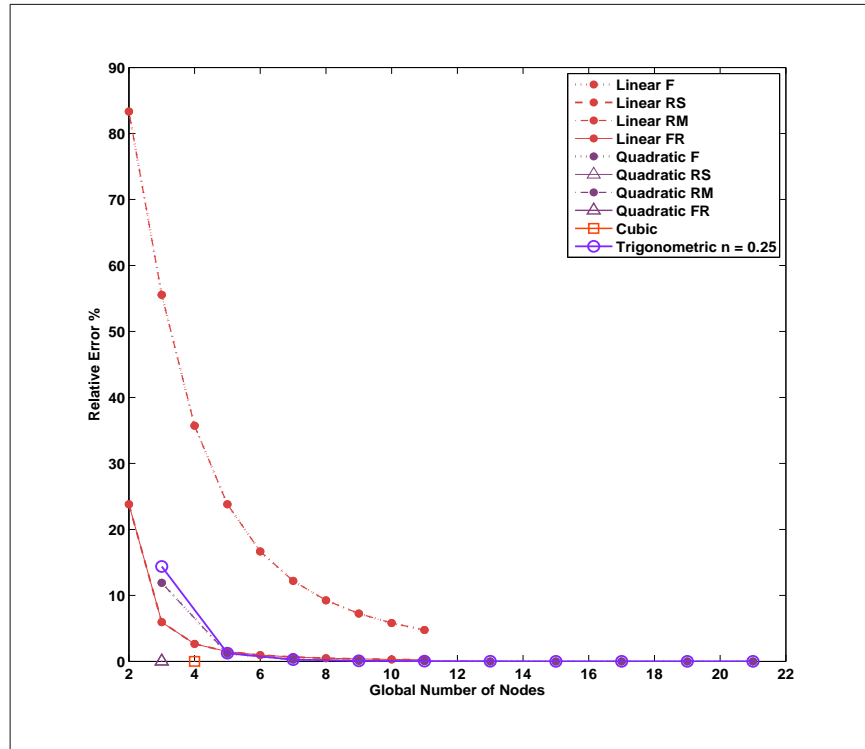


Figure 7.2: The cantilever straight beam results,  $L/h = 4$ , The relative error between the analytical and the finite element results for tip deflection.

Similar results are presented for beams of  $L/h = 10$  and  $L/h = 50$  in Figs. 7.3 and 7.4. Results for fully integrated (F) and reduced membrane integration (RM) of the linear basis functions have been removed in order to present the rest of results with higher resolution. Trigonometric basis functions do not show any shear locking behaviour. The reason has been provided by Prathap and Bhashyam [10] which indicates that they can not lock. However, it must be noted that these results do not suggest that, overall, trigonometric basis functions show a higher performance than reduced shear or fully reduced integrated polynomials in contrast with Heppler and Hansen [6], which could be due to the use of a different integration scheme for trigonometric basis functions.

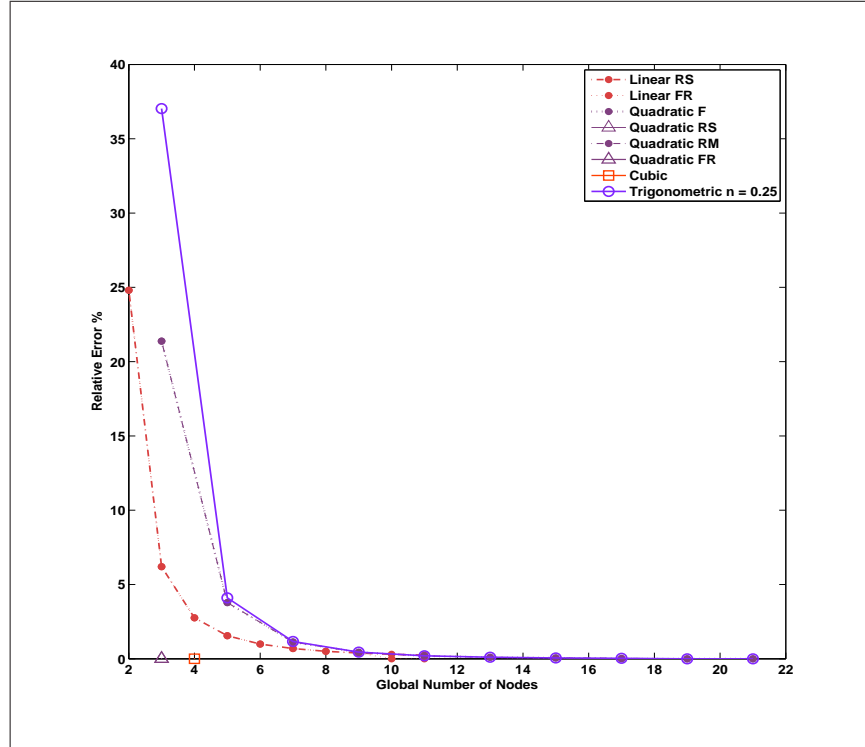


Figure 7.3: The cantilever straight beam results,  $L/h = 10$ , The relative error between the analytical and the finite element results for tip deflection.

## 7.2 Quarter Circle Beam with Radial Tip Load

The next test is that of a tip loaded cantilever curved beam with a quarter of a circle arc length. The details of the geometry are given in Fig. 7.5. The material properties are  $E = 10^7$ psi,  $\nu = 0.3$  and  $k = 5/6$ . An analytical solution to this problem can be found in Lee and Sin [8] or in Young and Budynas [14] as

$$w_{max} = \frac{\pi PR^3}{4EI} + \frac{\pi PR}{4kGA} \quad (7.3)$$

Three different load condition has been considered and, for brevity, results for the radial deflection at the free end of the beam is shown in Fig. 7.6. Obviously, fully reduced integration scheme gives the best results and as it was expected the reduced membrane integration scheme results in faster convergence and lower error in compare with reduced shear integration scheme. In curved beams membrane locking



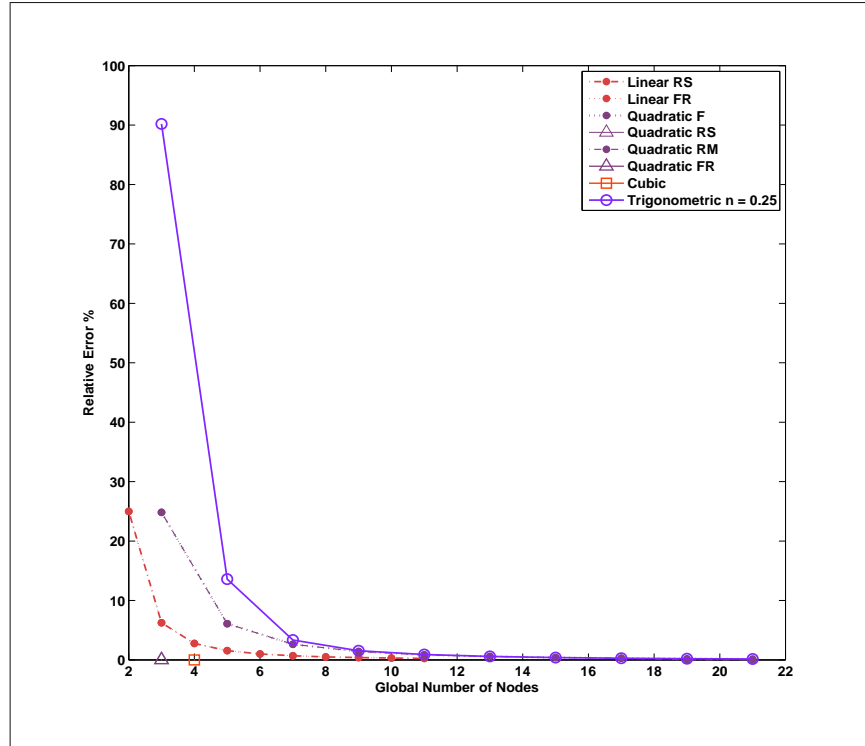


Figure 7.4: The cantilever straight beam results,  $L/h = 50$ , The relative error between the analytical and the finite element results for tip deflection.

is more serious than the shear locking therefore these results agrees with the theoretical predictions, see Appendix B.

In order to examine the p-convergence of the results for tangential and radial displacements the degrees of the basis functions polynomials have been increased from 1 to 10. The results , Fig. 7.7, show a very fast convergence in compare with h-convergence as it was expected from theory. A full integration scheme has been used to monitor the worst case from previous simulations. Results show the convergence when a 4<sup>th</sup>-degree polynomial basis functions is employed.

An additional set of simulation has been performed to show the effects of geometric nonlinearity. The input force has been reduced to 0.5 and nine quadratic elements with fully reduced integration scheme, which gave a perfect solution previously, has been employed. An incremental (Euler) method has been used where load is applied in several steps. The results are presented in Figs. 7.8 and 7.9. As it is clear, the nonlinear solution starts to diverge from the linear solution after a few increment, however the error remains small as the correction has not yet been used.

A last simulation has been performed using a combination of an Euler method with a Newton-Raphson correction steps. The force-displacement curve is shown in Fig. 7.10. At two major turning points are identified, while the solution method is probably not able to catch the true curve due to its intrinsic shortcomings in following turning points or more precisely points where tangent stiffness becomes parallel to the displacement axis.

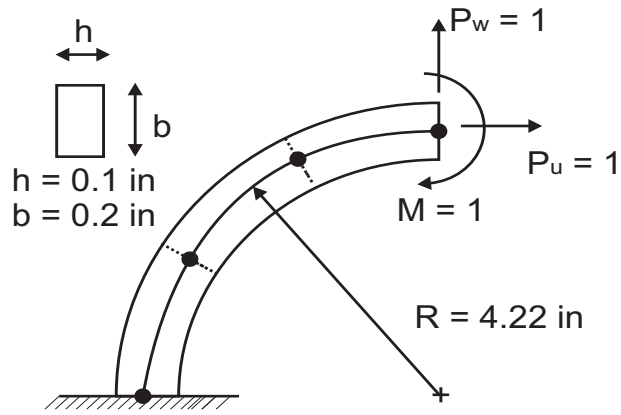


Figure 7.5: The quarter circle curved cantilever beam.

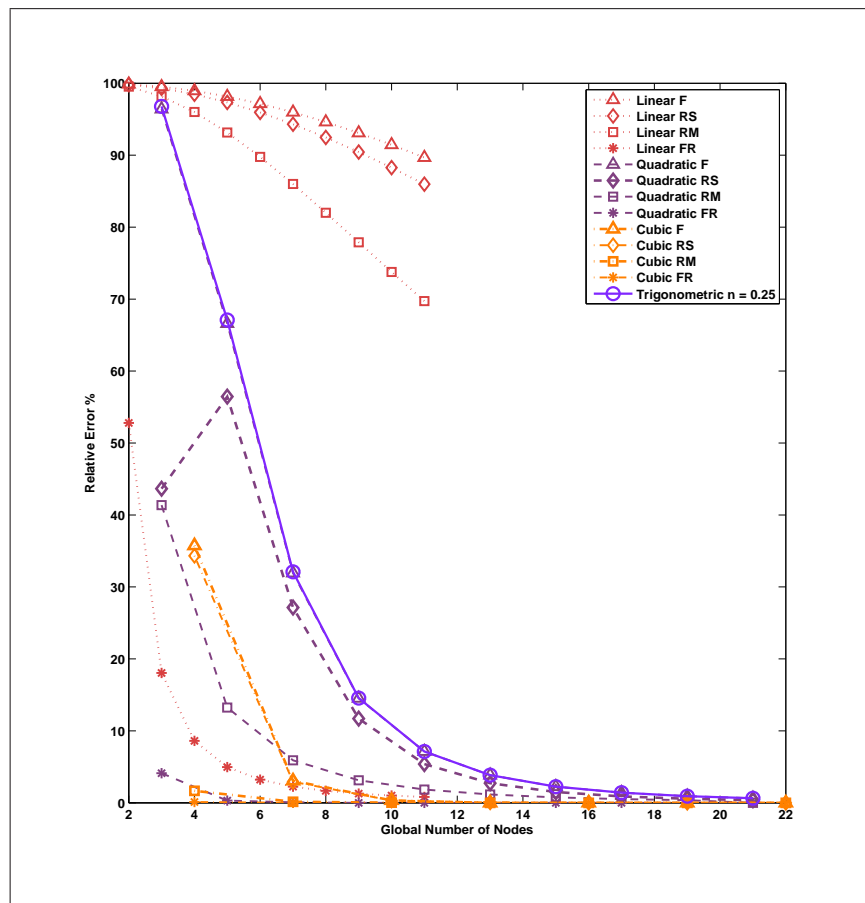


Figure 7.6: The cantilever curved beam results for different types of basis function in compare with each other. The advantage of fully reduced integration is significant.

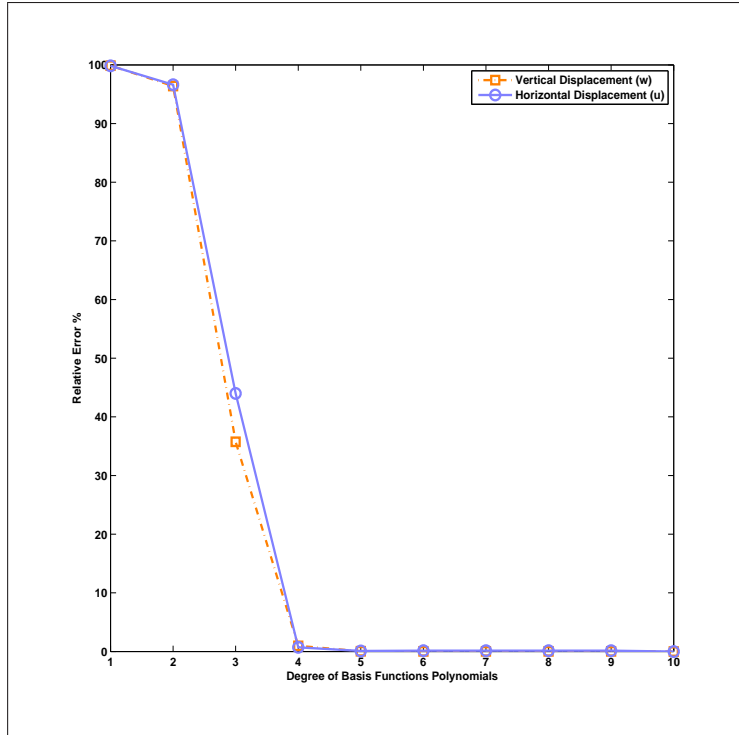


Figure 7.7: The p-convergence of the results for the cantilever curved beam. A full integration scheme has been used to monitor the worst case from previous simulations.

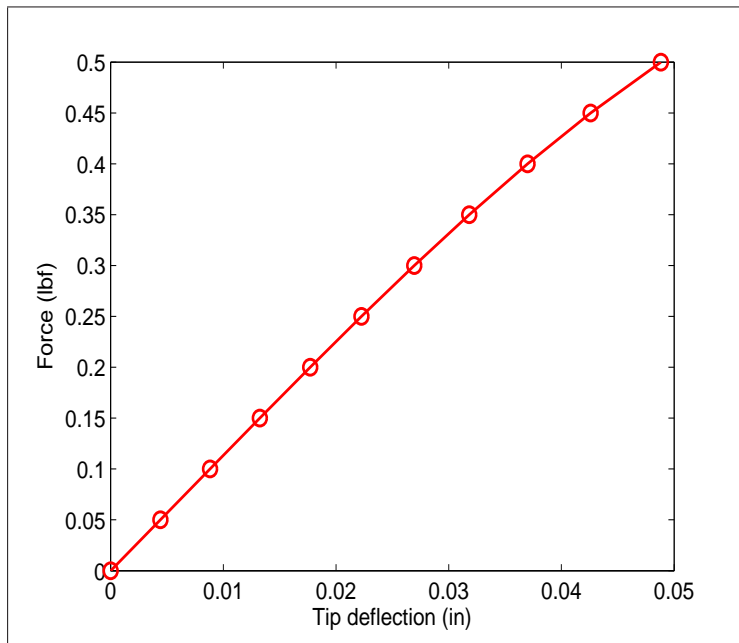


Figure 7.8: The nonlinear force-displacement curve using an incremental (Euler) method for a quarter circle beam.

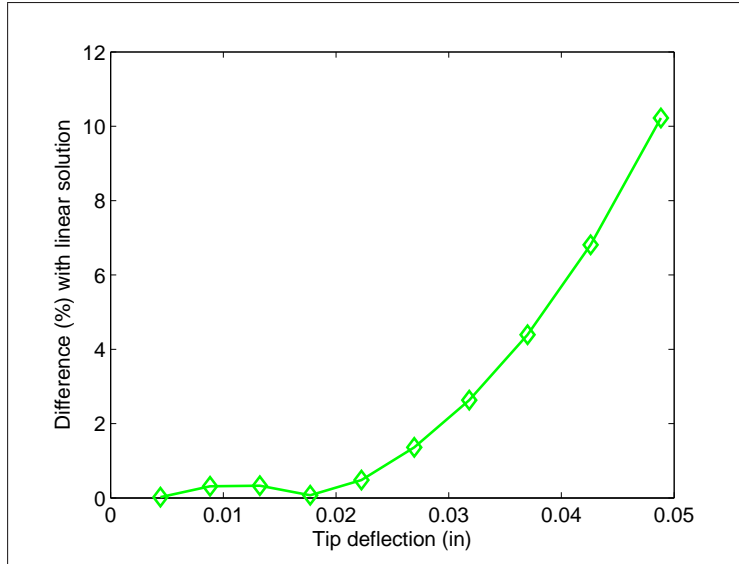


Figure 7.9: The difference between nonlinear and linear solution (%).

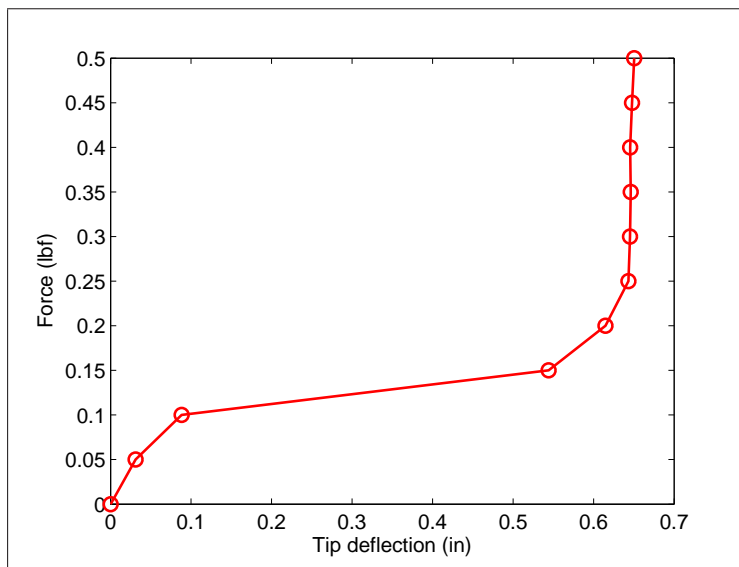


Figure 7.10: The force-displacement curve implementing a combined Euler and Newton-Raphson methods.

### 7.3 Discussion

In full analysis of geometric nonlinearities an incremental method does not give an accountable results except in case of weak nonlinearities. Although, it is necessary to compare the method with an available nonlinear solution to the problem. When considering the available solutions the solution has converged to the solution. The Newton-Raphson is not promising in nonlinear analysis especially when turning and bifurcation points are present in the true response of the structure. It might have predicted the wrong results and have given an incorrect force-displacement curve. That is an important defect in the Newton-Raphson or Euler method in analyzing geometric nonlinearities.

# Chapter 8

## Conclusions

### 8.1 Conclusions

An important geometric nonlinearity which is still under investigation in various problems of structural mechanics has been examined. Timoshenko beam analysis has been considered to be the one dimensional version of the Reissner-Mindlin plate theory, and a nonlinear form of the strain-displacement relation has been explored. A method previously introduced by other researchers has been followed to derive the strain-displacement relations, where the linear and nonlinear parts have been divided apart. The linear, nonlinear and stress stiffness matrices have been derived and convergence of the method has been tested in case of available analytical solutions. The locking phenomenon has been fully investigated and the effects of using reduced integration schemes have been demonstrated. The trigonometric shape functions has shown to be able to catch the true results, although lack of an exact integration schemes is clearly seen. Fully reduced Lagrangian basis functions show the best performance, while trigonometric basis functions do not lock. In full nonlinear analysis, Euler method could follow the force-displacement path for very small loads. Combining Newton-Raphson method to the Euler method have made the method able to follow the force-displacement path in severer cases, although it should be verified with an appropriate analytical solution.

### 8.2 Future Works

The possible future works are summarized here as:

- It would be a good work to collect all available basis functions for beam analysis and compare their performance when using them in finite element analysis of Euler-Bernoulli or Timoshenko beams.
- Computation time has not been considered, while it should be taken into account in order to choose an appropriate shape function if the computational cost is of importance.
- The methods presented here in formulating Timoshenko beams could be developed to some other problems in structural mechanics.
- Comparing Timoshenko beams with Euler-Bernoulli beams in nonlinear analysis could be done in future.
- In full nonlinear analysis other methods should be considered as the current method is not stable and not able to follow the true path adequately.
- Implementing Arc-length method and general displacement control methods could resolve the issues.

# Appendix A

## Source codes

The main files are attached here, however the full versions of the source codes should be found in the companion CD.

### A.1 Quadrilateral Bilinear Element

```
1 %-----%
2 % S.Amir Mousavi-Lajimi %
3 % Copyright 1' 2009 %
4 %-----%
5 % This code is developed to calculate displacement and force vectors for
6 % a plane stress problem explained in the report
7 % Element : Isoparametric Quadrilateral Bilinear Element
8 % The driver file is : TwoD4NIsoQuad.m
9 clear all;
10 close all;
11 % Include Global variables
12 Vars2D4NIQ;
13 % Include Input variables
14 Input2D4NIQ;
15
16 for e=1:nele
17 % Calculate the element stiffness matrix 8*8
18 Ke = ElementStiffness(Cmat,ngpoints,e,xI,yI);
19 Kg = Assembly(Ke,e);
20 K = Kg + K; % save assembled matrix
21 end
22
23 Pf = zeros(gnodes*2,1) ;
24 Pforce = [9,7.5,0;18,7.5,0] ;
25 for i=1:size(Pforce(:,1))
26 nnum = Pforce(i,1) ;
27 v = [2*nnum-1,2*nnum];
28 if any(Pf(v)≠0)
29 error('Loads matrix specifies the same entry of force twice');
30 end
31 Pf(v)= Pforce(i,2:3)' ;
32 end
33 % Apply displacement boundary conditions
34 [Kbc, fbc, srow] = BCs(K,Pf,dbc);
35
36 % Extract displacement vectors
```

```

37 d1 = Kbc \ fbc;
38
39 % Postprocess
40 [d,F]=Postprocess(d1,srow,dbc,K);

```

```

1 % Calculates the element stiffness matrix
2 function Ke = ElementStiffness(Cmat,ngpoints,e,xI,yI)
3 Vars2D4NIQ;
4 Ke = zeros(neqe,neqe);
5 [qweights,qpoints] = GaussQuad(ngpoints);
6
7 for i=1:length(qweights)
8     for j=1:length(qweights)
9         [BNe,detJe] = PNmat2D4NIQ(qpoints(j),qpoints(i),xI(e,:),yI(e,:));
10        Ke = qweights(j)* qweights(i)* (BNe)' * Cmat * (BNe)* detJe*t + Ke;
11    end
12 end
13
14 end

```

```

1 %----- calculates the derivatives of shape functions -----%
2 function [BNe,detJe,dNdr,dNds,JeInv] = PNmat2D4NIQ(r,s,x,y)
3 Vars2D4NIQ;
4 Ne1r = -(1/4)*(1-s) ;
5 Ne2r = (1/4)*(1-s) ;
6 Ne3r = (1/4)*(1+s) ;
7 Ne4r = -(1/4)*(1+s) ;
8 dNdr = [Ne1r;Ne2r;Ne3r;Ne4r];
9 Ne1s = -(1/4)*(1-r) ;
10 Ne2s = -(1/4)*(1+r);
11 Ne3s = (1/4)*(1+r) ;
12 Ne4s = (1/4)*(1-r) ;dNds = [Ne1s;Ne2s;Ne3s;Ne4s];
13 PNe = zeros(3,neqe);
14 PNe = [Ne1r,0,Ne2r,0,Ne3r,0,Ne4r,0;
15        0,Ne1s,0,Ne2s,0,Ne3s,0,Ne4s;
16        Ne1s,Ne1r,Ne2s,Ne2r,Ne3s,Ne3r,Ne4s,Ne4r];
17 Je = zeros(2,2);
18 j = 1;
19 k = 1;
20 p = 1;
21 for I = 1:4;
22     Je(j,k) = x(I)* PNe(j,p) + Je(j,k);
23     Je(j+1,k) = x(I)* PNe(j+1,p+1) + Je(j+1,k);
24     Je(j,k+1) = y(I)* PNe(j,p) + Je(j,k+1);
25     Je(j+1,k+1) = y(I)* PNe(j+1,p+1) + Je(j+1,k+1);
26     p = p + 2;
27 end
28 detJe = det(Je) ;
29 JeInv = inv(Je) ;
30 BNe = zeros(3,8) ;
31 BNe1 = zeros(3,2,4) ;
32
33 for I=1:4
34 BNe1(:, :, I) = [JeInv(1,1)*dNdr(I)+JeInv(1,2)*dNds(I), 0;
35                0,JeInv(2,1)*dNdr(I)+JeInv(2,2)*dNds(I);
36                JeInv(2,1)*dNdr(I)+JeInv(2,2)*dNds(I), JeInv(1,1)*dNdr(I)+JeInv(1,2)*dNds(I)];
37 end
38
39 BNe = [BNe1(:, :, 1), BNe1(:, :, 2), BNe1(:, :, 3), BNe1(:, :, 4)];
40 end

```

## A.2 Quadratic Triangular Element

```
1 %-----%
2 % S.Amir Mousavi-Lajimi %
3 % Copyright 1' 2009 %
4 %-----%
5 % Computes the global displacements and force vectors
6 % Element : Quadratic Triangular Element
7
8 clear all;
9 close all;
10 % Input data
11 Input ;
12 for e=1:nele
13 Ke = ElementStiffness(Cmat,ngpoints,e,XI,YI);
14 K = Assembly(Ke,e) + K;
15 end
16
17 % Apply displacement boundary conditions
18 [Kbc, fbc, krow, dbc] = BCs(K,Pf);
19 % Extract displacement vectors
20 d1 = Kbc \ fbc;
21 % Postprocess
22 [F,d] = Postprocess(K,d1,krow,dbc);
```

```
1 %-----%
2 % S.Amir Mousavi-Lajimi %
3 % Copyright 1' 2009 %
4 %-----%
5 % Calculate the element stiffness matrix
6 function Ke = ElementStiffness(Cmat,ngpoints,e,XI,YI)
7 % Include Global variables
8 Vars2D6NT;
9 Ke = zeros(neqe,neqe);
10 [qweights,qpoints] = GaussTri(ngpoints);
11
12 for i=1:length(qweights)
13 [dNe] = dN2D6NT(qpoints(i,1),qpoints(i,2),qpoints(i,3)) ;
14 [BNe,detJe] = Bmatrices(dNe,XI(e,:),YI(e,:)) ;
15 Ke = qweights(i)*BNe'*Cmat*BNe*(detJe/2)*t + Ke ;
16 end
17 end
```

```
1 % Computes the gradients of shape functions : B
2 function [BNe,detJe] = Bmatrices(dNe,xI,yI)
3 % Include Global variables
4 Vars2D6NT;
5 % Initialize the Jacobian
6 Jx1=0;
7 Jx2=0;
8 Jx3=0;
9 Jy1=0;
10 Jy2=0;
11 Jy3=0;
12 for i=1:6
13 Jx1 = dNe(i,1)*xI(i)+Jx1;
14 Jx2 = dNe(i,2)*xI(i)+Jx2;
15 Jx3 = dNe(i,3)*xI(i)+Jx3;
16 Jy1 = dNe(i,1)*yI(i)+Jy1;
17 Jy2 = dNe(i,2)*yI(i)+Jy2;
```



```

18     Jy3 = dNe(i,3)*yI(i)+Jy3;
19 end
20 Je = zeros(3,3);
21 Je = [1 ,1 ,1 ;
22       Jx1 ,Jx2 ,Jx3 ;
23       Jy1 ,Jy2 ,Jy3 ];
24
25 detJe = det(Je);
26 dLdxy = inv(Je)*[0,0;1,0;0,1];
27
28 dNdx = zeros(6);
29 dNdy = zeros(6);
30 for I=1:6
31     dNdx(I) = dNe(I,1)*dLdxy(1,1)+dNe(I,2)*dLdxy(2,1)+dNe(I,3)*dLdxy(3,1);
32     dNdy(I) = dNe(I,1)*dLdxy(1,2)+dNe(I,2)*dLdxy(2,2)+dNe(I,3)*dLdxy(3,2);
33 end
34 % Initialize BNe
35 BNe1 = zeros(3,2,6);
36 BNe = zeros(3,12);
37 for I=1:6
38     BNe1(:, :, I) = [dNdx(I),0;
39                     0,dNdy(I);
40                     dNdy(I),dNdx(I)];
41 end
42 BNe = [BNe1(:, :, 1),BNe1(:, :, 2),BNe1(:, :, 3),BNe1(:, :, 4),BNe1(:, :, 5),BNe1(:, :, 6)];
43 end

```

### A.3 Linear Tetrahedral Element

```

1  %-----%
2  % S.Amir Mousavi-Lajimi %
3  % Copyright l' 2009 %
4  %-----%
5  % Computes the global displacement and force vectors
6  % Computes the element displacement vectors and stresses
7  % Element : Linear Tetrahedral Element
8  clear all;
9  close all;
10 % Include all variables
11 Vars3D4NLinT;
12 % Call input file
13 Input;
14
15 for e=1:nele
16 % element stiffness matrix
17 Ke = ElementStiffness(Cmat,e);
18 % Assemble global stiffness matrix
19 Kg = Assembly(Ke,e);
20 % Save global stiffness matrix
21 K = Kg + K;
22 end
23
24 % Compute and assemble point forces
25 Pf = Pforces;
26 % Apply displacement boundary conditions
27 [Kbc, fbc, krow] = BCs(K, Pf);
28 % Extract displacement vectors
29 d1 = Kbc \ fbc;
30 % Postprocess
31 [F, d, de, Fe, sigma] = Postprocess(K, d1, krow, BmNe);

```

```

1 % Calculates the element stiffness matrix
2 function Ke = ElementStiffness(Cmat,e)
3 % Include Global variables
4 Vars3D4NLinT;
5 Ke = zeros(neqe,neqe) ;
6 % Calculate the derivate of shape functions and Jacobian
7 [detJe,dNdx,dNdy,dNdz] = dN3D4NLinT(XI,YI,ZI,e);
8 BNe = Bmatrices(dNdx,dNdy,dNdz) ;
9 Ke = BNe' * Cmat * BNe * Vol(e) + Ke ;
10
11 end

```

```

1 % Computes the gradient of shape functions B
2 function BNe = Bmatrices(dNdx,dNdy,dNdz)
3 % Include Global variables
4 Vars3D4NLinT;
5 % Initialize BNe
6 BNe1 = zeros(6,3,4);
7 BNe = zeros(6,12);
8 for I=1:enodes
9     BNe1(:, :, I) = [dNdx(I)    ,0    ,0    ;
10                    0    ,dNdy(I) ,0    ;
11                    0    ,0    ,dNdz(I) ;
12                    dNdy(I) ,dNdx(I) ,0    ;
13                    0    ,dNdz(I) ,dNdy(I) ;
14                    dNdz(I) ,0    ,dNdx(I) ];
15 end
16 BNe = [BNe1(:, :, 1), BNe1(:, :, 2), BNe1(:, :, 3), BNe1(:, :, 4)];
17 BmNe(:, :, e) = BNe;
18 end

```

## A.4 Nonlinear Timoshenko Beam Analysis

```

1 %-----%
2 % S.Amir Mousavi-Lajimi %
3 % Copyright 1' 2009 %
4 %-----%
5 % The master .m file
6 % TIMOSHENKO BEAM
7 % NON/LINEAR FEA
8 close all;
9 clear all;
10
11 % Input the geometric and material parameters
12 InputTBFEA;
13
14 % Discretize the domain and define the essential boundary conditions
15 MeshTBFEA;
16
17 % Initialize the matrices and vectors
18 InitializeTBFEA;
19
20 % Choosing type of analysis, solution method and integration order
21 MethodsTBFEA;
22
23 % Define the solution method parameters
24 ParametersTBFEA;
25

```

```

26 % master loop on force
27 count      = 0;
28 sfe        = zeros(gdof,NOINC);
29 sfi        = zeros(gdof,NOINC);
30 sd         = zeros(gdof,NOINC);
31 dhat       = zeros(gdof-size(e_bc(:,1)),1);
32 reactions  = zeros(gdof,1);
33 siter(1)= 0;
34 for ff=dP:dP:P
35     count = count + 1 ;
36     % Update external force vector
37     fe     = dfe + fe ; %+ reactions ;
38     % Set up the stiffness matrices
39     StiffnessTBFEA;
40     AssemblyTBFEA;
41
42     % Apply boundary conditions
43     [Khat,dfehat,DOFBC] = BCsTBFEA(K,dfe,gdof,e_bc);
44
45     % Solve the linear system for displacement variation
46     dhat     = linsolve(Khat,dfehat);
47
48     % Calculate current complete displacement including boundary conditions
49     [dd,reactions] = GatherTBFEA(dhat,DOFBC,gdof,e_bc);
50
51     % Update global displacement vector
52     d         = d + dd ;
53     %sd(:,count+1) = d(:);
54     if NRFLAG == 1
55         % Implement Newton-Raphson method
56         NRaphsonTBFEA;
57     end
58     sd(:,count+1) = d(:);
59
60     % Save force and displacement vectors
61     sfe(:,count+1) = fe(:) + reactions(:) ;
62     sfi(:,count+1) = fi(:) ;
63 end
64
65 % Postprocessing the outputs
66 PostprocessTBFEA;

```

```

1 function [BL,BN,NP,dw,J] = BmatricesTBFEA(enod,gp,x,w,alpha,rho)
2 % THE LAGRANGE POLYNOMIAL IS GENERATED BASED ON THE NUMBER OF NODES PER
3 % ELEMENT FOR AN ISOPARAMETRIC ELEMENT
4 % xi   : ELEMENT COORDINATE
5 % enod : NODES PER ELEMENT
6 % gp   : GAUSS POINT
7 % w    : A VECTOR WHICH HAS THE w COMPONENT OF DISPLACEMENT VECTOR FOR AN
8 %        ELEMENT
9 % BL   : CONTAINS THE LINEAR B-MATRICES (= BPL * N)
10 %      BL(:, :, 1) = BL0, BL(:, :, 2) = BL1, BL(:, :, 3) = BL2
11 % BN   : CONTAINS THE NONLINEAR B-MATRICES (= BPL * NP)
12 %      BN(:, :, 1) = BN0, BL(:, :, 2) = BN1, BL(:, :, 3) = BN2
13 % N    : KEEPS THE VECTOR OF SHAPE FUNCTIONS WHERE EACH COMPONENT OF THAT
14 %        CORRESPONDS TO A NODE IN ELEMENT
15 % Nmat : THE MATRIX OF BASIS FUNCTIONS
16 % J    : THE JACOBIAN TRANSFORMATION
17 % NP   : CONTAINS THE DERIVATIVE OF THE BASIS FUNCTIONS WITH RESPECT TO THE
18 %        PHYSICAL COORDINATE SYSTEM (x) IN A ROW VECTOR IN SUCH A WAY THAT
19 %        EACH BLOCK IS LIKE : [0 dNi 0] WHERE I VARIES BETWEEN ONE AND THE
20 %        NUMBER OF NODES PER ELEMENT
21 % rho  : THE RECIPROCAL OF THE RADIUS OF CURVATURE
22 % alpha: COEFFICIENT OF LAME COEFFICIENT

```

```

23 xi      = zeros(1,enod) ;
24 xi(1)   = -1           ;
25 xi(enod) = 1           ;
26 if enod > 2
27     for i=2:(enod - 1)
28         xi(i) = xi(1) + (i-1)*(2/(enod - 1)) ;
29     end
30 end
31 N      = zeros(1,enod) ;
32 for i=1:enod
33     M = 1;
34     for j=1:enod
35         if j≠i
36             N(i) = ((gp - xi(j))/(xi(i) - xi(j)))*M;
37             M = N(i);
38         end
39     end
40 end
41 dN      = zeros(1,enod) ;
42 dNa     = zeros(1,enod) ;
43
44 if enod==2
45     dN(1) = 1/(xi(1) - xi(2));
46     dN(2) = 1/(xi(2) - xi(1));
47 elseif enod>2
48     for i=1:enod
49         dN(i) = 0;
50         for j=1:enod
51             %FF = 0;
52             if j≠i
53                 M = 1;
54                 for k=1:enod
55                     if (k≠j) && (k≠i)
56                         dNa(i) = (1/(xi(i) - xi(j))) * ...
57                             ((gp - xi(k))/(xi(i) - xi(k))) * M;
58                         M = dNa(i)*(xi(i) - xi(j));
59                     end
60                 end
61                 FF = dNa(i) ;
62                 dN(i) = dN(i) + FF ;
63             end
64         end
65     end
66 end
67
68 J = 0;
69 for i=1:enod
70     J = x(i) * dN(i) + J;
71 end
72
73 Nmat = zeros(3,enod*3);
74 p = 0;
75 for i=1:3
76     k=1;
77     for j=1:3:(enod*3)
78         Nmat(i,j+p) = N(k);
79         k = k + 1;
80     end
81     p = p + 1;
82 end
83
84 NP = zeros(1,3*enod);
85 i=1;
86 for j=2:3:(3*enod)
87     NP(j) = dN(i);
88     i = i + 1;

```

```

89 end
90 NP = NP*inv(J);
91
92 % Linear B-matrices
93 BL = zeros(2,3*enod,3);
94 BL0 = zeros(2,3*enod) ;
95 BL1 = zeros(2,3*enod) ;
96 BL2 = zeros(2,3*enod) ;
97 p=0;
98 for i=1:enod
99
100 BL0(1:2,i+p:i+p+2) =...
101 [(1/alpha)*inv(J)*dN(i) , rho*N(i) , 0 ;
102 -rho*N(i) , (1/alpha)*inv(J)*dN(i) , N(i) ];
103
104 BL1(1:2,i+p:i+p+2) =...
105 [-(rho/alpha)*inv(J)*dN(i) , -(rho^2)*N(i) , (1/alpha)*inv(J)*dN(i) ;
106 (rho^2)*N(i) , -(rho/alpha)*inv(J)*dN(i) , -rho*N(i) ];
107
108 BL2(1:2,i+p:i+p+2) =...
109 [(rho^2/alpha)* inv(J) * dN(i), (rho^3)*N(i) , -(rho/alpha)*inv(J)*dN(i);
110 -(rho)^3)*N(i) , ((rho^2)/alpha)*inv(J)*dN(i), (rho^2)*N(i) ];
111
112 p = p + 2;
113 end
114 BL(:, :, 1) = BL0(:, :);
115 BL(:, :, 2) = BL1(:, :);
116 BL(:, :, 3) = BL2(:, :);
117
118 % Nonlinear B-matrices
119
120 dw = 0;
121 for i=1:enod
122 dw = dN(i)*w(i) + dw;
123 end
124
125 BPN0 = [(1/(alpha^2)) * inv(J) * dw ;
126 0 ];
127 BPN1 = [((-2*rho)/(alpha^2)) * inv(J) * dw ;
128 0 ];
129 BPN2 = [(3*rho^2)/(alpha^2)) * inv(J) * dw ;
130 0 ];
131
132 BN0 = BPN0 * NP;
133 BN1 = BPN1 * NP;
134 BN2 = BPN2 * NP;
135
136 BN(:, :, 1) = BN0(:, :);
137 BN(:, :, 2) = BN1(:, :);
138 BN(:, :, 3) = BN2(:, :);
139
140 end

```

```

1 % TIMOSHENKO BEAM
2 % NON/LINEAR FEA
3 % COMPUTING THE STIFFNESS MATRICES
4 % ed : ELEMENT DISPLACEMENT MATRIX : edof by nele WHERE EACH COLUMN KEEPS
5 % THE CORRESPONDING DISPLACEMENT VECTOR OF AN ELEMENT FOR CURRENT INCREMENT
6 % ex : ELEMENT NODAL POSITION MATRIX enod by nele
7
8 ed = zeros(edof,nele);
9 ex = zeros(enod,nele);
10 for e=1:nele
11 for lnID=1:enod

```

```

12     gnID = LG(e,lnID);
13     ex(lnID,e) = Xn(gnID);
14     for ldof=1:ndof
15         ed(3*(lnID - 1)+ldof,e) = d(ndof*(gnID - 1) + ldof);
16     end
17 end
18 end
19 KT = zeros(edof,edof,nele);
20
21 for en=1:nele
22     KLinear = KLinearTBFEA(D, alpha, rho, Iyy, Ayz, ex, ed, en, ngpoints, enod, edof);
23     if AFLAG == 1
24         KNLinear = KNLinearTBFEA(D, alpha, rho, Iyy, Ayz, ex, ed, en, ngpoints, enod, edof, zb);
25     end
26     if AFLAG == 2
27         KNLinear = KNLinearTBFEA(D, alpha, rho, Iyy, Ayz, ex, ed, en, ngpoints, enod, edof, zb);
28         KStress = KStressTBFEA(D, alpha, rho, Iyy, Ayz, ex, ed, en, ngpoints, enod, edof, zb);
29     end
30     KT(:, :, en) = KLinear + KNLinear + KStress;
31 end

```

```

1 function KNLinear = KNLinearTBFEA(D, alpha, rho, Iyy, Ayz, ex, ed, en, ngpoints, enod, edof, zb)
2 % TIMOSHENKO BEAM
3 % NON/LINEAR FEA
4 % COMPUTING THE NONLINEAR STIFFNESS MATRICES BOTH SHEAR AND MEMBRANE
5 % CONTRIBUTIONS
6 % BL : CONTAINS THE LINEAR B-MATRICES (= BPL * N)
7 %     BL(:, :, 1) = BL0, BL(:, :, 2) = BL1, BL(:, :, 3) = BL2
8 % w : A VECTOR WHICH HAS THE w COMPONENT OF DISPLACEMENT VECTOR FOR AN
9 %     ELEMENT
10 % ed : ELEMENT DISPLACEMENT MATRIX : edof by nele WHERE EACH COLUMN KEEPS
11 %     THE CORRESPONDING DISPLACEMENT VECTOR OF AN ELEMENT FOR CURRENT INCREMENT
12 % ex : ELEMENT NODAL POSITION MATRIX enod by nele
13 % NA : NOT ASSIGNED (NOT USED)
14
15 KNLinear = zeros(edof,edof);
16 KNLmembrane = zeros(edof,edof);
17 KNLshear = zeros(edof,edof);
18 w = ed(2:3:size(ed(:,en)),en);
19 [SGW,SGP] = GaussN(ngpoints(1));
20 [MGW,MGP] = GaussN(ngpoints(2));
21
22 DM = zeros(2,2);
23 DM(1,1) = D(1,1);
24
25 DS = zeros(2,2);
26 DS(2,2) = D(2,2);
27
28
29 for ip=1:size(MGP)
30     [BL,BN,NA,dw,J] = BmatricesTBFEA(enod,MGP(ip),ex(:,en),w,alpha,rho);
31     for iL=1:3
32         for iN=1:3
33             if (iL - 1 + iN - 1) == 0
34                 zcoef1 = Ayz;
35                 zcoef2 = zb * Ayz;
36             elseif (iL - 1 + iN - 1) == 1
37                 zcoef1 = zb * Ayz;
38                 zcoef2 = Iyy;
39             elseif (iL - 1 + iN - 1) == 2
40                 zcoef1 = Iyy;
41                 zcoef2 = 0;
42             else
43                 zcoef1 = 0;

```

```

44         zcoef2 = 0           ;
45     end
46     KNLmembrane = alpha * (BL(:, :, iL)' * DM * BN(:, :, iN) + ...
47                          BN(:, :, iN)' * DM * BL(:, :, iL) + ...
48                          BN(:, :, iN)' * DM * BN(:, :, iN)) * zcoef1 + ...
49     rho * (BL(:, :, iL)' * DM * BN(:, :, iN) + ...
50            BN(:, :, iN)' * DM * BL(:, :, iL) + ...
51            BN(:, :, iN)' * DM * BN(:, :, iN)) * zcoef2 + ...
52     KNLmembrane;
53 end
54 end
55 KNLLinear = KNLmembrane * MGW(ip) + KNLLinear;
56 end
57
58 for ip=1:size(SGP)
59     [BL,BN,NA,dw,J] = BmatricesTBFEA(enod,MGP(ip),ex(:,en),w,alpha,rho);
60     for iL=1:3
61         for iN=1:3
62             if (iL - 1 + iN - 1) == 0
63                 zcoef1 = Ayz           ;
64                 zcoef2 = zb * Ayz     ;
65             elseif (iL - 1 + iN - 1) == 1
66                 zcoef1 = zb * Ayz     ;
67                 zcoef2 = Iyy          ;
68             elseif (iL - 1 + iN - 1) == 2
69                 zcoef1 = Iyy          ;
70                 zcoef2 = 0            ;
71             else
72                 zcoef1 = 0            ;
73                 zcoef2 = 0            ;
74             end
75             KNLshear = alpha * (BL(:, :, iL)' * DS * BN(:, :, iN) + ...
76                              BN(:, :, iN)' * DS * BL(:, :, iL) + ...
77                              BN(:, :, iN)' * DS * BN(:, :, iN)) * zcoef1 + ...
78             rho * (BL(:, :, iL)' * DS * BN(:, :, iN) + ...
79                  BN(:, :, iN)' * DS * BL(:, :, iL) + ...
80                  BN(:, :, iN)' * DS * BN(:, :, iN)) * zcoef2 + ...
81             KNLshear;
82         end
83     end
84     KNLLinear = KNLshear * SGW(ip) + KNLLinear;
85 end
86 KNLLinear = KNLLinear * J;
87
88
89 end

```

```

1 function [dd,reactions] = GatherTBFEA(dhat,DOFBC,gdof,e_bc)
2 dd = zeros(gdof,1);
3 reactions = zeros(gdof,1);
4
5 for i=1:gdof
6     if DOFBC(i) == 0
7         for p=1:size(e_bc(:,1))
8             rindex = 3 * (e_bc(p,1) - 1) + e_bc(p,2);
9             if i == rindex
10                dd(i) = e_bc(p,3) ;
11                reactions(i) = dhat(rindex) ;
12            end
13        end
14    elseif DOFBC(i) == 1
15        dd(i) = dhat(i) ;
16    end
17 end

```

```

1 % TIMOSHENKO BEAM
2 % NON/LINEAR FEA
3 % COMPUTES THE IMPROVED SOLUTION (DISPLACEMENT) USING MODIFIED
4 % NEWTON-RAPHSON METHOD
5 % RES      : THE DIFFERENCE BETWEEN FORCE CALCULATED FOR CURRENT
6 %           DISPLACEMENT AND EXTERNAL FIXED FORCE
7 % MAXITER  : MAXIMUM ITERATION IN CORRECTION STEPS
8 % CONST    : USED TO CHECK THE CONVERGENCE OF THE RESULT
9 % CFLAG    : USED TO SHOW CONVERGENCE
10
11
12 CFLAG      = 0           ;
13 CONST      = 0.01       ;
14 RES        = zeros(gdof,1) ;
15 Condition  = 1          ;
16 MAXITER    = 6           ;
17 sRES       = zeros(MAXITER,1);
18 for iter=1:MAXITER
19     dcurrent = d           ;
20     StaticForceTBFEA           ;
21     AssemblyNR                 ;
22     g          = zeros(size(fi)) ;
23     g(DOFBC)   = fi(DOFBC) - fe(DOFBC);
24     sRES(iter) = norm(g)       ;
25     if Condition < CONST
26         CFLAG = 1;
27         break
28     else
29         [KmodNR,gmod,DOFBC] = BCsTBFEA(KN,g,gdof,e_bc);
30
31         % Solve the linear system for displacement variation
32         ddhat = linsolve(-KmodNR,g);
33         % Calculate current complete displacement including boundary conditions
34         [dd,reactions] = GatherTBFEA(ddhat,DOFBC,gdof,e_bc);
35
36         % Update global displacement vector
37         d          = d + dd           ;
38         Condition = norm(d - dcurrent)/norm(d) ;
39         sdNR(:,sum(siter)+iter) = d(:);
40     end
41 end
42 siter(count+1) = iter;
43 if CFLAG==1
44     fprintf('Converged! Condition is : %f\n',Condition);
45 end
46
47 if CFLAG==0
48     fprintf('failed to converge\n')
49 end

```



# Appendix B

## Locking

It has been recognized that in certain cases the predictions for displacements might be completely useless, apart from common problems with the finite element analysis in mechanics such as inaccurate results or slow convergence. The phenomenon is characterized by a severe underestimation of the displacements, i.e. the structure is too stiff [12]. This problem is called locking referring to the intuitive understanding that the structure locks itself against deformations. In the main, a critical parameter plays a significant role in locking manifestation. For example, in the case of transverse shear locking of plate or beam elements this parameter is the slenderness of the plate or beam, and in volumetric locking it is the bulk modulus.

The phenomenon can be described with the presence of some spurious stresses. These artificial stresses do not appear in exact solutions of the problem. The simulated stresses augments the, artificial, internal energy of the system resulting in an extra stiffness. The aforesaid critical parameter now may lead this energy to inf as it approaches a large number. Therefore, the behavior of the structure is dominated by this spurious energy, and drastic changes happen in the finite element predictions.

The shear locking occurs if the length to thickness, aspect, ratio of a beam becomes very large, that is the beam gets very thin. Therefore, the critical parameter in the case of shear locking is the aspect ratio of the element (i.e. no property of the underlying mathematical problem itself).

Shell and beam elements are susceptible to membrane locking as well. It refers to a situation where pure bending deformations are accompanied by spurious membrane stresses. The membrane locking only appears if the elements are actually curved. Therefore, Quadratic elements usually show strong membrane locking in any situation. Membrane locking can be observed in the most simple case for a quadratic (three-node) beam element. When imposing a constant bending moment it can be observed that the membrane strains and stresses (i.e. the normal forces in the beam) are non-zero. The reason for the fact that the pure bending deformation cannot be represented exactly is that the exact solution for the displacements contains trigonometric functions which are of course not included in the shape functions.

Several methods have been developed to overcome the problem of locking phenomenon. One simple method is to use a different shape function for rotation which is one order lower than the one for deflection [11]. Another method is to use the same order of interpolation for both rotation and deflection, but evaluate the energy in a proper manner such as reduced order integration. Several definitions are available for reduced order integration, such as the exactness of certain integrals, the satisfaction of certain tests, or simply evaluating the integrals using lower number of Gauss points than the required number of them for exact integration.

# Bibliography

- [1] G. R. Cowper, “The shear coefficient in timoshenko’s beam theory,” *Journal of Applied Mechanics*, no. 33, pp. 335–340, 1966.
- [2] G. Dahlquist and Å. Björck, *Numerical Methods in Scientific Computing: Volume 1*. Philadelphia, PA: SIAM, 2008.
- [3] C. A. J. Fletcher, *Computational Galerkin Methods*. New York: Springer-Verlag, 1984.
- [4] K. K. Gupta and J. L. Meek, *Finite Element Multidisciplinary Analysis*, ser. AIAA Education Series. Reston, VA: AIAA, 2000.
- [5] J. S. Hansen and G. R. Heppler, “A mindlin shell element which satisfies rigid body requirements,” *AIAA Journal*, vol. 23, pp. 288–295, 1985.
- [6] G. R. Heppler and J. S. Hansen, “Performance of trigonometric basis function finite elements in timoshenko beams,” in *AIAA/ASME/ASCE/AHS 28th Structures, Structural Dynamics and Materials Conference*. Monterey, California: AIAA, 1987.
- [7] V. N. Kaliakin, *Introduction to Approximate Solution Techniques, Numerical Modeling, and Finite Element Methods*. New York: Marcel Dekker, Inc., 2002.
- [8] P. G. Lee and H. C. Sin, “Locking-free curved beam element based on curvature,” *International Journal for Numerical Methods in Engineering*, vol. 37, no. 6, pp. 989–1007, 1994.
- [9] A. Pica, R. D. Wood, and E. Hinton, “Finite element analysis of geometrically nonlinear plate behaviour using a mindlin formulation,” *Computers & Structures*, vol. 11, pp. 203–215, 1980.
- [10] G. Prathap and G. R. Bhashyam, “Reduced integration and the shear-flexible beam element,” *International Journal for Numerical Methods in Engineering*, vol. 18, pp. 195–210, 1982.
- [11] J. N. Reddy, *An introduction to nonlinear finite element analysis*. Oxford, New York: Oxford University Press, 2004.
- [12] R. Wüchner, *Advanced Finite Element Methods, Course Notes*. Munich: Technical University of Munich, 2009.
- [13] R. D. Wood and O. C. Zienkiewicz, “Geometrically nonlinear finite element analysis of beams, frames, arches and axisymmetric shells,” *Computers & Structures*, vol. 7, pp. 725–735, 1977.
- [14] W. C. Young and R. Budynas, *Roark’s Formulas for Stress and Strain*, 7th ed. New York: McGraw-Hill, 2002.

# Index

Hamilton's Principle, 9

beam element, 31

bilinear quadrilateral element, 12

boundary conditions

Dirichlet, 6

Neumann, 6

compatibility condition, 7

compatibility equations, 7

constitutive relations, 5

curved beams, 29

Galerkin method, 7

gauss quadrature

triangular elements, 16

Lamé coefficient, 29

linear tetrahedral element, 17

locking, 58

membrane, 37

shear, 37

nonlinearities

category, 25

geometric, 26

material, 26

plane strain, 5

plane stress, 5

Principle of Minimum Potential Energy, 8

quadratic triangular element, 14

Reduced Numerical Integration, 37

Reissner-Mindlin plate theory, 27

shear correction factor, 33

solution procedures, 26

tangent stiffness matrix, 34

Timoshenko beam, 27

trigonometric basis functions, 31

variational methods, 7