Numerically Generated Tangent Stiffness Matrices for Geometrically Non-Linear Structures

Sonia Lebofsky

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

Master of Science in Aeronautics and Astronautics

University of Washington

2013

Reading Committee:

Eli Livne, Chair

Keith Holsapple

Program Authorized to Offer Degree:
Aeronautics & Astronautics

University of Washington

Abstract

Numerically Generated Tangent Stiffness Matrices for Geometrically Non-Linear Structures

Sonia Lebofsky

Chair of the Supervisory Committee:
Professor Eli Livne
UW Aeronautics & Astronautics

The aim of this thesis is to develop a general numerical solution method for geometrically non-linear structures. Most common work involves tedious derivations of analytic tangent stiffness matrices. The major objective of the current work is to develop a numerically generated tangent stiffness matrix that allows for a general and easily implementable solution method. The thesis begins with the definition of the tangent stiffness matrix and a discussion of the Newton-Raphson incremental-iterative method typically used to solve geometrically non-linear problems. This is followed by a detailed description of how the tangent stiffness matrix is numerically generated using complex variable differentiation to approximate sensitivities. The thesis proceeds with details of the solution method applied to three different structural elements: 3D truss, membrane, and 3D beam. These discussions include numeric examples for each type of structure, the results of which are compared with the literature and ANSYS solutions. The results from the present work show that solutions obtained using the general numerically generated tangent stiffness matrix are accurate. While computational effort is increased, the method is especially attractive in the context of research involving small finite element models.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I wish to thank, first and foremost, my supervisor Dr. Eli Livne for his guidance and encouragement during the length of my graduate studies and the development of this thesis work. His extensive knowledge of aircraft structures and design was an inspiration and a true motivator.

I would also like to thank my parents, Teresa and Paul, and my partner Paul Lambert. I am immeasurably grateful for their valuable advice, acceptance, and unwavering support.

Chapter 1
INTRODUCTION

1.1 Background and Motivation

When analyzing flexible structures, such as high aspect ratio airplane wings, a typical linear elastic analysis may not adequately capture the true response of the structure. In linear elastic analysis, deflections are assumed to be small relative to member size, meaning there is a relatively small difference between the undeformed and deformed shape of the structure. The static equilibrium equations for the structure are based on the undeformed geometry and the strains are linear functions of the displacements, resulting in linear equilibrium equations. However, when the displacements and rotations of the structure become large, it is necessary to include the effects of geometric stiffness in order to obtain results that are accurate and physically representative. Geometric stiffness is a function of the internal force in the member and the change in location of its nodes. For a structure to remain in static equilibrium as it undergoes finite deformations, forces are developed in the members to balance externally applied force. These internal forces caused by the change in geometry result in an increase in stiffness, known as "stress stiffening" or "geometric stiffening". To solve structures of this nature, it is more accurate to refer the static equilibrium equations to the deformed configuration. The strains are then non-linear functions of the displacements, resulting in non-linear equilibrium equations. This kind of problem is called geometrically non-linear.

Analysis of geometrically non-linear problems typically starts with implementation of a finite element method (FEM) that results in a set of non-linear algebraic equations, $R(u) = P$, where P is the externally applied load and R is a non-linear

function of the nodal displacements. This set of non-linear equations is linearized by dividing the problem into incremental load steps, resulting in the relation

$$[K_T]\Delta u = \Delta P \tag{1.1}$$

where $[K_T]$ is the tangent stiffness matrix, the calculation of which is crucial for finding a solution to the geometrically non-linear problem. To increase the accuracy of the linearized approximation, the incremental method is combined with an iterative method in order to converge on the exact solution. The tangent stiffness matrix is calculated at each iteration.

Extensive research and development has been done regarding the derivation of the non-linear governing equations and the tangent stiffness matrices for various structural finite elements, and many textbooks have been written on the topic of non-linear finite elements, including [1], [3], [4], and [22]. Previous work is primarily focused on analytic derivations starting from FEM variational methods, such as in [12], [13], [14], and [15] where the principal of minimum potential energy is used to derive the non-linear equations, or in [2] and [21], where the non-linear equations are derived using the principal of virtual work. Additionally, existing assumptions may be used in conjunction with FEM, such as in [16] where conventional beam-column theory is employed in the derivation for a non-linear space frame. The work done in [9] and [19] involves derivation of the geometric stiffness matrix beginning with a load perturbation of the equilibrium equations. In all these previous works, the non-linear finite element equations and resulting geometric or tangent stiffness matrices are arrived at through time-consuming, tedious, and complicated analytic derivations. With the advent of more sophisticated computing power, and in an effort to allow for the derivation of more complex finite elements, work has been done in utilizing numerical methods in the derivations. Symbolic algebra is used in [8] to carry out the derivations that are too complex to do analytically. Some work has also been done using derivative approximation methods, such as a forward finite difference Frechét

derivative used in [7] and a complex variable derivative method used in [5]. All of these methods, however, still lack a level of ease and generality that would allow for the creation of a simple and general computer program to solve geometrically non-linear problems.

The motivation for the present work is to present a method for numerically generating tangent stiffness matrices in a manner that results in a solution method for geometrically non-linear structures that is completely general and easily implementable as a computer program. The solution technique begins with a load perturbation of the non-linear force equilibrium equations, similar to that done in [9] and [19]. As long as the method for determining internal member forces is known – which it typically is from linear elastic analysis – then the tangent stiffness matrix is calculated numerically in exactly the same manner for every finite element type. While there may be computational costs for larger structures, proceeding in this manner avoids tedious and complicated analytic derivations or unnecessary assumptions unique to element types that could limit solution accuracy.

1.2 Report Structure

The present work proceeds with Chapter 2 in which the tangent stiffness matrix is derived and an incremental-iterative method for solving the geometrically nonlinear problem is introduced, specifically the Newton-Raphson method. The details for using complex variable derivatives to approximate sensitivities in order to numerically calculate the tangent stiffness matrix is then introduced and derived.

Chapters 3, 4, and 5 present the solution process for three different structural elements: 3D truss, membrane, and 3D beam elements. Each chapter contains a detailed discussion of how the elements are defined in space, and how the internal forces for each element type are calculated, which is a necessary process to proceed with the numerical generation of the tangent stiffness matrix. Additionally, each chapter contains numeric examples comparing solutions found using the numerical

tangent stiffness matrix with results obtained in literature using analytically calculated tangent stiffness matrices or using commercial finite element software. The present solution method was coded in MATLAB in order to solve the examples. A pseudocode for the solution method is provided in Appendix A.

Finally, Chapter 6 is a concluding discussion of the results and possible directions for future improvements and research.

Chapter 2
SOLUTION METHOD

2.1 Tangent stiffness matrix

The solution method for a geometrically non-linear problem begins with a linearization of the non-linear equilibrium equations, such as the general force equilibrium equations given by,

$$\mathbf{P_{ext}} = \mathbf{F_{int}}(\mathbf{u}) \tag{2.1}$$

where $\mathbf{F_{int}}$ is a vector of nodal internal member forces, which are functions of the nodal degree of freedom (DOF) displacements[1], $\mathbf{u}$. $\mathbf{P_{ext}}$ is a vector of externally applied loads.[2] Assuming there is a known set of nodal displacement DOFs, $\mathbf{u_0}$, that satisfies equation (2.1), the equilibrium equations can be linearized by perturbing the force about this known solution point. A small perturbation of the externally applied load corresponds to a perturbation in the nodal DOF displacements and equation (2.1) becomes

$$\mathbf{P_{ext}} + \mathbf{dP} = \mathbf{F_{int}}(\mathbf{u_0} + \mathbf{du}) \tag{2.2}$$

A first-order taylor series expansion of the right hand side of equation (2.2) results in

[1]The term "displacements" is used in this chapter to denote both linear displacements and rotations.

[2]Generally, the external forces may be follower forces and therefore will also be dependant on the nodal displacement degrees of freedom. However, for the purpose of simplicity, follower forces are not considered in this derivation.

$$\mathbf{P_{ext}} + \mathbf{dP} = \mathbf{F_{int}}(\mathbf{u_0}) + \left.\frac{\partial \mathbf{F_{int}}}{\partial \mathbf{u}}\right|_{u=u_0} \mathbf{du} \tag{2.3}$$

Since $\mathbf{u_0}$ is a solution that satisfies equation (2.1), equation (2.3) reduces to

$$\mathbf{dP} = \left.\frac{\partial \mathbf{F_{int}}}{\partial \mathbf{u}}\right|_{u=u_0} \mathbf{du} = [K_T(\mathbf{u_0})]\mathbf{du} \tag{2.4}$$

where

$$[K_T(\mathbf{u})] = \frac{\partial \mathbf{F_{int}}}{\partial \mathbf{u}} = \begin{bmatrix} \frac{\partial F_{int,1}}{\partial u_1} & \frac{\partial F_{int,1}}{\partial u_2} & \cdots & \frac{\partial F_{int,1}}{\partial u_m} \\ \frac{\partial F_{int,2}}{\partial u_1} & \frac{\partial F_{int,2}}{\partial u_2} & \cdots & \frac{\partial F_{int,2}}{\partial u_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_{int,m}}{\partial u_1} & \frac{\partial F_{int,m}}{\partial u_2} & \cdots & \frac{\partial F_{int,m}}{\partial u_m} \end{bmatrix} \tag{2.5}$$

is the tangent stiffness matrix, with m being the total number of nodal DOFs in the structure. From equation (2.5) it can be seen that the tangent stiffness matrix is essentially a matrix of sensitivities. In particular, it is the sensitivities of the internal member forces to perturbations in the nodal displacement DOFs of the system. The tangent stiffness matrix contains information regarding both the linear elastic and geometric stiffness of the structure.

2.2 Incremental-iterative solution technique: Netwon-Raphson method

As with any set of non-linear equations, the most effective way to solve the geometrically non-linear problem is to use an incremental-iterative technique. For the present work a Newton-Raphson method is used. The externally applied load is divided into increments, $\Delta\mathbf{P}$, and at each increment a Newton-Raphson iteration is run in order to converge on the equilibrium condition, as illustrated in Figure 2.1.

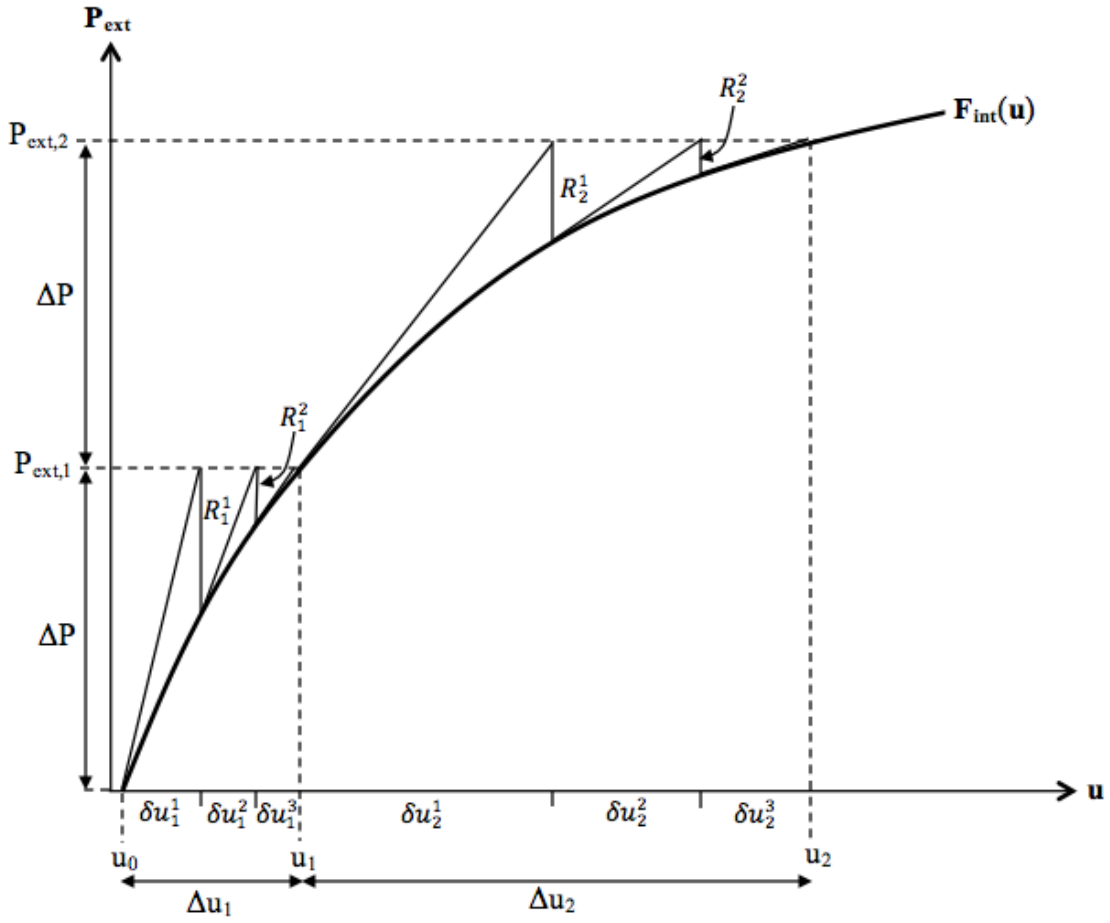


Figure 2.1: Newton-Raphson incremental-iterative solution method

Given an initial configuration for the structure, and thus the initial nodal DOF displacement vector $\mathbf{u_0}$, the initial tangent stiffness matrix, $[K_T(\mathbf{u_0})]$, can be calculated. The externally applied load is then incremented by some small increment, $\Delta\mathbf{P}$, and the linearized system of equations from equation (2.4) is solved for the nodal incremental displacement due to the external load:

$$\delta\mathbf{u_1^1} = [K_T(\mathbf{u_0})]^{-1}\Delta\mathbf{P} = [K_T(\mathbf{u_0})]^{-1}\mathbf{P_{ext,1}} \tag{2.6}$$

Here the subscripts represent the increment number, and the superscripts represent the iteration number.

The internal member forces at the updated configuration, $\mathbf{F_{int}}(\mathbf{u_0} + \delta\mathbf{u_1^1})$, are calculated, and the equilibrium condition from equation (2.1) is checked. That is, for equilibrium to be satisfied,

$$\mathbf{P_{ext,1}} - \mathbf{F_{int}}(\mathbf{u_0} + \delta\mathbf{u_1^1}) = 0 \tag{2.7}$$

However, since a linear approximation was used to solve for the nodal displacements, it is likely that there is an unbalance in the force equilibrium, otherwise known as a residual force $\mathbf{R}$. The equilibrium equation is then,

$$\mathbf{P_{ext,1}} - \mathbf{F_{int}}(\mathbf{u_0} + \delta\mathbf{u_1^1}) = \mathbf{R_1^1} \tag{2.8}$$

and iteration proceeds to converge on a solution for the nodal DOF displacements that results in the residual being as close to zero as possible. This is done by calculating the updated tangent stiffness matrix from the updated nodal displacements and solving for the nodal displacement increments due to the residual force,

$$\delta\mathbf{u_1^2} = [K_T(\mathbf{u_0} + \delta\mathbf{u_1^1})]^{-1}\mathbf{R_1^1} \tag{2.9}$$

The updated internal member forces, $\mathbf{F_{int}}(\mathbf{u_0} + \delta\mathbf{u_1^1} + \delta\mathbf{u_1^2})$, are again calculated, and the equilibrium equation becomes,

$$\mathbf{P_{ext,1}} - \mathbf{F_{int}}(\mathbf{u_0} + \delta\mathbf{u_1^1} + \delta\mathbf{u_1^2}) = \mathbf{R_1^2} \tag{2.10}$$

If the residual is not close to zero within an acceptable tolerance, then the process is repeated. If the residual is approximately zero, then the equilibrium condition has been achieved and the new updated equilibrium configuration due to the externally applied load of $\mathbf{P_{ext,1}}$ is $\mathbf{u_1} = \mathbf{u_0} + (\delta\mathbf{u_1^1} + \delta\mathbf{u_1^2} + \cdots + \delta\mathbf{u_1^i}) = \mathbf{u_0} + \Delta\mathbf{u_1}$, where i is the number of iterations required to achieve the equilibrium condition.

The solution process continues by again incrementing the externally applied load such that $\mathbf{P_{ext,2}} = \mathbf{P_{ext,1}} + \mathbf{\Delta P}$, and proceeding with the iteration on the linear system of equations. This is repeated until the full applied load is reached, with the final solution being $\mathbf{u_j} = \mathbf{u_0} + \Delta\mathbf{u_1} + \Delta\mathbf{u_2} + \cdots + \Delta\mathbf{u_j}$, where j is the total number of load increments.

While it is one of the simplest iterative methods to implement, the Newton-Raphson method is not the most numerically robust. In particular, for buckling problems, or structures with snap-through tendencies, there may be a point at which the tangent stiffness matrix becomes singular and the solution 'blows up'. For this reason, more sophisticated incremental-iterative methods are often used, including the arc length method, the generalized displacement control method as presented in [21], or the BFGS iteration method as presented in [1]. However, for this current work, only simple structural problems are analyzed for which the Newton-Raphson method is acceptable.

2.3 Numerically Generated Tangent Stiffness Matrix

Calculating the tangent stiffness matrix is integral to finding a solution to the geometrically non-linear problem. In order to develop a solution method that is general and easily implementable, the present work focuses on a method for calculating the tangent stiffness matrix numerically.

As shown in equation (2.5), the elements of the tangent stiffness matrix are sensitivities of the internal member forces to perturbations in the global nodal DOF displacements of the structure. There are many different numerical differentiation

techniques that could be employed to calculate these sensitivities, most common being any finite difference method, such as the central difference method. However, while finite difference methods are typically easy to implement, step size often becomes an issue with these methods. It is desirable to choose a smaller step size in order to obtain greater accuracy, but with a finite difference method a step size that is too small can result in subtractive cancellation round-off error and numerical instability. This type of error can be completely avoided, however, if the first derivative is estimated using a complex variable differentiation method.

An approximation of the first derivative using complex variable analysis begins by defining a complex function, $f(z) = f(x + iy) = u(x, y) + iv(x, y)$, where u and v are real functions. Combining the Cauchy-Riemann equations derived from complex function differentiation,

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \tag{2.11a}$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} \tag{2.11b}$$

and the definition of a derivative, yields

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = \lim_{h \to 0} \frac{v(x + i(y + h)) - v(x + iy)}{h}. \tag{2.12}$$

Since we are dealing with real structures, where the forces and degrees of freedom are real values, we restrict ourselves to the real axis, meaning that $y = 0$, so $v(x, y) = v(x) = 0$ and $f(x + iy) = f(x) = u(x)$. Equation (2.12) then becomes

$$\frac{\partial f(x)}{\partial x} = \lim_{h \to 0} \frac{v(x + ih)}{h} = \lim_{h \to 0} \frac{Imag[f(x + ih)]}{h}. \tag{2.13}$$

To approximate the derivative, h is assumed to be a very small finite step size, and equation (2.13) becomes

$$\frac{\partial f(x)}{\partial x} \approx \frac{Imag[f(x+ih)]}{h}. \tag{2.14}$$

Detailed derivation of the 'complex-step derivative approximation' and discussion regarding its numerical robustness compared to other approximation methods is presented in [10], [11], and [20]. Of particular importance is that complex variable differentiation is relatively easy to implement while still allowing for a very small step size that translates into very high accuracy without the problem of round-off errors.

Equation (2.14) can then be used to approximate the sensitivities of the internal member forces to perturbations in the global nodal DOFs, and thus used to find the components of the tangent stiffness matrix. The code, instead of being written using real variables, is written using complex variables for all entities involved.

To achieve this, the first step is to numerically perturb each global nodal DOF. This is done by defining a nodal displacement vector populated with zeros, except for the component of the vector corresponding to the perturbed DOF, which is updated from zero to $+ih$. This is equivalent to saying that only one particular DOF has been slightly perturbed while the rest have remained fixed. The resulting vector is the nodal displacement perturbation vector,

$$\mathbf{du}_{perturbed} = \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ 0 + ih \\ \vdots \\ 0 \end{Bmatrix} \tag{2.15}$$

where h is the chosen step size for the complex variable derivative approximation. With the global nodal perturbation vector established, the global DOFs for the

structure are updated as follows:

$$\mathbf{u}_{perturbed} = \mathbf{u} + \mathbf{du}_{perturbed} \tag{2.16}$$

The updated nodal DOFs in the local element axis system can then be calculated. The specific calculation of the local nodal DOF motions from the global nodal DOF motions is dependant on the particular finite element being used in the structural problem, and is typically a known procedure from linear-elastic analysis. It is discussed in further detail for three particular element types in following chapters of this work.

Next, the internal member forces resulting form the perturbed local nodal DOF motions are calculated. Since the internal structural forces are a result of elastic deformation in the member, the linear-elastic equations are used to calculate the forces as follows:

$$[k_E]\mathbf{u}_{perturbed,\, local} = \mathbf{f}_{\mathbf{int},perturbed} \tag{2.17}$$

where $[k_E]$ is the linear-elastic stiffness matrix for the particular structural element, and $\mathbf{f_{int}}$ are the internal elastic forces referred to in the local element axis system. Since $\mathbf{f_{int}}$ are the forces resulting from a complex variable perturbation, the values of the force vector will themselves be complex values (the linear-elastic stiffness matrix is of course real valued, but complex algebra is used in the computer code). Note that while large motions are allowed, it is assumed that in their own local coordinates, elements deform only by small deformations. This justifies using linear force-deformation relations in those element coordinate systems.

The local internal forces are transformed from the element local axis system to the global axis system using a transformation matrix, $[T]$,

$$\mathbf{F}_{\mathbf{int},perturbed} = [T]^T \mathbf{f}_{\mathbf{int},perturbed} \tag{2.18}$$

resulting in a complex valued vector representing the nodal member forces in the global axes due to a complex variable perturbation of one of the global nodal DOFs.

Equation (2.14) is used to numerically approximate the sensitivities of these member forces to the perturbation of the specific DOF as follows:

$$\frac{\partial \mathbf{F}_{\mathbf{int},perturbed}}{\partial u_p} = \frac{Imag[\mathbf{F}_{\mathbf{int},perturbed}]}{h} = \begin{Bmatrix} \frac{\partial F_{int,1}}{\partial u_p} \\ \frac{\partial F_{int,2}}{\partial u_p} \\ \vdots \\ \frac{\partial F_{int,m}}{\partial u_p} \end{Bmatrix} \tag{2.19}$$

where m is the total number of DOFs in the system, and p is the index corresponding to the particular perturbed DOF.

The vector resulting from equation (2.19) becomes the column of the tangent stiffness matrix corresponding to the particular perturbed DOF, specifically the p^{th} column. The above process is repeated for each global nodal DOF perturbation until all the columns of the tangent stiffness matrix have been generated. This is easily executed in a code by creating a loop that runs through each of the global DOFs and constructing the corresponding $[K_T]$ column. This process for numerically generating $[K_T]$ is the same for all structures, regardless of element type, resulting in the desired generality. Note that the resulting tangent stiffness matrix is in general not symmetric.

Chapter 3

TRUSS ELEMENTS

The following is a detailed derivation of the present solution method applied to geometrically non-linear three-dimensional structures constructed from truss elements. The derivation is followed by examples with calculated results that are compared with results from the literature.

3.1 Solution Method

A simple two-noded truss element is used for analyzing geometrically non-linear three-dimensional truss structures. This element carries axial load only, so in the local element axis has only one degree of freedom at each node. However, when transformed into the global axis system, each node has three degrees of freedom - displacement in the X, Y, and Z directions, as illustrated in Figure 3.1.

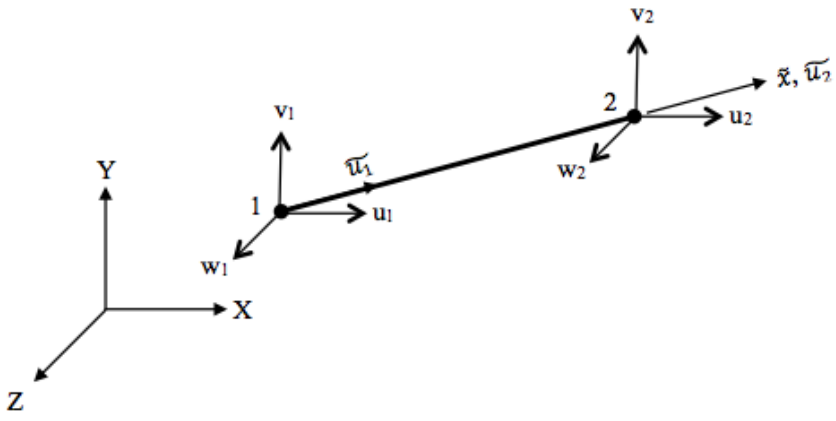


Figure 3.1: Truss element nodal degrees of freedom in local and global axis systems.

Element Definition

The geometry and movement of the truss element in three-dimensional space is completely defined by the location of the two nodes. The node locations are obtained from the location of the nodes in global coordinates. The coordinates are initially defined for the undeformed structure, and as the structure deforms the incremental displacements at the nodes are determined by solving the linear equation

$$[K_T]\mathbf{\Delta u} = (\mathbf{P_{ext}} - \mathbf{F_{int}}) \tag{3.1}$$

where,

$$\mathbf{\Delta u} = \{\Delta X_1 \quad \Delta Y_1 \quad \Delta Z_1 \quad \cdots \quad \Delta X_m \quad \Delta Y_m \quad \Delta Z_m\}^T$$

and m is the total number of nodes in the structure. The global coordinates are updated from the incremental nodal displacements as illustrated in Figure 3.2.

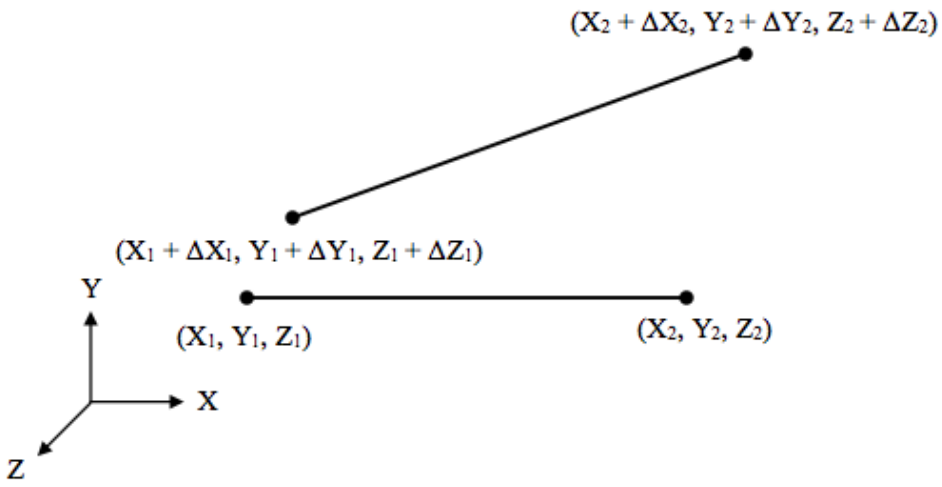


Figure 3.2: Updating of truss element node locations from underformed to deformed configuration.

With the global node locations determined, each truss element in the structure is completely defined in space.

Force Calculation and Solution Process

For a geometrically non-linear structure, finding the increment in internal member forces as the structure changes geometry is instrumental to the solution process. The process of finding the internal member forces is also an integral component of the numerical tangent stiffness generation. The internal forces are a result of linear elastic deformation, which is determined by the motion of the nodes relative to the local truss element axis, $\tilde{\mathbf{x}}$.

The displacements of the nodes relative to the element axis are found by calculating the displacement of node 2 relative to node 1 in the local axis system of each element. Therefore, at node 1 the relative displacement is

$$\tilde{u}_1 = 0 \tag{3.2}$$

The displacement of node 2 relative to node 1 is then simply the extension of the element in the local axis direction, which is the same as the change in element length. Therefore, at node 2 the relative displacement is

$$\tilde{u}_2 = L_{new} - L_{initial} \tag{3.3}$$

where the length of the element, L, is calculated from the node locations in global coordinates as follows:

$$L = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2}$$

Thus, the local elastic deformation vector for the truss element is,

$$\mathbf{u}_{elastic} = \begin{Bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ L_{new} - L_{initial} \end{Bmatrix} \tag{3.4}$$

and is illustrated in Figure 3.3.

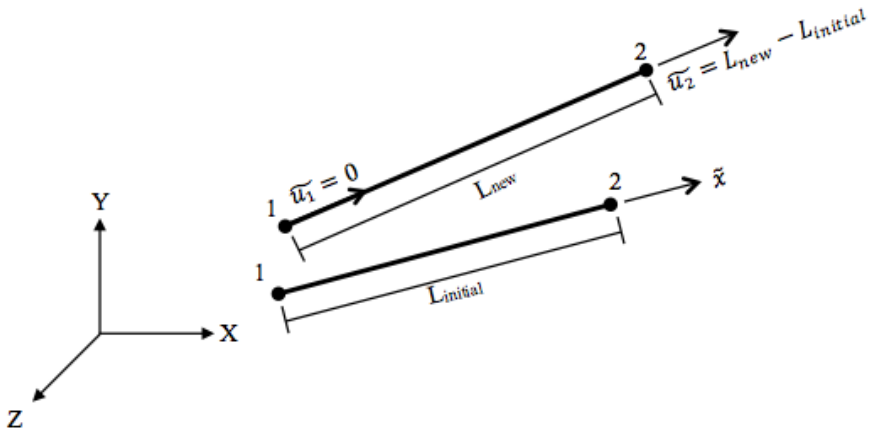


Figure 3.3: Elastic deformation of truss element in the local axis system.

The internal forces in the truss element resulting from the elastic deformation are then found using the linear-elastic equation

$$[k_E]\mathbf{u}_{elastic} = \mathbf{f}_{\mathbf{element}} \tag{3.5}$$

where $[k_E]$ is the local linear-elastic stiffness matrix for the truss element, which is given as

$$[k_E] = \frac{AE}{L_{initial}} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{3.6}$$

where A is the cross-sectional area of the truss element, and E is Young's Modulus for the material.

The forces calculated in equation (3.5) are due only to the change in geometry of the structure and do not include the effects of any pre-existing stresses in the elements. Therefore, if the structure was prestressed, thus having pre-existing internal element forces, it is necessary to account for them by adding them to the calculated forces. The pre-existing internal forces do not change as the structure deforms, but do contribute to the overall nodal forces in the structure.

The local element forces, $\mathbf{f_{element}}$, are transformed into the global coordinate system as follows:

$$\begin{Bmatrix} F_{x,1} \\ F_{y,1} \\ F_{z,1} \\ F_{x,2} \\ F_{y,2} \\ F_{z,2} \end{Bmatrix}_{global} = [T]^T \begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix}_{element} \tag{3.7}$$

where $[T]$ is the transformation matrix from the local axis system to the global axis system. The transformation matrix for the truss element is made up of direction cosines calculated from the global node coordinates, and typically has the following form:

$$[T] = \begin{bmatrix} mx & my & mz & 0 & 0 & 0 \\ 0 & 0 & 0 & mx & my & mz \end{bmatrix} \tag{3.8}$$

where the direction cosines are,

$$mx = \frac{X_2 - X_1}{L}; \quad my = \frac{Y_2 - Y_1}{L}; \quad mz = \frac{Z_2 - Z_1}{L} \tag{3.9}$$

The global forces are then assembled into the full internal force vector for the structure, $\mathbf{F_{int}}$, and the unbalanced load can then be calculated in order to check for equilibrium, as discussed in Section 2.2. The process for calculating the internal member forces as outlined above is also required for numerically generating the tangent stiffness matrix, which is done using a nodal displacement perturbation vector, as described in Section 2.3.

3.2 Numeric Results

The solution method for the present work involving the numerically generated tangent stiffness matrix is implemented in a MATLAB code, which is then used to obtain results for two different truss structure examples. The same truss structures are solved by Levy and Spillers in [9], using an analytically derived geometric stiffness matrix which is added to the linear-elastic stiffness matrix to create the tangent stiffness matrix. For both examples, the final updated nodal coordinates of the structure are compared.

3.2.1 Biot's Two-Bar Truss

The Biot two-bar truss is a classical example of a linearly unstable structure which can only be solved by considering the deformed equilibrium, meaning a tangent stiffness matrix is required.

The structure is made up of two truss elements, with cross-sectional area of $0.0127\,in^2$, initial length of $200\,in$, and a Young's Modulus of $E = 10,000,000\,psi$. A force of $70\,lbs$ is applied at the centre node in the downward direction. Additionally, the truss elements are initially prestressed with each element carrying an initial axial force of $1000\,lbs$. The problem is divided into 10 increments, such that each time equilibrium is achieved in the deformed configuration, the external load is increased by $\Delta P_{ext} = 7\,lbs$, until a final load of $70\,lbs$ is reached. At that point, the final node locations of the structure are output. The result is compared in Table 3.1 with the

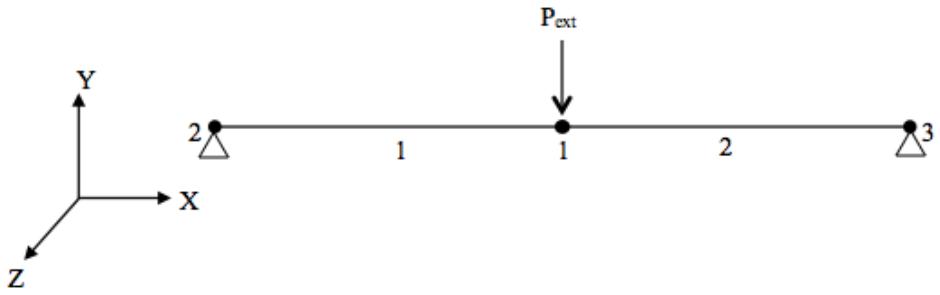


Figure 3.4: Biot's two-bar truss structure.

final nodal positions obtained by Levy using an analytically derived $[K_T]$ in example 3.6.1 in [9]. Only the updated coordinates of node 1 are compared, as node 1 is the only node that is not constrained.

Table 3.1: Comparison of final global node coordinates in Biot's truss problem.

	Levy Analytic $[K_T]$	Numeric $[K_T]$
X	0.20000000×10^{3}	0.20000000×10^{3}
Y	-0.65565399×10^{1}	-0.65564547×10^{1}
Z[1]	0	0

The Y-coordinate result from the numerical $[K_T]$ solution method differs from the analytic $[K_T]$ solution by only 0.0013%.

[1]The truss structure analyzed by Levy is two dimensional. The Z-coordinate is assumed to be zero for all nodes

3.2.2 Cablenet

The above Biot's truss problem was quite small, with only three nodes and two elements. The following example, a prestressed 'cablenet', is analyzed to determine how the numerical method behaves with a slightly larger problem.

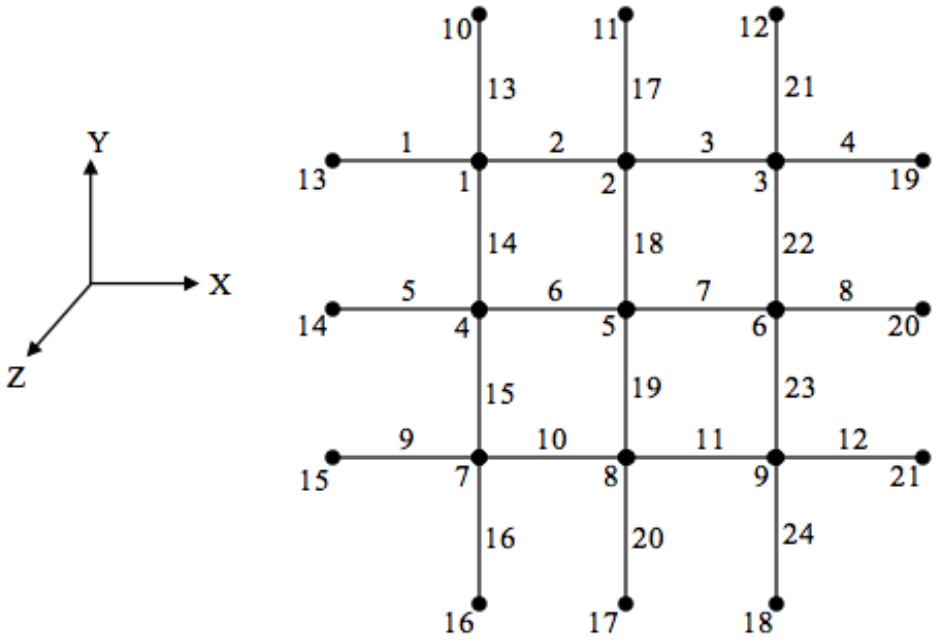


Figure 3.5: Prestressed cablenet made up of space truss elements.

The structure has 21 nodes and is made up of 24 space truss elements, with cross-sectional area of 0.25 in^2, initial length of 60 in, and a Young's Modulus of $E = 30,000,000\,psi$. The structure is constrained in all three degrees of freedom at the 12 outside nodes. A total force of $10,000\,lbs$ is applied downward at the centre node (node 5 in Figure 3.5). The truss elements are prestressed, with each element carrying an axial force of $20,000\,lbs$. The problem is divided into 20 increments, such that each time equilibrium is achieved in the deformed configuration, the external load

is increased by $\Delta P_{ext} = 500\ lbs$. It should be noted that dividing the problem into more increments, with $\Delta P_{ext} = 50\ lbs$ for example, results in a negligible increase in accuracy, but a more than 700% increase in run time. The final coordinate locations of the unconstrained nodes are outputted and compared in Table 3.2 with the results obtained by Levy using an analytically derived tangent stiffness matrix in example 3.6.2 in [9].

Again, even for a slightly larger problem, the method using a numerically calculated tangent stiffness matrix produces the same results as the method using an analytic tangent stiffness matrix, within a percentage difference of less than 0.02%.

Since the results are almost identical, what might be of more interest is the difference in the computational cost. For the above cablenet truss structure problem, a MATLAB code is written which implements the solution process using the geometric stiffness matrix that is analytically derived by Levy in [9] to calculate the tangent stiffness matrix. Both this code and the code in which the tangent stiffness matrix is calculated numerically are run, and the run times are measured using the tic/toc MATLAB function. The numerical $[K_T]$ code takes 2.739 seconds to reach a final solution, whereas the analytic $[K_T]$ code takes only 0.463 seconds to reach the final solution. While this is an almost 500% increase in run time, for such a small problem as this one and others typically analyzed in a research setting, the run time is still quite fast. However, it is easy to see that for much larger problems that may be encountered in industry, the increase in run time would be very significant.

Table 3.2: Comparison of final global node coordinates in prestessed cablenet truss problem

Node		Levy Analytic $[K_T]$	Numeric $[K_T]$	Percent Difference, %
1	X	0.60000097×10^{2}	0.60000097×10^{2}	0
	Y	0.17999990×10^{3}	0.17999990×10^{3}	0
	Z	-0.13456794×10^{1}	-0.13455144×10^{1}	0.01226
2	X	0.12000000×10^{3}	0.12000000×10^{3}	0
	Y	0.17994911×10^{3}	0.17994914×10^{3}	0
	Z	-0.26924913×10^{1}	-0.26921606×10^{1}	0.01228
3	X	0.17999990×10^{3}	0.17999990×10^{3}	0
	Y	0.17999990×10^{3}	0.17999990×10^{3}	0
	Z	-0.13456794×10^{1}	-0.13455144×10^{1}	0.01226
4	X	0.60050886×10^{2}	0.60050860×10^{2}	0
	Y	0.12000000×10^{3}	0.12000000×10^{3}	0
	Z	-0.26924913×10^{1}	-0.26921606×10^{1}	0.01228
5	X	0.12000000×10^{3}	0.12000000×10^{3}	0
	Y	0.12000000×10^{3}	0.12000000×10^{3}	0
	Z	-0.71144376×10^{1}	-0.71131884×10^{1}	0.01756
6	X	0.17994911×10^{3}	0.17994914×10^{3}	0
	Y	0.12000000×10^{3}	0.12000000×10^{3}	0
	Z	-0.26924913×10^{1}	-0.26921606×10^{1}	0.01228
7	X	0.60000097×10^{2}	0.60000097×10^{2}	0
	Y	0.60000097×10^{2}	0.60000097×10^{2}	0
	Z	-0.13456794×10^{1}	-0.13455144×10^{1}	0.01226
8	X	0.12000000×10^{3}	0.12000000×10^{3}	0
	Y	0.60050886×10^{2}	0.60050860×10^{2}	0
	Z	-0.26924913×10^{1}	-0.26921606×10^{1}	0.01228
9	X	0.17999990×10^{3}	0.17999990×10^{3}	0
	Y	0.60000097×10^{2}	0.60000097×10^{2}	0
	Z	-0.13456794×10^{1}	-0.13455144×10^{1}	0.01226

Chapter 4
MEMBRANE ELEMENTS

The following is a detailed derivation of the present solution method applied to geometrically non-linear structures constructed from membrane elements. The derivation is followed by an example with calculated results that are compared with results from the literature.

4.1 Solution Method

A three-noded constant strain triangle (CST) element is used for analyzing geometrically nonlinear membrane structures. This element carries in-plane load only, and so in the local element axis system has six degrees of freedom, two at each of the three nodes, corresponding to planar translational motion. Membrane structures do not carry bending loads and locally there is no out of plane motion or rotation. However, the elements may be assembled in three-dimensional space, and when transformed into the global axis system, each node will have three degrees of freedom - displacement in the X, Y, and Z directions, as illustrated in Figure 4.1.

Element Definition

The geometry and movement of the membrane CST element in three-dimensional space is completely defined by the location of the three nodes. The node locations are obtained from the global nodal coordinates. The coordinates are initially defined for the undeformed structure, and as the structure deforms the incremental displacements at the nodes are determined by solving the linear equation

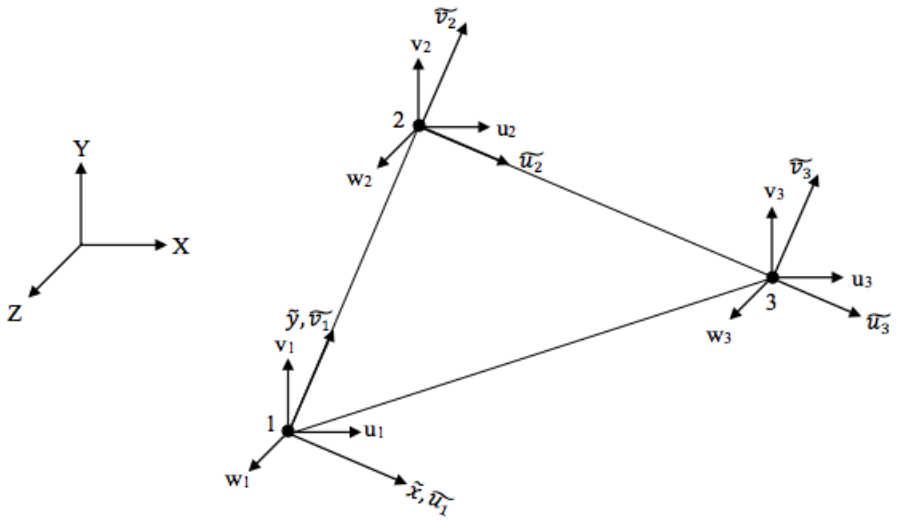


Figure 4.1: Membrane element nodal degrees of freedom in local and global axis systems.

$$[K_T]\mathbf{\Delta u} = (\mathbf{P_{ext}} - \mathbf{F_{int}}) \tag{4.1}$$

where,

$$\mathbf{\Delta u} = \{\Delta X_1 \quad \Delta Y_1 \quad \Delta Z_1 \quad \cdots \quad \Delta X_m \quad \Delta Y_m \quad \Delta Z_m\}$$

and m is the total number of nodes in the structure. The global coordinates are updated from the incremental nodal displacements as illustrated in Figure 4.2.

With the global node locations determined, each membrane element in the structure is completely defined in space.

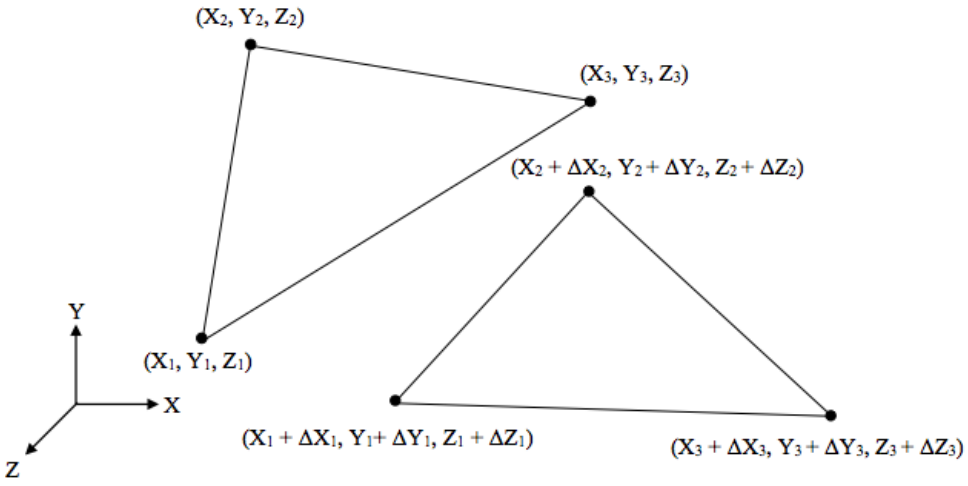


Figure 4.2: Updating of membrane element node locations from undeformed to deformed configuration

Force Calculation and Solution Process

Calculating the internal member forces is an instrumental step in both the solution process and the numerical tangent stiffness matrix generation. The internal member forces are a result of the elastic deformation of the element, which is determined by the motion of the element nodes in the local element axis system.

The displacements of the nodes in the local element axis system are found by first establishing the local axis system and determining the local coordinates, as illustrated in Figure 4.3. A detailed derivation for finding point 4 and determining the lengths of all the element sides that are then equivalent to the node coordinates is given by S. S. Rao on pages 319 – 320 of [18].

The elastic deformation is then found by calculating the displacement of node 2 and node 3 relative to node 1 in the local axis system. Therefore, the displacement of node 1 is

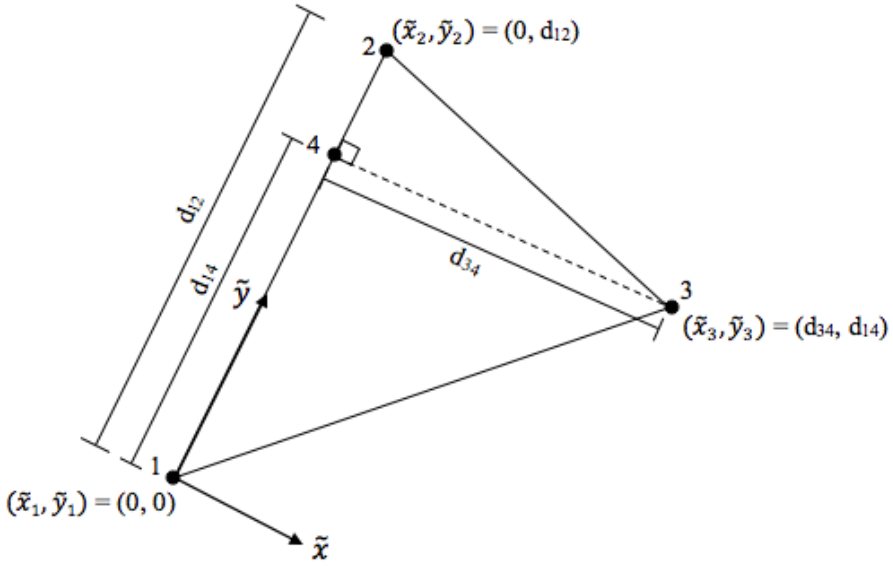


Figure 4.3: Membrane element nodal coordinates in local coordinate system.

$$\mathbf{u_1} = \begin{Bmatrix} \tilde{u}_1 \\ \tilde{v}_1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{4.2}$$

The displacement of node 2 relative to node 1 is then the change in length of the side connecting nodes 1 and 2, and is given by

$$\mathbf{u_2} = \begin{Bmatrix} \tilde{u}_2 \\ \tilde{v}_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ d_{12,new} - d_{12,initial} \end{Bmatrix} \tag{4.3}$$

The displacement of node 3 relative to node 1 is equivalent to the change in local coordinates of the node, and is given by

$$\mathbf{u_3} = \begin{Bmatrix} \tilde{u}_3 \\ \tilde{v}_3 \end{Bmatrix} = \begin{Bmatrix} d_{34,new} - d_{34,initial} \\ d_{14,new} - d_{14,initial} \end{Bmatrix} \tag{4.4}$$

Combining equations (4.2), (4.3), and (4.4) gives the local elastic deformation for the membrane element,

$$\mathbf{u}_{elastic} = \begin{Bmatrix} \mathbf{u_1} \\ \mathbf{u_2} \\ \mathbf{u_3} \end{Bmatrix} \quad (6 \times 1) \tag{4.5}$$

The internal member forces in the membrane element resulting from the elastic deformation are then found using the linear-elastic equation

$$[k_E]\mathbf{u}_{elastic} = \mathbf{f_{element}} \tag{4.6}$$

where $[k_E], (6 \times 6)$, is the local linear-elastic stiffness matrix for the CST membrane element, which is typically given as

$$[k_E] = tA[B]^T[D][B] \tag{4.7}$$

where t is the thickness of the element, and A is the area of the CST element which is calculated from the local node coordinates using the following element area equation:

$$A = \left(\frac{1}{2}\right)((\tilde{x}_3 - \tilde{x}_2)(\tilde{y}_2 - \tilde{y}_1) - (\tilde{x}_2 - \tilde{x}_1)(\tilde{y}_3 - \tilde{y}_2)) \tag{4.8}$$

The matrix $[D]$ is the constituent matrix for the element. Since membranes are typically thin, the constitutive relations used are for a plane stress assumption, and $[D]$ is typically given for an isotropic material as

$$[D] = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{4.9}$$

where E is the Young's Modulus and ν is the Poisson's ratio for the material.

The matrix $[B]$ is a matrix derived from the strain-displacement relations for the element, such that $\epsilon = [B]\mathbf{u}$ (where ϵ is the strain), and for the CST element is typically given as

$$[B] = \frac{1}{2A}\begin{bmatrix} \tilde{y}_{32} & 0 & -\tilde{y}_{31} & 0 & \tilde{y}_{21} & 0 \\ 0 & -\tilde{x}_{32} & 0 & \tilde{x}_{31} & 0 & -\tilde{x}_{21} \\ -\tilde{x}_{32} & \tilde{y}_{32} & \tilde{x}_{31} & -\tilde{y}_{31} & -\tilde{x}_{21} & \tilde{y}_{21} \end{bmatrix} \tag{4.10}$$

where,

$$\left.\begin{aligned} \tilde{x}_{ij} &= \tilde{x}_i - \tilde{x}_j \\ \tilde{y}_{ij} &= \tilde{y}_i - \tilde{y}_j \end{aligned}\right\}$$

Again, as with the truss element, the forces calculated in equation (4.6) are due only to the change in geometry of the structure and do not include the effects of any pre-existing stresses in the elements. Therefore, if the structure was prestressed, thus having pre-existing internal forces, it is necessary to account for them by adding them to the calculated forces.

The local element forces at the element nodes, $\mathbf{f_{element}}$, are transformed into the global coordinate system as follows:

$$\begin{Bmatrix} F_{x,1} \\ F_{y,1} \\ F_{z,1} \\ F_{x,2} \\ F_{y,2} \\ F_{z,2} \\ F_{x,3} \\ F_{y,3} \\ F_{z,3} \end{Bmatrix}_{global} = [T]^T \begin{Bmatrix} f_{x,1} \\ f_{y,1} \\ f_{x,2} \\ f_{y,2} \\ f_{x,3} \\ f_{y,3} \end{Bmatrix}_{element} \tag{4.11}$$

where $[T]$ is the transformation matrix from the local coordinate system to the global coordinate system. The transformation matrix for the CST membrane element is made up of direction cosines for the local element axes calculated from the global node coordinates and element side lengths, and typically has the form:

$$[T] = \begin{bmatrix} l_{34} & m_{34} & n_{34} & 0 & 0 & 0 & 0 & 0 & 0 \\ l_{12} & m_{12} & n_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & l_{34} & m_{34} & n_{34} & 0 & 0 & 0 \\ 0 & 0 & 0 & l_{12} & m_{12} & n_{12} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & l_{34} & m_{34} & n_{34} \\ 0 & 0 & 0 & 0 & 0 & 0 & l_{12} & m_{12} & n_{12} \end{bmatrix} \tag{4.12}$$

where the direction cosines are,

$$\begin{aligned} l_{34} &= \frac{X_3 - X_4}{d_{34}}; \quad m_{34} = \frac{Y_3 - Y_4}{d_{34}}; \quad n_{34} = \frac{Z_3 - Z_4}{d_{34}} \\ l_{12} &= \frac{X_2 - X_1}{d_{12}}; \quad m_{12} = \frac{Y_2 - Y_1}{d_{12}}; \quad n_{12} = \frac{Z_2 - Z_1}{d_{12}} \end{aligned} \tag{4.13}$$

The global forces are then assembled into the full internal force vector for the structure, $\mathbf{F_{int}}$, and the unbalanced load can then be calculated in order to determine if the equilibrium condition has been satisfied, as discussed in Section 2.2. The process for calculating the internal member forces as outlined above is also required for numerically generating the tangent stiffness matrix, which is done using a nodal displacement perturbation vector, as described in Section 2.3.

4.2 Numeric Results

The solution method for the present work involving the numerically generated tangent stiffness matrix is implemented in a MATLAB code, which is then used to obtain results for membrane structures. The prestressed cablenet truss structure that was solved in Section 3.2.2 is remodelled using CST membrane elements and solved using the current MATLAB code. The same structure is solved by Levy in example 6.6.3 in [9] using an analytically derived tangent stiffness matrix, and the results for the final node locations are compared.

4.2.1 Flat Stretched Membrane

The structure is made up of 25 nodes and 32 membrane elements, with thickness $t = 0.004167\ in$, Young's Modulus of $E = 30,000,000\ psi$, and Poisson's Ratio of $\nu = 0.3$. Like the cablenet truss, the structure is constrained in all global degrees of freedom at the 16 outside nodes. A total force of $10,000\ lbs$ is applied downward at the centre node (node 5 in Figure 4.4). The problem is divided into 5 increments, meaning each time equilibrium is achieved the external load is increased by $\Delta P_{ext} = 2000\ lbs$. The final coordinate locations of the unconstrained nodes are outputted and compared in Table 4.1 with the Levy results.

From the results it can be seen that the method using a numerically derived tangent stiffness matrix yields the same results as the method using an analytic tangent stiffness matrix, within a percentage difference of less than 0.05%.

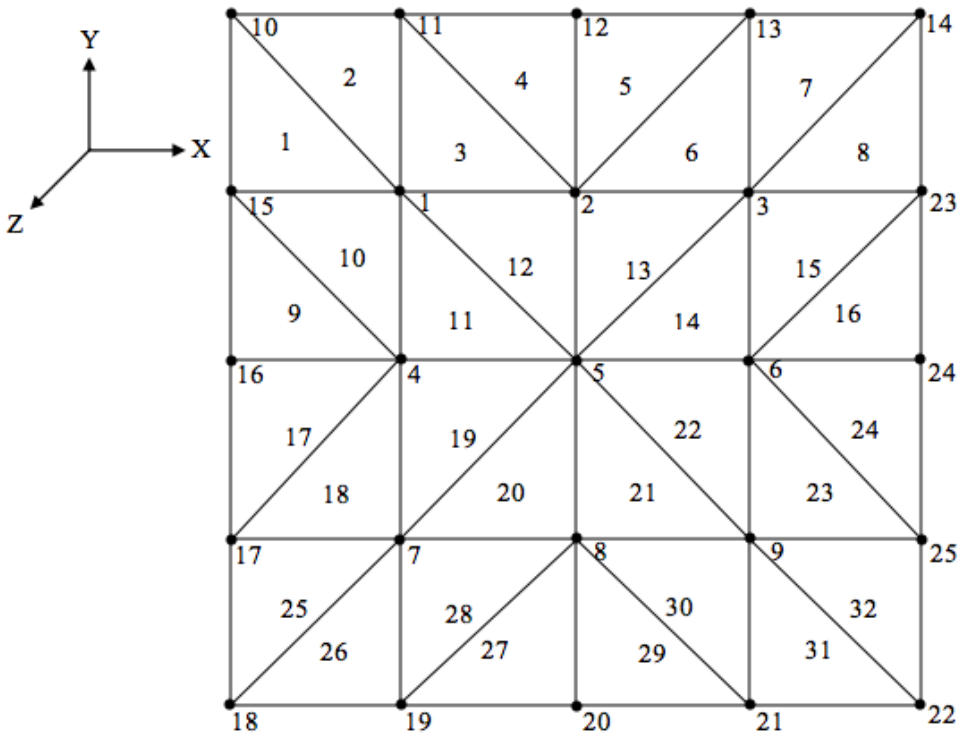


Figure 4.4: Flat stretched membrane modelled using CLT elements.

Table 4.1: Comparison of final global node coordinates in prestessed flat stretched membrane problem.

Node		Levy Analytic $[K_T]$	Numeric $[K_T]$	Percent Difference, %
1	X	-0.59985×10^2	-0.59985×10^2	0
	Y	0.59985×10^2	0.59985×10^2	0
	Z	-0.14309×10^1	-0.14315×10^1	0.0419
2	X	-0.25607×10^{-15}	0.00000×10^2	0
	Y	0.59983×10^2	0.59983×10^2	0
	Z	-0.26046×10^1	-0.26045×10^1	0.0038
3	X	0.59985×10^2	0.59985×10^2	0
	Y	0.59985×10^2	0.59985×10^2	0
	Z	-0.14309×10^1	-0.14315×10^1	0.0419
4	X	-0.59983×10^2	-0.59983×10^2	0
	Y	-0.82747×10^{-15}	0.00000×10^2	0
	Z	-0.26045×10^1	-0.26052×10^1	0.0269
5	X	0.39495×10^{-16}	-0.00000×10^2	0
	Y	0.36615×10^{-16}	-0.00000×10^2	0
	Z	-0.66418×10^1	-0.66404×10^1	0.0211
6	X	0.59983×10^2	0.59983×10^2	0
	Y	0.82668×10^{-15}	-0.00000×10^2	0
	Z	-0.26045×10^1	-0.26052×10^1	0.0269
7	X	-0.59985×10^2	-0.59985×10^2	0
	Y	-0.59985×10^2	-0.59985×10^2	0
	Z	-0.14309×10^1	-0.14315×10^1	0.0419
8	X	0.27357×10^{-15}	-0.00000×10^2	0
	Y	-0.59983×10^2	-0.59983×10^2	0
	Z	-0.26046×10^1	-0.26045×10^1	0.0038
9	X	0.59985×10^2	0.59985×10^2	0
	Y	-0.59985e2	-0.59985×10^2	0
	Z	0.14309×10^1	0.14315×10^1	0.0419

Chapter 5
BEAM ELEMENTS

The following is a detailed derivation of the present solution method applied to geometrically non-linear structures constructed from three-dimensional beam elements. The derivation is followed by examples with calculated results that are compared with results from the literature and commercial FEM software.

5.1 Solution Method

A two-noded three-dimensional beam element is used for analyzing geometrically nonlinear beam structures. Beam elements are capable of transferring axial load, shear load, and bending loads, so at each node there are six degrees of freedom, resulting in twelve degree of freedom motions total for the element. In particular, at each node there are three displacement degrees of freedom and three rotational degrees of freedom, in both the local element and global axis systems, as illustrated in Figure 5.1.

The beam element is a significantly more complex than the truss and membrane elements previously analyzed because it has not only nodal linear displacements, but also nodal rotations. It is mainly the small nodal rotations that contribute to the linear elastic forces in the beam element. However, as the whole structure deflects due to the applied load, the beam element also undergoes finite rigid body rotations in space. In other words, the beam element is a co-rotational element, and it is necessary to distinguish between and separate the small relative rotations from the finite rotations at the nodes. When analyzing the beam element it is necessary to keep track of not only the global coordinates of each node, but also the nodal finite

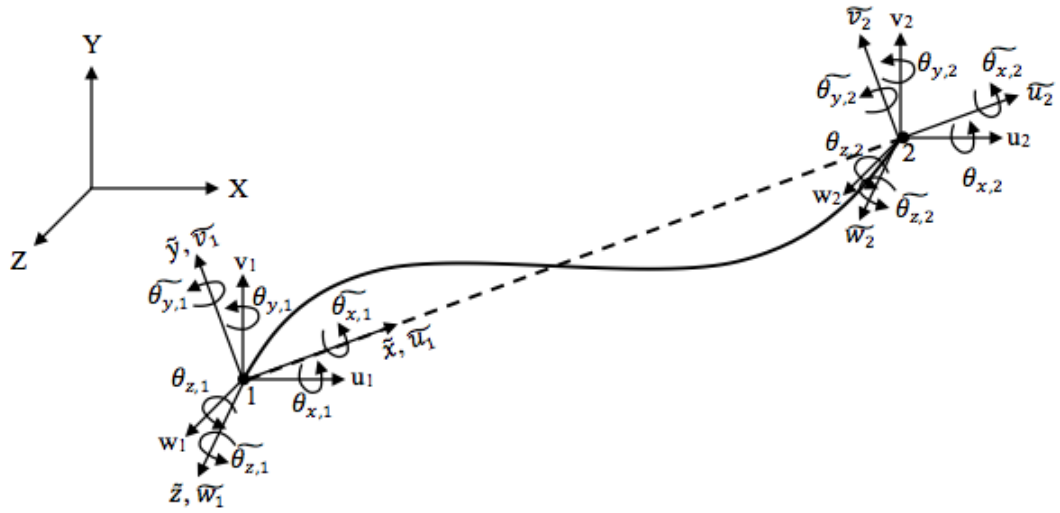


Figure 5.1: Beam element nodal degrees of freedom in local and global axes systems.

rotations. This is the primary difficulty of the beam element, and there have been many different attempts at addressing this issue. For example, in [21], Yang and Kuo use six separate axis systems to track and locate the beam element, while in [9] and [19], Levy and Spillers use primarily rotation matrices to position the element in space. The present method uses considerations from both these approaches.

Element Definition

The geometry and movement of the three-dimensional beam element in space is completely defined by the location of the nodes and two orthogonal axis systems: one defining the orientation of the left end cross-section and one defining the orientation of the right end cross-section. This allows for a complete description of the beam, including both the rigid body motion and the elastic deformation.

The node locations are obtained from the global node coordinates. The coordinates are initially defined for the undeformed structure. As the structure deforms,

the incremental displacements and incremental rotations at the nodes are determined by solving the linear equation

$$[K_T]\mathbf{\Delta u} = (\mathbf{P_{ext}} - \mathbf{F_{int}}) \tag{5.1}$$

where,

$$\begin{aligned}\mathbf{\Delta u} = \{&\Delta X_1 \quad \Delta Y_1 \quad \Delta Z_1 \quad \Delta\theta_{x,1} \quad \Delta\theta_{y,1} \quad \Delta\theta_{z,1} \quad \cdots \\ &\cdots \quad \Delta X_m \quad \Delta Y_m \quad \Delta Z_m \quad \Delta\theta_{x,m} \quad \Delta\theta_{y,m} \quad \Delta\theta_{z,m}\}^T,\end{aligned}$$

and m is the total number of nodes in the structure. The motion in global coordinates is updated from the incremental nodal displacements as illustrated in Figure 5.2.

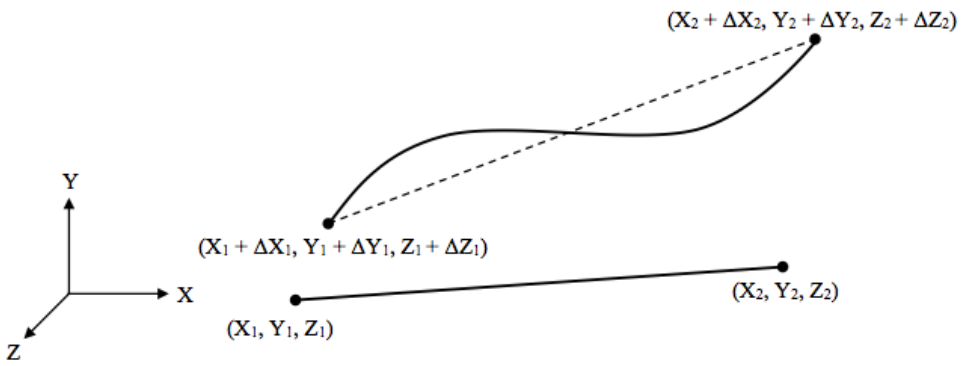
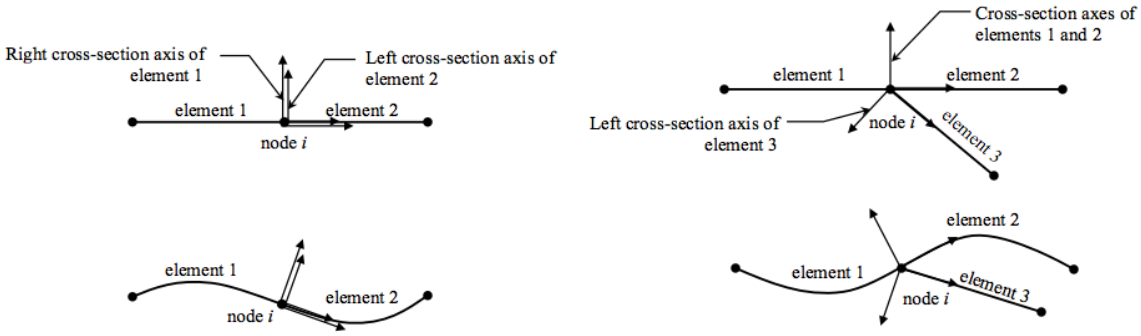


Figure 5.2: Updating of beam element global node locations from undeformed to deformed configuration.

Updating nodal rotations and tracking the cross-section axis systems is more complicated. As indicated, each element has a left and right cross-section axis system that

defines the rotation of the two element nodes. For a single long structure that is made of many beam elements, such as a cantilevered beam, the left cross-section of one element and the right cross-section of the adjoining element will be aligned and will rotate together as the structure deforms, as illustrated in Figure 5.3a. In this case, the nodal rotation is uniquely defined by the rotation of the cross-sections. However, for structures where two or more elements at different orientations join at a node, as illustrated in Figure 5.3b, the cross-section axes will no longer be aligned and the rotations of the cross-sections alone will not define the rotation of the node.



(a) Aligned Cross-Section Axes

(b) Unaligned Cross-Section Axes

Figure 5.3: Alignment of beam element left and right cross-section axes in multi-element structures.

In order to address this problem and define the rotation of a node in a multi-element structure, a single nodal rotation axis system is defined at each node. From this axis system a rotation matrix for the node is determined, $[L_{iG}]$, where the subscript 'iG' indicates that the matrix represents a rotation from the global coordinate system to the nodal rotation coordinate system for node i. Initially the nodal rotation axis system is aligned with the global axis system, but as the structure deforms the nodal rotation matrix is updated based on the incremental node rotations from the solution to equation (5.1). These rotations are small finite rotations.

For small finite rotations, a rotation matrix can usually be approximated by using small angle approximations and neglecting second order terms as follows:

$$[L] = \begin{bmatrix} 1 & d\psi & -d\theta \\ -d\psi & 1 & d\phi \\ d\theta & -d\phi & 1 \end{bmatrix} \tag{5.2}$$

where $\begin{Bmatrix} d\phi \\ d\theta \\ d\psi \end{Bmatrix}$ are small Euler angles, which in the case of the beam element are equivalent to the small-angle rotations at the node, $\begin{Bmatrix} \Delta\theta_x \\ \Delta\theta_y \\ \Delta\theta_z \end{Bmatrix}$. However, because equation (5.2) is only a first order approximation, it does not satisfy the properties of a transformation matrix. Specifically, $[L]^T[L] \neq [I]$, and use of this approximated rotation matrix would introduce errors into the solution. Therefore, to update the nodal rotation matrix from the small-angle rotations, a rotation matrix that has all the precise transformation matrix properties needs to be constructed. This is done using the Euler Theorem, which states that every coordinate axis rotation can be represented by a unique single angular rotation, α, about one axis, $\mathbf{n} = \{n_1 \quad n_2 \quad n_3\}$, where α, n_1, n_2, and n_3 are called the Euler parameters. Using the Euler parameters, a complete and correct rotation matrix can be constructed as follows:

$$[R] = \begin{bmatrix} \cos\alpha + (1-\cos\alpha)n_1^2 & (1-\cos\alpha)n_1n_2 - n_3\sin\alpha & (1-\cos\alpha)n_1n_3 + n_2\sin\alpha \\ (1-\cos\alpha)n_2n_1 + n_3\sin\alpha & \cos\alpha + (1-\cos\alpha)n_2^2 & (1-\cos\alpha)n_2n_3 - n_1\sin\alpha \\ (1-\cos\alpha)n_3n_1 - n_2\sin\alpha & (1-\cos\alpha)n_3n_2 + n_1\sin\alpha & \cos\alpha + (1-\cos\alpha)n_3^2 \end{bmatrix} \tag{5.3}$$

The Euler parameters are constructed from the small-angle rotations as follows:

$$\Delta\alpha = \sqrt{\Delta\theta_x^2 + \Delta\theta_y^2 + \Delta\theta_z^2} \tag{5.4}$$

$$\begin{Bmatrix} n_1 \\ n_2 \\ n_3 \end{Bmatrix} = \frac{1}{\sqrt{\Delta\theta_x^2 + \Delta\theta_y^2 + \Delta\theta_z^2}} \begin{Bmatrix} \Delta\theta_x \\ \Delta\theta_y \\ \Delta\theta_z \end{Bmatrix} \tag{5.5}$$

Using equations (5.4) and (5.5) in equation (5.3) results in an incremental nodal rotation matrix, $[R(\Delta\alpha)]$, which is then used to update the full nodal rotation matrix as follows:

$$[L_{iG}]_{updated} = [R(\Delta\alpha)][L_{iG}] \tag{5.6}$$

From the updated nodal rotation matrix the new left and right cross-section axis systems can be determined. The cross-section axes are defined as three orthogonal basis vectors with components *relative to the nodal rotation matrix*. The 1-axis basis vector is normal to the cross-section, and the 2-axis and 3-axis basis vectors point in the principal directions of the cross-section, as illustrated in Figure 5.4.

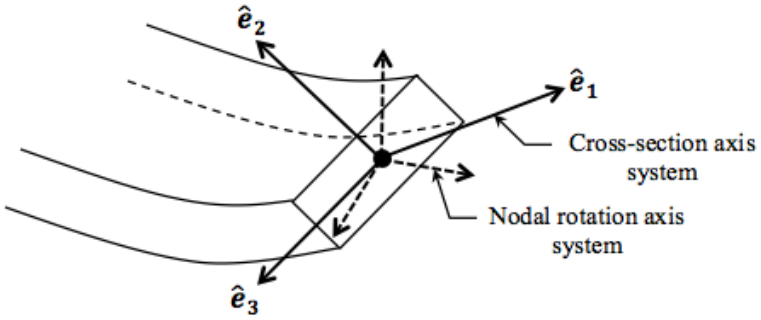


Figure 5.4: Beam element cross-section axis system basis vectors.

These basis vectors are used to construct a rotation matrix for the cross-section,

$[L_{Li}]$ for the left node and $[L_{Ri}]$ for the right node, with each basis vector being a row in the matrix. The subscripts 'Li' and 'Ri' indicate that the rotation matrices represent a rotation from the nodal rotation coordinate system at node i to the cross-section coordinate system.

As the structure deforms the cross-sections rotate with the nodes, but the relative relationship between the nodal rotation axes and the cross-section axes does not change, meaning the rotation matrices $[L_{Li}]$ and $[L_{Ri}]$ are constants throughout the solution process. Therefore, with the rotations from global to nodal and from nodal to cross-section known, finding the orientation of the left and right cross-sections with respect to the global axis system is achieved by a simple rotation matrix calculation, as follows:

$$[L_{LG}] = [L_{Li}][L_{iG}] \tag{5.7}$$

$$[L_{RG}] = [L_{Ri}][L_{iG}] \tag{5.8}$$

where the subscripts 'LG' and 'RG' indicate that the rotation matrices represent a rotation from the global coordinate system to the left and right cross-section coordinate systems.

With the global node locations and global cross-section axis systems determined, each beam element in the structure is completely defined in space.

Force Calculation and Solution Process

Calculating the internal member forces is an instrumental step in both the solution process and the tangent stiffness matrix generation. The internal forces are a result of elastic deformation and not rigid body motion, so for the beam element the elastic deformation must be extracted from the full motion of the beam. This is achieved by defining a beam element reference axis and then determining the motion of the

nodes relative to that reference axis. The beam element reference axis defines the rigid body motion of the beam element. Motions of the nodes relative to this beam element reference axis system determines the elastic forces and moments in the element.

The beam element axis system is made up of the three basis vectors $\hat{\mathbf{e}}_{\mathbf{B1}}$, $\hat{\mathbf{e}}_{\mathbf{B2}}$, and $\hat{\mathbf{e}}_{\mathbf{B3}}$, as illustrated in Figure 5.5.

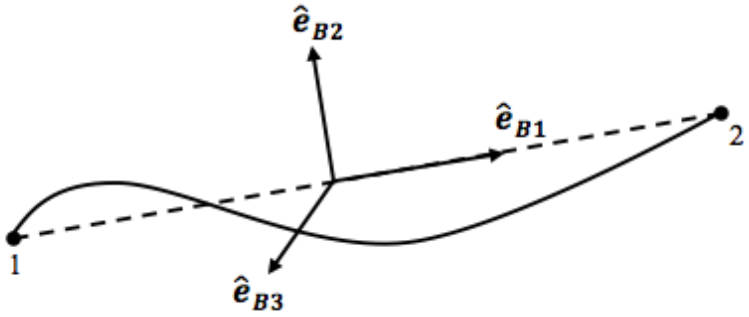


Figure 5.5: Beam element reference axis system.

The vector $\hat{\mathbf{e}}_{\mathbf{B1}}$, which is the beam element reference axis, is typically defined as pointing from the left node to the right node, and can be easily calculated from the node locations in global coordinates as follows:

$$\hat{\mathbf{e}}_{\mathbf{B1}} = \frac{1}{L}\begin{Bmatrix} X_2 - X_1 \\ Y_2 - Y_1 \\ Z_2 - Z_1 \end{Bmatrix} \tag{5.9}$$

$$\text{where} \quad L = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2}$$

The question then becomes how to define the $\hat{\mathbf{e}}_{\mathbf{B2}}$ and $\hat{\mathbf{e}}_{\mathbf{B3}}$ basis vectors. If the beam element did not bend or twist along its length, then the end cross-sections would remain aligned and the two basis vectors would simply align with the principal axes of the cross-sections. However, typically the left and right cross-sections do not remain

rotationally aligned. It is logical to use some combination of the two cross-section axis systems to define the beam element axis system. One possibility is to align the $\hat{\mathbf{e}}_{\mathbf{B2}}$ and $\hat{\mathbf{e}}_{\mathbf{B3}}$ vectors with only the left cross-section or only the right cross-section. In the present work, however, an average of the two cross-sections is used, in order to capture all motions of the element. For interest, a comparison of all three methods is presented in Appendix B.

First, the 2-axis of the left cross-section, $\hat{\mathbf{e}}_{\mathbf{L2}}$, and the 2-axis of the right cross section, $\hat{\mathbf{e}}_{\mathbf{R2}}$, are extracted from the cross-section rotation matrices, $[L_{LG}]$ and $[L_{RG}]$. Taking the cross-product of a cross-section 2-vector with $\hat{\mathbf{e}}_{\mathbf{B1}}$ results in a vector perpendicular to the beam reference axis. To find $\hat{\mathbf{e}}_{\mathbf{B3}}$ the cross-product is calculated for both cross-sections and the average is taken as follows:

$$\hat{\mathbf{e}}_{\mathbf{B3}} = \left(\frac{1}{2}\right)\left[(\hat{\mathbf{e}}_{\mathbf{B1}} \times \hat{\mathbf{e}}_{\mathbf{L2}}) + (\hat{\mathbf{e}}_{\mathbf{B1}} \times \hat{\mathbf{e}}_{\mathbf{R2}})\right] \tag{5.10}$$

Once $\hat{\mathbf{e}}_{\mathbf{B1}}$ and $\hat{\mathbf{e}}_{\mathbf{B3}}$ have been calculated, $\hat{\mathbf{e}}_{\mathbf{B2}}$ is found by simply taking a cross-product of $\hat{\mathbf{e}}_{\mathbf{B1}}$ and $\hat{\mathbf{e}}_{\mathbf{B3}}$ as follows:

$$\hat{\mathbf{e}}_{\mathbf{B2}} = \hat{\mathbf{e}}_{\mathbf{B3}} \times \hat{\mathbf{e}}_{\mathbf{B1}} \tag{5.11}$$

The result is an orthogonal axis system for the beam element. A rotation matrix, $[L_{BG}]$, is constructed from the beam element axes by using each of the basis vectors as a matrix row. The subscript 'BG' indicates that the rotation matrix represents a rotation from the global coordinate system to the beam element coordinate system.

With the beam element axis system established, it is possible to determine the movement of the nodes relative to the beam reference axis, and therefore the elastic deformation. The small-angle rotations of the nodes relative to the beam reference axis are determined by calculating rotations from the beam reference axis to the

cross-section reference axis, represented by the rotation matrices $[L_{LB}]$ and $[L_{RB}]$ for the left and right ends respectively. These matrices are found by noting that a full rotation from global axis to cross-section axis can be done through a sequence of rotations, from global to beam and then beam to cross-section, as follows:

$$\begin{aligned} [L_{LG}] &= [L_{LB}][L_{BG}] \\ [L_{RG}] &= [L_{RB}][L_{BG}] \end{aligned} \tag{5.12}$$

and therefore,

$$\begin{aligned} [L_{LB}] &= [L_{LG}][L_{BG}]^T \\ [L_{RB}] &= [L_{RG}][L_{BG}]^T \end{aligned} \tag{5.13}$$

Thus, with the global cross-section rotation matrices and beam element rotation matrices already defined, the relative cross-section rotation matrices are easily calculated.

The relative nodal rotations are assumed to be small angles, so $[L_{LB}]$ and $[L_{RB}]$ are assumed to have the form of Equation (5.3). The Euler parameters for the rotation can then be backed out of the rotation matrix as follows:

$$\left.\begin{aligned} -(R_{23} - R_{32}) &= 2n_1 \sin\alpha \\ -(R_{31} - R_{13}) &= 2n_2 \sin\alpha \\ -(R_{12} - R_{21}) &= 2n_3 \sin\alpha \end{aligned}\right\} \quad \text{From Equation (5.3)}$$

$$4(\sin\alpha)^2(n_1^2+n_2^2+n_3^2) = (R_{23}-R_{32})^2+(R_{31}-R_{13})^2+(R_{12}-R_{21})^2$$

$$n_1^2+n_2^2+n_3^2 = 1 \quad \text{by definition}$$

$$\therefore \quad \sin\alpha = \frac{1}{2}\sqrt{(R_{23}-R_{32})^2+(R_{31}-R_{13})^2+(R_{12}-R_{21})^2}$$

$$\text{Let} \quad \lambda = \sqrt{(R_{23}-R_{32})^2+(R_{31}-R_{13})^2+(R_{12}-R_{21})^2}$$

Therefore,

$$\alpha = \sin^{-1}\left(\frac{\lambda}{2}\right) \tag{5.14}$$

$$\mathbf{n} = \begin{Bmatrix} n_1 \\ n_2 \\ n_3 \end{Bmatrix} = \begin{Bmatrix} \frac{-(R_{23}-R_{32})}{\lambda} \\ \frac{-(R_{31}-R_{13})}{\lambda} \\ \frac{-(R_{12}-R_{21})}{\lambda} \end{Bmatrix} \tag{5.15}$$

The small-angle rotation of the left node with respect to the beam element reference axis is therefore,

$$\theta_{\mathbf{L}} = \begin{Bmatrix} \theta_{Lx} \\ \theta_{Ly} \\ \theta_{Lz} \end{Bmatrix} = \alpha_L \mathbf{n}_L \tag{5.16}$$

where,

$$\alpha_L = \sin^{-1}\left(\frac{\lambda_L}{2}\right) \tag{5.17}$$

$$\mathbf{n}_L = \begin{Bmatrix} \frac{-(L_{LB,23}-L_{LB,32})}{\lambda_L} \\ \frac{-(L_{LB,31}-L_{LB,13})}{\lambda_L} \\ \frac{-(L_{LB,12}-L_{LB,21})}{\lambda_L} \end{Bmatrix} \tag{5.18}$$

$$\lambda_L = \sqrt{(L_{LB,23}-L_{LB,32})^2 + (L_{LB,31}-L_{LB,13})^2 + (L_{LB,12}-L_{LB,21})^2}$$

Similarly the small-angle rotation of the right node with respect to the beam element reference axis is as follows:

$$\theta_{\mathbf{R}} = \begin{Bmatrix} \theta_{Bx} \\ \theta_{By} \\ \theta_{Bz} \end{Bmatrix} = \alpha_B \mathbf{n}_B \tag{5.19}$$

where,

$$\alpha_R = \sin^{-1}\left(\frac{\lambda_R}{2}\right) \tag{5.20}$$

$$\mathbf{n}_R = \begin{Bmatrix} \frac{-(L_{RB,23}-L_{RB,32})}{\lambda_R} \\ \frac{-(L_{RB,31}-L_{RB,13})}{\lambda_R} \\ \frac{-(L_{RB,12}-L_{RB,21})}{\lambda_R} \end{Bmatrix} \tag{5.21}$$

$$\lambda_R = \sqrt{(L_{RB,23}-L_{RB,32})^2 + (L_{RB,31}-L_{RB,13})^2 + (L_{RB,12}-L_{RB,21})^2}$$

The displacements of the nodes relative to the beam element reference axis are found by calculating the displacement of the right node relative to the left node in the beam axis coordinate system. Therefore, at the left node the relative displacement is

$$\mathbf{u_L} = \begin{Bmatrix} u_{Lx} \\ u_{Ly} \\ u_{Lz} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \tag{5.22}$$

The displacement of the right node relative to the left node is found by transforming the known node locations from the global axis to the local beam axis and subtracting off the original length of the element. Because the beam element reference axis is assumed to pass through the left and right nodes, the y and z components of the displacement of the right end in the element reference axis system will be zero and there will be only an x component of displacement. The result is the extension of the element in the direction of the beam element reference axis, and therefore at the right node the relative displacement is

$$\mathbf{u_R} = [L_{BG}] \begin{Bmatrix} X_R - X_L \\ Y_R - Y_L \\ Z_R - Z_L \end{Bmatrix} - \begin{Bmatrix} L_{original} \\ 0 \\ 0 \end{Bmatrix} \tag{5.23}$$

Note that the global node coordinates are transformed to the local beam element axis system by premultiplying with the $[L_{BG}]$ matrix. This is because the $[L_{BG}]$ matrix gives the rotation from the global axes to the beam axes, and is therefore equivalent to a transformation matrix from global to local coordinate systems.

Combining equations (5.16), (5.19), (5.22), and (5.23), gives the local elastic deformation vector for a beam element,

$$\mathbf{u}_{elastic} = \begin{Bmatrix} \mathbf{u_L} \\ \theta_\mathbf{L} \\ \mathbf{u_R} \\ \theta_\mathbf{R} \end{Bmatrix} \tag{5.24}$$

The forces and moments in the beam element resulting from the elastic deformation are then found using the linear-elastic equation for beam elements as follows:

$$[k_E]\mathbf{u}_{elastic} = \mathbf{f}_{\mathbf{element}} \tag{5.25}$$

where $[k_E]$ is the local linear-elastic stiffness matrix for the element, which for an Euler-Bernoulli beam is given by equation (5.26). It should be noted that the linear-elastic stiffness matrix for any kind of beam element can be easily slotted in to this solution process. For example, see [17], page 79, for a beam element stiffness matrix accounting for transverse shear effects.

$$[k_E] = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\ & \frac{12EI_{zz}}{L^3} & 0 & 0 & 0 & \frac{6EI_{zz}}{L^2} & 0 & -\frac{12EI_{zz}}{L^3} & 0 & 0 & 0 & \frac{6EI_{zz}}{L^2} \\ & & \frac{12EI_{yy}}{L^3} & 0 & -\frac{6EI_{yy}}{L^2} & 0 & 0 & 0 & -\frac{12EI_{yy}}{L^3} & 0 & -\frac{6EI_{yy}}{L^2} & 0 \\ & & & \frac{GI_{xx}}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GI_{xx}}{L} & 0 & 0 \\ & & & & \frac{4EI_{yy}}{L} & 0 & 0 & 0 & \frac{6EI_{yy}}{L^2} & 0 & \frac{2EI_{yy}}{L} & 0 \\ & & & & & \frac{4EI_{zz}}{L} & 0 & -\frac{6EI_{zz}}{L^2} & 0 & 0 & 0 & \frac{2EI_{zz}}{L} \\ Symmetric & & & & & & \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & \frac{12EI_{zz}}{L^3} & 0 & 0 & 0 & -\frac{6EI_{zz}}{L^2} \\ & & & & & & & & \frac{12EI_{yy}}{L^3} & 0 & \frac{6EI_{yy}}{L^2} & 0 \\ & & & & & & & & & \frac{GI_{xx}}{L} & 0 & 0 \\ & & & & & & & & & & \frac{4EI_{yy}}{L} & 0 \\ & & & & & & & & & & & \frac{4EI_{zz}}{L} \end{bmatrix} \tag{5.26}$$

The local element forces and moments, $\mathbf{f_{element}}$, are transformed into components in the global coordinate system as follows:

$$\begin{Bmatrix} F_{xL} \\ F_{yL} \\ F_{zL} \\ M_{xL} \\ M_{yL} \\ M_{zL} \\ F_{xR} \\ F_{yR} \\ F_{zR} \\ M_{xR} \\ M_{yR} \\ M_{zR} \end{Bmatrix}_{gobal} = \begin{bmatrix} [L_{BG}]^T & [0] & [0] & [0] \\ [0] & [L_{BG}]^T & [0] & [0] \\ [0] & [0] & [L_{BG}]^T & [0] \\ [0] & [0] & [0] & [L_{BG}]^T \end{bmatrix} \begin{Bmatrix} f_{xL} \\ f_{yL} \\ f_{zL} \\ M_{xL} \\ M_{yL} \\ M_{zL} \\ f_{xR} \\ f_{yR} \\ f_{zR} \\ M_{xR} \\ M_{yR} \\ M_{zR} \end{Bmatrix}_{element} \tag{5.27}$$

The global forces and moments are then assembled into the full internal force vector for the structure, $\mathbf{F_{int}}$, and the unbalanced load can then be calculated in order to check for equilibrium, as discussed in Section 2.2. The process for calculating the internal member forces is also required for numerically generating the tangent stiffness matrix, which is done using a nodal displacement perturbation vector as described in Section 2.3.

5.2 Numeric Results

The solution method for the present work involving the numerically generated tangents stiffness matrix is implemented in a MATLAB code, which is then used to solve several examples of geometrically non-linear three-dimensional beam problems. The results are compared with results from the corresponding examples in the literature or using ANSYS commercial FEM software.

5.2.1 Elastica Cantilevered Beam

The elastica cantilevered beam is a typical large deflection planar Euler-Bernoulli problem. The problem consists of a long thin cantilevered beam subject to a tip load, P_y, and a tip moment, M_z, that increases with a given multiple of the tip load. Two different cases are analyzed: one where the tip load remains vertical (non-follower force) and one where the tip force changes direction and remains perpendicular to the beam element reference axis (follower force). For both cases, the problem is solved using the present MATLAB code and the tip force versus the vertical deflection of the beam tip is plotted. The same problems are solved by John Lau in [6] using a closed form solution to the beam differential equations. The results from Lau's paper are plotted for comparison.

Non-Follower Force

The applied tip force, P_y, remains in the global Y-direction throughout the solution process. The plotted results are shown in Figure 5.7.

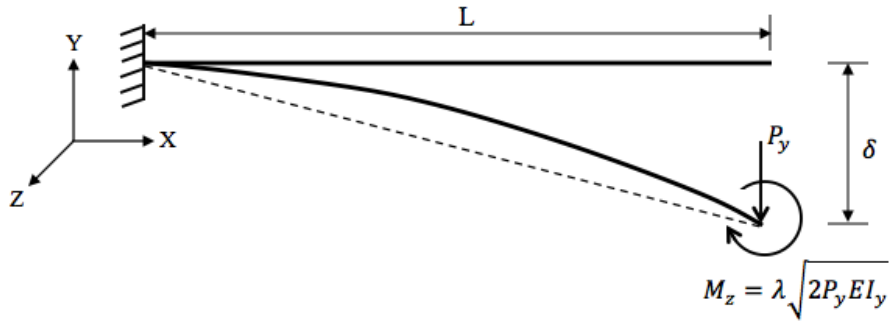


Figure 5.6: Elastica beam problem with non-follower applied tip force and tip moment.

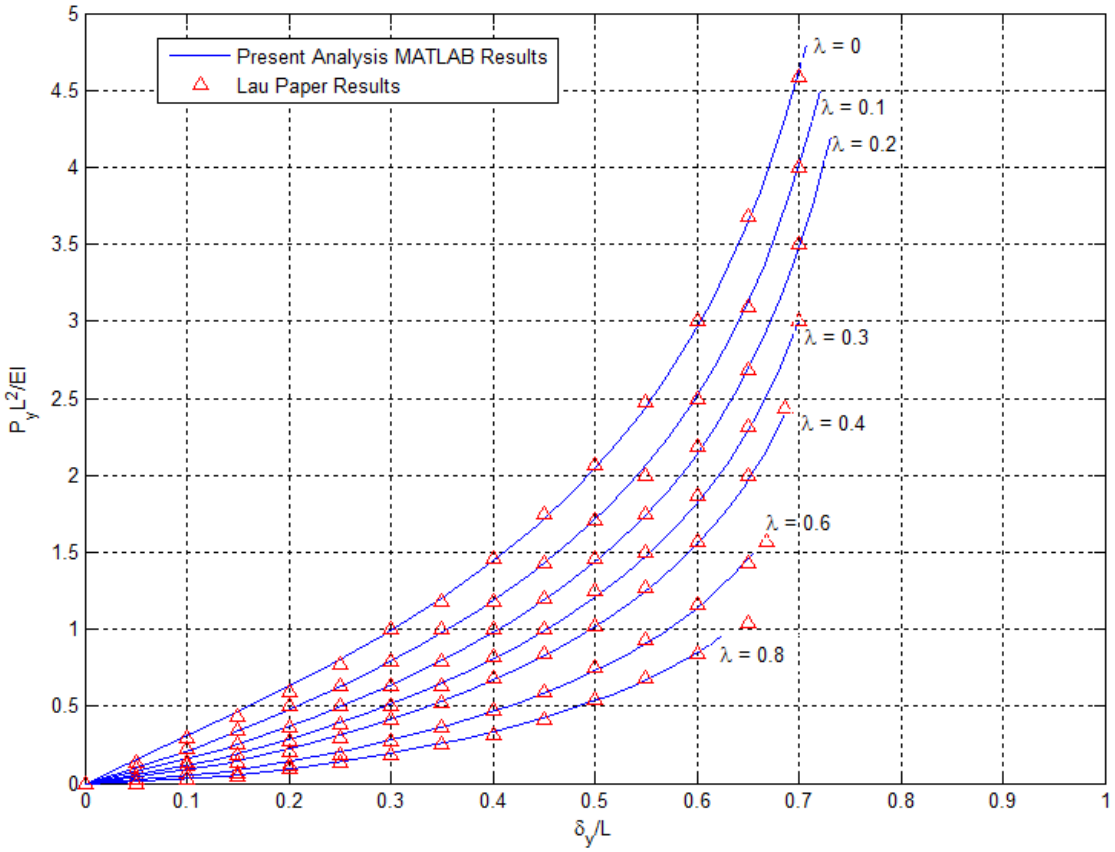


Figure 5.7: Force-deflection results for elastica beam problem with increasing non-follower applied tip force and tip moment.

Follower Force

The applied tip force, P_y, changes direction throughout the solution process to remain perpendicular to the beam element reference axis . The plotted results are shown in Figure 5.9.

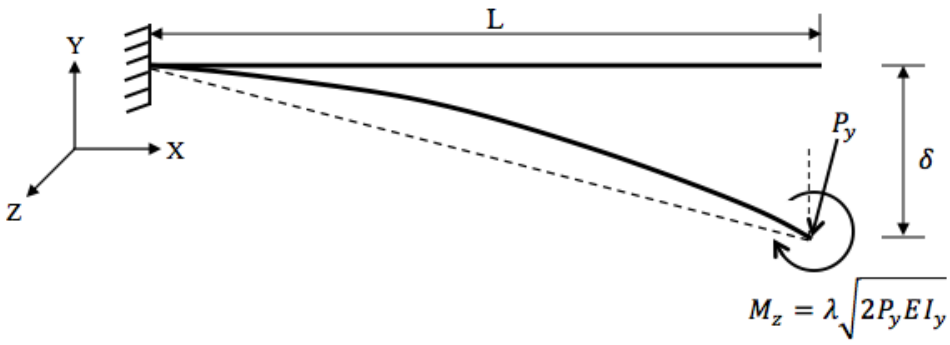


Figure 5.8: Elastica beam problem with follower applied tip force and tip moment.

It can be seen from the two elastica beam examples that the results from the present solution method using a numerically generated tangent stiffness matrix matches well with the direct closed from solution to the beam differential equations, showing that no accuracy is lost with the general numerical method.

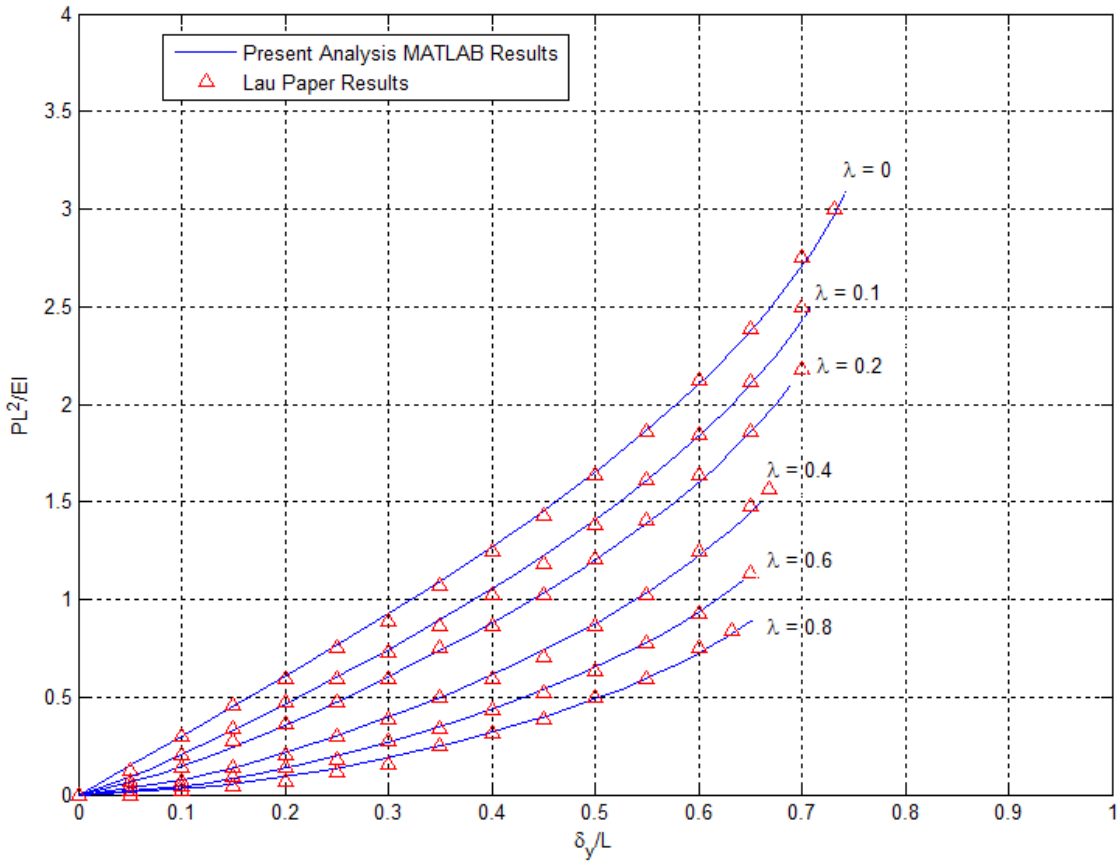


Figure 5.9: Force-deflection results for elastica beam problem with increasing follower applied tip force and tip moment.

5.2.2 Space Frame with Out-Of-Plane Loading

The elastica beam problem solved in the previous example, while a good example of large deflection, is only representative of in-plane loading and deflection. The following example shows how the present solution method behaves when there is both in-plane loading and out-of-plane loading and deflection. The structure shown in Figure 5.10 is subjected to an applied vertical tip load of $F_Y = 20\ lb$, as well as a simultaneously applied lateral tip load of $F_Z = 10\ lb$. The structure is solved using the MATLAB code for the present solution method, as well as using ANSYS. The plotted force-deflection results for both the vertical tip displacement and the lateral tip displacement are shown in Figure 5.11 and Figure 5.12.

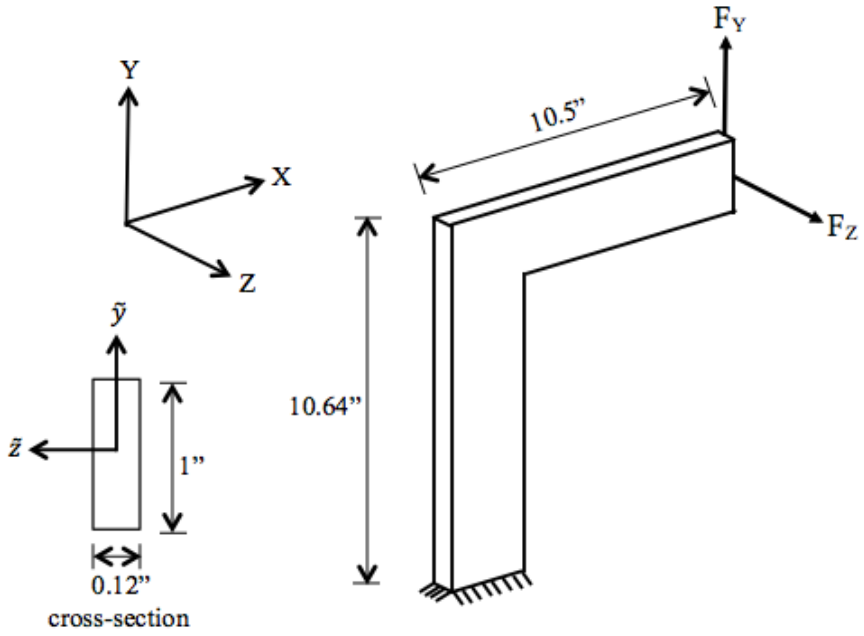


Figure 5.10: Space frame structure with a combination of in-plane and out-of-plane loading.

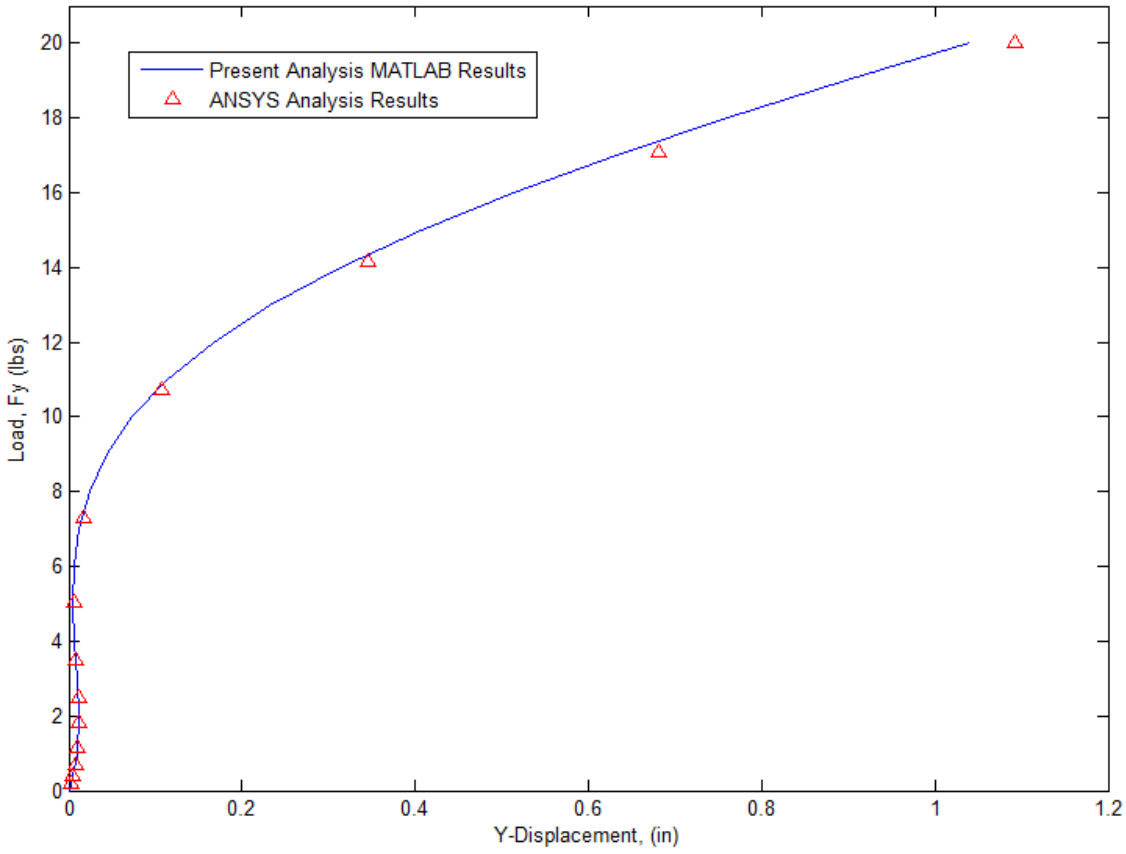


Figure 5.11: Force-deflection results showing vertical tip deflection for space frame structure with combined in-plane and out-of-plane loading.

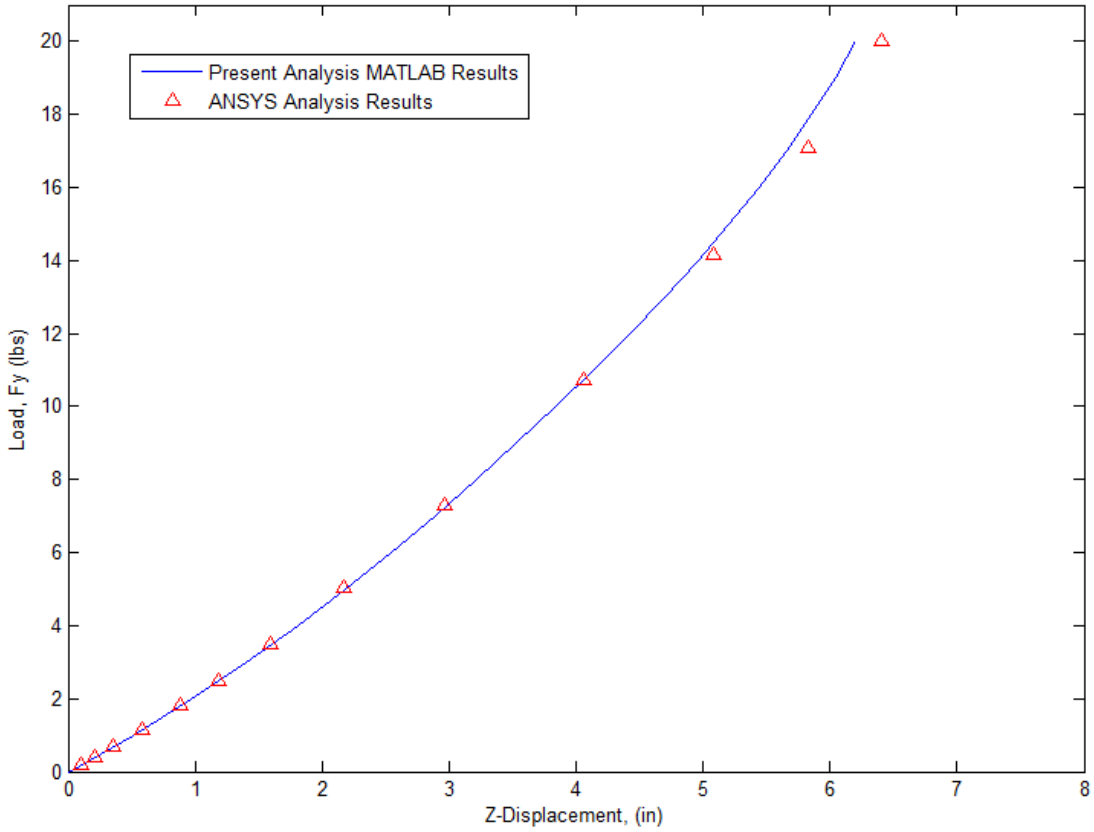


Figure 5.12: Force-deflection results showing lateral tip deflection for space frame structure with combined in-plane and out-of-plane loading.

It can be seen from the results that the present solution method using a numerically generated tangent stiffness matrix agrees well with the commercial finite element software results, showing that the general numerical method is capable of accurately capturing out-of-plane motion.

Chapter 6
CONCLUSION

To obtain an accurate representation of the large deflection of a structure, it is necessary to include geometric stiffness in the formulation of the problem. This leads to non-linearity and increased complexity that is not present in a simple linear-elastic analysis. Extensive work has already been done to derive the equations for geometrically non-linear problems. Solutions have been developed for many different kinds of structural elements, from simple truss elements to space frames, which have added complexity due to non-commutative finite rotations. Most work that has been done so far involves the use of typical finite element derivations, such as variational principals and minimization of potential energy, to derive analytic expressions that are solved using an iterative linearization technique. In order to obtain the required tangent or geometric stiffness matrices, tedious analytic calculations are required. In addition to the laboriousness of the analytic derivation, the complexity of the calculations limits analysis to relatively simple element types and structures. Some attempts to alleviate these inherent problems with analytic derivation have been made by introducing techniques involving numerical approximations of derivatives to estimate the specific derivatives required in the derivation of the tangent and geometric stiffness matrices. However, these techniques still lack ease and generality.

The present work introduces a method to numerically generate the tangent stiffness matrix for a structure in a way that is general and automatic, regardless of the specific structural element being analyzed. Since the method involves calculating the sensitivities of the element force to perturbations in the global degrees of freedom of the structure, all that is required is the knowledge of how to calculate the internal

element forces caused by the elastic deformations, which is typically already known from linear-elastic analysis. The sensitivities are numerically estimated using complex variable differentiation, which is straight-forward to implement and less sensitive to step-size than finite difference derivative approximation methods. In this way, a general and easily implemented solution method is developed that can be used to evaluate the large deflections and rotations of a geometrically non-linear structure made up of any structural elements.

Several structures were solved using the present method and compared with results from literature or from a commercial finite element analysis software. Static analysis was performed on three-dimensional truss structures, membrane structures, and beam structures, and final values for nodal location or force-displacement plots were compared. In all cases the results obtained from the method using a general numerically generated tangent stiffness matrix were the same as the results obtained from analytical methods, to within very small differences. This shows that no information is lost in generalizing the solution method and that the numerically generated tangent stiffness matrix, while using derivative approximations to obtain sensitivities, is just as accurate as the tedious analytically derived equations.

The greatest limitation with the present method, however, is the extensive computational cost. The numerical technique for calculating the sensitivities requires a loop in which each of the global degrees of freedom is individually perturbed and the internal element forces calculated. As the number of global degrees of freedom increases, the computational time required to execute this loop grows. Additionally, in the Newton-Rhapson method the tangent stiffness matrix is calculated at every iteration. Since the matrix is generated from a numerical approximation, it may take more iterations to converge on the equilibrium condition than with the analytically derived matrix. There are, however, several improvements that could be implemented to alleviate the computational time issue. For example, much work has been done to find more efficient iterative techniques, beyond the Newton-Rhapson method, that

would allow convergence to occur more quickly. This includes methods such as the displacement-control method or the arc-length method. Furthermore, since the technique for numerically calculating the tangent stiffness matrix involves perturbing each degree of freedom independently, it would be possible to use parallel computing to execute the loop thus speeding up the computation time. Overall, the present technique is attractive in a research setting where finite element problems of relatively small scale are to be solved, and non-linear finite element codes using different elements are to be produced quickly.

Future directions for the present work would include implementing some of the above mentioned strategies to speed up computation time, making the computational cost more comparable to analytic methods. Additionally, this method could be easily implemented for other non-linear structural problems, such as problems involving non-linear materials or non-linear forces. With the introduction of aerodynamic forcing, this method could be implemented in aeroelastic analysis for more general and automatic solution methods involving non-linear structures.

BIBLIOGRAPHY

[1] K. J. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, 1996.

[2] Klaus-Jurgen Bathe and Said Bolourchi. Large displacement analysis of three-dimensional beam structures. *International Journal for Numerical Methods in Engineering*, 14:961–986, 1979.

[3] Ted Belytschko, Wing Kam Liu, and Brian Moran. *Nonlinear Finite Elements for Continua and Structures*. Wiley, 1st edition, 2000.

[4] M. A. Crisfield. *Non-Linear Finite Element Analysis of Solids and Structures*. Wiley, 1996.

[5] Sanghaun Kim, Junghyun Ryu, and Maenghyo Cho. Numerically generated tangent stiffness matrices using the complex variable derivative method for nonlinear structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 200:403–413, 2011.

[6] John H. Lau. Large deflection of beams with combined loads. *Journal of the Engineering Mechanics Division*, 108(EM1):180–185, 1982.

[7] Yongjoo Lee and K. C. Park. Numerically generated tangent stiffness matrices for nonlinear structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 191:5833–5846, 2011.

[8] Robert Levy, Cheng-Wei Lin, Erez Gal, and Yeong-Bin Yang. Geometric stiffness of space frames using symbolic algebra. *International Journal of Structural Stability and Dynamics*, 3(3):335–353, 2003.

[9] Robert Levy and William R. Spillers. *Analysis of Geometrically Nonlinear Structures*. Kluwer Academic Publishers, 2nd edition, 2003.

[10] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso. The complex-step derivative approximation. *ACM Transactions on Mathematical Software*, 29(3):245–262, 2003.

[11] Joaquim R. R. A. Martins, Ilan M. Kroo, and Juan J. Alonso. An automated method for sensitivity analysis using complex variables. In *AIAA paper 2000-0689, 38th Aerospace Sciences Meeting*, 2000.

[12] J. L. Meek and S. Loganathan. Geometrically non-linear behaviour of space frame structures. *Computers & Structures*, 31(1):35–45, 1989.

[13] J. L. Meek and Hoon Swee Tan. Geometrically nonlinear analysis of space frames by an incremental iterative technique. *Computer Methods in Applied Mechanics and Engineering*, 47:261–282, 1984.

[14] G. Narayanan and C. S. Krishnamoorthy. An investigation of geometric non-linear formulations for 3d beam elements. *International Journal of Non-Linear Mechanics*, 25(6):643–662, 1990.

[15] Cenap Oran. Tangent stiffness in plane frames. *Journal of the Structural Division*, 99(ST6):973–985, 1973.

[16] Cenap Oran. Tangent stiffness in space frames. *Journal of the Structural Division*, 99(ST6):987–1001, 1973.

[17] J. S. Przemieniecki. *Theory of Matrix Structural Analysis*. Dover Publication, 2012.

[18] S. S. Rao. *The Finite Element Method in Engineering*. Pergamon Press, 2nd edition, 1989.

[19] W. R. Spillers. Geometric stiffness matrix for space frames. *Computers & Structures*, 36(1):29–37, 1990.

[20] W. Squire and G. Trapp. Using complex variables to estimate derivatives of real functions. *SIAM Rev.*, 40(1):110–112, 1998.

[21] Yeong-Bin Yang and Shyh-Rong Kuo. *Theory & Analysis of Nonlinear Framed Structures*. Prentice Hall, 1994.

[22] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Butterworth-Heinemann, sixth edition, 2005.

Appendix A

SOLUTION METHOD PSEUDOCODE

% Define and initialize all parameters and values required for analysis

Define: Total number of nodes and elements in structure.

Define: Geometric properties; eg. length, section area, moments of inertia, etc.

Define: Material properties; eg. Young's Modulus, shear modulus, etc.

Initialize all axis systems, as required (for beam element):

- Node rotation axis aligned with global axis.
- Left and right cross-section orientation relative to node rotation axis.
- Calculate beam reference axis system from initial cross-sections.

Define: Single point constraint boundary conditions for structure.

Define: Final total applied loading = F_{full}.

Define: Total number of load steps = NLS.

Calculate: $\Delta P = F_{full}/$(No. of load steps)

Define: Initial external and internal loads.

$\mathbf{P_{ext}} = \mathbf{0}$

$\mathbf{F_{int}} = \mathbf{0}$ % If structure is prestressed, then $F_{int} \neq \mathbf{0}$

Define: Convergence tolerance = tol.

Define: Complex variable differentiation step size = h

% Begin incremental solution method

for $i = 1 : NLS$ **do**

$\mathbf{P_{ext},}_i = \mathbf{P_{ext},}_{(i-1)} + \Delta P$

$\mathbf{P_{unbalanced}} = \mathbf{P_{ext},}_i - \mathbf{F_{int},}_{(i-1)}$

while $|\mathbf{P}_{\mathbf{unbalanced}}| \leq tol$ or $n \leq$ prescribed no. of iterations **do**

% Numerically generate tangent stiffness matrix.

for $dof = 1 :$ No. of nodes **do**

$\mathbf{u}_{perturbed} = \mathbf{0}$

$\mathbf{u}_{perturbed}(dof) = \mathbf{u}_{perturbed}(dof) + ih$ % Perturb particular dof.

$coordinates_{perturbed} = coordinates^{(n-1)} + \mathbf{u}_{perturbed}$

% If beam element, update nodal rotation matrices

$[L_{iG}]_{perturbed} = R[\Delta\alpha_{perturbed}][L_{iG}]^{(n-1)}$

for $el = 1 :$ No. of elements **do**

Calculate: elastic deformation for element = $\mathbf{u}_{\mathbf{elastic},perturbed}$

$\mathbf{f}_{\mathbf{element},perturbed} = [k_E]\mathbf{u}_{\mathbf{elastic},perturbed}$

Calculate: transformation matrix, $[T]$

$\mathbf{F}_{\mathbf{element},perturbed} = [T]^T\mathbf{f}_{\mathbf{element},perturbed}$

Assemble into full internal force vector, $\mathbf{F}_{\mathbf{int},perturbed}$

end for

$[K_T]_{dof^{th}\ column} = \frac{Imag[\mathbf{F}_{\mathbf{int}},perturbed]}{h}$

end for

$\mathbf{u}^n = [K_T]^{-1}\mathbf{P}_{\mathbf{unbalanced}}$

$coordinates^n = coordinates^{(n-1)} + \mathbf{u}^n$

% If beam element, update nodal rotation matrices

$[L_{iG}]^n = R[\Delta\alpha][L_{iG}]^{(n-1)}$

for $el = 1 :$ No. of elements **do**

Calculate: elastic deformation for element = $\mathbf{u}_{\mathbf{elastic}}$

$\mathbf{f}_{\mathbf{element}} = [k_E]\mathbf{u}_{\mathbf{elastic}}$

Calculate: transformation matrix, $[T]$

$\mathbf{F}_{\mathbf{element}} = [T]^T\mathbf{f}_{\mathbf{element}}$

Assemble into full internal force vector, $\mathbf{F}_{\mathbf{int},i}^{n}$

end for

Recalculate: $\mathbf{P_{unbalanced}} = \mathbf{P_{ext,i}} - \mathbf{F_{int,}}_{i}^{n}$

end while

$coordinates^{(n-1)} = coordinates^{n}$

$[L_{iG}]^{(n-1)} = [L_{iG}]^{n}$ % For beam element

$\mathbf{P_{ext,}}_{(i-1)} = \mathbf{P_{ext,}}_{i}$

$\mathbf{F_{int,}}_{(i-1)} = \mathbf{F_{int,}}_{i}^{n}$

end for

Appendix B

BEAM ELEMENT AXIS SYSTEM FORMULATION COMPARISON

When defining the beam element reference axis system, it is typical to define the $\hat{\mathbf{e}}_{\mathbf{B1}}$ basis vector as passing through the left and right nodes. The question then becomes how to define the $\hat{\mathbf{e}}_{\mathbf{B2}}$ and $\hat{\mathbf{e}}_{\mathbf{B3}}$ basis vectors based on the left and right end cross-section axes. For this thesis, an average of the two cross-sections was used. However, as mentioned on page 9 of Section 5, this is not the only solution. Alternate methods include using only the left cross-section axes to define the basis vectors, or using only the right cross-section axes. Changing between these different beam element axis definition methods is as simple as changing one line in the solution code. In particular, when the average is used (as in the present work), the $\hat{\mathbf{e}}_{\mathbf{B3}}$ basis vector is found using the following single equation:

$$\hat{\mathbf{e}}_{\mathbf{B3}} = \left(\frac{1}{2}\right)\left[(\hat{\mathbf{e}}_{\mathbf{B1}} \times \hat{\mathbf{e}}_{\mathbf{L2}}) + (\hat{\mathbf{e}}_{\mathbf{B1}} \times \hat{\mathbf{e}}_{\mathbf{R2}})\right] \tag{B.1}$$

If only the left cross-section axis system is used, the single equation above is changed to the following:

$$\hat{\mathbf{e}}_{\mathbf{B3}} = \hat{\mathbf{e}}_{\mathbf{B1}} \times \hat{\mathbf{e}}_{\mathbf{L2}} \tag{B.2}$$

Similarly, if only the right cross-section axis system is used, $\hat{\mathbf{e}}_{\mathbf{B3}}$ is found using the following equation:

$$\hat{\mathbf{e}}_{\mathbf{B3}} = \hat{\mathbf{e}}_{\mathbf{B1}} \times \hat{\mathbf{e}}_{\mathbf{R2}} \tag{B.3}$$

For all three methods, the $\hat{\mathbf{e}}_{\mathbf{B2}}$ basis vector is found the same way, using the following equation:

$$\hat{\mathbf{e}}_{\mathbf{B2}} = \hat{\mathbf{e}}_{\mathbf{B3}} \times \hat{\mathbf{e}}_{\mathbf{B1}} \tag{B.4}$$

It is of interest to see how the different methods for formulating the beam element axis system compare. This is done by using all three methods to find the beam element reference axes for the space frame structure problem that was analyzed in Section 5.2.2. Three different cases were analyzed: one where only two beam elements were used, one where 10 beam elements were used, and one where 20 beam elements were used. The force-deflection results for the three different cases, using the three different beam element reference axis definitions, are shown in Figures B.1, B.2, and B.3 below.

As can be seen from the results, for a structure that is made up of very few elements the results are not accurate when only one of the end cross-sections is used to formulate the beam element axis system. This is because with fewer elements, the elements themselves are longer and the nodes are further apart, meaning there is likely to be more bend and twist between the two nodes that needs to be taken into consideration. This is why the average of the two cross-sections yields the best results. This is further confirmed by noting that as the number of elements increases, and therefore the distance between the left and right ends decreases, the results using all three methods converge on the most accurate solution. This is because there is less distance between the two cross-sections for bend and twist to occur, and so the cross-section axes are closer to aligned. The conclusion to be drawn from this, is that

as long as enough elements are used, it is simple to swap out different formulations for the beam element axis system within the solution method code, thus emphasizing the generality and simplicity of the code itself.

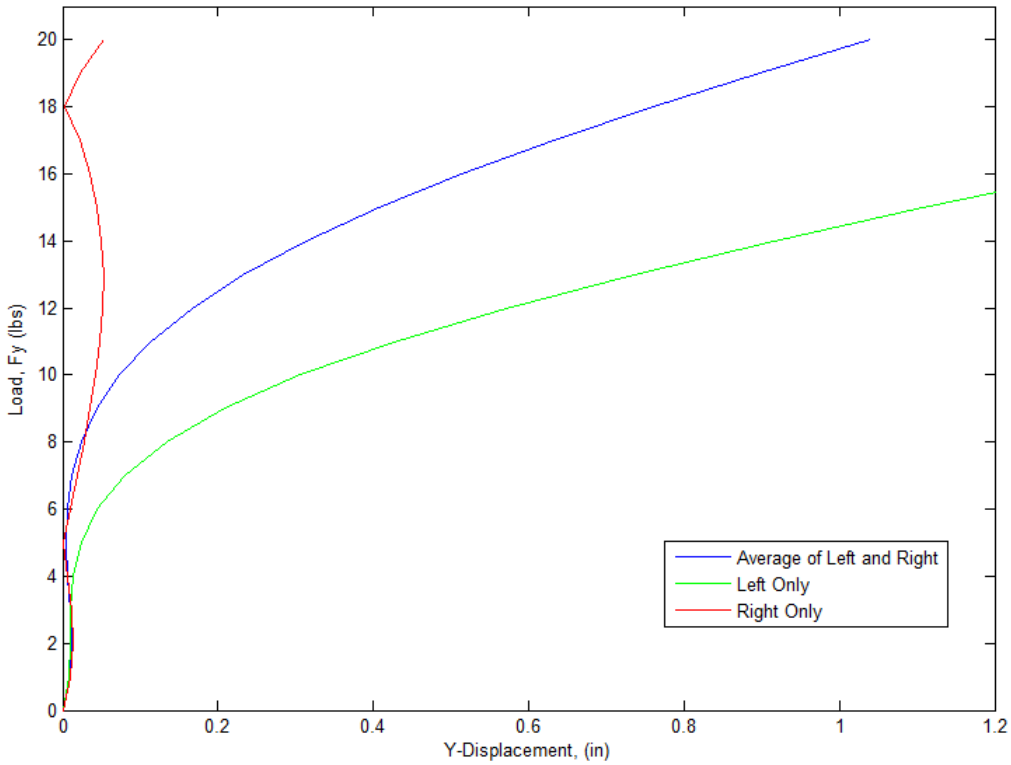


Figure B.1: Comparison of the different beam element axis system definition methods used to find the vertical tip deflection for the space frame structure made up of 2 elements

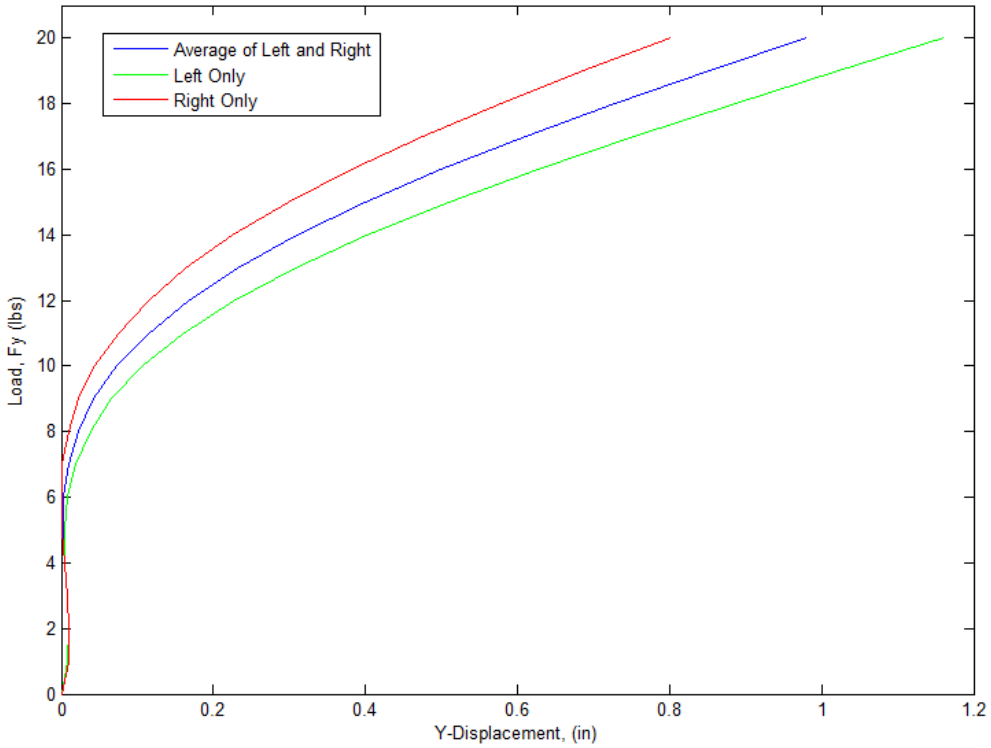


Figure B.2: Comparison of the different beam element axis system definition methods used to find the vertical tip deflection for the space frame structure made up of 10 elements

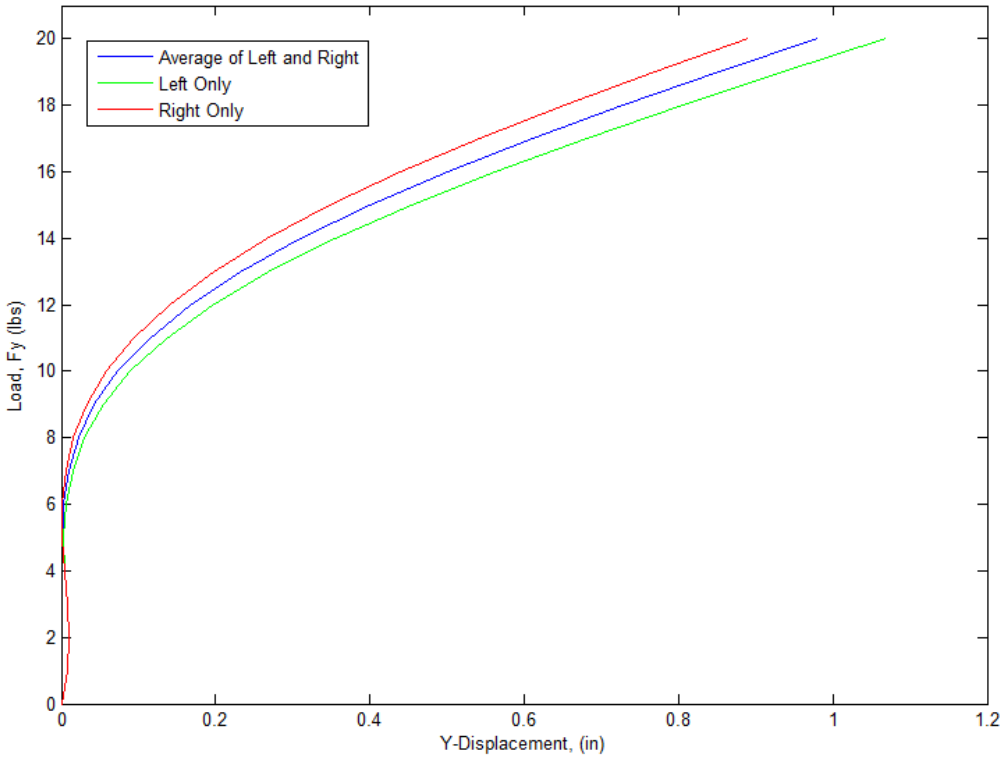


Figure B.3: Comparison of the different beam element axis system definition methods used to find the vertical tip deflection for the space frame structure made up of 20 elements