

Integration of Finite Element Method with
Multibody System Dynamics using Symbolic Computation

by

Yiteng Liang

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement for the degree of

Master of Applied Science

in

Systems Design Engineering

Waterloo, Ontario, Canada, 2017

© Yiteng Liang 2017

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

A general procedure integrating the finite element method with multibody system dynamics using symbolic computation is presented. It takes advantage of both the nodal formulation and the graph-theoretical approach, so it avoids the tedious modal analysis and can automatically reduce the number of system equations. The validity and expected performance are demonstrated mathematically and philosophically, which are tested by several general examples.

A flexible beam is created with a finite element formulation, which is derived according to the classical Bernoulli-Euler beam theory. A comprehensive procedure of deriving kinetic and kinematic equations for the flexible beam is provided, which starts from inspecting the diversity of shape functions to the generalization of inertia and elastic terms. The geometrical nonlinear formulation is revisited and specifically developed for the symbolic integration, and explicit expressions that are rare in existing publications for geometrical nonlinearity are derived for three-dimensional applications.

A detailed process is described for the transformation of the equations of motion of the finite-element flexible beam to a form required by the graph-theoretical approach. The beam model is implemented in MapleSim (a multi-physics simulation environment based on symbolic computation and graph theory) and made compatible with other multibody components in its library. Example problems are shown to validate the functionality of the implemented model. The dynamic stiffening effect due to geometrical nonlinearity is explicitly explained with its significance especially in fast and large overall-motion problems. Comparison results also show that the nodal approach shows great potential using the symbolic computation, which runs 70% faster and is more convenient to manipulate than the modal approach. The general procedure of the finite element formulation for the flexible beam can be followed to derive proper formulations of plate and solid elements if using appropriate displacement field assumptions, and the transformation to graph-theoretical equations is also similar.

Acknowledgments

First, I would like to thank my supervisor Dr. John McPhee for granting me such an excellent opportunity to further my academic career at the University of Waterloo. His valuable suggestions and generous support are the keys to making my research work meaningful.

I would like to thank my previous colleague Dr. Joydeep Banerjee who provided me with the critical technical support that helped me overcome countless obstacles and significantly enhanced my creative thinking ability.

I would like to thank Maplesoft, Toyota, and NSERC for their generous funding so that I could focus on my research work seamlessly. I would like to especially thank Dr. Chad Schmitke and Dr. Orang Vahid from Maplesoft for their kindness to offer significant technical support.

Finally, I would like to give my appreciation to my parents who keep inspiring me for the constant pursuit of knowledge and financially sponsoring my daily life.

Table of Contents

Author's Declaration	ii
Abstract	iii
Acknowledgements	iv
List of Figures	ix
List of Tables	x
1 Introduction	1
2 Literature Review	4
2.1 Approaches in Flexible Multibody System Dynamics	4
2.2 Finite Element Method	7
2.2.1 Beam Element Types	8
2.2.2 Nodal and Modal Approaches	10
2.3 Symbolic Computation	11
2.4 Symbolic Flexible Multibody Systems using Finite Element Method	13
3 Finite Element Modelling of Flexible Beams	15
3.1 Conventional Finite Element Formulation	17
3.1.1 Displacement Field and Element Shape Functions	17
3.1.2 Coordinate Systems	24
3.1.3 Position Vector to Arbitrary Point	26
3.1.4 Connectivity and Boundary Condition Matrices	29
3.1.5 Mass Matrix	31
3.1.6 Stiffness Matrix	34
3.1.6.1 Axial Deformation	34
3.1.6.2 Bending Deformation	36
3.1.6.3 Torsional Deformation	38

3.1.6.4	Superposition	39
3.1.7	Examination of the Linear Stiffness Matrix	40
3.1.8	Quadratic Velocity Force Vector	41
3.1.9	Generalized Gravitational Force Vector	43
3.1.10	Generalized External Force Vector	44
3.2	Geometrically Nonlinear Finite Element Formulation	45
3.2.1	Global Position Vector with Foreshortening	46
3.2.2	Modifications of the Mass Matrix and Force Vectors	50
3.2.3	Unchanged Stiffness Matrix	52
4	Finite Element Model in Symbolic Multibody System Dynamics	54
4.1	Governing Equations of Motion	55
4.2	Implementation into Library of Symbolic Multibody Systems	59
4.2.1	Finite-element Beam Body Element	60
4.2.2	Finite-element Beam Arm	62
4.3	Illustrative Example Problem	63
5	Example Problems	69
5.1	Cantilever Tapered Beam	69
5.2	Spin-Up Beam	72
5.3	Planar Slider-crank Mechanism with Flexible Crankshaft and Connecting Rod	78
5.4	Spatial Flexible Manipulator	81
5.5	Comparison of the current Finite-element Beam and the Analytical Beam Models in MapleSim	85
6	Conclusions and Future Work	88
6.1	Overview of the Current Development	88
6.2	Future Development and Extensions	89
	References	91
	A Appendix	97

A.1 Inertia Shape Integrals	97
A.2 Jacobian Matrix for the Illustrative Example Problem	101

List of Figures

3.1	DoFs of a 3D beam with 2 elements in body-fixed frame.	16
3.2	Deformation figures of a node. (a), (b), and (c) indicate displacements along x-, y-, and z-axis directions. (d), (e), and (f) indicate infinitesimal rotations about x-, y-, and z-axis directions.	16
3.3	A 3D beam element with nodal DoFs	18
3.4	Tilted cross-section	23
3.5	Finite-element coordinate systems for general problems defined by Shabana [1]	25
3.6	Selection of finite-element beam coordinate systems	26
3.7	Bending curvature	36
3.8	Twist angle (note R is the radius of the cross-section)	39
3.9	Shortened longitudinal displacement caused by bending deflection	46
3.10	Foreshortening of an infinitesimal curve	47
3.11	Accumulated shortenings of the elements	48
4.1	Multibody systems of interconnected rigid and flexible bodies (RB and FB refer to rigid body and flexible body). Note: the body frame of reference of a rigid body is located on the centre of mass.)	58
4.2	Pendulum flexible beam with tip point mass	63
4.3	Linear graph of the pendulum flexible beam with tip point mass	63
4.4	Axes of a revolute joint	66
5.1	Cantilever tapered cylindrical beam	71
5.2	Tip deflections of the tapered and uniform beams	72
5.3	Planar spin-up beam	72
5.4	Angular speed corresponding to the prescribed angular motion in Eq. (5.7)	73
5.5	Tip deflection of the spin-up beam	74
5.6	Tip deflection of the spin-up stiffer beam	75
5.7	Newly-defined prescribed angular motion	76
5.8	Tip in-plane deflection of the newly-defined spin-up beam	77

5.9	Tip out-of-plane deflection of the newly-defined spin-up beam	78
5.10	Slider-crank mechanism	79
5.11	Deformation of the midpoint of the connecting rod	80
5.12	Position of the slider block	80
5.13	Schematic of the spatial manipulator with a flexible link	81
5.14	Tip deformation magnitudes	82
5.15	Tip axial deformation about the x_b axis (Analytical and MapleSim FEM results converge)	83
5.16	Tip transverse deformation about the local y_b and z_b axis (all results converge)	83
5.17	Tip transverse deformation for different numbers of elements (MapleSim FEM)	84
5.18	Tip transverse deformation for different numbers of elements (ADAMS/Nastran)	85

List of Tables

5.1	CPU time corresponding to number of elements	78
-----	--	----

1 Introduction

With computer-aided engineering (CAE) software now widely used to solve computational mechanics problems, engineering design has become more accurate and efficient. Multibody system dynamics is one of the typical applications of computational mechanics. It can be found in various fields of applications, such as robotics, automotive, and aerospace. A considerable extension of the classic multibody systems dynamics is one wherein rigid and flexible bodies are interconnected with each other. For this important branch, vast effort has been spent on combining the continuum mechanics techniques with the equations describing the interconnected gross motion. Numerous approaches have been developed to model the flexibility, and all of them follow the same basic framework underlying each formulation. Due to the fact that a deformable body has infinite degrees of freedom, a physical discretization method should be used to reduce the infinite number to a finite number. One of the discretization approaches, which is widely used, is the finite element method (FEM). The finite element method shows strong ability to construct approximating deformation functions for flexible bodies with arbitrary shapes.

Realizing the integration of multibody system dynamics with the finite element method to predict the dynamics of flexible multibody systems has taken decades of work in engineering and mathematical fields, but there still remain several knotty problems to be tackled. Certain software packages have already commercialized the integration [2] and made great contributions to engineering development; however, its potential is highly restrained by numerical computation. Inconvenience in the modelling subroutines and long computational time are the most critical factors that result in low efficiency and high cost of engineering design and analysis. The advent of symbolic programming enables faster simulation code, via which the development of flexible multibody system dynamics has made predominant achievements [3, 4, 5] that show better performance over numerically-based simulations. Currently, the development using symbolic programming to integrate multibody system dynamics with the finite element method is at the initial stages, and publications in this area are scarce. Even though certain researchers have shown simple integration examples

using the Modelica symbolic programming language [6], the detailed process of the implementation is not provided.

The method described in this thesis is a general integration of the finite element method with multibody system dynamics, using symbolic computation technology (Maple/MapleSim [7]). The dynamic equations of a finite-element flexible beam based on a nodal formulation [1] are explicitly derived in symbolic form first, which are modified specifically to realize the integration. Newly-defined explicit expressions regarding the geometric stiffening non-linearity are derived from the classic Bernoulli-Euler beam theory, which is specific for the nodal formulation. In addition, the graph-theoretical approach [4] is used to construct the flexible multibody system equations. With this approach, highly-optimized equations of motion are generated. The development of the symbolic finite-element beam is considered as the first step; further extensions to plates and even arbitrarily-shaped deformable bodies can be created in the future by following the same routine. The integration is expected to become a powerful and efficient simulation tool in the future.

Six chapters are covered in this thesis. A brief literature review is presented in Chapter 2 focusing on the connection of flexible multibody system dynamics and the finite element method and the advantages of using symbolic computation as the programming basis. In Chapter 3, the detailed formulation of the finite-element beam element based on the Bernoulli-Euler theory is constructed using a nodal approach. The steps of derivation are general so that they can be used to derive the formulation of plate or shell elements when given the corresponding theoretical assumptions. A specific nodal formulation regarding the geometric stiffening nonlinearity is built following the classical beam formulation. After that, in Chapter 4, a general approach of integrating the multibody system dynamics with the finite element method is discussed. Following that, a recent approach based on graph theory, which is directly built on the symbolic computation technique, is briefly introduced. A general procedure showing how to tailor and transfer the finite-element equations to the graph-theoretical system equations is proposed. An illustrative example displaying the details of the proposed implementing procedure is provided at the end of the

chapter. In Chapter 5, the functionality of the implemented three-dimensional symbolic finite-element beam is tested. The results obtained are compared with those recorded in the literature or from models built with numerically-based commercial software packages. Accuracy, CPU time, and user-friendliness are some of the key factors concerned in the comparison of symbolically-based and numerically-based models. Several conclusions are drawn in Chapter 6, and future related work is identified.

2 Literature Review

2.1 Approaches in Flexible Multibody System Dynamics

There are two main approaches, which differ in the selection of coordinate systems, used to formulate the dynamic equations of motion of deformable bodies in flexible multibody system dynamics: the floating frame of reference method [8] and the absolute nodal coordinate method [9]. Even though both approaches can be used to construct equations of motion for the same multibody systems, the forms of the equations are entirely different from each other and possess respective merits and defects.

The floating frame of reference (FFR) method is currently the mainstream approach adopted by most commercial software packages dealing with flexible multibody dynamics problems. This method defines the configuration of the flexible body in terms of two sets of coordinates: reference coordinates and elastic coordinates. Reference coordinates define the location and orientation of a selected body-fixed frame. Elastic coordinates describe the body deformation with respect to the body-fixed frame. The potential energy defined using these local elastic coordinates yields a simple expression based on the assumption of small strains. The expression of the kinetic energy as well as the virtual work of the forces, however, are highly nonlinear since they are written in terms of the coupled sets of reference and elastic coordinates. As a result, the floating frame of reference method will generate a simple, and usually constant, stiffness matrix but a deformation-dependent mass matrix and other nonlinear inertia force terms. Despite the defects mentioned above, one of the main reasons for this method to be popular is because conventional discretization and coordinate-reduction methods can be easily applied [1]. This approach is widely used and has proved efficient in modelling flexible bodies with large overall motion while undergoing small deformation.

The absolute nodal coordinate formulation (ANCF) method, according to its name, uses coordinates with respect to the inertial frame directly to define all the configuration of the

deformable body. Its conceptual difference in the formulation makes it effective in modelling deformable bodies with large deformation since it uses no infinitesimal rotations as deformable coordinates. The difference also comes with the fact that due to the elimination of the coupling of reference and elastic coordinates, the mass matrix becomes constant, which no longer depends on the deformation of the body. However, using absolute coordinates will bring complexity into the formulation of continuum mechanics because the stiffness matrix turns out to be nonlinear even in the case of a linear elastic body with small deformation.

A number of test problems have been developed to compare these two methods [10, 11, 12]. Most of the cases show a good agreement between the two approaches, and some cases also indicate that a divergence of results emerge when the deformation or vibration of the deformable body is large to some extent [10]. Although the absolute nodal coordinate method has been used by many researchers to analyze flexible multibody problems, certain researchers still doubt the theoretical correctness of its formulation [13], which is one of the reasons to limit its broad application.

Having defined the coordinates of deformable bodies in multibody system dynamics, one needs to follow a specific approach to construct the system-level governing equations of motion. A conventional method that has been widely applied in commercial software packages uses all of the coordinates corresponding to all rigid and deformable bodies to assemble the entire system equations of motion, which is referred to as absolute coordinate formulation [14]. Notice that this method should be differentiated from the absolute nodal coordinate formulation used to define flexible bodies even though their names are similar. The system equations developed by the absolute coordinate formulation will be in the form of differential-algebraic equations (DAEs), and all joint reaction forces and moments will appear as Lagrange multipliers [15] inside the equations. This method is very straight forward and thus easy to be manipulated and implemented in computer systems. The main disadvantage of this method is that it always generates the whole number of DAEs including all of the reactions even though some of them are not desired, resulting in

unnecessary time cost.

With the advent of the graph-theoretical approach for flexible multibody systems, the way of assembling the system equations of motion becomes more intelligent. The graph-theoretical approach was initiated by Andrews and Kesavan [16] only to deal with unconstrained particles in a three-dimensional system. Later on, it was extended by McPhee [17] and pushed forward by other researchers (Andrews, Richard, Anderson [18]) to treat general two- and three-dimensional rigid multibody systems. Forces and moments are used as through variables and displacements and rotations are used as across variables. It was not until Shi [4] who used virtual work of the flexible body as a through variable that the graph-theoretical approach was used to analyze flexible multibody system dynamics. Following a specific spanning-tree selection scheme explored by McPhee [19], the graph-theoretical approach can produce fewer system equations by eliminating undesired joint coordinates to save simulation time.

Shi [20] specifically formulated the terminal equations of a flexible beam with geometrically nonlinearity based on the floating frame of reference method and used one of the classical variational methods [21], the Rayleigh-Ritz method, to discretize the deformation variables into generalized elastic coordinates over the entire length of the beam. The shape functions relating the generalized coordinates and the deformation variables are sets of polynomials with appropriate orders. This formulation is easy to obtain the inertia and elastic forces, which are extracted from the integration of the virtual work of a generic cross-section along the entire length. However, as pointed out by [22], the generalized coordinates from the Rayleigh-Ritz variation lack physical meanings as they only serve to reduce the infinite number of degrees of freedom to a calculable finite number. In addition, the classical variational methods unavoidably possess the disadvantage of difficulty of properly constructing the approximating functions for arbitrarily-shaped deformable bodies, which limits their usage when modelling more complex flexible components.

2.2 Finite Element Method

Having been developed for more than six decades, the finite element method is proved capable of constructing approximate functions for arbitrary domains, which has been applied in various fields of engineering application and analysis. Among one of the applications, its massive use in modelling flexible components in multibody system dynamics has been commercialized as different CAE software packages to solve complex engineering problems. The basic process of integrating the finite element method with multibody system dynamics is introduced in [1]. The finite element method can be viewed as a special case of the Rayleigh-Ritz method wherein the deformable body is divided into small regions called elements. The deformable body is separated into a number of finite elements that are interconnected at nodal points on their boundaries. The deformation variables are primarily the displacements of the selected nodal points. The displacement field of the element is adequately described by using interpolating polynomials on specific types of deformation, which are used to derive the shape functions associated with the element. Therefore, the undesired features of the Rayleigh-Ritz method pointed out in the previous section can be avoided by the finite element method using nodal coordinates to directly describe the displacements and rotations at the selected nodal points on the deformable body.

The types of classical finite elements that are used in the static and dynamic analysis of deformable bodies can generally be divided into two categories: isoparametric elements and non-isoparametric elements. As defined in [23], an isoparametric element has a shape function which can interpolate both the location and deformation of the generic points of the element. The isoparametric element uses only nodal displacements as coordinates and does not employ rotations or slopes as nodal coordinates. Examples of these elements are the rectangular, triangular, solid, and tetrahedral elements. The non-isoparametric element, on the other hand, uses infinitesimal rotations as nodal coordinates. These elements include beam, plate, and shell elements. The non-isoparametric element is widely used to develop computational procedures for the large rotation and deformation problems. Among these non-isoparametric elements, the beam element, which includes several de-

rived versions with various complexity of formulations, is the fundamental one that is used massively to model mechanisms with deformable links such as robotic arms, connecting rods, propellers, and so forth.

2.2.1 Beam Element Types

The beam element types are differentiated mainly by the different beam theories used to create the element formulation associated with the theoretical assumptions. Sometimes special nonlinear terms are added to the formulation of a selected type of element in order to capture specific physical behaviour.

The simplest classical beam theory is known as Bernoulli-Euler beam theory [24]. The key assumption under this theory is the plane sections of the beam initially perpendicular to the centroid axis will remain plane and perpendicular to the axis after deformation. This assumption necessarily implies that the shear strain and stress associated with the shear force is zero. As pointed out in [24], this formulation ignoring shear stresses can be used for load cases wherein stresses due to bending moment are the most significant; it is, however, acceptable only for long slender beams as errors incurred in displacements by ignoring shear effects are of the order of H/L^2 , where H is the depth of the beam and L is the length. Another critical assumption of the Bernoulli-Euler beam theory states that accurate description of the beam deformation will only be guaranteed under small deformation with small strains. Based on empirical data, it is estimated that the Bernoulli-Euler beam theory allows the maximum deformation ratio of the tip deformation over the beam length to be 10%, and errors will become intolerable if beyond this limit. A simplified version of the Bernoulli-Euler beam element used in certain applications assumes that the plane sections remain in plane but perpendicular to the undeformed centroid axis, in which the rotary inertia of the cross-sections is neglected. This simplified version further reduces the complexity of the mass and stiffness matrices so that it saves time when used to model thin beams of which the rotary inertia of the cross section is negligible.

In cases where the beam is relatively short or deep, shear effects can not be negligible. An

analytical beam theory with shear effects included is the Timoshenko beam theory [25]. The assumption that the plane section remains plane and perpendicular to the neutral axis is relaxed to allow plane sections to undergo a shear strain. The rotation of the cross-section described by the Bernoulli-Euler beam theory is reduced by an amount of the shear strain. Initiated from the Timoshenko beam theory, several derivatives with improvements in different aspects have been developed by numerous researchers [26, 27].

Regarding flexible multibody dynamics, the Bernoulli-Euler beam element is usually the first choice considering that the deformable components in multibody systems are often slender links. However, certain cases involving large and fast rigid-body motion of the flexible body lead to the inexact modelling of the deformation by using the conventional Bernoulli-Euler beam formulation. Theoretically, the assumption of small strains omits higher order terms in the full strain energy expression [28]. This hypothesis decouples different types of deformation (bending, axial, and torsion) in reality, and the uncoupled bending and axial deformation result in the failure of modelling the dynamic stiffening effect. This effect, usually referred to as geometric elastic nonlinearity [29], is critical in flexible multibody dynamics with the deformable components undergoing large and fast rigid-body motions, which can also be described as the foreshortening effect [30].

A benchmark problem concerning this effect was initiated by Kane, Ryan, and Banerjee [30] who developed a planar spin-up flexible beam to analyze the significance of the geometric elastic nonlinearity. A number of mathematical modifications based on the original Bernoulli-Euler beam formulation have been developed to cover geometric elastic nonlinearity. Bakr and Shabana [31] expanded the original formulation by adding the classical geometric stiffness matrix [32] to the traditionally constant stiffness matrix. Afterwards, Mayo et al. [29] improved the formulation by developing a new geometric stiffness matrix and a vector of nonlinear elastic forces with more general formulations than that of Bakr and Shabana. However, all these methods suffer from computational inefficiency due to the necessity to include many axial deformation modes to approximate the displacement field. An alternative formulation derived later on by Mayo and Dominguez [33] manu-

ally added a foreshortening term in the expression of the axial displacement field. This approach distinguished the shortening-induced and strain-induced displacements, and will yield the traditionally constant stiffness matrix. The nonlinearity was moved from the elastic forces to inertia, external, and reactive forces. This approach is proved more computationally efficient [34] and widely used in numerous examples [35, 36] to approximate the foreshortening effect.

2.2.2 Nodal and Modal Approaches

The nodal approach, also referred to as transient analysis in some commercial software packages, generates the system equations of motion of the flexible multibody dynamics which contain the entire set of the elastic nodal coordinates of the flexible body. The results provide the exact values of the nodal displacements after solving the system equations. It may require a significant number of nodal coordinates to obtain an adequate representation of large-scale nonlinear mechanical systems. If using the nodal approach, the large computational cost could be a serious issue. For this reason, it is necessary to reduce the number of coordinates to obtain a solution with a reasonable amount of computer time. The modal transformation approach has been used extensively in structural dynamics [37, 38] to reduce the problem dimensionality.

Most researchers as well as commercial software packages utilize the modal approach as the primary selection to solve flexible multibody dynamics problems, but the modal approach has its disadvantages. It unavoidably requires the user to spend time on mode shape analysis [39] or modal analysis. References about modal analysis always state that the mode shapes should be selected in a way that a good approximation for the displaced shape can be obtained, but none of them has given a gold-standard selection method for general problems. Therefore, efficient selection of the appropriate mode shapes often requires sufficient empirical knowledge. Even so, when faced with a new problem, a number of trials of selections are still necessary to refine the approximation of the displaced shape before reaching convergence. Additionally, for optimization problems where the geometrical and material parameters of the flexible body are frequently changed, using the modal approach

seems to be time-consuming due to the fact that changing parameters always results in a new set of mode shapes to be numerically calculated and manually selected.

With the advent of symbolic computation whose features are introduced in the following section, the integration of finite element method with flexible multibody dynamics using the nodal approach shows great potential on several aspects. It tackles the problem of redundant creation of new flexible bodies when the parameters are changed. It eliminates numerical errors associated with the mode-shape approximation. As the modal approach often requires a large number of elements to obtain a set of exact mode shapes, the nodal approach only needs a small number since it fully utilizes the shape function which directly describes the deformation of the material points. These facts will be presented in this thesis.

2.3 Symbolic Computation

Since the 1970s, many multibody systems programs based on numerical programming have been developed to assist a wide range of applications in a given modelling and analysis field. Although each of them was said to be a general purpose multibody software package, they are all restricted by numerical programming on both modelling and analysis. It was emphasized by Samin and Fiset [\[3\]](#) of the distinction between modelling and analysis of multibody systems. The modelling phase is a process which, once and for all, derives the entire system equations of motion describing a multibody application for a given set of system parameters and generalized coordinates. This phase can be based on analytical, numerical, or symbolical approaches. Once the system equations have been set up, they can repeatedly be used by the subsequent process, which is called the analysis phase, to generate desired results. The conventional way of implementing these two phases in a computer program is via a purely numerical manner. Since the analysis phase generally involves time integration of the complicated multibody systems equations where a closed-form solution can hardly be obtained, numerical solvers become mandatory for this step. However, the modelling phase does not necessarily rely on the numerical approaches.

Essentially based on symbolic computation, symbolic programming subtly avoids common shortcomings that normally belong to numerical computation. As described by Samin and Fisette, symbolic computation takes advantage of both the numerical and manual techniques. The manual derivation of system equations, using a pen or some general purpose programs, was a natural way before the computer era. It always plays an enlightening role for one to learn the process of deriving system equations. Although the manual generation is limited to very simple systems and always subject to human error, it has the strength to yield optimized mathematical equations in terms of arithmetical and trigonometrical expressions. If based on numerical generation, the modelling phase requires numerous calls of subroutines. Each subroutine is used to build some specific parts of the equations or specific vector/tensor operations. However, the multiple calls sometimes become annoying when parameters of the system should frequently be changed. Many excessive zero-value quantities, for large systems, are taken into account and treated equally with the non-zero ones. A manual generation would eliminate all the unnecessary arithmetical operations beforehand so that an optimal form of system questions can be generated. Symbolic programs manipulate only arithmetical operators ($+$, $-$, $*$, $/$, etc.) as well as strings of alphanumeric characters ($\cos \theta$, d_{ij} , π , etc.) to produce the equations in an analytical form. From the modelling point of view, the symbolic generation is performed only once, and it is not necessary to assign values to the symbolic parameters throughout the calculation unless and until a numeric approximation is needed. For instance, $\pi/4$ can remain as $\pi/4$ throughout the modelling phase, resulting in an exact result by eliminating accumulated round-off errors.

Also, the symbolic simplification of expressions enables higher efficiency and makes it easy to sweep parameters, optimize values, and study the behaviour of the system. If only considering the simulation time, comparison tests [40] shows that a symbolic multibody model performs a time simulation about five to ten times faster than a purely numerical one with all other things being equal. Many other features of the symbolic computation in different applications can be found in [41].

2.4 Symbolic Flexible Multibody Systems using Finite Element Method

Although symbolic computation has already become a mature technology used in flexible multibody simulation programs [42], the integration with the finite element method is still rare.

There are two ways preferred by symbolic multibody programs for modelling the flexible body: a finite segment approach in which the beam is replaced by a sequence of rigid segments interconnected by equivalent springs [43, 44], or an assumed displacement field method using polynomials up to different orders (same as the Rayleigh-Ritz variation used in [45, 46]) to directly describe the displacement field corresponding to different types of deformation. Both methods agree nicely with the symbolic technique because they allow symbolic expressions of the deformation equations associated with the deformable bodies to be directly generated. However, neither of the two methods can be easily extended to model arbitrarily-shaped deformable bodies. Another applicable method uses the input of a set of assumed mode shapes resulting from a preprocessed modal analysis [47], which follows the same procedure as employed in numerical programs and cannot avoid the tedious mode-shape selection step.

Ferretti [48] modelled a three-dimensional finite-element thin beam with the classical Bernoulli-Euler beam formulation using Dymola based on the Modelica programming language [49]. It displays the advantage of using symbolic programming to build up the finite element equations. It is known that crafting finite-element models involves encoding a number of large and complex matrices, such as the mass and stiffness matrices. For instance, developers can use symbolic programming to just correctly write out the symbolic expression of the position vector of an arbitrary point on the flexible body and let it integrate symbolically with other symbols representing the density and cross-sectional area to obtain the whole inertia term. Even though using the nodal approach, the comparison results show that the symbolic finite-element models created in Dymola are 100 times faster than the same model simulated in ANSYS. However, the content focuses mainly on the

finite-element matrices and generalized forces referred from [1] while the actual process of the implementation of the finite element method into the system is ambiguously described.

As mentioned in the preceding section, Shi [4] used the graph-theoretical approach and the Rayleigh-Ritz method to create a library consisting of multibody system components and an analytical flexible beam, which is based on symbolic computation as well. The library has now been modified and integrated into MapleSim [7]. Shi [50] also pointed out the possibility of using the finite element method for the discretization of the flexible body. With the benefit of the graph-theoretical approach, this technique can be used to construct even more complex flexible multibody systems with the finite element method.

3 Finite Element Modelling of Flexible Beams

The theory of Bernoulli-Euler beam is applied to obtain the inertia and stiffness terms of the flexible beam. In the beginning, it is necessary to highlight what is meant by the classical Bernoulli-Euler beam. The deformation patterns of the Bernoulli-Euler beam stays in between those of the slender prismatic bar and Timoshenko beam. Three principal assumptions are made for the Bernoulli-Euler beam according to Astley [24]. First, the classical Bernoulli-Euler beam model assumes that the internal energy of the beam is entirely due to bending strains and stresses, so it neglects transverse shear deformations. In this case, the cross-sections remain plane and will always be perpendicular to the reference axis during deformation. Second, for small deformations, axial deformation, bending, and twist can be decoupled and looked at separately. The last one is that the twist is regarded independent, and the Saint-Venant method [22] is used to establish the torsional stiffness.

The Bernoulli-Euler beam can be adequately modelled by using the finite element method. The finite element method can be viewed as a special case of discretization where the deformable body is divided into small regions called elements. The deformable body is separated by imaginary lines into a number of finite elements that are assumed to be interconnected at nodal points on their boundaries. The displacements of the selected nodal points are the basic unknowns of the problem, which are also referred to as the deformation degrees of freedom (DoFs).

Generally, a spatial finite beam element, which is used the most widely, has one node located at each tip-end. Sometimes, however, a beam element can have more than two nodes to obtain greater accuracy with higher computational cost. For any three-dimensional beam elements with an arbitrary number of nodes, there are always six deformation degrees of freedom (three displacements and three infinitesimal rotations) engaged with each node in a Cartesian coordinate system. A beam distributed by two elements is shown in Figure 3.1, and the body-fixed frame of reference is located at the left tip-end with the x-axis pointing along the neutral axis to the right. At Node 1, there are three displacement

vectors denoted by u_1, u_2, u_3 and three rotation vectors denoted by u_4, u_5, u_6 respectively. It is identical for the other nodes, and the same routine will apply if more beam elements are used.

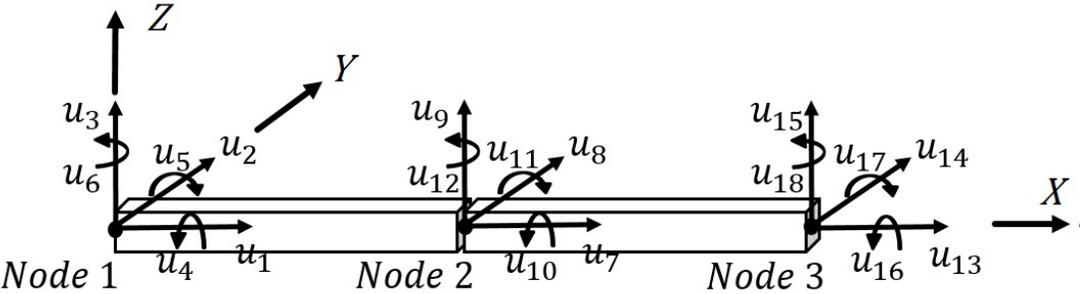


Figure 3.1: DoFs of a 3D beam with 2 elements in body-fixed frame.

Classical deformations of an arbitrary node are shown in Figure 3.2, and they are corresponding to axial deformation, torsion, two lateral bending translations and rotations. The deformation degrees of freedom of one node can be clearly observed from the six deformation figures.

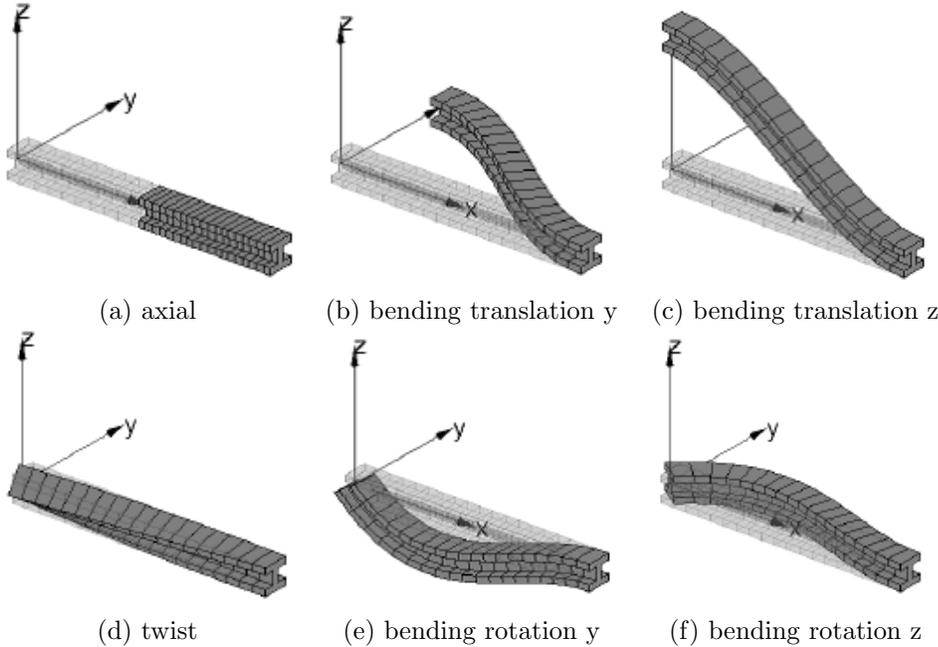


Figure 3.2: Deformation figures of a node. (a), (b), and (c) indicate displacements along x -, y -, and z -axis directions. (d), (e), and (f) indicate infinitesimal rotations about x -, y -, and z -axis directions.

Summing up all of the statements above, some key assumptions can be generalized when using the conventional Bernoulli-Euler beam element:

1. The beam is applicable in any arbitrary overall motion with small deformation.
2. There is no coupling between axial deformation, lateral bendings, and twist.
3. Shear due to bending or warping due to torsion are ignored.

The classical Bernoulli-Euler beam can be modelled by using the conventional finite element method formulation. However, certain cases where the overall motion of the beam is not only large but also fast will result in a physical effect called dynamic stiffening. This phenomenon happens even when the motion is slow if the beam is not stiff enough. The physical behaviour of dynamic stiffening is that a beam will become harder to bend during spinning since the axial deformation caused by centripetal force is counteracting the bending. This effect can not be correctly modelled by using the conventional formulation that completely decouples bending and axial deformation, so it brings the need to adopt a modified formulation. Several nonlinear formulations can be used to capture the dynamic stiffening effect, of which the most efficient one is to add an additional foreshortening term in the displacement field [33]. The foreshortening term can be interpreted as that bending can cause axial shortening due to the inextensibility of the beam. By properly adding the foreshortening term, the beam becomes geometrically nonlinear which leads to generating highly nonlinear inertia terms. The conventional finite element and the geometrically nonlinear formulations will be presented in Section 3.1 and 3.2, respectively.

3.1 Conventional Finite Element Formulation

3.1.1 Displacement Field and Element Shape Functions

In the standard finite element formulation for non-isoparametric elements, nodal coordinates at each selected node are used to describe displacements and infinitesimal rotations. Between these selected points the displacement field within the element can be adequately

described by using interpolating polynomials. Shape functions are derived from these polynomials after several transformation steps.

The concept of interpolation is to select a function $f(x)$ from a given class of functions in such a way that the function passes through the given data points, which in this case are the nodal points. Therefore, exact and interpolated curves match at the endpoints (the nodes) but may differ elsewhere.

Classical procedures of the derivation of the shape functions can be found in many finite element textbooks and papers, which are derived from Lagrange polynomials with varying orders to describe axial deformation, twist, and bending [51]. However, the classical way makes it hard to comprehend the meaning of orders of deformations for new finite element users. Another method provided by Shabana [1] derives the identical shape functions as used by Lagrange polynomials, but it follows a more straightforward path by deriving them from Taylor polynomials.

In Shabana [1], a simple example of deriving the shape functions for a two-dimensional beam is given. Let us consider a more complex one, and take the three-dimensional beam element shown in Figure 3.3 for example.

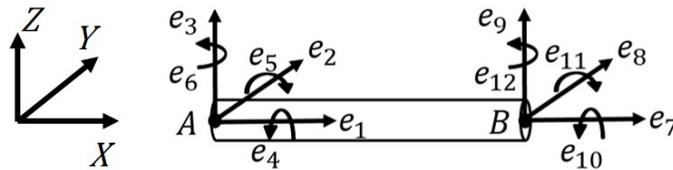


Figure 3.3: A 3D beam element with nodal DoFs

In this beam element, there are two nodal points A and B each with six degrees of freedom: three translational coordinates in X , Y , and Z directions (XYZ is the body-fixed frame), two infinitesimal rotational coordinates describing the slope at the nodal point, and the other one defining the twist angle. The total number of coordinates for this element is 12, denoted as e_1, e_2, \dots, e_{12} . One may describe the displacement field \mathbf{w} in three-dimensional

space plus torsion within the element by the following deformations described by Taylor polynomials:

$$\left. \begin{aligned} w_1 = u &= a_0 + a_1x \\ w_2 = v_y &= a_2 + a_3x + a_4(x)^2 + a_5(x)^3 \\ w_3 = v_z &= a_6 + a_7x + a_8(x)^2 + a_9(x)^3 \\ w_4 = \phi &= a_{10} + a_{11}x \end{aligned} \right\} \quad (3.1)$$

where u and ϕ are the axial deformation and twist about X direction, respectively, and v_y and v_z are the transverse deformation in Y and Z directions. A linear interpolation is used to describe the axial deformation and twist, while a cubic function is used to describe the transverse deformation. x is the distance from the nodal point A to an arbitrary point on the X -axis. The coefficients a_0, a_1, \dots, a_{11} are to be determined by applying boundary conditions.

Eq. (3.1) can be written in a matrix form

$$\mathbf{w} = \mathbf{X}\mathbf{a} \quad (3.2)$$

where \mathbf{X} is the matrix

$$\mathbf{X} = \begin{bmatrix} 1 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & x & x^2 & x^3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & x & x^2 & x^3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x \end{bmatrix} \quad (3.3)$$

and \mathbf{a} is the time-dependent vector whose components are given by

$$\mathbf{a} = \left[a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_7 \ a_8 \ a_9 \ a_{10} \ a_{11} \right]^{\mathbf{T}} \quad (3.4)$$

It is important to notice that \mathbf{a} is a constant vector in static analysis, while it is a function of time in dynamic analysis.

If the length of the beam element is l , then the nodal point A is located at $x = 0$, and

nodal point B is located at $x = l$. We can also write all of the coordinates in a vector form as

$$\mathbf{e} = \left[e_1 \ e_2 \ e_3 \ e_4 \ e_5 \ e_6 \ e_7 \ e_8 \ e_9 \ e_{10} \ e_{11} \ e_{12} \right]^T \quad (3.5)$$

where $e_1, e_2, e_3, e_7, e_8, e_9$ are the translational coordinates as shown in Figure 3.3, e_5, e_6, e_9, e_{11} are the slopes (rotational coordinates) at the nodal points, and e_4, e_{10} are the torsional coordinates at the nodal points A and B , respectively. Notice that \mathbf{e} is time-variant in dynamic analysis. To determine the coefficients $a_i, i = 0, 1..11$ in Eq. (3.1), we impose the following boundary conditions:

$$\left. \begin{aligned} w_1(0) = e_1, w_2(0) = e_2, w_3(0) = e_3, \phi(0) = e_4, w'_2(0) = e_5, w'_3(0) = e_6 \\ w_1(l) = e_7, w_2(l) = e_8, w_3(l) = e_9, \phi(l) = e_{10}, w'_2(l) = e_{11}, w'_3(l) = e_{12} \end{aligned} \right\} \quad (3.6)$$

where $(')$ denotes partial differentiation with respect to the spatial coordinate x .

Using the representation of Eq. (3.2) and the boundary conditions of Eq. (3.6), one can write

$$\mathbf{e} = \bar{\mathbf{X}}\mathbf{a} \quad (3.7)$$

where the 12×12 matrix $\bar{\mathbf{X}}$ is defined as

$$\bar{\mathbf{X}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & l & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & l & l^2 & l^3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & l & l^2 & l^3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 & 2l & 3l^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2l & 3l^2 & 0 & 0 \end{bmatrix} \quad (3.8)$$

Therefore, the vector of the coefficients \mathbf{a} can be determined in terms of the nodal coordinates \mathbf{e} as

$$\mathbf{a} = \bar{\mathbf{X}}^{-1} \mathbf{e} \quad (3.9)$$

where $\bar{\mathbf{X}}^{-1}$ is the inverse of $\bar{\mathbf{X}}$. Using Eqs. (3.2) and (3.9), the displacement field of the beam element can be written in terms of the nodal coordinates as $\mathbf{w} = \mathbf{X} \bar{\mathbf{X}}^{-1} \mathbf{e}$, or

$$\mathbf{w} = \widehat{\mathbf{S}} \mathbf{e} \quad (3.10)$$

where $\widehat{\mathbf{S}}$ is widely known as the element shape function matrix and defined as $\mathbf{X} \bar{\mathbf{X}}^{-1}$. Using

Eq. (3.8), the space-dependent shape function matrix of the beam element is derived as

$$\widehat{\mathbf{S}} = \begin{bmatrix} 1-\xi & 0 & 0 & 0 \\ 0 & 1 - 3(\xi)^2 + 2(\xi)^3 & 0 & 0 \\ 0 & 0 & 1 - 3(\xi)^2 + 2(\xi)^3 & 0 \\ 0 & 0 & 0 & 1-\xi \\ 0 & 0 & [-\xi + 2(\xi)^2 - (\xi)^3]l & 0 \\ 0 & [\xi - 2(\xi)^2 + (\xi)^3]l & 0 & 0 \\ \xi & 0 & 0 & 0 \\ 0 & 3(\xi)^2 - 2(\xi)^3 & 0 & 0 \\ 0 & 0 & 3(\xi)^2 - 2(\xi)^3 & 0 \\ 0 & 0 & 0 & \xi \\ 0 & 0 & [(\xi)^2 - (\xi)^3]l & 0 \\ 0 & [-(\xi)^2 + (\xi)^3]l & 0 & 0 \end{bmatrix}^T \quad (3.11)$$

in which $\xi = x/l$. Note that at $x = 0$, the elements of the shape function matrix associated with the coordinates of the second node are equal to zero, while at $x = l$, the elements of the shape function matrix associated with the coordinates of the first node are equal to zero.

The shape function matrix given in Eq. (3.11) works for any basic Bernoulli-Euler beam element if the beam is considered very thin and thus the rotary inertia of cross-section is neglected. The displacement fields defined in Eq. (3.1), which is used to derive the shape function matrix in Eq. (3.11), only describe the displacement for an arbitrary point C located on the neutral axis of the beam shown in Figure 3.4. However, it is not able to describe the displacement for a generic point P of the beam that offsets from the neutral axis. For this reason, any energy associated with the tilted cross-section of the beam will be lost, and the rotary inertia of the cross-section is ignored as a result. Therefore, if a beam has a small slenderness ratio, additional terms need to be added into the displacement fields defined in Eq. (3.1).

Generally, the rotary angle of the cross-section is approximated as the partial derivative

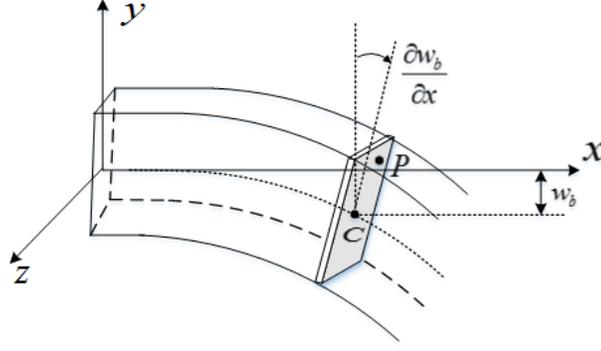


Figure 3.4: Tilted cross-section

of the transverse displacement w_b with respect to the longitudinal spatial coordinate x shown in Figure 3.4, which is the slope for small deformation assumption [51]. Using this definition, a modified displacement field which defines the displacement for a generic point P within the beam element can be written as

$$\left. \begin{aligned} w_1 &= u - yv'_y - zv'_z \\ w_2 &= v_y - z\phi \\ w_3 &= v_z + y\phi \\ w_4 &= \phi \end{aligned} \right\} \quad (3.12)$$

where u , v_y , v_z , and ϕ are defined in Eq. (3.1), and y and z are the other two spatial coordinates along Y and Z directions shown in Figure 3.3. Several terms have been added into the displacement field compared with Eq. (3.1). Notice that the additional terms added in the axial displacement w_1 is the foreshortening movement of P in axial direction due to the rotation of cross-section, but it is not the foreshortening term introduced later in Section 3.2 due to geometrical nonlinearity. Following the same steps as described above,

a modified shape function matrix can be generated as

$$\widehat{\mathbf{S}} = \begin{bmatrix} 1-\xi & 0 & 0 & 0 \\ 6[\xi - (\xi)^2]\eta & 1 - 3(\xi)^2 + 2(\xi)^3 & 0 & 0 \\ 6[\xi - (\xi)^2]\delta & 0 & 1 - 3(\xi)^2 + 2(\xi)^3 & 0 \\ 0 & -(1 - \xi)l\delta & (1 - \xi)l\eta & 1-\xi \\ [1 - 4\xi + 3(\xi)^2]l\delta & 0 & [-\xi + 2(\xi)^2 - (\xi)^3]l & 0 \\ [-1 + 4\xi - 3(\xi)^2]l\eta & [\xi - 2(\xi)^2 + (\xi)^3]l & 0 & 0 \\ \xi & 0 & 0 & 0 \\ 6[-\xi + (\xi)^2]\eta & 3(\xi)^2 - 2(\xi)^3 & 0 & 0 \\ 6[-\xi + (\xi)^2]\delta & 0 & 3(\xi)^2 - 2(\xi)^3 & 0 \\ 0 & -l\xi\delta & l\xi\eta & \xi \\ [-2\xi + 3(\xi)^2]l\delta & 0 & [(\xi)^2 - (\xi)^3]l & 0 \\ [2\xi - 3(\xi)^2]l\eta & [-(\xi)^2 + (\xi)^3]l & 0 & 0 \end{bmatrix}^T \quad (3.13)$$

where $\xi = x/l$, $\eta = y/l$, and $\delta = z/l$. This modified shape function matrix will be used to generate terms including the mass matrix, stiffness matrix, and force vectors of the Bernoulli-Euler beam element later in Section 3.1.3 - 3.1.10. Actually, users can choose either the form in Eq. (3.11) or the full expression in Eq. (3.13) according to different applications considering whether the rotary inertia of the cross-section is significant to the result.

Note the procedure outlined above for writing the displacement field in terms of the nodal coordinates of the element is general and can be used for any type of element with any type of nodal coordinates. This procedure also applies for planar analysis as well as spatial analysis.

3.1.2 Coordinate Systems

The coordinate system selection according to the floating frame of reference method defined by Shabana [1] is presented in this section. For a flexible body, Eq. (3.10) that indicates the

relation between the displacement field \mathbf{w} and nodal coordinates \mathbf{e} is defined with respect to any coordinate system that is initially parallel to the element coordinate system. Given a general finite-element body in the global coordinate system $\mathbf{X}_i\mathbf{Y}_i\mathbf{Z}_i$ shown in Figure 3.5, for instance, $x^iy^iz^i$ is the element coordinate system that translates and rotates with the element; that is, the origin of this coordinate system is rigidly attached to a point on the element. The XYZ system is a selected body coordinate system that need not be rigidly attached to a point on the body (floating frame of reference) [52]. For the beam shown in Figure 3.1, the body coordinate system is often attached to the tip end with the X axis initially parallel to the beam, which is referred to as the body-fixed frame. The $X^iY^iZ^i$ system is an intermediate element coordinate system whose origin is rigidly attached to the origin of the body coordinate system XYZ . The coordinates system $X^iY^iZ^i$ is assumed to have a fixed orientation with respect to the body coordinate system; that is, the $X^iY^iZ^i$ coordinate system translates and rotates with the body coordinate system. Moreover, it is assumed that the orientations of the axes $X^iY^iZ^i$ are selected in such a manner that they are initially parallel to the axes of the element coordinate system $x^iy^iz^i$. Therefore, Eq. (3.10) can be defined with respect to the $X^iY^iZ^i$ coordinate system. In general, each element has a unique intermediate element coordinate system associated with it.

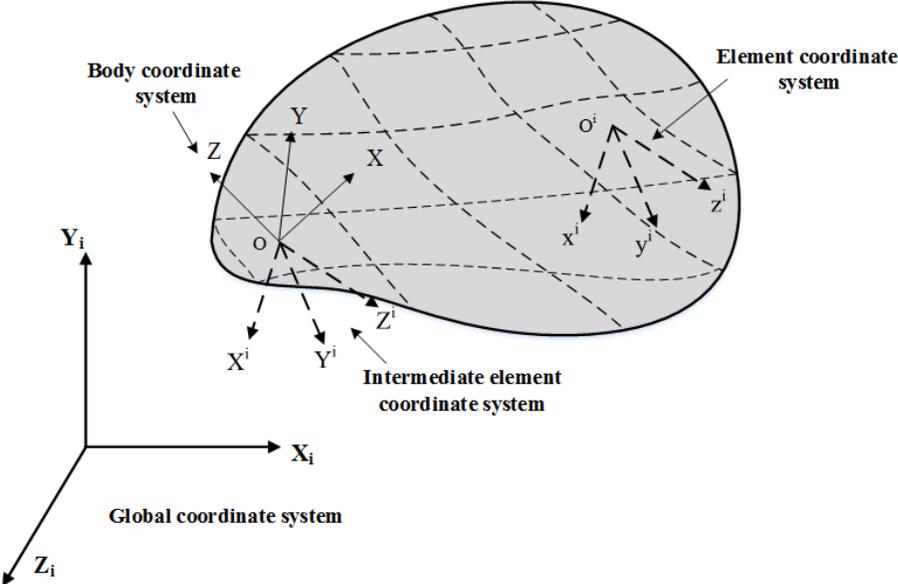


Figure 3.5: Finite-element coordinate systems for general problems defined by Shabana [1]

Using the coordinate system specification described above, the finite-element beam coordinate systems are selected in a simplified way in which the intermediate element coordinate system is parallel to the body-fixed frame (body coordinate system) shown in Figure 3.6. In this figure, P is an arbitrary point on the beam, and P' locates the position of P after deformation. \mathbf{u}_f^i is the local displacement vector, and \mathbf{u}^i is the local position vector used to locate an arbitrary point on the beam with respect to the intermediate element coordinate system. \mathbf{R} defines the translations of the body-fixed frame XYZ , and \mathbf{r}^i is the global position vector of an arbitrary point on the beam. All these quantities are defined in Section 3.1.3-3.1.6. It will be shown that by setting the intermediate element coordinate system parallel to the body-fixed frame, one can simplify the equations of these quantities.

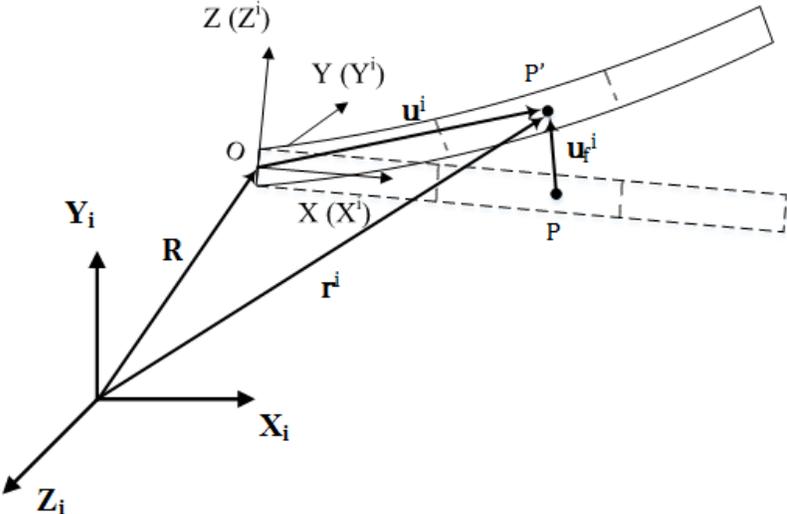


Figure 3.6: Selection of finite-element beam coordinate systems

3.1.3 Position Vector to Arbitrary Point

Since all of the nodal coordinates have their initial value when the body is in the undeformed state with respect to the body-fixed frame of reference, it is important to re-write the nodal coordinate vector \mathbf{e} in Eq. (3.4) as

$$\mathbf{e} = \mathbf{e}_0 + \mathbf{e}_f \tag{3.14}$$

where \mathbf{e}_0 represents the values of the coordinates in the undeformed state and \mathbf{e}_f represents the vector of the elastic nodal coordinates associated with the deformation of the element. When using finite elements, a deformable body is generally divided into more than one element. For an arbitrary element, a notation is often used to distinguish it from the others to avoid any confusion. Therefore, for an element i in a deformable body Eq. (3.14) is often written as

$$\mathbf{e}^i = \mathbf{e}_0^i + \mathbf{e}_f^i \quad (3.15)$$

where the superscript i refers to the element number in the finite element discretization of the deformable body.

For a general coordinate systems selection in which the intermediate element coordinate system is not parallel to the body-fixed frame, one needs to transfer the generalized nodal coordinates defined in the body-fixed frame to the nodal coordinates defined in the intermediate element coordinate system first:

$$\mathbf{e}^i = \bar{\mathbf{C}}^i \mathbf{q}_n^i \quad (3.16)$$

where $\bar{\mathbf{C}}$ is an orthogonal constant transformation matrix and \mathbf{q}_n^i is the vector of generalized nodal coordinates of the element i . The local position vector of an arbitrary point on the beam \mathbf{u}^i shown in Figure 3.6 can be determined in the body-fixed frame using Eq. (3.10) as

$$\mathbf{u}^i = \mathbf{C}^i \mathbf{w}_{i1..3}^i = \mathbf{C}^i \widehat{\mathbf{S}}_{1..3}^i \mathbf{e}_i^i = \mathbf{C}^i \mathbf{S}^i \mathbf{e}_i^i \quad (3.17)$$

where \mathbf{C}^i is a 3×3 transformation matrix that transforms the displacement field defined within the intermediate element coordinate system to the body-fixed coordinate system. Note that the shape function \mathbf{S}^i in Eq. (3.17) without the curved hat contains only the first three rows of $\widehat{\mathbf{S}}$ detailed in Eq. (3.13).

Substituting Eq. (3.16) into Eq. (3.17) yields

$$\mathbf{u}^i = \mathbf{C}^i \mathbf{S}^i \bar{\mathbf{C}}^i \mathbf{q}_n^i \quad (3.18)$$

Using this equation, one can define the position of an arbitrary point on the finite element with respect to the origin of the body coordinate system. The local position vector is expressed in terms of the set of nodal coordinates defined in the body-fixed frame. If setting the beam intermediate element coordinate system parallel to the body-fixed frame, both the constant transformation matrices \mathbf{C}^i and $\bar{\mathbf{C}}^i$ become identity matrices. In this case, Eq. (3.18) can be simplified as

$$\mathbf{u}^i = \mathbf{S}^i \mathbf{q}_n^i \quad (3.19)$$

where \mathbf{C}^i and $\bar{\mathbf{C}}^i$ are neglected. This simplified expression of the displacement vector will be used in subsequent sections when deriving energy terms and yield fundamental matrices.

In addition to the local position vector, we must also introduce a set of absolute Cartesian and rotational coordinates that, respectively, describe the location of the origin and orientation of the body-fixed frame in order to locate an arbitrary point in the flexible body with respect to the global coordinate system. Such coordinates can be referred to as reference coordinates of the body-fixed frame, and are expressed as

$$\mathbf{q}_r = [\mathbf{R}^T \quad \boldsymbol{\theta}^T]^T \quad (3.20)$$

where \mathbf{R} (shown in Figure 3.6) and $\boldsymbol{\theta}$ contain three absolute Cartesian and three rotational coordinates, respectively. Then one can uniquely define the global position vector of an arbitrary point on the i^{th} element of the flexible body with respect to the global coordinate system shown in Figure 3.6 as

$$\mathbf{r}^i = \mathbf{R} + \mathbf{A}(\boldsymbol{\theta}) \mathbf{u}^i \quad (3.21)$$

where \mathbf{A} is the conventional frame transformation matrix from the body-fixed frame to the global frame, which is a function of the rotational coordinates.

3.1.4 Connectivity and Boundary Condition Matrices

If a beam is discretized into n beam elements, following the manner given in Figure 3.1 that each element has two nodes and six degrees of freedom for every node, the beam will have $n + 1$ nodes and $6(n + 1)$ deformation degrees of freedom. Generally, a nodal coordinate vector \mathbf{q}_n will be defined right after setting the number of elements, and this vector contains all of the $6(n + 1)$ deformation degrees of freedom of the flexible body. Since we will write equations for each single element, we need to extract the corresponding element nodal coordinates \mathbf{q}_n^i from the total coordinates \mathbf{q}_n . Then the vector of element nodal coordinates can be written in terms of the total coordinates as

$$\mathbf{q}_n^i = \mathbf{B}_1^i \mathbf{q}_n \quad (3.22)$$

where \mathbf{B}_1^i is a constant Boolean transformation matrix, whose elements are either zeros or ones, which serves to express the connectivity of the element. Let us consider the example shown in Figure 3.1 where the two beam elements are rigidly connected at node 2. In this case, the vector \mathbf{q}_n^i contains 12 components representing the 12 nodal coordinates for each element, that is,

$$\mathbf{q}_n^1 = [u_{1..12}]^T, \quad \mathbf{q}_n^2 = [u_{7..18}]^T \quad (3.23)$$

which have the shared nodal coordinates $u_{7..12}$. However, the total coordinate vector \mathbf{q}_n defined for the assembled body is

$$\mathbf{q}_n = [u_{1..18}]^T \quad (3.24)$$

Therefore, the Boolean transformation of Eq. (3.22) can be recognized for the first and second element as

$$\mathbf{B}_1^1 = \begin{bmatrix} \mathbf{I}_{12}^{12} & \mathbf{0} \end{bmatrix}_{12}^{18}, \quad \mathbf{B}_1^2 = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{12}^{12} \end{bmatrix}_{12}^{18} \quad (3.25)$$

where the digits 12 and 18 on the right of the brackets indicate the number of rows and columns, respectively. Then the local position vector of the i^{th} element of Eq. (3.19) can

be written in terms of the total nodal coordinate vector using Eq. (3.22) as

$$\mathbf{u}^i = \mathbf{S}^i \mathbf{B}_1^i \mathbf{q}_n \quad (3.26)$$

However, this representation contains six rigid body modes using the shape function \mathbf{S}^i defined previously, and accordingly the rigid body motion of the elements with respect to the body reference is allowed. For instance, if u_1 , u_7 , and u_{13} in Figure 3.1 had the same value, the beam could have a longitudinal motion toward the positive x -direction with respect to the body-fixed frame. These rigid body modes should be eliminated by setting appropriate boundary conditions.

In rigid body analysis, it is often preferred to rigidly attach the origin of the body reference to the centre of mass. For a flexible body, its origin can be set very flexibly due to many factors in different cases. In the case of a flexible beam, the origin is generally fixed to one of the tip ends. Although many boundary conditions (e.g. simply supported beam, etc.) are considered when integrating the flexible body with different multibody systems via varying joint connections in conventional cases, we are primarily focusing on the condition where one of the tip ends of the beam is assumed to be fixed to the origin of the body reference (fixed-free). The main reason why we are interested in only the fixed-free boundary condition is because, within the computer implementation, we have various predefined joint connectors that can be used to integrate the flexible body to other multibody components.

In the finite element analysis, following the same derivation of Eq. (3.15), the total nodal coordinate vector of the flexible body can be written as

$$\mathbf{q}_n = \mathbf{q}_0 + \bar{\mathbf{q}}_f \quad (3.27)$$

where \mathbf{q}_0 is the vector of nodal coordinates in the undeformed state, and $\bar{\mathbf{q}}_f$ is the vector of nodal deformations defined as

$$\bar{\mathbf{q}}_f = \mathbf{B}_2 \mathbf{q}_f \quad (3.28)$$

where we introduce a linear transformation \mathbf{B}_2 that arises from imposing the boundary conditions, and \mathbf{q}_f contains all of the nodal deformation coordinates. For the beam with two elements and fixed-free condition, \mathbf{B}_2 is defined as

$$\mathbf{B}_2 = \begin{bmatrix} \mathbf{0}_6^6 & \mathbf{0}_6^{12} \\ \mathbf{0}_{12}^6 & \mathbf{I}_{12}^{12} \end{bmatrix} \quad (3.29)$$

where the digits on the bottom and top right of the entries indicate the number of rows and columns, respectively. One can easily notice that by multiplying \mathbf{q}_f with \mathbf{B}_2 , the first 6 deformation coordinates are forced to be zeros, which reflects the fixed-free boundary condition. \mathbf{B}_2 varies if different boundary conditions are considered.

Substituting Eqs. (3.28), (3.27), and (3.26) into Eq. (3.21), the expanded expression of the global position vector is obtained as

$$\mathbf{r}^i = \mathbf{R} + \mathbf{A}\mathbf{u}^i = \mathbf{R} + \mathbf{A}\mathbf{S}^i\mathbf{B}_1^i\mathbf{q}_n = \mathbf{R} + \mathbf{A}\mathbf{S}^i\mathbf{B}_1^i(\mathbf{q}_0 + \mathbf{B}_2\mathbf{q}_f) \quad (3.30)$$

where the position vector \mathbf{r}^i is written in terms of the reference and elastic coordinates of the flexible body.

3.1.5 Mass Matrix

One way to derive the mass matrix is to extract it from the expression of the kinetic energy of the elements. Developing the kinetic energy of the elements leads to the definition of the nonlinear terms that represent the inertia coupling between the reference motion of the body and the elastic deformation of the elements. All of the terms can be easily generated by using the definition of kinetic energy and the velocity vector from differentiating the position vector we have developed in the previous chapter. Differentiating the position vector of Eq. (3.30) with respect to time yields

$$\dot{\mathbf{r}}^i = \dot{\mathbf{R}} + \mathbf{A}(\boldsymbol{\omega} \times \mathbf{u}^i) + \mathbf{A}\mathbf{S}^i\mathbf{B}_1^i(\mathbf{B}_2\dot{\mathbf{q}}_f) \quad (3.31)$$

where $\boldsymbol{\omega}$ is the angular velocity vector defined in the local coordinate system. If using the function of skew-symmetric matrix, we can write

$$\boldsymbol{\omega} \times \mathbf{u}^i = -\tilde{\mathbf{u}}^i \boldsymbol{\omega} \quad (3.32)$$

where $\tilde{\mathbf{u}}$ indicates the skew-symmetric matrix generated from \mathbf{u} . The angular velocity vector $\boldsymbol{\omega}$ can be written in terms of the time derivatives of the reference rotational coordinates of the body as

$$\boldsymbol{\omega} = \bar{\mathbf{G}} \dot{\boldsymbol{\theta}} \quad (3.33)$$

where the Euler angles $[\alpha, \theta, \beta]^T$ with a sequence of 1-2-3 about the body-fixed frame can be used as the rotational coordinates in the rotational coordinates vector $\boldsymbol{\theta}$ (or Euler angle vector); the angular velocity vector transformation matrix $\bar{\mathbf{G}}$ is defined, in terms of the Euler angles, as

$$\bar{\mathbf{G}} = \begin{bmatrix} \cos \theta \cos \beta & \sin \beta & 0 \\ -\cos \theta \sin \beta & \cos \beta & 0 \\ \sin \theta & 0 & 1 \end{bmatrix} \quad (3.34)$$

Notice that, in reality, the rotational coordinates vector $\boldsymbol{\theta}$ can have different sequence of Euler angles, which causes shifts of entries related to body rotation in mass matrix and inertia forces. The sequence of 1-2-3 about the body-fixed frame is used in this article as an example. After the substitution of Eq. (3.32) and (3.33) into Eq. (3.31), the velocity vector can be re-written as

$$\dot{\mathbf{r}}^i = \dot{\mathbf{R}} - \mathbf{A} \tilde{\mathbf{u}}^i \bar{\mathbf{G}} \dot{\boldsymbol{\theta}} + \mathbf{A} \mathbf{S}^i \mathbf{B}_1^i (\mathbf{B}_2 \dot{\mathbf{q}}_f) \quad (3.35)$$

or in a partitioned form as

$$\dot{\mathbf{r}}^i = \begin{bmatrix} \mathbf{I} & -\mathbf{A} \tilde{\mathbf{u}}^i \bar{\mathbf{G}} & \mathbf{A} \mathbf{S}^i \mathbf{B}_1^i \mathbf{B}_2 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{R}} \\ \dot{\boldsymbol{\theta}} \\ \dot{\mathbf{q}}_f \end{bmatrix} \quad (3.36)$$

where \mathbf{I} is the 3×3 identity matrix.

Conventionally, the kinetic energy of the i^{th} element can be defined as

$$T^i = \frac{1}{2} \int_{V^i} \rho^i \dot{\mathbf{r}}^{iT} \dot{\mathbf{r}}^i dV^i = \frac{1}{2} \int_0^{l^i} \rho^i \dot{\mathbf{r}}^{iT} \dot{\mathbf{r}}^i A^i dx \quad (3.37)$$

where ρ^i and V^i are the mass density and volume, and A^i and l^i are the cross-section area and length of the i^{th} element, respectively. Note that the cross-section area A^i can vary and be a function of x in general cases.

Substituting Eq. (3.36) into Eq. (3.37) yields

$$T^i = \frac{1}{2} \dot{\mathbf{q}}^{iT} \mathbf{M}^i \dot{\mathbf{q}}^i \quad (3.38)$$

where \mathbf{q} is the total vector of generalized coordinates of the body defined as

$$\mathbf{q} = [\mathbf{q}_r^T \quad \mathbf{q}_f^T]^T = [\mathbf{R}^T \quad \boldsymbol{\theta}^T \quad \mathbf{q}_f^T] \quad (3.39)$$

and \mathbf{M}^i is the mass matrix of the i^{th} element, which can be written in the partitioned form as

$$\mathbf{M}^i = \int_0^{l^i} \rho^i \begin{bmatrix} \mathbf{I} & -\mathbf{A} \tilde{\mathbf{u}}^i \bar{\mathbf{G}} & \mathbf{A} \mathbf{S}^i \mathbf{B}_1^i \mathbf{B}_2 \\ \bar{\mathbf{G}}^T \tilde{\mathbf{u}}^{iT} \tilde{\mathbf{u}}^i \bar{\mathbf{G}} & \bar{\mathbf{G}}^T \tilde{\mathbf{u}}^i \mathbf{S}^i \mathbf{B}_1^i \mathbf{B}_2 \\ \text{sym} & & \mathbf{B}_2^T (\mathbf{S}^i \mathbf{B}_1^i)^T \mathbf{S}^i \mathbf{B}_1^i \mathbf{B}_2 \end{bmatrix} A^i dx \quad (3.40)$$

where the orthogonality of the rotation matrix, $\mathbf{A}^T \mathbf{A} = \mathbf{I}$, is used and omitted in the partitioned form. The mass matrix can also be simplified as

$$\mathbf{M}^i = \begin{bmatrix} \mathbf{m}_{rr}^i & \mathbf{m}_{r\theta}^i & \mathbf{m}_{rf}^i \\ & \mathbf{m}_{\theta\theta}^i & \mathbf{m}_{\theta f}^i \\ \text{sym} & & \mathbf{m}_{ff}^i \end{bmatrix} \quad (3.41)$$

where the subscripts r , θ , and f refer to the reference translation, reference rotation, and deformation of the body, respectively. The detailed derivation of each of the sub \mathbf{m} -

matrices of constant cross-section area A^i can be found in Appendix A.1.

After obtaining the mass matrix of an element, the total mass matrix of the body can be determined by summing up all of the element mass matrices defined as

$$\mathbf{M} = \sum_{i=1}^n \mathbf{M}^i \quad (3.42)$$

where n is the number of elements and \mathbf{M} is the total mass matrix of the flexible body.

3.1.6 Stiffness Matrix

The normal strain energy of the i^{th} element can be written as

$$U = \frac{1}{2} \int_{V^i} \sigma^{iT} \varepsilon^i dV = \frac{1}{2} \int_0^{l^i} \sigma^{iT} \varepsilon^i A^i dx \quad (3.43)$$

where σ^i and ε^i are the stress and strain vectors, respectively. Since the assumptions of Bernoulli-Euler beam is used, different deformations are looked at separately. Therefore, the strain can be categorized into three main classes: strain due to axial deformation, strain due to bending, and strain due to torsion. In this manner, the stress can also be derived accordingly by using proper constitutive equations.

3.1.6.1 Axial Deformation

For the axial deformation, the strain-displacement relationship for element i can be defined as

$$\varepsilon_{xx}^i = \frac{du_{fx}^i}{dx} \quad (3.44)$$

where u_{fx}^i is the x -component of the local displacement vector \mathbf{u}_f^i shown in Figure 3.6 defined using Eqs. (3.26)-(3.28) as

$$\mathbf{u}_f^i = \mathbf{S}^i \mathbf{B}_1^i \mathbf{B}_2 \mathbf{q}_f \quad (3.45)$$

For a slender flexible beam, the transverse strain due to the Poisson effect is neglected. Therefore, the constitutive relation for the axial deformation is defined as

$$\sigma_{xx}^i = E\varepsilon_{xx}^i \quad (3.46)$$

where E is the Young's modulus. Using Eq. (3.43), (3.44), and (3.46), the strain energy of element i due to axial deformation can be written as

$$U_a^i = \frac{1}{2} \int_0^{l^i} EA^i u_{fx}^i{}' u_{fx}^i{}' dx \quad (3.47)$$

where the subscript a indicates axial deformation, and the prime symbol $'$ means the differentiation with respect to x . Since u_{fx}^i is the x -component of the displacement vector \mathbf{u}_f^i , it can be expanded as

$$u_{fx}^i = \mathbf{S}_{k=1}^i \mathbf{B}_1^i \mathbf{B}_2 \mathbf{q}_f \quad (3.48)$$

where $\mathbf{S}_{k=1}^i$ is the 1st row of the shape function matrix \mathbf{S}^i defined in Eq. (3.17), or in a compact form as

$$u_{fx}^i = \mathbf{N}_x \mathbf{q}_f \quad (3.49)$$

where \mathbf{N}_x is defined as

$$\mathbf{N}_x = \mathbf{S}_{k=1}^i \mathbf{B}_1^i \mathbf{B}_2 \quad (3.50)$$

Substituting Eq. (3.49) into Eq. (3.47) leads to

$$U_a^i = \frac{1}{2} \mathbf{q}_f^T \left(\int_0^{l^i} EA^i \mathbf{N}_x^T \mathbf{N}_x' dx \right) \mathbf{q}_f \quad (3.51)$$

which can also be simplified as

$$U_a^i = \frac{1}{2} \mathbf{q}_f^T \mathbf{K}_{fa}^i \mathbf{q}_f \quad (3.52)$$

where \mathbf{K}_{fa}^i is the element stiffness matrix of the element i due to axial deformation defined as

$$\mathbf{K}_{fa}^i = \int_0^{l^i} EA^i \mathbf{N}_x^T \mathbf{N}_x' dx \quad (3.53)$$

3.1.6.2 Bending Deformation

Recall that the conventional Bernoulli-Euler beam assumes small deformation, so only the normal bending strain ε_{xx} is considered while the shear strain and transverse strain are assumed trivial which lead to zero stresses. Note that this is not true but comes out of simplifying a 3D problem to 2D while maintaining sufficient accuracy. This means that one can derive the stiffness matrices for the bending deformation about the y - and z -axis, respectively. Taking the bending deformation about z -axis for instance, the normal bending strain can be defined as

$$\varepsilon_{xx,b}^i = -y\kappa^i \quad (3.54)$$

where y is the distance from the neutral axis that can be either positive or negative, and κ^i is the reciprocal of the radius of curvature of the element defined as

$$\kappa^i = \frac{1}{\rho_r} = \frac{d\theta^i}{ds} \quad (3.55)$$

where the radius of curvature ρ_r , incremental angle of rotation $d\theta^i$, and infinitesimal deflected length ds are shown in Figure 3.7. The u_{fy}^i in Figure 3.7 is the y -component of the

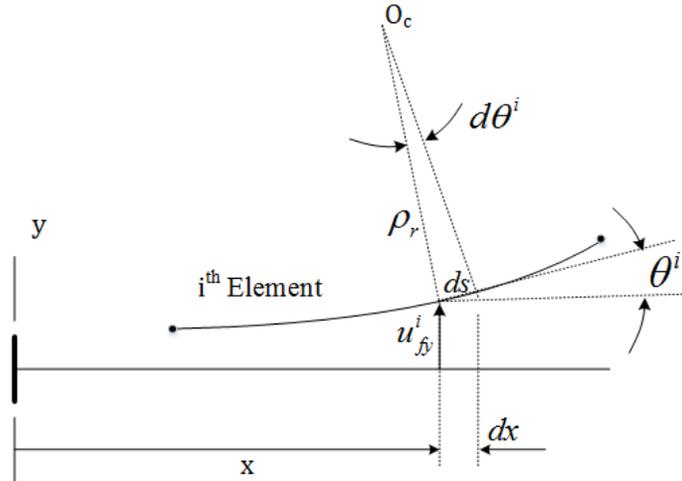


Figure 3.7: Bending curvature

displacement vector \mathbf{u}_f^i in Eq. (3.45), which indicates the bending displacement of an arbitrary point in the element along y direction. If the angle of rotation θ^i is subtle according to the small-deformation assumption of the Bernoulli-Euler beam, it can be approximated

by the slope of the bending defined as

$$\theta^i = \tan \theta^i = \frac{du_{fy}^i}{dx} \quad (3.56)$$

Furthermore, the infinitesimal deflected length ds is approximately equal to the original infinitesimal length dx , thus

$$ds = dx \quad (3.57)$$

Using Eqs. (3.54)-(3.57), one can derive the normal bending strain as

$$\varepsilon_{xx,b}^i = -y \frac{d^2 u_{fy}^i}{dx^2} \quad (3.58)$$

The constitutive relation defined in Eq. (3.46) is still valid for the normal bending strain and stress, so substituting Eq. (3.58) and Eq. (3.46) into Eq. (3.43) yields the strain energy of bending about z -axis as

$$U_{bz}^i = \frac{1}{2} \int_{V^i} \int E u_{fy}^{i''} u_{fy}^{i''} y^2 dV = \frac{1}{2} \int_0^{l^i} \int E u_{fy}^{i''} u_{fy}^{i''} y^2 dA dx \quad (3.59)$$

which can be simplified as

$$U_{bz}^i = \frac{1}{2} \int_0^{l^i} EI_y u_{fy}^{i''} u_{fy}^{i''} dx \quad (3.60)$$

where I_y is the second moment of area about y -axis defined as

$$I_y = \int y^2 dA \quad (3.61)$$

Following the same manner of Eq. (3.49)-(3.53), the stiffness matrix corresponding to the bending deformation about z -axis can be generalized as

$$\mathbf{K}_{fb,z}^i = \int_0^{l^i} EI_y \mathbf{N}_y^T \mathbf{N}_y'' dx \quad (3.62)$$

where \mathbf{N}_y is defined as

$$\mathbf{N}_y = \mathbf{S}_{k=2}^i \mathbf{B}_1^i \mathbf{B}_2 \quad (3.63)$$

where $\mathbf{S}_{k=2}^i$ is the 2^{nd} row of the shape function matrix \mathbf{S}^i defined in Eq. (3.17). Similarly, following the same routine, the stiffness matrix corresponding to the bending deformation about y -axis can be derived as

$$\mathbf{K}_{fb,y}^i = \int_0^{l^i} EI_z \mathbf{N}_z^{T''} \mathbf{N}_z'' dx \quad (3.64)$$

where \mathbf{N}_z is defined as

$$\mathbf{N}_z^i = \mathbf{S}_{k=3}^i \mathbf{B}_1^i \mathbf{B}_2 \quad (3.65)$$

where $\mathbf{S}_{k=3}^i$ is the 3^{rd} row of the shape function matrix \mathbf{S}^i defined in Eq. (3.17).

3.1.6.3 Torsional Deformation

When defining the torsional strain energy, it is generally more precise to rewrite the strain energy defined in Eq. (3.43) as

$$U_t^i = \frac{1}{2} \int_{V^i} \tau^i \gamma^i dV \quad (3.66)$$

where τ^i and γ^i are the torsional stress and strain of the i^{th} element, respectively. The relation of the torsional stress and strain is defined as

$$\tau^i = G \gamma^i \quad (3.67)$$

where G is the shear modulus. If assuming small strain, the torsional strain γ^i is generally expressed in terms of the twist angle as

$$\gamma^i = r \phi^{i'} \quad (3.68)$$

where r and $\phi^{i'}$ are, respectively, the distance from the neutral axis within a cross-section of the beam and the derivative of the twist angle ϕ^i with respect to the x coordinate, which are shown in Figure 3.8. Therefore, the torsional strain energy in Eq. (3.66) can be written as

$$U_t^i = \frac{1}{2} \int_{V^i} \int G \phi^{i'} \phi^{i'} r^2 dV = \frac{1}{2} \int_0^{l^i} \int G \phi^{i'} \phi^{i'} r^2 dA dx \quad (3.69)$$

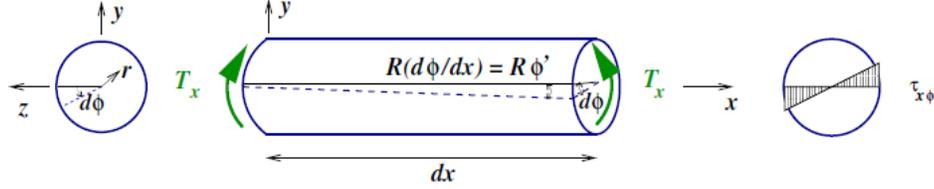


Figure 3.8: Twist angle (note R is the radius of the cross-section)

which can be simplified as

$$U_t^i = \frac{1}{2} \int_0^{l^i} GI_x \phi^{i'} \phi^{i'} dx \quad (3.70)$$

where the way of defining the second moment of area about x -axis is used as follows

$$I_x = \int r^2 dA = \int (z^2 + y^2) dA = I_y + I_z \quad (3.71)$$

Similar to the other types of deformation, the twist angle can be written in terms of the nodal coordinates as

$$\phi^i = \mathbf{N}_\phi \mathbf{q}_f \quad (3.72)$$

where \mathbf{N}_ϕ is defined as

$$\mathbf{N}_\phi = \mathbf{S}_{k=4}^i \mathbf{B}_1^i \mathbf{B}_2 \quad (3.73)$$

where $\mathbf{S}_{k=4}^i$ is the 4th row of the shape function matrix \mathbf{S}^i defined in Eq. (3.17). Following the same steps of Eq. (3.49)-(3.53), the stiffness matrix corresponding to the torsional deformation is obtained as

$$\mathbf{K}_\phi^i = \int_0^{l^i} GI_x \mathbf{N}_\phi^{T'} \mathbf{N}_\phi' dx \quad (3.74)$$

3.1.6.4 Superposition

The stiffness matrix for the i^{th} element can be generalized by simply summing up all the stiffness matrices corresponding to the different types of deformation as

$$\mathbf{K}_f^i = \mathbf{K}_{fa}^i + \mathbf{K}_{fb,y}^i + \mathbf{K}_{fb,z}^i + \mathbf{K}_\phi^i \quad (3.75)$$

and the total stiffness matrix of the entire beam can be determined by summing up all of the element stiffness matrices defined as

$$\mathbf{K}_f = \sum_{i=1}^n \mathbf{K}^i \quad (3.76)$$

where n is the number of elements and \mathbf{K}_f is the total stiffness matrix of the flexible beam.

3.1.7 Examination of the Linear Stiffness Matrix

If using the general nonlinear strain-displacement relations [28], the normal strain of the i^{th} element in the X direction can be written as

$$\varepsilon_{xx}^i = \frac{\partial w_1^i}{\partial x} + \frac{1}{2} \left[\left(\frac{\partial w_1^i}{\partial x} \right)^2 + \left(\frac{\partial w_2^i}{\partial x} \right)^2 + \left(\frac{\partial w_3^i}{\partial x} \right)^2 \right] \quad (3.77)$$

where $w_{1,3}^i$ are the first three translational components of the displacement field defined in Eq. (3.12). Substituting the whole terms of $w_{1,3}^i$ into ε_{xx}^i yields

$$\varepsilon_{xx}^i = u' - yv''_y - zv''_z + \frac{1}{2}[u' - yv''_y - zv''_z]^2 + \frac{1}{2}[v'_y - z\phi']^2 + \frac{1}{2}[v'_z + z\phi']^2 \quad (3.78)$$

where the prime $'$ indicates differentiation with respect to x and the superscript i of each deformation variable is neglected for convenience. Neglecting terms higher in order than second due to their insignificance, Eq. (3.78) is then reduced to

$$\varepsilon_{xx}^i = u' - yv''_y - zv''_z + \frac{1}{2}(u'^2 + v'^2_y + v'^2_z) \quad (3.79)$$

where the term $\frac{1}{2}u'^2$ is actually trivial, which is often neglected, since the axial deformation of a slender beam is small compared with bending deformations. Therefore, Eq. (3.79) is further reduced to

$$\varepsilon_{xx}^i = u' - yv''_y - zv''_z + \frac{1}{2}(v'^2_y + v'^2_z) \quad (3.80)$$

Substituting Eq. (3.80) into the strain energy defined in Eq. (3.44) yields

$$\begin{aligned}
U^i &= -\frac{1}{2} \int_V E \varepsilon_{xx}^i{}^2 dV \\
&= -\frac{1}{2} \int_V E (u'v'_y{}^2 + u'v'_z{}^2 + \frac{1}{2}v'_y{}^2v'_z{}^2 + 2yzv''_yv''_z \\
&\quad + \frac{1}{4}v'_y{}^4 + \frac{1}{4}v'_z{}^4 - zv''_zv'_y{}^2 - 2zu'v''_z - yv''_yv'_y{}^2 \\
&\quad + \underline{u'^2} + \underline{y^2v''_y{}^2} + \underline{z^2v''_z{}^2} - yv''_yv'_z{}^2 - 2yu'v''_y - zv''_zv'_z{}^2) dV \\
&= -\frac{1}{2} \int_V E (u'v'_y{}^2 + u'v'_z{}^2 + \frac{1}{2}v'_y{}^2v'_z{}^2 + \frac{1}{4}v'_y{}^4 \\
&\quad + \frac{1}{4}v'_z{}^4 + \underline{u'^2} + \underline{y^2v''_y{}^2} + \underline{z^2v''_z{}^2}) dV
\end{aligned} \tag{3.81}$$

where certain terms vanish because $\int_V ydV$, $\int_V zdV$, and $\int_V yzdV$ are equal to zero if the x -axis is coincident with the neutral axis of the element.

If looking at the types of deformation separately, as stated by the assumption of Bernoulli-Euler beam theory, the total linear normal strain energy of an element obtained in Chapter 3.1.6 can be generalized as

$$U^i = -\frac{1}{2} \int_V E (\underline{u'^2} + \underline{y^2v''_y{}^2} + \underline{z^2v''_z{}^2}) dV \tag{3.82}$$

which consists of the strain energy associated with axial and bending deformations that are taken separately to derive the corresponding stiffness matrices.

Therefore, comparing Eq. (3.82) and Eq. (3.81), it is noticed that the Bernoulli-Euler normal strain energy, by only keeping the underlined terms in Eq. (3.81), neglects all the coupled terms as well as some high-order terms.

3.1.8 Quadratic Velocity Force Vector

The quadratic velocity force vector is part of the inertia force that becomes significant when rotational rigid-body motion exists, and it consists of gyroscopic and Coriolis forces. The derivation of the quadratic velocity force vector is presented in this section by using virtual work of the inertia force of the deformable body. The virtual work of the inertia

force of the i^{th} element is defined as

$$\delta W^i = \int_{V^i} \rho \delta \mathbf{r}_i^T \ddot{\mathbf{r}}_i dV \quad (3.83)$$

where \mathbf{r}_i is the position vector defined in Eq. (3.30), and its virtual change can be written as

$$\delta \mathbf{r}_i = \delta \mathbf{R} - \mathbf{A} \tilde{\mathbf{u}}^i \delta \boldsymbol{\omega} + \mathbf{A} \mathbf{S}^i \mathbf{B}_1^i \mathbf{B}_2 \delta \mathbf{q}_f \quad (3.84)$$

where $\tilde{\mathbf{u}}^i$ is the skew symmetric matrix of \mathbf{u}^i . The acceleration vector $\ddot{\mathbf{r}}_i$, which is the derivative of the velocity vector defined in Eq. (3.35), is written as

$$\ddot{\mathbf{r}}_i = \ddot{\mathbf{R}} - \mathbf{A} \tilde{\mathbf{u}}^i \dot{\boldsymbol{\omega}} + \underline{\mathbf{A} \tilde{\boldsymbol{\omega}}^2 \mathbf{u}^i} + \underline{2\mathbf{A} \tilde{\boldsymbol{\omega}} (\mathbf{S}^i \mathbf{B}_1^i \mathbf{B}_2 \dot{\mathbf{q}}_f)} + \mathbf{A} (\mathbf{S}^i \mathbf{B}_1^i \mathbf{B}_2 \ddot{\mathbf{q}}_f) \quad (3.85)$$

Note that only the underlined parts of the acceleration vector are involved in deriving the quadratic velocity force vector, as the other terms are related to the kinetic energy that has been used to derive the mass matrix in Eq. (3.40). Therefore, using the virtual work of the inertia force and the underlined terms of the acceleration vector, the quadratic velocity force vector of the i^{th} element can be written as

$$\mathbf{Q}_v = \begin{bmatrix} \mathbf{Q}_{vr} \\ \mathbf{Q}_{v\theta} \\ \mathbf{Q}_{vf} \end{bmatrix} = \int_{V^i} \rho \begin{bmatrix} \mathbf{I} \\ -\tilde{\mathbf{u}}^{iT} \mathbf{A}^T \\ \mathbf{B}_2^T \mathbf{B}_1^{iT} \mathbf{S}^{iT} \mathbf{A}^T \end{bmatrix} [\underline{\mathbf{A} \tilde{\boldsymbol{\omega}}^2 \mathbf{u}^i} + \underline{2\mathbf{A} \tilde{\boldsymbol{\omega}} (\mathbf{S}^i \mathbf{B}_1^i \mathbf{B}_2 \dot{\mathbf{q}}_f)}] dV \quad (3.86)$$

or as

$$\mathbf{Q}_v = \begin{bmatrix} \mathbf{Q}_{vr} \\ \mathbf{Q}_{v\theta} \\ \mathbf{Q}_{vf} \end{bmatrix} = \int_0^{l^i} \rho A^i \begin{bmatrix} \mathbf{A} \tilde{\boldsymbol{\omega}}^2 \mathbf{u}^i + 2\mathbf{A} \tilde{\boldsymbol{\omega}} (\mathbf{S}^i \mathbf{B}_1^i \mathbf{B}_2 \dot{\mathbf{q}}_f) \\ -\tilde{\mathbf{u}}^{iT} \tilde{\boldsymbol{\omega}}^2 \mathbf{u}^i - 2\tilde{\mathbf{u}}^{iT} \tilde{\boldsymbol{\omega}} (\mathbf{S}^i \mathbf{B}_1^i \mathbf{B}_2 \dot{\mathbf{q}}_f) \\ \mathbf{B}_2^T \mathbf{B}_1^{iT} \mathbf{S}^{iT} [\tilde{\boldsymbol{\omega}}^2 \mathbf{u}^i + 2\tilde{\boldsymbol{\omega}} (\mathbf{S}^i \mathbf{B}_1^i \mathbf{B}_2 \dot{\mathbf{q}}_f)] \end{bmatrix} dl \quad (3.87)$$

The quadratic velocity force vector is a complex term that takes dramatically extra computational time to be integrated symbolically. Furthermore, it leads to little difference of the beam deformation and motion results if the overall rigid-body motion exists only to a small extent or the beam is stiff. Some cases are illustrated in Section 5 to clearly identify the

importance of the quadratic velocity vector under difference circumstances, and sometimes it can be deactivated to save computational time.

3.1.9 Generalized Gravitational Force Vector

The generalized gravitational force vector is derived in this section by using the virtual work formulation. The virtual work due to gravity can be expressed as

$$\delta W_g^i = \int_{V^i} \rho \mathbf{g}^T \delta \mathbf{r}_i dV \quad (3.88)$$

where \mathbf{g} is the gravity vector, which determines both the value and direction of the gravity. If the direction of the gravity is determined to be along the $-y$ axis, for example, then the gravity vector is defined as

$$\mathbf{g}_y = \begin{bmatrix} 0 & -g & 0 \end{bmatrix}^T \quad (3.89)$$

where g is the gravity value. Substituting the virtual change of the position vector $\delta \mathbf{r}_i$ defined in Eq. (3.84) into Eq. (3.88), it leads to

$$\delta W_g^i = \int_0^{t^i} \rho A^i \begin{bmatrix} \mathbf{g}^T & -\mathbf{g}^T \mathbf{A} \tilde{\mathbf{u}}^i & \mathbf{g}^T \mathbf{A} \mathbf{S}^i \mathbf{B}_1^i \mathbf{B}_2 \end{bmatrix} dx \begin{bmatrix} \delta \mathbf{R} \\ \delta \boldsymbol{\omega} \\ \delta \mathbf{q}_f \end{bmatrix} \quad (3.90)$$

or in a compact form as

$$\delta W_g^i = \mathbf{F}_g^T \begin{bmatrix} \delta \mathbf{R} \\ \delta \boldsymbol{\omega} \\ \delta \mathbf{q}_f \end{bmatrix} \quad (3.91)$$

where the \mathbf{F}_g is the generalized gravitational force vector defined as

$$\mathbf{F}_g = \begin{bmatrix} \mathbf{Q}_{gr} & \mathbf{Q}_{g\theta} & \mathbf{Q}_{gf} \end{bmatrix}^T = \int_0^{t^i} \rho A^i \begin{bmatrix} \mathbf{g}^T & -\mathbf{g}^T \mathbf{A} \tilde{\mathbf{u}}^i & \mathbf{g}^T \mathbf{A} \mathbf{S}^i \mathbf{B}_1^i \mathbf{B}_2 \end{bmatrix}^T dx \quad (3.92)$$

3.1.10 Generalized External Force Vector

In general, the external force applied on a body can either be a point load or a distributed load. If the load is distributed, the derivation of the generalized external force vector will be similar as that of obtaining the generalized gravitational force vector discussed in the previous section as the gravity is a typical distributed load. Complex distributed loads are beyond the scope of the research, so it is not covered in this article. A general approach of applying distributed loads on a flexible beam can be referred to [53]. In most cases of multibody system analysis, the external force is caused by one or several point loads. The derivation of the generalized external force vector resulting from a single point load is presented in this section. The force vector has three components, which are defined in the global coordinate system, that is,

$$\mathbf{P} = \begin{bmatrix} P_1 & P_2 & P_3 \end{bmatrix}^T \quad (3.93)$$

The virtual work of the force \mathbf{P} is defined as

$$\delta W_e = \mathbf{P}^T \delta \mathbf{r}_p \quad (3.94)$$

where $\delta \mathbf{r}_p$ is the virtual change of the global position vector of the point P where the load is acting on. Assuming the point P is on the p^{th} element, for example, then $\delta \mathbf{r}_p$ is written in a partitioned form as

$$\delta \mathbf{r}_p = \begin{bmatrix} \mathbf{I} & -\mathbf{A}\tilde{\mathbf{u}}^p & \mathbf{A}\mathbf{S}^p\mathbf{B}_1^p\mathbf{B}_2 \end{bmatrix} \begin{bmatrix} \delta \mathbf{R} \\ \delta \boldsymbol{\omega} \\ \delta \mathbf{q}_f \end{bmatrix} \quad (3.95)$$

where the superscript p indicates the unique terms defined at the point P in the p^{th} element. Substituting Eq. (3.95) into Eq. (3.94) yields

$$\delta W_e = \mathbf{F}_e^T \cdot \begin{bmatrix} \delta \mathbf{R} \\ \delta \boldsymbol{\omega} \\ \delta \mathbf{q}_f \end{bmatrix} = \mathbf{P}^T \begin{bmatrix} \mathbf{I} & -\mathbf{A}\tilde{\mathbf{u}}^p & \mathbf{A}\mathbf{S}^p\mathbf{B}_1^p\mathbf{B}_2 \end{bmatrix} \begin{bmatrix} \delta \mathbf{R} \\ \delta \boldsymbol{\omega} \\ \delta \mathbf{q}_f \end{bmatrix} \quad (3.96)$$

where \mathbf{F}_e is the generalized external force vector written as

$$\mathbf{F}_e = \begin{bmatrix} \mathbf{P}^T & -\mathbf{P}^T\mathbf{A}\tilde{\mathbf{u}}^p & \mathbf{P}^T\mathbf{A}\mathbf{S}^p\mathbf{B}_1^p\mathbf{B}_2 \end{bmatrix}^T \quad (3.97)$$

or, equivalently, one can write it as

$$\mathbf{F}_e = \begin{bmatrix} \mathbf{Q}_{er} & \mathbf{Q}_{e\theta} & \mathbf{Q}_{ef} \end{bmatrix}^T \quad (3.98)$$

One may notice that the rotational component of the generalized external force vector $\mathbf{Q}_{e\theta}$ is the generalized moment caused by the point load. If there is any additionally applied moment on the body, it can be directly applied on the rigid body and the corresponding node.

3.2 Geometrically Nonlinear Finite Element Formulation

As more and more requirements on simulation speed and accuracy have been brought forward, the conventional finite element method shows inadequacy to model rigid-flexible coupling dynamic systems. It is mentioned in Chapter 2.2 that several nonlinear formulation techniques have been invented to provide the flexible beam undergoing large overall motion with more accuracy, and revealing the dynamical stiffening effect is considered the core duty of these nonlinear terms. Among these formulations, the one that uses the addition of geometrically foreshortening term to include geometric stiffness terms in the equation of motion is regarded as the most efficient method. The biggest advantage of this formulation is that the stiffness matrices are linear, thus making it computationally efficient. The detailed formulation of the foreshortening method is structured in this chapter

based on the conventional finite-element-method terms defined in Chapter 3.1.

3.2.1 Global Position Vector with Foreshortening

During the derivation of the conventional finite element formulation based on the Bernoulli-Euler beam theory, the displacements corresponding to different types of deformation are looked at separately. The longitudinal displacement is only produced by the actual axial forces. The foreshortening, which is the shortened longitudinal displacement of a beam due to its in-extensibility undergoing bending deflection shown in Figure 3.9, cannot be captured by the conventional linear formulation. Considering the foreshortening effect, the

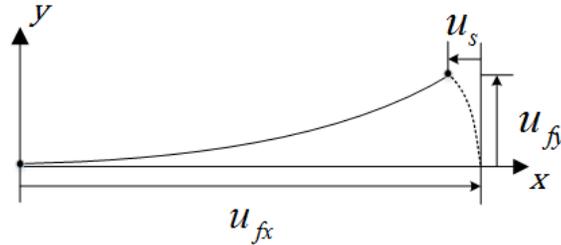


Figure 3.9: Shortened longitudinal displacement caused by bending deflection

longitudinal displacement of any point at the neutral axis, u_a , should be divided into the axial deformation produced by the actual axial forces, u_{fx} , and the shortening produced by the deflection u_s ,

$$u_a = u_{fx} + u_s \quad (3.99)$$

The foreshortening of an infinitesimal curve dl can be approximated, as shown by a 2-D case in Figure 3.10, by projecting the difference of $dl - dx$ onto the undeformed neutral

axis,

$$\begin{aligned}
ds &= (dl - dx) \cos \alpha \\
&= (dl - dx) \frac{dx}{dl} \\
&= \left(1 - \frac{dx}{dl}\right) dx \\
&= \left(1 - \frac{dx}{\sqrt{du_{fy}^2 + dx^2}}\right) dx \\
&= \left(1 - \frac{1}{\sqrt{u'_{fy}{}^2 + 1}}\right) dx
\end{aligned} \tag{3.100}$$

Assuming small values of u'_{fy} and using a set of Taylor series up to the second order, the

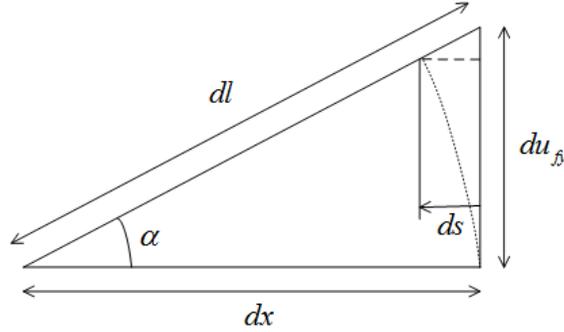


Figure 3.10: Foreshortening of an infinitesimal curve

expression of ds in Eq. (3.100) can be simplified as

$$ds = \frac{1}{2} u'_{fy}{}^2 dx \tag{3.101}$$

The total shortening accumulated from a reference point x_0 is obtained by integration as

$$u_s = -\frac{1}{2} \int_{x_0}^x u'_{fy}{}^2 dx \tag{3.102}$$

If one wants to know the shortening amount of an entire beam modelled with only a single element, for instance, one only needs to integrate u_s along the whole length as

$$u_s = -\frac{1}{2} \int_0^L \mathbf{q}_f^T \mathbf{N}_y^{1T'} \mathbf{N}_y^{1'} \mathbf{q}_f dx \tag{3.103}$$

where \mathbf{N}_y^1 is defined in Eq. (3.63) with $i = 1$. To calculate the total foreshortening of a beam discretized by more than one finite element, however, some critical factors must be considered ahead and modifications made correspondingly. When the beam has several finite elements each with a length of l , the foreshortening of the i^{th} element is written as

$$u_s^{ei} = -\frac{1}{2} \int_0^l \mathbf{q}_f^T \mathbf{N}_y^{iT'} \mathbf{N}_y^{i'} \mathbf{q}_f dx \quad (3.104)$$

where u_s^{ei} only represents the shortened amount produced in that element by its own deflection instead of the whole amount with respect to the base frame. Even though the nodal deformation coordinates, \mathbf{q}_f , represents the actual nodal deformation with respect to the base frame, the integration along a finite length l makes u_s^{ei} only able to represent a local shortening of a single element. Another attention should be paid to the integration of u_s^{ei} , as it cannot be integrated along the entire length L when using the finite element method. The shape function matrix of each element uses its own nodal deformation to estimate the deformation of an arbitrary point within that element, and it must be integrated within the same element. If integrating it along the entire beam length, it will yield completely wrong results. Actually, the total shortening accumulated by the finite elements located between the base frame and the i^{th} element should be calculated by the sum of the shortenings of the $i - 1$ elements plus that of itself shown in Figure 3.11. Therefore, the actual shortening

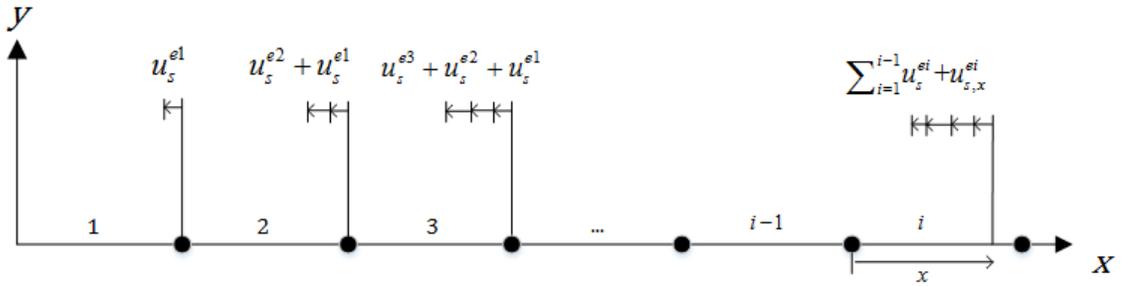


Figure 3.11: Accumulated shortenings of the elements

of an arbitrary point on the neutral axis of the i^{th} element is defined as

$$u_s^i = \sum_{j=1}^{i-1} u_s^{ej} + u_{s,x}^{ei} = \mathbf{B}_s^i \mathbf{q}_f \quad (3.105)$$

where the matrix \mathbf{B}_s^i is defined as

$$\mathbf{B}_s^i = -\mathbf{q}_f^T \mathbf{H}^i \quad (3.106)$$

where \mathbf{H}^i is a symmetric matrix and a function of the spatial coordinates defined as

$$\mathbf{H}^i = -\frac{1}{2} \left(\sum_{n=1}^{i-1} \int_0^l \mathbf{N}_y^{nT'} \mathbf{N}_y^{n'} dx + \int_0^x \mathbf{N}_y^{iT'} \mathbf{N}_y^{i'} dx \right) \quad (3.107)$$

Considering three-dimensional cases with bending deflection existing in both y and z directions, the matrix \mathbf{H}^i in Eq. (3.107) is modified for three-dimensional beams as

$$\mathbf{H}^i = -\frac{1}{2} \left[\sum_{n=1}^{i-1} \int_0^l (\mathbf{N}_y^{nT'} \mathbf{N}_y^{n'} + \mathbf{N}_z^{nT'} \mathbf{N}_z^{n'}) dx + \int_0^x (\mathbf{N}_y^{iT'} \mathbf{N}_y^{i'} + \mathbf{N}_z^{iT'} \mathbf{N}_z^{i'}) dx \right] \quad (3.108)$$

where \mathbf{N}_z^i is defined in Eq. (3.65). After obtaining the general foreshortening equation, the modified version of the displacement vector defined in Eq. (3.26) can be calculated as follows

$$\begin{aligned} \mathbf{u}^i &= \mathbf{S}^i \mathbf{B}_1^i \mathbf{q}_n + \begin{Bmatrix} u_s^i \\ 0 \\ 0 \end{Bmatrix} \\ &= \mathbf{S}^i \mathbf{B}_1^i (\mathbf{q}_0 + \mathbf{B}_2 \mathbf{q}_f) + \begin{Bmatrix} \mathbf{B}_s^i \mathbf{q}_f \\ 0 \\ 0 \end{Bmatrix} \\ &= \mathbf{S}^i \mathbf{B}_1^i \mathbf{q}_0 + (\mathbf{N}^i + \mathbf{B}_H^i) \mathbf{q}_f \end{aligned} \quad (3.109)$$

where \mathbf{B}_H^i is defined as

$$\mathbf{B}_H^i = \begin{bmatrix} \mathbf{B}_s^i & \mathbf{0} & \mathbf{0} \end{bmatrix}^T \quad (3.110)$$

which has the same dimension of \mathbf{N}^i defined as

$$\mathbf{N}^i = \begin{bmatrix} \mathbf{N}_x^i & \mathbf{N}_y^i & \mathbf{N}_z^i \end{bmatrix}^T \quad (3.111)$$

where \mathbf{N}_x^i is defined in Eq. (3.50).

Having the modified displacement vector, the global position vector of an arbitrary point of the flexible beam defined in Eq. (3.30) is re-defined as

$$\mathbf{r}^i = \mathbf{R} + \mathbf{A}\mathbf{u}^i = \mathbf{R} + \mathbf{A} [\mathbf{S}^i \mathbf{B}_1^i \mathbf{q}_0 + (\mathbf{N}^i + \mathbf{B}_H^i) \mathbf{q}_f] \quad (3.112)$$

Notice that the matrix \mathbf{B}_H^i is highly nonlinear, which is different from \mathbf{N}^i that is only dependent on the time-invariant spatial coordinates, because it is a function of both spatial coordinates and the time-variant nodal deformation coordinates \mathbf{q}_f .

3.2.2 Modifications of the Mass Matrix and Force Vectors

The steps to derive the mass matrix, stiffness matrix, and other necessary force vectors are identical as those of the conventional linear formulation developed in the previous chapters; one only needs to correctly write the modified global velocity vector $\dot{\mathbf{r}}^i$, acceleration vector $\ddot{\mathbf{r}}^i$, and the virtual change of the global position vector $\delta \mathbf{r}^i$ first.

The modified global velocity vector can be obtained by differentiating the global position vector defined in Eq. (3.112) with respect to time as

$$\dot{\mathbf{r}}^i = \dot{\mathbf{R}} - \mathbf{A} \tilde{\mathbf{u}}^i \boldsymbol{\omega} + \mathbf{A} (\mathbf{N}^i + 2\mathbf{B}_H^i) \dot{\mathbf{q}}_f \quad (3.113)$$

where $\tilde{\mathbf{u}}^i$ is the skew symmetric matrix of the local displacement vector defined in Eq. (3.112). The symmetry of the matrix \mathbf{H}^i is used during the differentiation of $\mathbf{B}_H^i \mathbf{q}_f$ with

respect to time as follows:

$$\begin{aligned}
\frac{d(\mathbf{B}_H^i \mathbf{q}_f)}{dt} &= \dot{\mathbf{B}}_H^i \mathbf{q}_f + \mathbf{B}_H^i \dot{\mathbf{q}}_f \\
&= \left[\dot{\mathbf{B}}_s^i \mathbf{q}_f \quad 0 \quad 0 \right]^T + \mathbf{B}_H^i \dot{\mathbf{q}}_f \\
&= \left[-\dot{\mathbf{q}}_f^T \mathbf{H}^i \mathbf{q}_f \quad 0 \quad 0 \right]^T + \mathbf{B}_H^i \dot{\mathbf{q}}_f \\
&= \left[-(\dot{\mathbf{q}}_f^T \mathbf{H}^i \mathbf{q}_f)^T \quad 0 \quad 0 \right]^T + \mathbf{B}_H^i \dot{\mathbf{q}}_f \\
&= \left[-\mathbf{q}_f^T \mathbf{H}^i \dot{\mathbf{q}}_f \quad 0 \quad 0 \right]^T + \mathbf{B}_H^i \dot{\mathbf{q}}_f \\
&= 2\mathbf{B}_H^i \dot{\mathbf{q}}_f
\end{aligned} \tag{3.114}$$

The global acceleration vector can be derived by differentiating the global velocity vector defined in Eq. (3.114) as

$$\ddot{\mathbf{r}}^i = \ddot{\mathbf{R}} - \mathbf{A} \ddot{\mathbf{u}}^i \dot{\omega} + \underline{\mathbf{A} \tilde{\omega}^2 \mathbf{u}^i} + \underline{2\mathbf{A} \tilde{\omega} (\mathbf{N}^i + 2\mathbf{B}_H^i) \dot{\mathbf{q}}_f} + \underline{\mathbf{A} \dot{\mathbf{B}}_H^i \dot{\mathbf{q}}_f} + \mathbf{A} (\mathbf{N}^i + 2\mathbf{B}_H^i) \ddot{\mathbf{q}}_f \tag{3.115}$$

where the underlined parts are used to derive the quadratic velocity force vector with the same reason explained for Eq. (3.85), and $\dot{\mathbf{B}}_H^i$ is calculated as

$$\dot{\mathbf{B}}_H^i = \left[\dot{\mathbf{q}}_f^T \mathbf{H}^i \quad 0 \quad 0 \right]^T \tag{3.116}$$

The virtual change of the position vector is similar to the global velocity vector, which is defined as

$$\delta \mathbf{r}_i = \delta \mathbf{R} - \mathbf{A} \tilde{\mathbf{u}}^i \delta \omega + \mathbf{A} (\mathbf{N}^i + 2\mathbf{B}_H^i) \delta \mathbf{q}_f \tag{3.117}$$

Using the way to derive the mass matrix described in Chapter 3.1.5 by defining the kinetic energy, all sub-terms in the mass matrix defined in Eq. (3.40) are changed except for \mathbf{m}_{rr}^i .

The geometrically nonlinear mass matrix is defined as

$$\mathbf{M}_{GN}^i = \int_0^{l^i} \rho^i \begin{bmatrix} \mathbf{I} & -\mathbf{A} \tilde{\mathbf{u}}^i & \mathbf{A} \mathbf{N}_{GN}^i \\ & \tilde{\mathbf{u}}^{iT} \tilde{\mathbf{u}}^i & \tilde{\mathbf{u}}^i \mathbf{N}_{GN}^i \\ \text{sym} & & \mathbf{N}_{GN}^{iT} \mathbf{N}_{GN}^i \end{bmatrix} A^i dx \tag{3.118}$$

where the subscript GN refers to geometrically nonlinear formulation and \mathbf{N}_{GN}^i is defined as

$$\mathbf{N}_{GN}^i = \mathbf{N}^i + 2\mathbf{B}_H^i \quad (3.119)$$

Although some of the sub-terms in \mathbf{M}_{GN}^i , which are only dependent on the displacement vector \mathbf{u}^i , maintain the same forms as the linear ones, one must pay attention that the newly-defined nonlinear displacement vector in Eq. (3.112) makes them internally different from before.

Following the same steps to derive the quadratic velocity force, generalized gravitational force, and generalized external force vectors described through Chapter 3.1.8-3.1.10, respectively, these force vectors are modified as

$$\mathbf{Q}_{v,NG} = \int_0^{l^i} \rho A^i \begin{bmatrix} \mathbf{A}\tilde{\omega}^2\mathbf{u}^i + 2\mathbf{A}\tilde{\omega}\mathbf{N}_{NG}^i\dot{\mathbf{q}}_f + \mathbf{A}\dot{\mathbf{B}}_H^i\dot{\mathbf{q}}_f \\ -\tilde{\mathbf{u}}^{iT}\tilde{\omega}^2\mathbf{u}^i - 2\tilde{\mathbf{u}}^{iT}\tilde{\omega}\mathbf{N}_{NG}^i\dot{\mathbf{q}}_f - \tilde{\mathbf{u}}^{iT}\dot{\mathbf{B}}_H^i\dot{\mathbf{q}}_f \\ \mathbf{N}_{GN}^{iT}(\tilde{\omega}^2\mathbf{u}^i + 2\tilde{\omega}\mathbf{N}_{NG}^i\dot{\mathbf{q}}_f + \dot{\mathbf{B}}_H^i\dot{\mathbf{q}}_f) \end{bmatrix} dx \quad (3.120)$$

$$\mathbf{F}_{g,NG} = \int_0^{l^i} \rho A^i \begin{bmatrix} \mathbf{g}^T & -\mathbf{g}^T\mathbf{A}\tilde{\mathbf{u}}^i & \mathbf{g}^T\mathbf{A}\mathbf{N}_{NG}^i \end{bmatrix}^T dx \quad (3.121)$$

$$\mathbf{F}_{e,NG} = \begin{bmatrix} \mathbf{P}^T & -\mathbf{P}^T\mathbf{A}\tilde{\mathbf{u}}^p & \mathbf{P}^T\mathbf{A}\mathbf{N}_{NG}^{i,P} \end{bmatrix}^T \quad (3.122)$$

Among these force vectors, the additional terms inside the quadratic velocity force vector are the most significant to let the beam show exact dynamic stiffening effect. It means that even if the mass matrix and other force vectors maintain their linear forms, only making the quadratic velocity force vector geometrically nonlinear would still lead to an accurate dynamic stiffening result. Discussion about the significance of these terms is provided in the example problems in Chapter 5.

3.2.3 Unchanged Stiffness Matrix

The benefit of the foreshortening method is the unchanged stiffness matrix even though the nonlinear global position vector has brought complexity into the mass matrix and force

vectors. It is proved in this section by checking whether the strain energy defined in Eq. (3.81) keeps the same form. As the longitudinal deformation is added by a foreshortening term shown in Eq. (3.99), the differentiation of the deformation variables defined in Eq. (3.1) with respect to x are modified as

$$\left. \begin{aligned} u'_{foreshort} &= u' - \frac{1}{2}(v'_y{}^2 + v'_z{}^2) \\ v'_y &= v'_y \\ v'_z &= v'_z \end{aligned} \right\} \quad (3.123)$$

where the differentiation of the bending deformation variables is not changed. Substituting Eq. (3.123) into the strain energy defined in Eq. (3.81), which is just replacing u' with $u'_{foreshort}$, one can show that the coupled and high-order terms are cancelled out with each other during derivation and the final form is calculated as

$$U^i_{foreshorten} = -\frac{1}{2} \int_V E(u'^2 + y^2 v''_y{}^2 + z^2 v''_z{}^2) dV \quad (3.124)$$

It is evident that the updated strain energy keeps exactly the same form as the linear one in Eq. (3.82). Therefore, the stiffness matrix defined for the conventional Bernoulli-Euler beam in Eq. (3.75) can still be used for the geometrically nonlinear beam without causing any trouble. One must understand that the nonlinear stiffness matrix derived from Eq. (3.124), though it has the same form as the linear one defined in Eq. (3.82), cannot be seen as obtained from linear superposition because the coupled and high-order terms in the full strain energy expression of Eq. (3.81) are cancelled out themselves instead of intentionally truncated during the derivation presented in Section 3.1.7. Moreover, since there is no intentional truncation, the stiffness matrix generated from Eq. (3.124) should have exact accuracy. This advantage will also help to save computational time during both equation preparation and calculation stages.

4 Finite Element Model in Symbolic Multibody System Dynamics

In this chapter, the formulation of a finite set of governing dynamic equations of motion of multibody systems that contain interconnected deformable and rigid bodies will be presented. Generally, the dynamic equations of motion of rigid bodies in multibody systems can be defined in terms of the mass of the body, the inertia tensor, and the generalized forces acting on the body. However, the dynamic equations of motion of structural systems require the definitions of the system mass and inertia matrices, stiffness matrices, as well as the vectors of generalized forces. In Chapter 3, these quantities for a flexible beam are developed by using the finite element method, and they will be used to construct the dynamic governing equations of deformable beams that undergo large translational and rotational motions.

The classical way of simulating multibody system dynamics in a computer program is to generate and solve the system dynamic governing equations in a purely numerical manner, as opposed to a symbolic computation of multibody system dynamics. The generation and solution of the governing equations can be regarded as the modelling and analysis phases described in Section 2.3. What really sets the symbolic computation apart is that it makes the modelling phase much more convenient even though its analysis phase unavoidably relies on numerical solvers. Unlike numerical computation that requires rebuilding of the model to obtain a new set of governing equations at each timestep, or when the system changes one or a few features such as body geometries or connections, symbolic computation is able to recall the equations pre-defined for each body and joint from the library and combine them automatically for every specifically-defined multibody system during the modelling phase. There is no need, for example, to go back to the modelling phase and re-construct a flexible body if its parameters have changed, since the equations of motion contain only the symbols of the parameters that can be given any numerical value before the analysis phase.

In order to integrate the finite-element flexible body with symbolic multibody system dynamics, one only needs to know how to convert the finite-element terms into the equations that can be called to configure any related systems and save them in the symbolic multibody systems library. The conversion will be introduced after a discussion of the general governing equations of motion.

4.1 Governing Equations of Motion

The kinematic constraints that describe mechanical joints as well as specified motions of a body can be formulated by using a set of nonlinear algebraic constraint equations

$$\mathbf{C}(\mathbf{q}, t) = \begin{bmatrix} C_1 & C_2 & \dots & C_{n_c} \end{bmatrix}^T = \mathbf{0} \quad (4.1)$$

where C_i are the linearly independent constraint equations, t is time, and $\mathbf{q} = \begin{bmatrix} \mathbf{q}_r & \mathbf{q}_f \end{bmatrix}^T$, of which the subscripts r and f refer, respectively, to the rigid and flexible coordinates.

Having determined the constraint equations, the kinetic energy of the flexible beam, and the virtual work of the internal and external forces, one can use Lagrange's equation [1] to write the equations of motion of the flexible body in the multibody system. To this end, the virtual work of the forces acting on the flexible beam is

$$\delta W = \delta W_i + \delta W_e \quad (4.2)$$

where δW is the total virtual work of all forces, δW_i is the virtual work of the elastic forces resulting from the deformation of the beam, and δW_e is the virtual work due to the externally applied forces, which may have various types of external loads and gravity effect depending on specific circumstances. Using the stiffness matrix \mathbf{K}_f defined in the preceding chapter, the virtual work of the elastic forces can be written as

$$\delta W_i = -\mathbf{q}^T \mathbf{K} \delta \mathbf{q} \quad (4.3)$$

where \mathbf{K} is defined as

$$\mathbf{K} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f \end{bmatrix} \quad (4.4)$$

In general, the virtual work of the externally applied forces can be written as

$$\delta W_e = \mathbf{F}_t \delta \mathbf{q} \quad (4.5)$$

where \mathbf{F}_t is the vector of the total generalized externally applied forces consisting of the generalized external force vector \mathbf{F}_e and the generalized gravitational force vector \mathbf{F}_g defined previously. Combining Eq. (4.2)-(4.5) leads to

$$\delta W = \delta \mathbf{q} (-\mathbf{K} \mathbf{q} + \mathbf{F}_t) \quad (4.6)$$

For a flexible body in the multibody system, Lagrange's equation takes the form

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\mathbf{q}}} \right)^T - \left(\frac{\partial T}{\partial \mathbf{q}} \right)^T + \mathbf{C}_q^T \boldsymbol{\lambda} = -\mathbf{K} \mathbf{q} + \mathbf{F}_t \quad (4.7)$$

where T is the kinetic energy of the body, \mathbf{C}_q is the constraint Jacobian matrix, and $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. Using the definition of the kinetic energy of Eq. (3.38), the first two terms on the left-hand side of Eq. (4.7) yield

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\mathbf{q}}} \right)^T - \left(\frac{\partial T}{\partial \mathbf{q}} \right)^T = \mathbf{M} \ddot{\mathbf{q}} + \dot{\mathbf{M}} \dot{\mathbf{q}} - \left[\frac{\partial}{\partial \mathbf{q}} \left(\frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} \right) \right]^T \quad (4.8)$$

where \mathbf{M} is the mass matrix defined in the preceding chapter. One can prove the equivalence [1]

$$\mathbf{Q}_v = \dot{\mathbf{M}} \dot{\mathbf{q}} - \left[\frac{\partial}{\partial \mathbf{q}} \left(\frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} \right) \right]^T \quad (4.9)$$

where \mathbf{Q}_v is the quadratic velocity force vector derived in the preceding chapter using the virtual work method. Using Eq. (4.7)-(4.9), the Lagrange's equation yields

$$\mathbf{M} \ddot{\mathbf{q}} + \mathbf{C}_q^T \boldsymbol{\lambda} = -\mathbf{K} \mathbf{q} + \mathbf{F}_t + \mathbf{Q}_v \quad (4.10)$$

For a flexible beam with $\mathbf{q} = \begin{bmatrix} \mathbf{R} & \boldsymbol{\theta} & \mathbf{q}_f \end{bmatrix}^T$, Eq. (4.10) can be written in a partitioned matrix form as

$$\begin{aligned}
& \begin{bmatrix} \mathbf{m}_{rr} & \mathbf{m}_{r\theta} & \mathbf{m}_{rf} \\ & \mathbf{m}_{\theta\theta} & \mathbf{m}_{\theta f} \\ \text{sym} & & \mathbf{m}_{ff} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{R}} \\ \dot{\boldsymbol{\omega}} \\ \ddot{\mathbf{q}}_f \end{Bmatrix} + \begin{bmatrix} \mathbf{C}_{\mathbf{R}}^T \\ \mathbf{C}_{\boldsymbol{\theta}}^T \\ \mathbf{C}_{\mathbf{q}_f}^T \end{bmatrix} \boldsymbol{\lambda} \\
= & \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & \mathbf{0} & \mathbf{0} \\ \text{sym} & & -\mathbf{K}_f \end{bmatrix} \begin{Bmatrix} \mathbf{R} \\ \boldsymbol{\theta} \\ \mathbf{q}_f \end{Bmatrix} + \begin{bmatrix} (\mathbf{F}_t)_r \\ (\mathbf{F}_t)_\theta \\ (\mathbf{F}_t)_f \end{bmatrix} + \begin{bmatrix} (\mathbf{Q}_v)_r \\ (\mathbf{Q}_v)_\theta \\ (\mathbf{Q}_v)_f \end{bmatrix}
\end{aligned} \tag{4.11}$$

where \mathbf{R} is the global position vector pointing from the inertia frame of reference to the body frame of reference, and $\boldsymbol{\omega}$ is the angular velocity vector defined with respect to the body frame of reference. All of the force terms on the right-hand side of Eq. (4.11) can be generalized as a combined force vector with a symbol \mathbf{Q} .

Eq. (4.10) is a set of second-order differential equations whose solution has to satisfy the algebraic constraint equations of Eq. (4.1). Both sets of equations are combined to become a set of differential algebraic equations that have to be solved simultaneously. In large-scale multibody systems, a closed-form solution of such set of equations becomes hard to obtain, so it is necessary to rely on numerical solvers.

For a general multibody system consisting of interconnected rigid and deformable components shown in Fig. 4.1, each of which satisfies the equations of motion of Eq. (4.10) except that rigid bodies do not have the stiffness term, the construction of the entire set of governing differential algebraic equations follows a basic assembly of all of the component-level

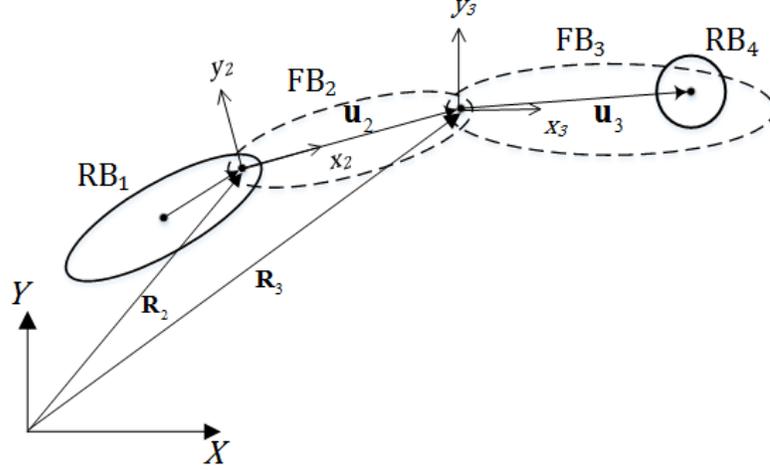


Figure 4.1: Multibody systems of interconnected rigid and flexible bodies (RB and FB refer to rigid body and flexible body). Note: the body frame of reference of a rigid body is located on the centre of mass.)

equations of motion as

$$\left. \begin{aligned} & \begin{bmatrix} \mathbf{M}_1 & \mathbf{0} & \mathbf{0} & \mathbf{C}_{\mathbf{q}_1}^T \\ \mathbf{0} & \ddots & \mathbf{0} & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_n & \mathbf{C}_{\mathbf{q}_n}^T \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}}_1 \\ \vdots \\ \ddot{\mathbf{q}}_n \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \mathbf{Q}_1 \\ \vdots \\ \mathbf{Q}_n \end{Bmatrix} \\ & \mathbf{C}(\mathbf{q}, t) = \mathbf{0} \end{aligned} \right\} \quad (4.12)$$

where \mathbf{M}_i , \mathbf{Q}_i , and \mathbf{q}_i ($i = 1, 2, \dots, n$) represent the mass matrix, generalized combined force vector, and the coordinates of the i^{th} body. Notice that \mathbf{Q}_i of rigid bodies do not have the elastic force vector, and $\mathbf{C}_{\mathbf{q}_i}^T$ is the global Jacobian matrix partially differentiated from the global constraint equations $\mathbf{C}(\mathbf{q}, t)$. Eq. (4.12) can be written in a simplified form as

$$\left. \begin{aligned} & \mathbf{M}^* \ddot{\mathbf{q}} + \mathbf{C}_{\mathbf{q}}^T \lambda = \mathbf{Q}^* \\ & \mathbf{C}(\mathbf{q}, t) = \mathbf{0} \end{aligned} \right\} \quad (4.13)$$

where the star superscript means system-level matrix and vector.

The global position vector, $\mathbf{r} = \mathbf{R} + \mathbf{A}\mathbf{u}$, of an arbitrary point on a flexible beam defined in the preceding chapter is used with the position vectors of rigid bodies to derive the global

constraint equations $\mathbf{C}(\mathbf{q}, t)$. The centre of mass of a rigid body is often where the body frame of reference is located while a flexible body will follow the rule of floating frame of reference [1], as shown in Fig. 4.1, and the inertia is defined with respect to it. For a flexible beam, the body frame of reference is often fixed to one of its tip-end with x-axis in-line with the neutral axis of the beam.

4.2 Implementation into Library of Symbolic Multibody Systems

The implementation of the finite element flexible body into the symbolic multibody system dynamics is achieved by using MapleSim, which is a multi-domain physical modelling environment built on Maple’s technology that provides users with a general-purpose symbolic computation environment. The implemented finite-element beam model in MapleSim is referred to as “MapleSim FEM” in Chapter 5. Having been integrated with DynaFlex Pro [54], MapleSim has its own multibody systems dynamics library from where users can select different types of multibody components (bodies, drivers, joints, etc.) and configure all kinds of open- or closed-loop mechanisms. The modelling phase, which is mentioned in the preceding section, of multibody systems inside MapleSim is based on the technology of graph theory that adopts virtual work [4] to generate the global constraint equations and system governing equations of motion. In applying the graph-theory method to a system, each multibody component is mathematically modelled separately, which is independent of other system components, in the form of one or more terminal equations. The terminal equations basically contain the virtual work, generalized force and torque, and position vectors of the component. Moreover, each component is graphically represented by one or more graph edges between certain vertices or points of the interconnection of the component to other system components. Using graph theory, the result is a linear graph that can generate equations to describe the interconnection of all the system components. The terminal equations and graph edges associated with each component are used together to generate system equations in a manner similar to applying the principle of virtual work to flexible multibody system dynamics and formulating constraint equations, which is equivalent to traditional kinematic summations from a vector polygon.

For a flexible body component in multibody systems, what needs to be formulated are two sets of terminal equations with terminal edges associated with each. The first one is referred to as the body element that defines the virtual work and generalized forces and torques, and the second one is referred to as an “arm” that serves to locate any arbitrary point on the body where an imparted load (external load, joint reaction, etc.) is applied with respect to the body-fixed frame of reference. Following graph theory, the two sets of terminal equations and edges associated with the finite element flexible beam are presented by manipulating its definitions stated in the preceding chapter as follows.

4.2.1 Finite-element Beam Body Element

In practice, when defining the terminal equations for the body element of a flexible component, they are separated into two parts which are, respectively, corresponding to rigid-body mode and flexible-body mode since the total coordinates associated with a flexible body are divided as $\mathbf{q} = \begin{bmatrix} \mathbf{q}_r & \mathbf{q}_f \end{bmatrix}^T$. Whether the coordinates are used in the system equations depends on the tree and co-tree selection technique [19] used by the graph-theoretical approach. Dependent coordinates at joints can be eliminated using appropriate tree selection unless certain reaction results are needed. The Lagrange multipliers in the kinetic equations (4.13) depend on the co-tree selection. This will be discussed further in Section 4.3. The extreme condition is that there is no Lagrange multiplier in the kinetic equations, and the system governing equations have the form:

$$\mathbf{M}^* \ddot{\mathbf{q}} = \mathbf{Q}^* \quad (4.14)$$

where the total coordinates \mathbf{q} are independent, and there is no constraint equation.

Terminal Equations:

- Rigid-body Mode:

$$\mathbf{F}_{rb} = -(\mathbf{m}_{rr} \ddot{\mathbf{R}} + \mathbf{m}_{r\theta} \dot{\boldsymbol{\omega}} + \mathbf{m}_{rf} \ddot{\mathbf{q}}_f) + \mathbf{Q}_{gr} + \mathbf{Q}_{vr} \quad (4.15)$$

$$\mathbf{T}_{rb} = -(\mathbf{m}_{\theta r}\ddot{\mathbf{R}} + \mathbf{m}_{\theta\theta}\dot{\boldsymbol{\omega}} + \mathbf{m}_{\theta f}\ddot{\mathbf{q}}_f) + \mathbf{Q}_{g\theta} + \mathbf{Q}_{v\theta} \quad (4.16)$$

$$\delta W_{rb} = \mathbf{F}_{rb}^T \delta \mathbf{R} + \mathbf{T}_{rb}^T \delta \boldsymbol{\theta} \quad (4.17)$$

where the subscript rb refers to rigid-body mode, the \mathbf{m} terms are the sub-mass matrices of Eq. (4.11), and the \mathbf{Q}_g and \mathbf{Q}_v terms are, respectively, the components of the gravitational force vector of Eq. (3.92) and the quadratic velocity force vector of Eq. (3.87). Notice that the virtual work of the rigid-body mode does not include any virtual work caused by imparted loads to the body because it is taken care of by the existing force drivers and the dependent virtual work element [50].

- Flexible-body Mode:

In the virtual work of the flexible-body mode, however, the virtual work of imparted loads must be taken into account; otherwise, there is no existing element able to transfer the corresponding forces and torques into the generalized ones in terms of the flexible coordinates \mathbf{q}_f . The virtual work associated with the flexible-body mode is written as

$$\delta W_{fb} = -\delta \mathbf{R}^T \mathbf{m}_{fr}\ddot{\mathbf{R}} + \delta \boldsymbol{\theta}^T \mathbf{m}_{f\theta}\dot{\boldsymbol{\omega}} + \delta \mathbf{q}_f^T (-\mathbf{m}_{ff}\ddot{\mathbf{q}}_f - \mathbf{K}_f \mathbf{q}_f + \mathbf{Q}_{gf} + \mathbf{Q}_{vf} + \mathbf{Q}_{ef}) \quad (4.18)$$

where the subscript fb refers to flexible-body mode and all the other terms are defined in the preceding chapter. The terminal equation of the generalized force can be written as

$$\mathbf{F}_{fb} = -(\mathbf{m}_{fr}\ddot{\mathbf{R}} + \mathbf{m}_{f\theta}\dot{\boldsymbol{\omega}} + \mathbf{m}_{ff}\ddot{\mathbf{q}}_f) - \mathbf{K}_f \mathbf{q}_f + \mathbf{Q}_{gf} + \mathbf{Q}_{vf} + \mathbf{Q}_{ef} \quad (4.19)$$

Notice that the force \mathbf{P} defined in Eq. (3.94) that was used to derive \mathbf{Q}_{ef} can be either the external loads or reaction forces, which is automatically assigned by the force drivers and the dependent virtual work element.

Terminal Graph: same definition of the flexible body element [50], which is represented by the rigid-body coordinates \mathbf{R} in vector form.

4.2.2 Finite-element Beam Arm

Terminal Equations:

$$\mathbf{r}_P = \mathbf{u} \quad (4.20)$$

$$\dot{\mathbf{r}}_P = \boldsymbol{\omega} \times \mathbf{u} + \mathbf{N}^i \dot{\mathbf{q}}_f \quad (4.21)$$

$$\delta \mathbf{r}_P = \delta \boldsymbol{\theta} \times \mathbf{u} + \mathbf{N}^i \delta \mathbf{q}_f \quad (4.22)$$

$$\ddot{\mathbf{r}}_P = \dot{\boldsymbol{\omega}} \times \mathbf{u} + \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{u} + 2\boldsymbol{\omega} \times \mathbf{N}^i \dot{\mathbf{q}}_f + \mathbf{N}^i \ddot{\mathbf{q}}_f \quad (4.23)$$

$$\mathbf{A}_P: \text{rotation matrix by 1-2-3 sequence of } \boldsymbol{\theta}_P = \mathbf{N}^{i'} \mathbf{q}_f \quad (4.24)$$

$$\boldsymbol{\omega}_P = \mathbf{N}^{i'} \dot{\mathbf{q}}_f \quad (4.25)$$

$$\dot{\boldsymbol{\omega}}_P = \mathbf{N}^{i'} \ddot{\mathbf{q}}_f \quad (4.26)$$

$$\delta \boldsymbol{\theta}_P = \mathbf{N}^{i'} \delta \mathbf{q}_f \quad (4.27)$$

where the subscript P refers to the distal point P where the imparted action is located; \mathbf{u} is the local position vector of the point P defined by either Eq. (3.26) or Eq. (3.112) depending on whether the formulation is linear or nonlinear, which is used, for example, to represent \mathbf{u}_2 and \mathbf{u}_3 in Figure 4.1; $\boldsymbol{\omega}$ is the angular velocity vector with respect to the body-fixed frame; \mathbf{N}^i is defined in Eq. (3.111) or Eq. (3.119) depending on the formulation selection; \mathbf{A}_P is the rotation transformation matrix of the frame of reference whose origin is at the distal point P with respect to the body-fixed frame; $\boldsymbol{\omega}_P$ and $\dot{\boldsymbol{\omega}}_P$ are, respectively, the corresponding angular velocity and angular acceleration of the reference frame at the distal point P relative to the body-fixed frame. Since the deformation has been assumed small, the slopes of point P , estimated by $\boldsymbol{\theta}_P$, are used as Euler angles to evaluate the rotation matrix \mathbf{A}_P . For more accuracy, high-fidelity approaches of calculating the rotation matrix should be used [20].

Terminal Graph: same definition of the flexible arm element [50], which is represented by \mathbf{r}_P in vector form.

4.3 Illustrative Example Problem

To better understand the procedure of using the terminal equations defined for each component to assemble the system equations of motion, a simple example is presented in this section. Shown in Fig. 4.2 is a flexible beam rigidly attached to a revolute joint on the ground, and a point mass with negligible moment of inertia is rigidly attached to the free tip-end of the beam.

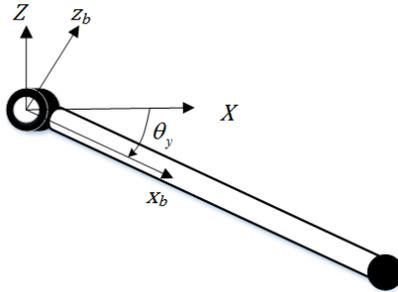


Figure 4.2: Pendulum flexible beam with tip point mass

As shown in Fig. 4.3, the linear graph of the system is drawn first, where edges h_4 and w_5 represent, respectively, the revolute joint and weld joint between the correspondingly connected components. Edge m_1 represents the beam body element in the rigid-body mode, and edge r_3 represents it in the flexible-body mode. Edge m_2 represents the point mass at the tip. A set of system equations, which are called “cutset” and “circuit” equations [19] in graph theory, will be derived into which the terminal equations are called and substituted.

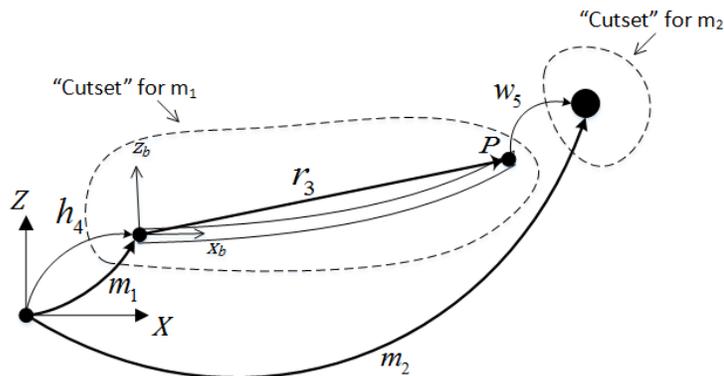


Figure 4.3: Linear graph of the pendulum flexible beam with tip point mass

The cutset equation can be simply looked at as the summation of the total virtual work of a component (or a “cutset”) being equal to zero, which also means that, cancelling the virtual change of coordinates $\delta\mathbf{q}$, the summations of the total forces and torques on a component are, respectively, equal to zero. Graph theory defines this step as projecting the equations onto the motion space, which is introduced later in this section. If edges m_1 , r_3 , and m_2 are selected in the tree (bold lines in Figure 4.3), the “cutset” equation for the flexible beam corresponding to the “cutset” for m_1 drawn in the graph can be written in terms of pre-defined forces and torques as

$$\mathbf{F}_{h_4} + \mathbf{F}_{m_1} - \mathbf{F}_{w_5} = 0 \quad (4.28)$$

$$\mathbf{T}_{h_4} + \mathbf{T}_{m_1} - \mathbf{T}_{w_5} = 0 \quad (4.29)$$

where the arrow pointing to the “cutset” leads to positive work and pointing out leads to negative work. Similarly, the “cutset” equation for the point mass corresponding to the “cutset” for m_2 can be written as

$$\mathbf{F}_{w_5} + \mathbf{F}_{m_2} = 0 \quad (4.30)$$

Notice that the point mass without moment of inertia does not have any torques associated with it. What is left is to call the terminal equations defined for the forces and torques and substitute them in Eq. (4.28)-(4.30). The \mathbf{F}_{m_1} and \mathbf{T}_{m_1} are defined in Eq. (4.15) & (4.16) for the flexible beam in the rigid-body mode. According to graph theory, the other terms are defined as follows:

$$\begin{aligned} \mathbf{F}_{h_4} &= \begin{bmatrix} F_{h_x} & F_{h_y} & F_{h_z} \end{bmatrix}^T \\ \mathbf{F}_{w_5} &= \begin{bmatrix} F_{w_x} & F_{w_y} & F_{w_z} \end{bmatrix}^T \\ \mathbf{T}_{h_4} &= \begin{bmatrix} T_{hx} & T_{hz} \end{bmatrix}^T \\ \mathbf{F}_{m_2} &= -m_s(\ddot{\mathbf{R}}_{m_2} - \mathbf{g}) \\ \mathbf{T}_{m_2} &= -\mathbf{I}_{m_s} \cdot \dot{\boldsymbol{\omega}}_{m_2} - \boldsymbol{\omega}_{m_2} \times \mathbf{I}_{m_s} \cdot \boldsymbol{\omega}_{m_2} - \mathbf{r}_{m_2} \times \mathbf{F}_{m_2} \\ \mathbf{T}_{w_5} &= -\mathbf{T}_{m_2} \end{aligned} \quad (4.31)$$

where the components in \mathbf{F}_{h_4} and \mathbf{F}_{w_5} are the reaction forces at the corresponding joints; \mathbf{T}_{h_4} contains the reaction torques at the revolte joint about X and Z axes shown in Figure 4.2; m_s is the mass of the point mass; $\ddot{\mathbf{R}}_{m_2}$ is the global acceleration vector of the point mass; \mathbf{I}_{m_s} represents the moment of inertia of the point mass, which is equal to zero; $\boldsymbol{\omega}_{m_2}$ is the angular velocity of the point mass; \mathbf{g} is the gravity vector; \mathbf{r}_{m_2} is the global position vector that starts from the inertia frame to the the weld joint and can be written, using the ‘‘circuit’’ equation, as

$$\mathbf{r}_{m_2} = \mathbf{R}_{m_1} + \mathbf{A}_{b/i}\mathbf{r}_3 \quad (4.32)$$

where \mathbf{R}_{m_1} is the global position vector of the body-fixed frame of the flexible beam which is a null vector in this case, and $\mathbf{A}_{b/i}$ is the rotation matrix defined by using θ_x , θ_y , and θ_z as Euler angles in a 1-2-3 sequence (θ_x , θ_y , and θ_z are the rotations about the X , Y , and Z axes in Figure 4.2); \mathbf{r}_3 is the arm terminal equation defined in Eq. (4.20) for the point P located at the beam tip end connecting the point mass shown in Figure 4.3. If one uses one element and adopts the conventional finite-element formulation to model the flexible beam, then the arm \mathbf{r}_3 and the nodal deformation coordinates \mathbf{q}_f are, respectively, defined as

$$\mathbf{r}_3 = \mathbf{N}^i(\mathbf{q}_0 + \mathbf{B}_2\mathbf{q}_f) \quad (4.33)$$

$$\mathbf{q}_f = \begin{bmatrix} q_{f1} & \dots & q_{f12} \end{bmatrix} \quad (4.34)$$

where \mathbf{q}_0 is defined in Eq. (3.27).

Substituting all of the terminal equations along with their derivations of Eq. (4.31)-(4.34) into the ‘‘cutset’’ equations of Eq. (4.28)-(4.30) and assembling with the flexible-mode terminal equation of Eq. (4.19) equal to zero ($\mathbf{F}_{fb} = \mathbf{0}$), the dynamic equations can be obtained in matrix form as identical to the first equation of Eq. (4.13) whose key terms are defined as follows:

$$\mathbf{M}^* = \begin{bmatrix} \mathbf{m}_{rr} & \mathbf{m}_{r\theta} & \mathbf{m}_{rf} & \mathbf{0} \\ \mathbf{m}_{\theta r} & \mathbf{m}_{\theta\theta} & \mathbf{m}_{\theta f} & \mathbf{0} \\ \mathbf{m}_{fr} & \mathbf{m}_{f\theta} & \mathbf{m}_{ff} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & m_s\mathbf{I} \end{bmatrix} \quad (4.35)$$

$$\mathbf{q} = \left[\mathbf{R}_{m_1} \quad \theta \quad \mathbf{q}_f \quad \mathbf{R}_{m_2} \right]^T \quad (4.36)$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{0}_6^6 & \mathbf{0} & \mathbf{0}_6^3 \\ \mathbf{0} & -\mathbf{K}_f & \mathbf{0} \\ \mathbf{0}_3^6 & \mathbf{0} & \mathbf{0}_3^3 \end{bmatrix} \mathbf{q} + \begin{bmatrix} \mathbf{Q}_g \\ \mathbf{0}_3^1 \end{bmatrix} + \begin{bmatrix} \mathbf{Q}_v \\ \mathbf{0}_3^1 \end{bmatrix} + \begin{bmatrix} \mathbf{0}_6^1 \\ \mathbf{Q}_{ef} \\ \mathbf{0}_3^1 \end{bmatrix} \quad (4.37)$$

$$\boldsymbol{\lambda} = \left[\mathbf{F}_{h_4} \quad \mathbf{T}_{h_4} \quad \mathbf{F}_{w_5} \right] \quad (4.38)$$

where 21 differential equations and 29 unknowns (DoFs) indicated in both \mathbf{q} and $\boldsymbol{\lambda}$ are combined; \mathbf{I} is 3×3 identity matrix; the subscripts and superscripts of the zero matrices indicate, respectively, the row and column dimensions. The elements of the Jacobian matrix $\mathbf{C}_{\mathbf{q}}$ can be found in Appendix A.2. Notice that, using the graph-theoretical approach, there is no additional step to differentiate the global constraint equations $\mathbf{C}(\mathbf{q}, t)$ with respect to \mathbf{q} to get $\mathbf{C}_{\mathbf{q}}$ since the term $\mathbf{C}_{\mathbf{q}}\boldsymbol{\lambda}$ automatically appears in the dynamic equation after substitution of terminal equations. The co-tree selection, which consists of edges h_4 and w_5 , results in the reactions in Eq. (4.38) associated with the revolute and weld joints appearing in the system equations.

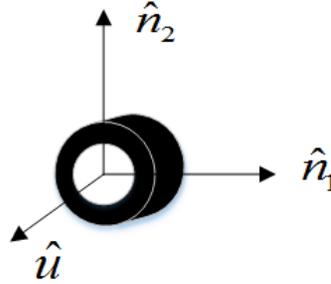


Figure 4.4: Axes of a revolute joint

There is one more step to use the “circuit” equation and the terminal equations of the finite-element arm to obtain $\mathbf{C}(\mathbf{q}, t)$ and assemble the whole system equations of motion. By graph-theoretical definition, the constraint equations are obtained by projecting the “circuit” equations of the co-tree joints, h_3 and w_5 in this problem, onto their reaction space [19]. The reaction space can be seen as the restricted translational and rotational directions of motion by the corresponding joint, and the other unrestricted motion direc-

tions compose the motion space. Taking the revolute joint in Figure 4.4 for example, its translational reaction space is defined by the translations along the joint axis \hat{u} , about which the joint rotates, and the two orthogonal axes \hat{n}_1 and \hat{n}_2 ; its rotational reaction space is defined by the restricted rotations about the two orthogonal axes \hat{n}_1 and \hat{n}_2 . In short, the “circuit” equation is similar to the traditional kinematic summations from vector polygon (multiplication of rotation matrices for three-dimensional rotations) to construct the constraint equations. The “circuit” equations for the h_3 and w_5 joints can be written as

$$\left. \begin{aligned} \mathbf{r}_h &= -\mathbf{r}_{m_1} \\ \mathbf{A}_h &= \mathbf{A}_{m_1}^T \\ \mathbf{r}_w &= -\mathbf{r}_3 - \mathbf{r}_{m_1} + \mathbf{r}_{m_2} \end{aligned} \right\} \quad (4.39)$$

Substituting the terminal equations defined for each term in Eq (4.39) yields

$$\left. \begin{aligned} \mathbf{r}_{h_4} &= -\mathbf{R}_{m_1} \\ \mathbf{A}_h &= \mathbf{A}_{b/i}^T \\ \mathbf{r}_{w_5} &= -\mathbf{r}_3 - \mathbf{R}_{m_1} + \mathbf{R}_{m_2} \end{aligned} \right\} \quad (4.40)$$

where \mathbf{r}_3 is defined in Eq. (4.33). By projecting each equation in Eq.(4.40) onto the reaction space of the corresponding joint, the 8 constraint equations $\mathbf{C}(\mathbf{q}, t)$ are obtained with the final expression:

$$\mathbf{C}(\mathbf{q}, t) = \begin{bmatrix} \mathbf{R} \\ \theta_x \\ \theta_z \\ \mathbf{R}_{m_s} - \mathbf{A}_{b/i} \mathbf{N}^i(\mathbf{q}_0 + \mathbf{B}_2 \mathbf{q}_f) \end{bmatrix} \quad (4.41)$$

According to the selection of expected results by the user, the implemented graph-theory technology can choose a different set of edges in the tree to eliminate unnecessary equations associated with dependent joint coordinates in the system equations. For this example problem, if the user does not expect any reactions to be shown in the results, MapleSim will select h_4 , r_3 , and w_5 in the tree and leave no joint edges in the co-tree. In this way, all coordinates in the system equations are independent and there is also no reaction $\boldsymbol{\lambda}$.

Therefore, the whole system equations will be in an ODE form as Eq. (4.14), which saves computational time.

5 Example Problems

In this chapter, a few example problems solved by the finite-element beam implemented in MapleSim are presented to validate the model. These examples include both planar and spatial multibody systems with flexible beam components. The results are compared with results from either the literature or with other codes or models. In addition, the performance of the finite-element models using symbolic computation is compared with that using numerical computation in terms of CPU time as well as complexity of practical manipulation, to further discuss the benefits of the symbolic computation. At the end, comparison of different features of the implemented finite-element beam and the built-in analytical beam in MapleSim is briefly described.

5.1 Cantilever Tapered Beam

In this section, the way of modelling a basic tapered beam and its deformation under cantilever condition is presented. The basic tapered beam has a tapered cross-sectional area and correspondingly varying second moment of area, but the material properties are assumed homogeneous along the entire beam. Therefore, modelling the tapered beam involves integrating the inertia and stiffness terms for each element with the cross-sectional area A^i and second moment of area $I_{y/z}^i$ to be a function of x instead of constant values.

Generally, if modelling a tapered cylindrical beam, for instance, it is the radius that is assumed to be linearly decreasing (or increasing) along the length of the beam, and thus the radius can be written as

$$r = a_1 + a_2x \quad (5.1)$$

where the radius r is represented by a linear polynomial function with coefficients a_i ($i = 1, 2$), which is dependent on the spatial coordinate x . Using circular area and radius relationship, the cross-sectional area can be represented by a second-order polynomial

function written as

$$A = \pi r^2 = a_3 + a_4x + a_5x^2 \quad (5.2)$$

In the same manner, the second moment of area about y - and z -axis can be expressed by a fourth-order polynomial function as

$$I_y = I_z = \frac{\pi r^4}{4} = a_6 + a_7x + a_8x^2 + a_9x^3 + a_{10}x^4 \quad (5.3)$$

Since all of these variables become functions of x , they can take part in the integrations along x . However, the integrations described in Chapter 3 are all element-level calculation, which integrates a term along an element length. This restricts the functions of (5.1)-(5.3) to be used directly since these functions are defined with respect to the entire beam. Only a minor modification is needed to allow these functions to be used directly in the integration on the element level, and the corresponding functions of the i^{th} element can be generally defined as

$$f^i(x^i) = f(x + L_{i-1}) \quad (5.4)$$

where $f(x)$ represents all the functions of Eq. (5.1)-(5.3), and L_{i-1} is the length accumulated from the first element to the element before the current one. If the beam is distributed by elements with equivalent length l , L_{i-1} can be defined as

$$L_{i-1} = l \cdot (i - 1) \quad (5.5)$$

A tapered beam with cross-sectional area varying from $0.005m^2$ to $0.001m^2$ is modelled, of which the radius is linearly decreasing along the beam. Accordingly, the cross-sectional area and the second moment of area defined with respect to the entire beam are written as follows:

$$\left. \begin{aligned} r &= 0.03984 - 0.007351x \\ A &= 0.005 - 1.8426 \cdot 10^{-3}x + 1.6976 \cdot 10^{-4}x^2 \\ I_y = I_z &= 1.9894 \cdot 10^{-6} - 1.4663 \cdot 10^{-6}x + 4.0528 \cdot 10^{-7}x^2 \\ &\quad - 4.9785 \cdot 10^{-8}x^3 + 2.2934 \cdot 10^{-9}x^4 \end{aligned} \right\} \quad (5.6)$$

Substituting the modified equations using Eq. (5.4) of Eq. (5.6) into the inertia, force, and stiffness integrals defined in Chapter 3 will yield the mass and stiffness matrices and force vectors defined for the tapered beam. The beam is subject to the cantilever condition under a constant 1000 N force toward $-z$ direction shown in Figure 5.1. The other properties are: length $L = 3\text{ m}$, mass density $\rho = 3000\text{ kg/m}^3$, Young's modulus $E = 7 \times 10^{10}\text{ N/m}^2$, and shear modulus $G = 2.7 \times 10^{10}\text{ N/m}^2$. There is no gravity associated with the beam, so the only focus is the deflection change due to the tapered section under the constant load.

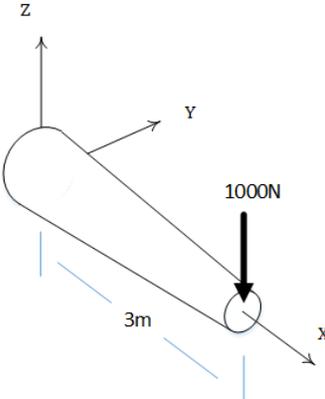


Figure 5.1: Cantilever tapered cylindrical beam

Five elements, which have been tested to satisfy convergence for this problem, are used to model both the tapered and uniform ($A = 0.005\text{ m}^2$) beams in MapleSim FEM. The tapered beam deflection result is validated by using ADAMS/Nastran. The deflection results compared with each other are shown in Figure 5.2.

The tapered beam deflection predicted by MapleSim FEM is almost exactly the same as the one predicted by ADAMS/Nastran with a root-mean-square error of 0.603%. One can notice that the tapered beam is so much softer than the uniform beam that leads to larger deflection, and the natural frequency has also changed due to different mass distribution. The latter one is critical in dynamic analysis, although static analysis often neglects this.

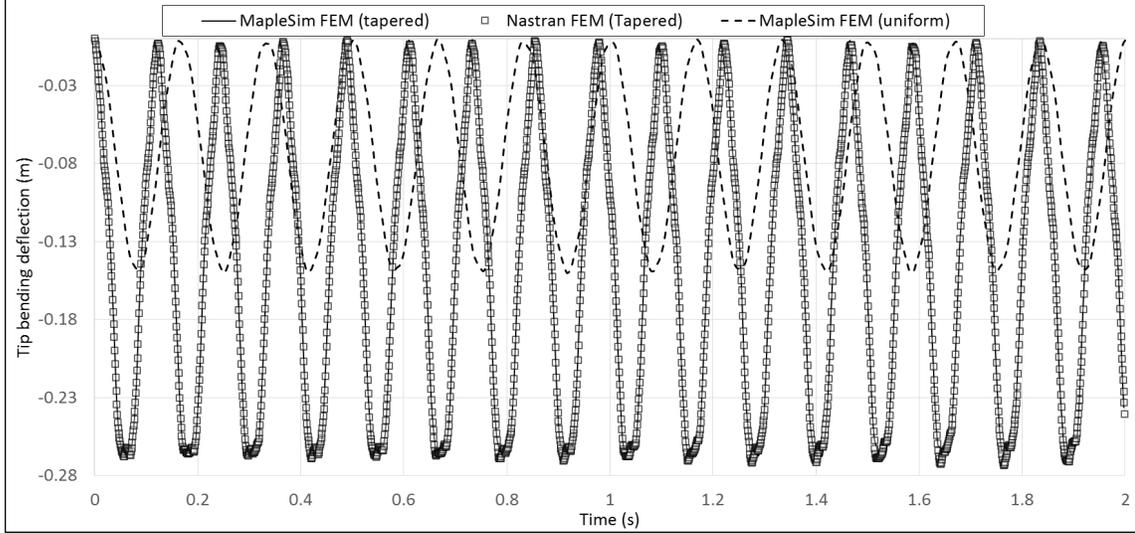


Figure 5.2: Tip deflections of the tapered and uniform beams

5.2 Spin-Up Beam

The planar rotating flexible beam is a benchmark problem for testing dynamic analysis of flexible bodies under large overall motion [30]. The very basic planar rotating beam problem is shown in Figure 5.3. The beam is rigidly attached to a rotating hub which is driven with an angular displacement $\theta(t)$ about the Y axis, and the gravity is assumed zero. The focus is the transverse deformation of the tip in the beam's body-fixed frame.

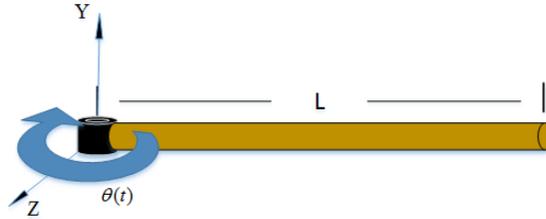


Figure 5.3: Planar spin-up beam

The geometric and material properties for the flexible beam are: length $L = 10m$, cross-sectional area $A = 0.0004m^2$, mass density $\rho = 3000kg/m^3$, second moment of area $I_y = I_z = 2 \times 10^{-7}m^4$, Young's modulus $E = 7 \times 10^{10}N/m^2$, and shear modulus $G = 2.7 \times$

$10^{10} N/m^2$, which are also used by [30]. The prescribed motion of the hub is also the same:

$$\theta(t) = \begin{cases} \frac{\omega_s}{T_s} \left[\frac{t^2}{2} + \left(\frac{T_s}{2\pi}\right)^2 (\cos(\frac{2\pi t}{T_s}) - 1) \right] & t < T_s \\ \omega_s (t - \frac{T_s}{2}) & t \geq T_s \end{cases} \quad (5.7)$$

where ω_s is set to 6.0 rad/s , and T_s to 15 s . Differentiating the prescribed angular displacement with respect to time, the angular speed is obtained and plotted in Figure 5.4. The rotation of the hub speeds up at the beginning and slows down until reaching steady motion after T_s , and then spins at the constant angular speed ω_s .

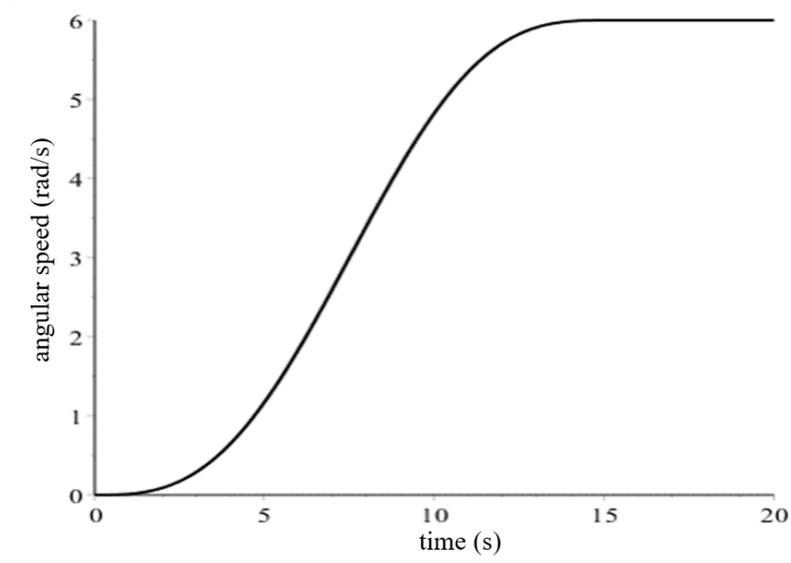


Figure 5.4: Angular speed corresponding to the prescribed angular motion in Eq. (5.7)

Five elements are used to model the beam in MapleSim FEM, and both conventional and geometrically nonlinear formulation are used separately to predict the deformation. The results are validated by using the analytical beam model implemented in MapleSim, which uses the nonlinear formulation presented in Shi et al. [20]. Using the classical variational method, the analytical beam uses Taylor polynomials as defined by Eq. (3.1) directly to derive the inertia and stiffness equations. The order of the polynomials for different types of deformation can be predefined by setting the “elastic coordinates”. Taking axial deformation for example, using one elastic coordinate indicates that the power is zero. The derivation of the analytical beam’s equations also involves order truncation, which aims

at eliminating high-order terms in the inertia and stiffness equations. In this example problem, the elastic coordinates used for axial, bending about two lateral directions, and twist are, respectively 1, 3, 3, 1; any term with order higher than two is truncated. The tip transverse deflection of the beam predicted by the three formulations is shown in Figure 5.5.

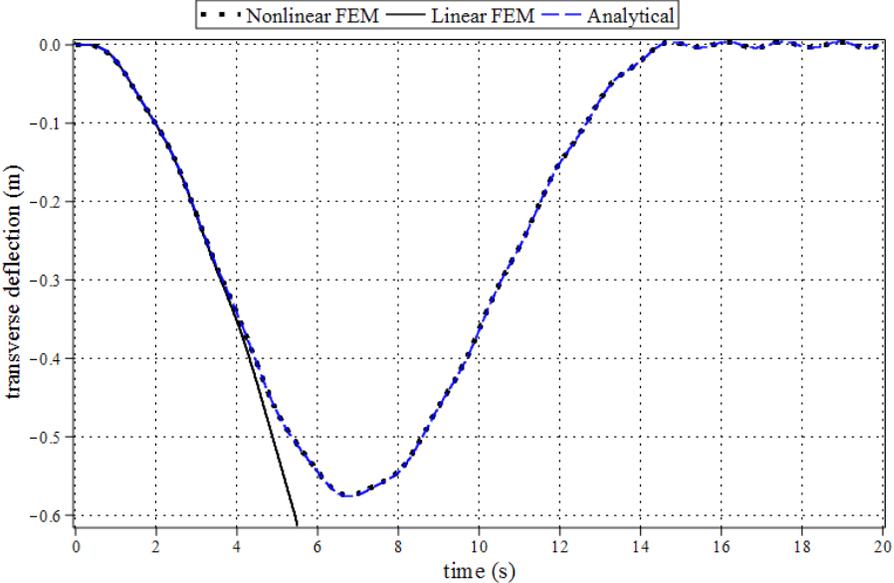


Figure 5.5: Tip deflection of the spin-up beam

The peak deflection magnitude of the nonlinear FEM (geometrically nonlinear formulation) is $0.573 m$, and the result is almost exactly the same as predicted by the analytical beam and the result from [33] with a root-mean-square error of around 0.05%. However, the linear FEM (conventional formulation) fails to predict the correct deformation after about $t = 4 s$. The reason for this is because the lack of nonlinearity inside the conventional FEM is unable to capture the dynamic stiffening effect [30] when the beam spins with a relatively fast angular speed.

For structural analysis in a steady state, or dynamic analysis with small and slow overall motion, or with stiffer flexible bodies, the dynamic stiffening that is largely associated with centrifugal and Coriolis effects is negligible. In this case, the linear functions inside the conventional formulation are sufficient to model the deformation of the body. The explanation can be verified by applying another set of properties to the beam, which makes the

beam stiffer while the motion is kept the same. The stiffer properties are: length $L = 3m$, cross-sectional area $A = 0.005m^2$, mass density $\rho = 3000kg/m^3$, second moment of area $I_y = I_z = 1.989 \times 10^{-6}m^4$, Young's modulus $E = 7 \times 10^{10}N/m^2$, and shear modulus $G = 2.7 \times 10^{10}N/m^2$. The tip transverse deflection of the stiffer beam predicted by the three formulations is shown in Figure 5.6.

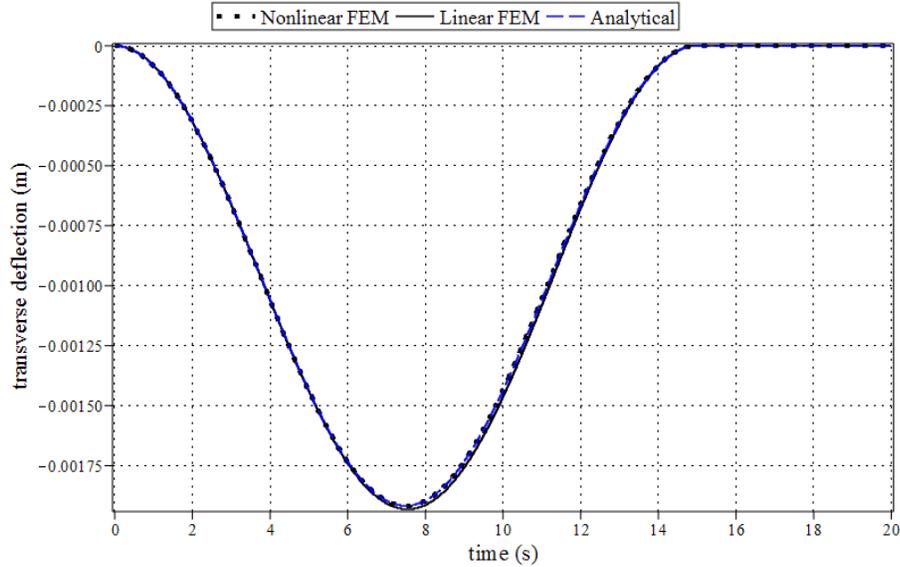


Figure 5.6: Tip deflection of the spin-up stiffer beam

Although slight differences exist between the linear FEM and the other two, they converge to each other very well since the dynamic stiffening effect becomes negligible for the stiffer beam.

In order to visualize the dynamic stiffening effect more clearly and extend the traditional benchmark spin-up beam example, another test of the spin-up beam has been set up by applying a different set of properties and prescribed angular motion while taking gravity into account. The newly defined properties are: length $L = 1m$, cross-sectional area $A = 7.853 \times 10^{-5}m^2$, mass density $\rho = 2700kg/m^3$, second moment of area $I_y/I_z = 4.909 \times 10^{-10}m^4$, Young's modulus $E = 7.2 \times 10^{10}N/m^2$, shear modulus $G = 2.7 \times 10^{10}N/m^2$, and gravity $g = 9.81m/s^2$. The prescribed angular motion is described in terms of angular speed shown in Figure 5.7. The beam is initially undeformed.

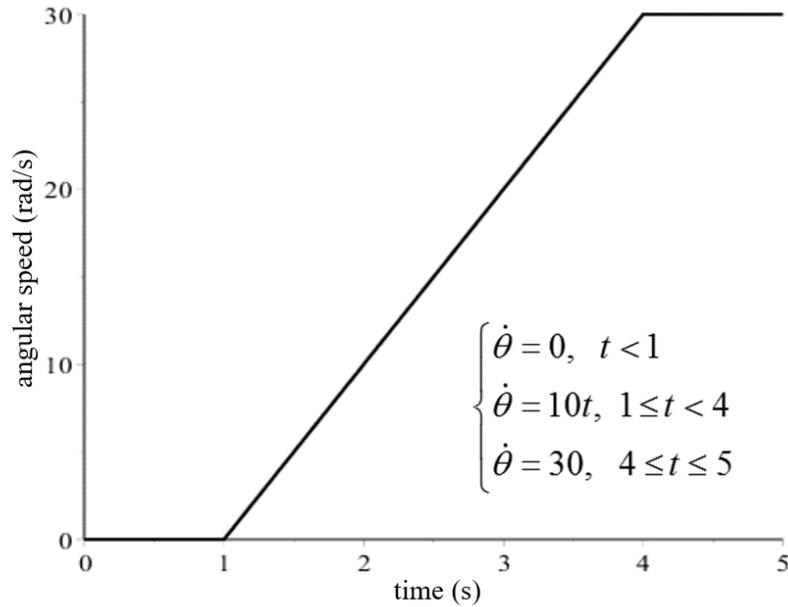


Figure 5.7: Newly-defined prescribed angular motion

The tip transverse deflection under the newly defined parameters and condition predicted by the three formulations is shown in Figure 5.8 and Figure 5.9. The former figure shows the horizontal deflection (in the plane of spinning), and the latter one shows the vertical deflection (out of the plane of spinning) along the direction of gravity. Looking at both figures, the deformation predicted by the three models starts to diverge at about $t = 2s$, and this is when the dynamic stiffening effect becomes significant as the angular speed reaches a critical value. It is mainly the Coriolis force that causes the horizontal deflection during spinning, and the Coriolis force gets larger when the angular speed increases. It can be clearly seen, from Figure 5.8, that the nonlinear beam models are able to resist bending by the increasing Coriolis force since the beam turns to be stiffer in practice during fast rotation. On the other hand, the deformation of the linear beam model subject to the Coriolis force is increasingly bigger until the angular speed becomes a constant value at $t = 4s$. Even after the angular speed becomes constant, the linear beam still has larger deformation than the other two “stiffer” nonlinear beams.

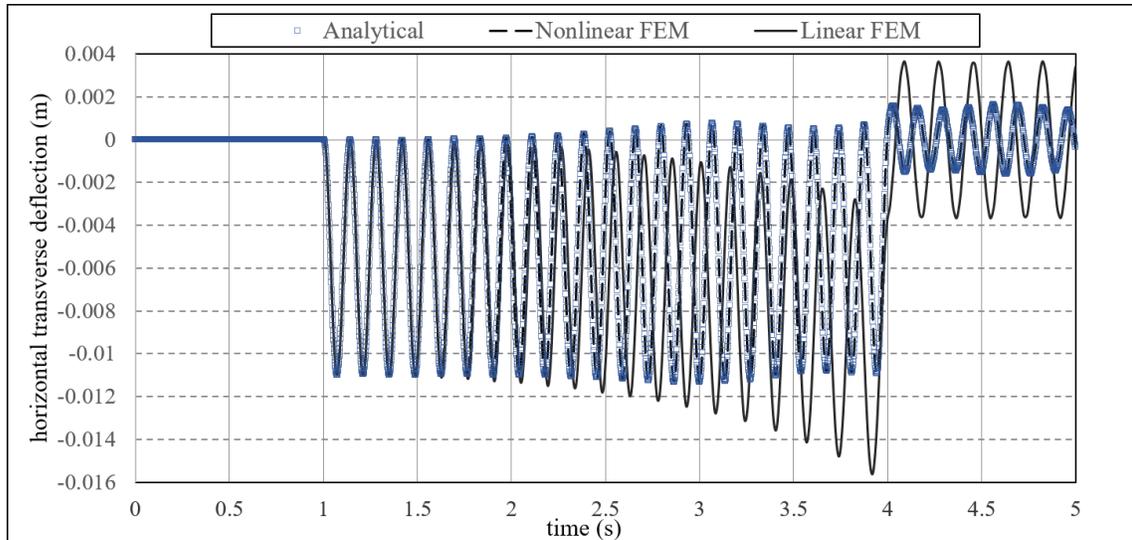


Figure 5.8: Tip in-plane deflection of the newly-defined spin-up beam

From Figure 5.9 one can see how the nonlinear beams resist gravity during fast spinning, which matches the physical behaviour in reality. Since the conventional formulation assumes decoupled deformation as stated in Chapter 3, its out-of-plane deformation is only caused by gravity so that it is always in the same pattern however the beam spins within the plane perpendicular to the gravity. The result is that the resistance to gravity due to the centrifugal effect, which is shown by the nonlinear beams, does not appear on the linear beam.

The CPU time of 4000 time steps corresponding to different number of elements is also recorded for the re-defined spin-up beam case. The CPU time can be divided into two categories, of which the first one is the time used for setting up the system equations (analysis time), and the second one is used for numerical time-integration (simulation time). The CPU time have been recorded for using 1 to 5 elements, which are shown in Table 5.1. The RMS errors of being compared with results of using 5 elements are given in the table. It is obvious that the computational cost associated with the nonlinear FEM increases dramatically with the number of elements while the cost with the linear FEM increases gently. However, the convergence for both FEM models in this case is reached at using 3 elements. Although one can use more than 3 elements to refine the results, it is efficient

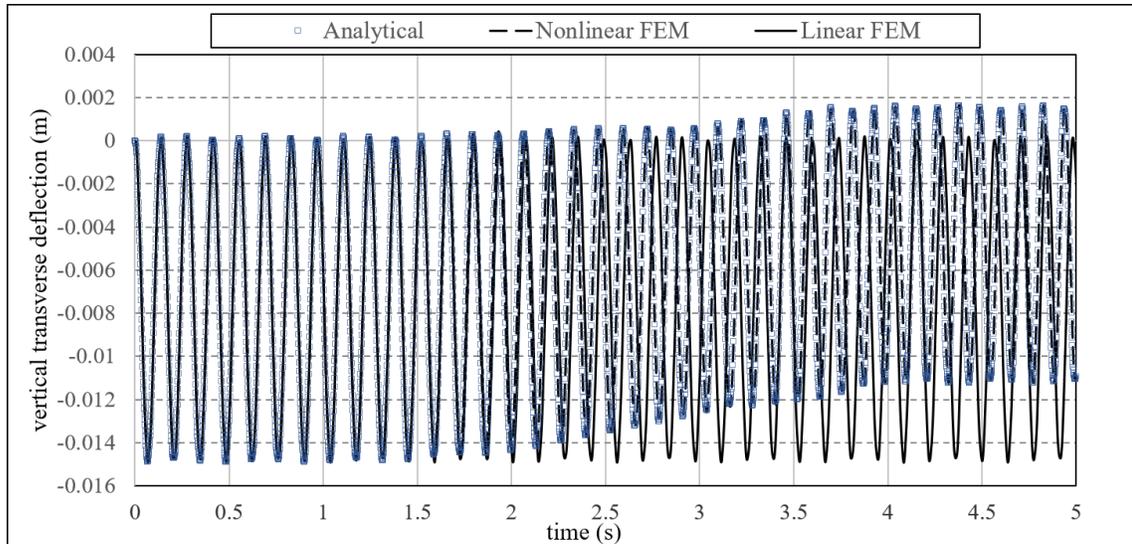


Figure 5.9: Tip out-of-plane deflection of the newly-defined spin-up beam

to use 2-3 elements to quickly get numerical results which are quite accurate. More CPU time information will be provided in the next few example problems for reference.

Table 5.1: CPU time corresponding to number of elements

	1	2	3	4	5
Linear FEM	(1.3; 0.55)	(2.9; 2.1)	(6.3; 7.1)	(10.5; 15.1)	(18; 30.1)
Nonlinear FEM	(6.1; 1.25)	(41; 7.2)	(105; 45.7)	(190; 146.3)	(308; 362)
RMS Error	0.21%	0.043%	0.039%	0.028%	-

Note: Values in the parentheses indicate (analysis time (s); simulation time (s))

5.3 Planar Slider-crank Mechanism with Flexible Crankshaft and Connecting Rod

This slider-crank mechanism example is used to test the functionality of the finite-element flexible beam with a closed-loop mechanism and to help further demonstrate the usage of the conventionally linear and the geometrically nonlinear formulations. This mechanism is taken from [10] as shown in Figure 5.10. The flexible crankshaft has a length of 0.152 m , a cross-sectional area of $7.854 \times 10^{-5}\text{ m}^2$, a second moment of area of $4.909 \times 10^{-10}\text{ m}^4$, a mass density of 2770 kg/m^3 , and a Yonge's modulus of $1.0 \times 10^9\text{ N/m}^2$. The connecting

rod has a length of 0.304 m , and has the same dimensions and material properties as the crankshaft with the exception of the Yonge's modulus, which is $0.5 \times 10^8\text{ N/m}^2$. The slider is assumed massless. The crankshaft of the system is assumed to be driven by the following torque expressed in N/m :

$$M(t) = \begin{cases} 0.01(1 - e^{-t/0.167}) & t < 0.70\text{ s} \\ 0 & t \geq 0.70\text{ s} \end{cases} \quad (5.8)$$

The crankshaft is modelled using one element, and the connecting rod is modelled using

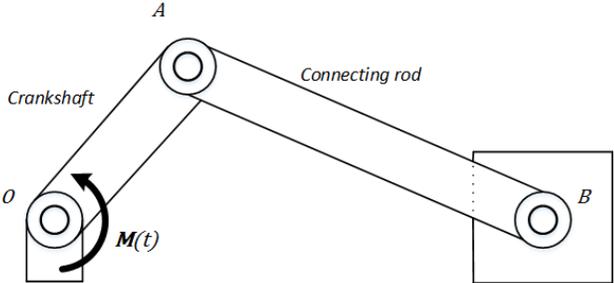


Figure 5.10: Slider-crank mechanism

two elements. Figure 5.11 shows the midpoint deformation results of the connecting rod measured with respect to its body-fixed frame obtained by using the two FEM formulations along with the result recorded in [10]. It is seen that the three sets of results agree well with each other. The nonlinear FEM generates a very small difference of the deformation because the connecting rod is quite short so that the centrifugal and Coriolis effects discussed in the preceding section are negligible. In addition, the difference between the result in [10] and the others is probably due to that the beam model in the reference adopts a modal approach. This difference will be discussed particularly in the next example problem.

The position of the slider block is plotted in Figure 5.12. It is shown that the influence of the slight difference of the deformation to the position of the slider block is trivial.

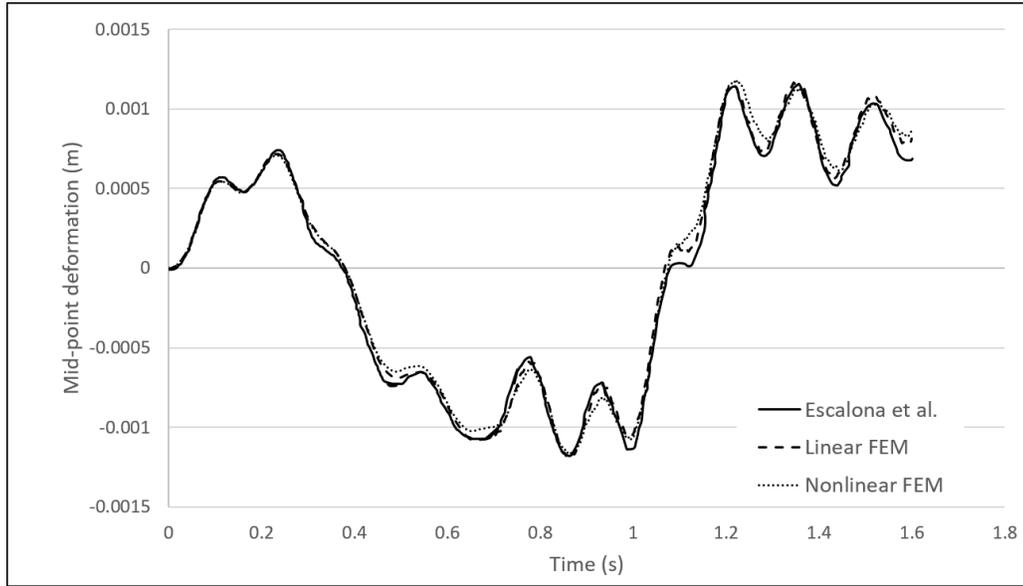


Figure 5.11: Deformation of the midpoint of the connecting rod

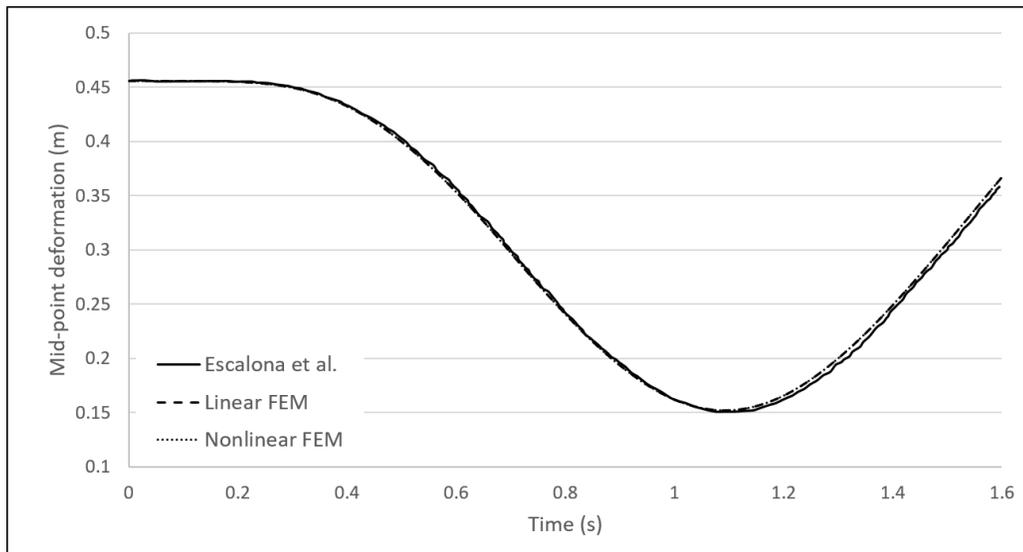


Figure 5.12: Position of the slider block

5.4 Spatial Flexible Manipulator

In this section, a spatial manipulator with a flexible link, shown in Figure 5.13, is modelled by the MapleSim nonlinear FEM and validated by the analytical beam model in MapleSim. The results are compared with that generated by ADAMS/Nastran which is based on a numerical modal approach. This example is used to present a comprehensive comparison in various aspects between the symbolic FEM (with nodal approach) and the numerical FEM with modal approach.

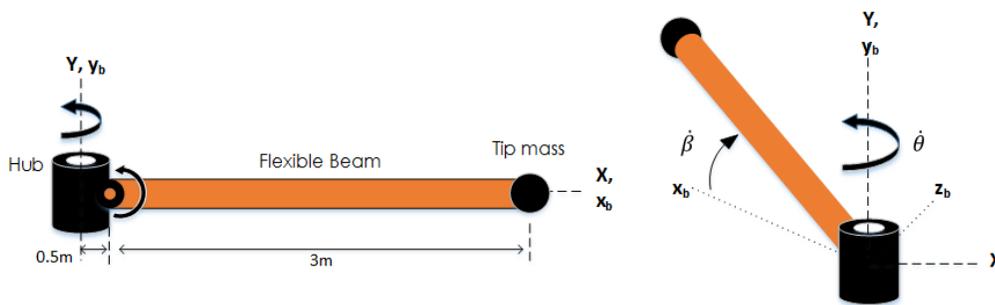


Figure 5.13: Schematic of the spatial manipulator with a flexible link

The mechanism shows that the flexible beam is attached to a hub that can rotate about a vertical Y axis. The beam can also rotate about its body-fixed horizontal z_b axis. There is also a tip mass attached to the beam which adds additional inertia and weight. The spatial motion of the manipulator can be prescribed as: when $t < 1s$ there is no rotation so that the beam is static which is the same as a cantilever beam under gravity; when $1s \leq t < 2s$ the beam is spinning only about the Y axis like a planar spinning beam; when $2s \leq t < 3s$ the beam has spatial motion of rotating about both Y and z_b axis. The prescribed angular motions in both directions can be defined in terms of angular velocity as

$$\dot{\theta} = \begin{cases} 0, & t < 1s \\ \omega_{\theta}, & 1s \leq t < 3s \end{cases} \quad \dot{\beta} = \begin{cases} 0, & t < 2s \\ \omega_{\beta}, & 2s \leq t < 3s \end{cases} \quad (5.9)$$

where ω_{θ} is set to $\pi \text{ rad/s}$, and ω_{β} is set to $\pi/2 \text{ rad/s}$. The parameters of the flexible beam are: length $L = 3m$, cross-sectional area $A = 0.005m^2$, mass density $\rho = 3000 \text{ kg/m}^3$, second moment of area $I_y = I_z = 1.989 \times 10^{-6}m^4$, Young's modulus $E = 7.0 \times 10^{10} \text{ N/m}^2$,

shear modulus $G = 2.7 \times 10^{10} \text{ N/m}^2$, and gravity $g = 9.81 \text{ m/s}^2$. The rigid hub has a radius of 0.5 m , and the tip mass has a mass of 30 kg .

The “CBEAM” formulation and nonlinear element formulation “LAGR” embedded in Nastran are selected, respectively, to construct the beam properties and element equations, and 10 elements are used to generate the mode shapes of the flexible beam before assembling the mechanism in ADAMS. On the other hand, two elements are used for the MapleSim FEM. Both results are validated by the analytical model mentioned before. The tip deflection magnitudes predicted by the three models are plotted in Figure 5.14. The results of all three models are converged with small difference observed between ADAMS/Nastran and the other two after $t = 2.5 \text{ s}$ with a root-mean-square error of 0.23%.

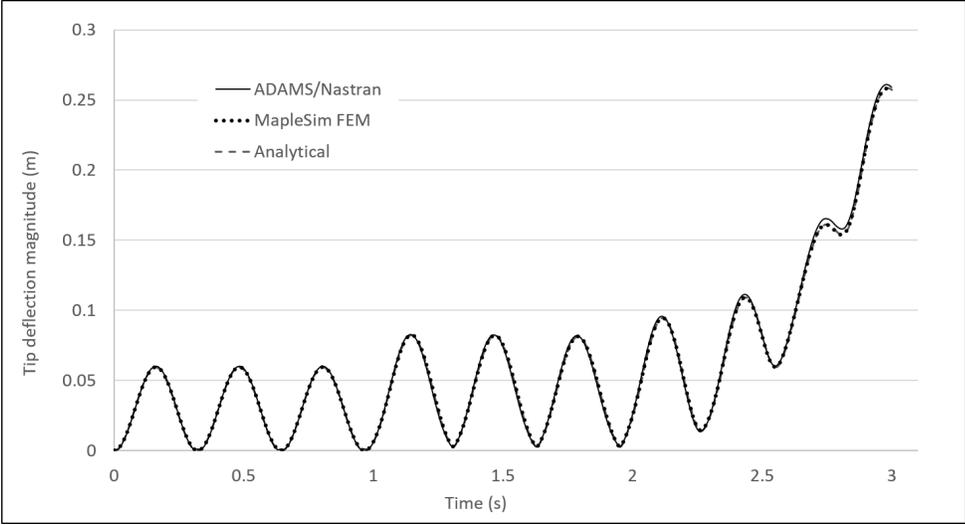


Figure 5.14: Tip deformation magnitudes

This negligible difference can be explained by the different values of axial foreshortening predicted by ADAMS/Nastran and the other two models shown in Figure 5.15, as the tip transverse deflections from three models are converged exactly shown in Figure 5.16. Different approaches of nonlinear formulation between Nastran and MapleSim FEM and the fact that Nastran forces the selection of lumped mass distribution when generating the mode shapes could both result in the small divergence of the axial deformation. Since the nonlinear formulation embedded in the analytical model uses the same foreshortening

approach as the MapleSim FEM, they generate the same foreshortening value. However, this tiny difference will not lead to any obvious divergence of the overall deformation.

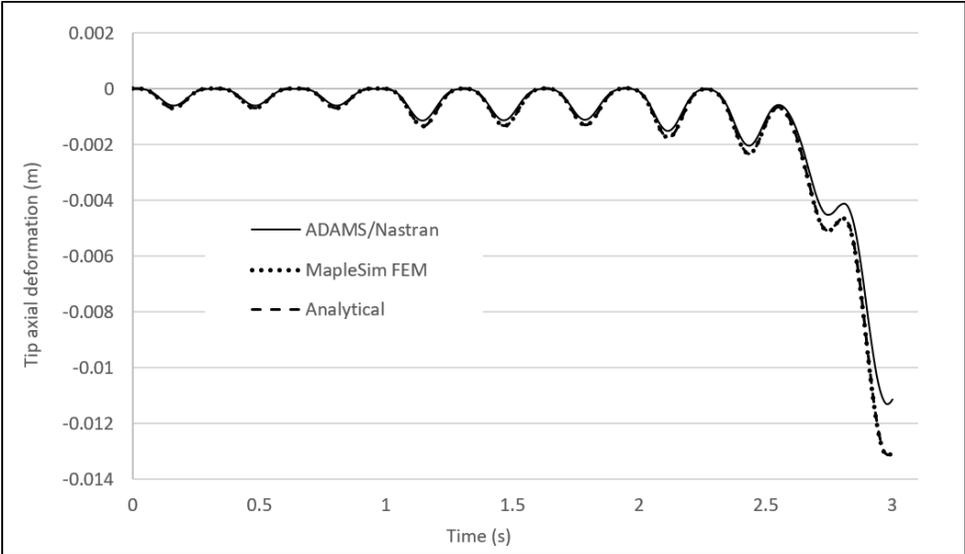


Figure 5.15: Tip axial deformation about the x_b axis (Analytical and MapleSim FEM results converge)

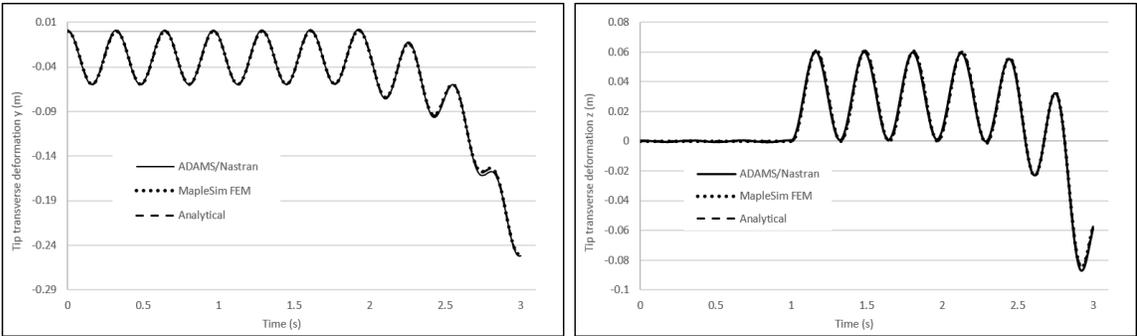


Figure 5.16: Tip transverse deformation about the local y_b and z_b axis (all results converge)

Although the modal approach (Nastran) and the nodal approach (MapleSim FEM) both display comparable accuracy when modelling this example, a large difference exists between them in terms of ability to converge. For such an example where low-frequency flexible modes are dominating, only one element is sufficient for the MapleSim FEM to reach convergence shown in Figure 5.17. This is because the displacement field of an element described by the shape functions defined in Chapter 3 is accurate enough to directly

predict the actual deformation of the flexible body under simple conditions.

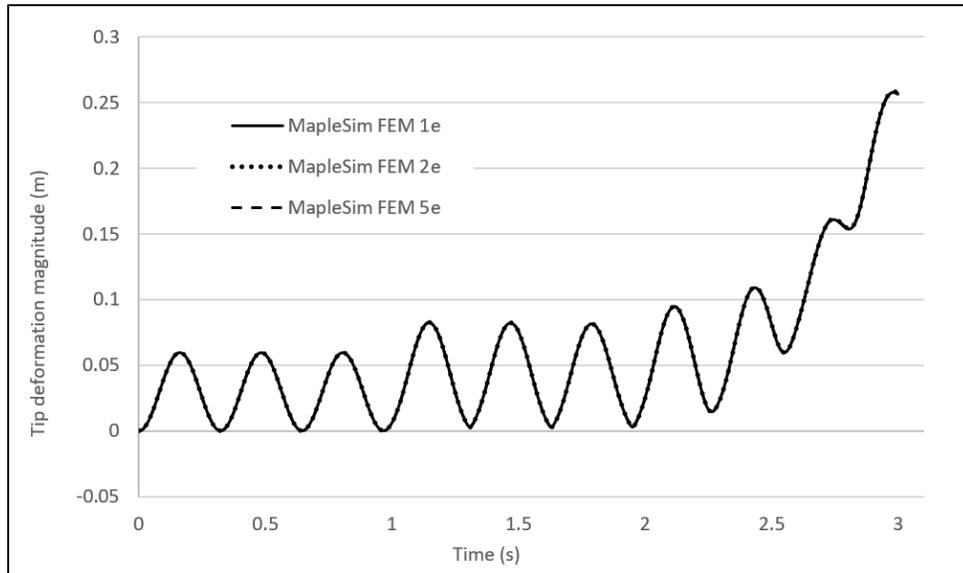


Figure 5.17: Tip transverse deformation for different numbers of elements (MapleSim FEM)

Unlike the nodal approach, the modal approach usually relies on a larger number of elements in order to generate accurate mode shapes. The ability to reach convergence for Nastran is shown in Figure 5.18. It is obvious that using the mode shapes generated by 3 elements is insufficient to predict accurate deformation results, so at least 6 elements are usually used at the beginning to obtain a rough estimation. Even 10 or more elements have to be selected in order to refine the results afterwards.

Associated with the larger number of elements and selection of mode shapes is always a higher cost. When considering only the CPU time, it will take less than 10 s for MapleSim FEM to finish one round of calculation when using one element for this example, which includes both equation set-up and time integration. However, it will take about 2.6 minutes for ADAMS just to finish the time integration when using 6 elements for this example. Other than that, if the user wants to change the number of elements in MapleSim FEM, he or she can just type a different number before each run without doing anything else when using the nodal approach. The modal approach needs the user to redefine the flexible body

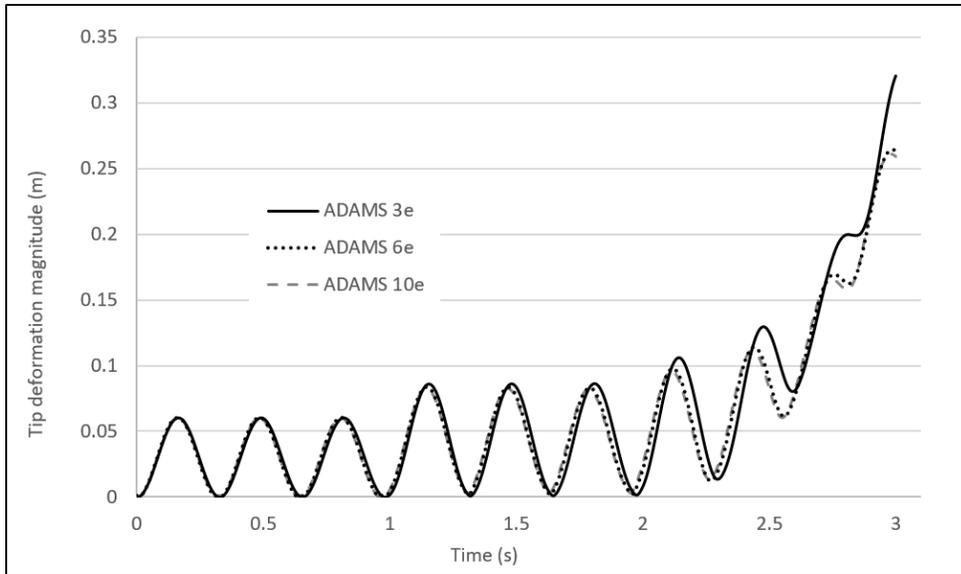


Figure 5.18: Tip transverse deformation for different numbers of elements (ADAMS/Nastran)

in Nastran and generate a new set of mode shapes, which has to go through many manual operations. The convenience of manipulation can be hardly compared only based on time.

Moreover, since the symbolic computation is able to generate the equations of motion as a pre-processor, parameters other than the number of elements, such as dimensional and material properties, can be easily changed afterwards. For this example, if using the symbolic approach, the user can modify the beam density, length, cross-sectional area, Young's modulus and so forth very conveniently. This allows the user to have more time to analyze many different results associated with varying parameters, which will benefit control and design engineers to a great extent.

5.5 Comparison of the current Finite-element Beam and the Analytical Beam Models in MapleSim

Compared with the well-developed analytical beam model in MapleSim, the currently implemented finite-element beam model has less fidelity due to the classical beam theory used to create the finite-element formulation, which neglects higher-order terms or assumes lin-

earity in the equations.

Recalling that the finite-element beam model is formulated based on the classical Bernoulli-Euler beam theory, which uses the traditional displacement field approach, twist is regarded free and decoupled from axial and bending deformations. The orientation of an arbitrary cross-section with respect to the body-fixed frame is indicated by using the twist angle and bending slopes as Euler angles. However, the equations of the analytical beam include the coupling between different deformations by using the complete 2^{nd} order ($O(2)$) displacement field approach [20] that yields a full elastic rotation matrix indicating the orientation of a cross-section with respect to the body-fixed frame. The difference of the assumed displacement fields and elastic rotation matrices between the two implemented beam models leads to different kinematic and elastic equations. An example in [20], which models a three-dimensional spin-up beam with an offset tip mass, shows the importance of the $O(2)$ terms in coupled bending and twist scenarios. In practice, when modelling a beam for which the coupling between bending and twist is significant, the finite-element beam model is insufficient to predict the orientation of one tip end with respect to the other, even though the bending and axial deformations are predicted correctly. Modelling of a golf shaft, for instance, requires a high-fidelity model that can accurately predict both the deformation and orientation of the golf head with respect to the grip. This is because a small difference of the golf head orientation will result in a large change of the golf ball trajectory.

The finite-element beam is expected to have some advantages over the analytical beam in certain cases. As stated in Chapter 2, the generalized coordinates from the Rayleigh-Ritz variation used by the analytical beam model lack physical meaning. Unlike the nodal coordinates of the finite-element model, these Rayleigh-Ritz coordinates cannot represent physical displacements by themselves. Since the nodal coordinates of the finite-element model are able to represent physical displacements, nodal strains and stresses can be calculated conveniently. In addition, the classical variational methods unavoidably possess the disadvantage of difficulty in properly constructing the approximating functions for arbitrarily-shaped deformable bodies, which limits their usage when modelling more com-

plex flexible components. Using the coordinate systems described in Section 3.1.2, for example, the finite-element beam model can be extended to model curved beams conveniently since the orientations of all elements are not necessarily co-linear by using the intermediate element coordinate system.

6 Conclusions and Future Work

6.1 Overview of the Current Development

The objective of integrating the finite element method with multibody system dynamics using symbolic computation has been successfully completed using the nodal formulation and a graph-theoretical approach. It takes advantage of both the nodal formulation over a modal formulation, and the graph-theoretical approach over the conventional absolute coordinate formulation. The former gets rid of the tedious modal analysis, and the latter helps to reduce the number of system equations to ensure the efficiency of the nodal approach. The validity and expected performance are explained mathematically and philosophically, which are also backed up by tests through several general examples.

The finite element method used for a flexible beam model, due to its simplicity and relatively wide application in multibody system dynamics, has been developed first. The classical *Bernoulli-Euler* beam theory is used to construct the finite-element formulation. Considering the critical influence of the geometrical stiffening nonlinearity under fast and large motion problems, an upgraded version simply referred to as the nonlinear formulation is developed with the addition of a foreshortening term to the displacement field. The nonlinear formulation is specifically and explicitly derived for the nodal finite-element formulation, which is rarely found in publications. A systematic procedure for formulating the flexible beam equations is presented for the linear and nonlinear versions. Starting from deriving the element shape function matrix based on assumed displacement fields, the position vector of an arbitrary point on the deformable body is generated. The position vector plays the key to deriving the other critical terms relating the inertia and stiffness of the deformable body that are used to build the system dynamic equations. This general procedure can be followed to derive formulations of plate and solid elements if using appropriate displacement field assumptions.

The equations of motion of the finite-element flexible beam are transferred to a form re-

quired by the graph-theoretical approach, and then the beam model is implemented in MapleSim and made compatible with other multibody system components in the default library. A general procedure is developed for the transformation. Since the transformation is based on the original equations of motion of the flexible beam, which are similar for any type of elements, the procedure can be followed later to create other types of elements.

Example problems validate the functionality of the finite-element beam model implemented. The deformation results for simple load conditions and closed-loop mechanisms and the results regarding the geometrical stiffening are convergent with the results recorded from the literature and obtained from other models. The dynamic stiffening effect is explicitly explained with its significance in fast and large motion problems. Comparison results show that the nodal approach is well-suited to symbolic computation, which runs faster and is more convenient to manipulate than the numerical modal approach.

6.2 Future Development and Extensions

The implementation of a basic symbolic finite-element beam in MapleSim is the initial stage. The future work can be divided into two categories. The first one is to improve the current finite-element beam model in various aspects, and the second one is to extend the presented method to other types of elements.

For improving the finite-element beam, increasing the complexity of the assumed displacement field described in Section 3.1.1 could be the next step. Since the coupling of deformations and the full expression of the elastic rotation matrix become necessary when accuracy is required to obtain the orientation of one tip end with respect to the other, the method in [20] can be used to modify the formulation. Furthermore, the function of modelling tapered beams with continuous cross-sectional areas should be upgraded to model beams with discrete cross-sectional areas. For example, the cross-sectional area and second moment of area can be defined by piecewise functions. The material quantities also need to allow inputs of functions for the purpose of modelling non-homogeneous materials. If shear is needed to model thick beams, Timoshenko beam theory can be used to modify

the formulation.

Considering the benefits of the graph-theoretical approach and symbolic computation, the extension from the beam element to those with more complex geometries in the future seems promising. The code of a finite-element plate can be incorporated first. It is expected that the additional effort will be spent on defining the boundary conditions. Different from a beam which is often connected at its tip ends or sometimes at points on the longitudinal axis, the connections of a plate are more flexible and user-defined most of the time. The ultimate object is to develop solid elements used to model flexible bodies with arbitrary shapes. The formulation is expected to be easier since most of the solid elements are isoparametric types described in Section 2.2 and do not have coordinates incorporated with nodal rotations. However, the total number of degrees of freedom generated by the nodal approach is an outstanding issue to be resolved. It can be worthwhile to develop a specific modal analysis or any coordinate reduction techniques, such as nodal condensation or subtraction approaches, which fully take the advantages of the symbolic computation.

References

- [1] A. A. Shabana, *Dynamics of Multibody Systems*. Cambridge university press, 2013.
- [2] W. Schiehlen, *Advanced Multibody System Dynamics: Simulation and Software Tools*, vol. 20. Springer Science & Business Media, 2013.
- [3] J.-C. Samin and P. Fiset, *Symbolic Modeling of Multibody Systems*, vol. 112. Springer Science & Business Media, 2013.
- [4] P. Shi and J. McPhee, “Dynamics of flexible multibody systems using virtual work and linear graph theory,” *Multibody System Dynamics*, vol. 4, no. 4, pp. 355–381, 2000.
- [5] J. McPhee, C. Schmitke, and S. Redmond, “Dynamic modelling of mechatronic multibody systems with symbolic computing and linear graph theory,” *Mathematical and Computer Modelling*, vol. 10, no. 1, pp. 1–23, 2004.
- [6] M. Tiller, *Introduction to Physical Modeling with Modelica*, vol. 615. Springer Science & Business Media, 2012.
- [7] J. Hřebíček, M. Řezáč, *et al.*, “Modelling with maple and maplesim,” in *Proceedings 22nd European conference modelling and simulation ECMS*, pp. 60–66, 2008.
- [8] A. Shabana and R. Schwertassek, “Equivalence of the floating frame of reference approach and finite element formulations,” *International Journal of Non-linear Mechanics*, vol. 33, no. 3, pp. 417–432, 1998.
- [9] A. A. Shabana and R. Y. Yakoub, “Three dimensional absolute nodal coordinate formulation for beam elements: theory,” *Journal of Mechanical Design*, vol. 123, no. 4, pp. 606–613, 2001.
- [10] J. Escalona, H. Hussien, and A. Shabana, “Application of the absolute nodal coordinate formulation to multibody system dynamics,” Tech. Rep. No. MBS97-1-UIC, Illinois University at Chicago Circle Department of Mechanical Engineering, 1997.

- [11] M. Berzeri, M. Campanelli, and A. A. Shabana, "Definition of the elastic forces in the finite-element absolute nodal coordinate formulation and the floating frame of reference formulation," *Multibody System Dynamics*, vol. 5, no. 1, pp. 21–54, 2001.
- [12] M. Dibold, J. Gerstmayr, and H. Irschik, "A detailed comparison of the absolute nodal coordinate and the floating frame of reference formulation in deformable multibody systems," *Journal of Computational and Nonlinear Dynamics*, vol. 4, no. 2, p. 021006, 2009.
- [13] O. A. Bauchau, S. Han, A. Mikkola, and M. K. Matikainen, "Comparison of the absolute nodal coordinate and geometrically exact formulations for beams," *Multibody System Dynamics*, vol. 32, no. 1, pp. 67–85, 2014.
- [14] D. Garcia-Vallejo, J. Escalona, J. Mayo, and J. Dominguez, "Describing rigid-flexible multibody systems using absolute coordinates," *Nonlinear Dynamics*, vol. 34, no. 1-2, pp. 75–94, 2003.
- [15] P. E. Nikravesh and G. Gim, "Systematic construction of the equations of motion for multibody systems containing closed kinematic loops," *Journal of Mechanical Design*, vol. 115, no. 1, pp. 143–149, 1993.
- [16] G. Andrews and H. Kesavan, "The vector-network model: a new approach to vector dynamics," *Mechanism and Machine Theory*, vol. 10, no. 1, pp. 57–75, 1975.
- [17] J. J. McPhee, "On the use of linear graph theory in multibody system dynamics," *Nonlinear Dynamics*, vol. 9, no. 1-2, pp. 73–90, 1996.
- [18] G. C. Andrews, M. J. Richard, and R. J. Anderson, "A general vector-network formulation for dynamic systems with kinematic constraints," *Mechanism and Machine Theory*, vol. 23, no. 3, pp. 243–256, 1988.
- [19] J. J. McPhee, "Automatic generation of motion equations for planar mechanical systems using the new set of branch coordinates," *Mechanism and Machine Theory*, vol. 33, no. 6, pp. 805–823, 1998.

- [20] P. Shi, J. McPhee, and G. Heppler, “A deformation field for euler–bernoulli beams with applications to flexible multibody dynamics,” *Multibody System Dynamics*, vol. 5, no. 1, pp. 79–104, 2001.
- [21] B. A. Finlayson, *The Method of Weighted Residuals and Variational Principles*, vol. 73. SIAM, 2013.
- [22] R. D. Cook *et al.*, *Concepts and Applications of Finite Element Analysis*. John Wiley & Sons, 2007.
- [23] A. A. Shabana, “Flexible multibody dynamics: review of past and recent developments,” *Multibody System Dynamics*, vol. 1, no. 2, pp. 189–222, 1997.
- [24] R. J. Astley, *Finite Elements in Solids and Structures. An Introduction*. Chapman & Hall (Springer), 1992.
- [25] J. Hutchinson, “Shear coefficients for timoshenko beam theory,” *Journal of Applied Mechanics*, vol. 68, no. 1, pp. 87–92, 2001.
- [26] H. Ma, X.-L. Gao, and J. Reddy, “A microstructure-dependent timoshenko beam model based on a modified couple stress theory,” *Journal of the Mechanics and Physics of Solids*, vol. 56, no. 12, pp. 3379–3391, 2008.
- [27] Z. Friedman and J. B. Kosmatka, “An improved two-node timoshenko beam finite element,” *Computers & Structures*, vol. 47, no. 3, pp. 473–481, 1993.
- [28] I. H. Shames, *Energy and Finite Element Methods in Structural Mechanics*. CRC Press, 1985.
- [29] J. Mayo, J. Dominguez, and A. Shabana, “Geometrically nonlinear formulations of beams in flexible multibody dynamics,” *Journal of Vibration and Acoustics*, vol. 117, no. 4, pp. 501–509, 1995.
- [30] T. Kane, R. Ryan, and A. Banerjee, “Dynamics of a cantilever beam attached to a moving base,” *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 2, pp. 139–151, 1987.

- [31] E. Bakr and A. A. Shabana, “Geometrically nonlinear analysis of multibody systems,” *Computers & Structures*, vol. 23, no. 6, pp. 739–751, 1986.
- [32] R. W. Clough, J. Penzien, and D. Griffin, “Dynamics of structures,” *Journal of Applied Mechanics*, vol. 44, p. 366, 1977.
- [33] J. Mayo and J. Dominguez, “A finite element geometrically nonlinear dynamic formulation of flexible multibody systems using a new displacements representation,” *Journal of Vibration and Acoustics*, vol. 119, no. 4, pp. 573–581, 1997.
- [34] U. Lugić, M. A. Naya, J. A. Pérez, and J. Cuadrado, “Implementation and efficiency of two geometric stiffening approaches,” *Multibody System Dynamics*, vol. 20, no. 2, pp. 147–161, 2008.
- [35] J.-C. Piedbuef and B. Moore, “On the foreshortening effects of a rotating flexible beam using different modeling methods*,” *Mechanics of Structures and Machines*, vol. 30, no. 1, pp. 83–102, 2002.
- [36] J. Liu and J. Hong, “Geometric stiffening effect on rigid-flexible coupling dynamics of an elastic beam,” *Journal of Sound and Vibration*, vol. 278, no. 4, pp. 1147–1162, 2004.
- [37] M. C. Bampton and R. R. Craig, JR, “Coupling of substructures for dynamic analyses,” *AIAA Journal*, vol. 6, no. 7, pp. 1313–1319, 1968.
- [38] L. Meirovitch, *Methods of Analytical Dynamics*. Courier Corporation, 2010.
- [39] R. R. Craig, “Coupling of substructures for dynamic analyses: an overview,” in *Proceedings of AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference and exhibit*, pp. 1573–1584, 2000.
- [40] P. Fisette, T. Postiau, L. Sass, and J. C. Samin, “Fully symbolic generation of complex multibody models*,” *Mechanics of Structures and Machines*, vol. 30, pp. 31–82, Jan. 2002.
- [41] A. Boyle and B. Caviness, “Future directions for research in symbolic computation,” tech. rep., Society for Industrial and Applied Mathematics, Philadelphia, 1990.

- [42] N. Docquier, A. Poncelet, P. Fisette, *et al.*, “Robotran: a powerful symbolic generator of multibody models,” *Mech. Sci.*, vol. 4, no. 1, pp. 199–219, 2013.
- [43] J. Connelly and R. Huston, “The dynamics of flexible multibody systems: A finite segment approach - i. Theoretical aspects,” *Computers & Structures*, vol. 50, no. 2, pp. 255 – 258, 1994.
- [44] J. Connelly and R. Huston, “The dynamics of flexible multibody systems: A finite segment approach - ii. Example problems,” *Computers & Structures*, vol. 50, no. 2, pp. 259–262, 1994.
- [45] C. Jen, D. Johnson, and F. Dubois, “Numerical modal analysis of structures based on a revised substructure synthesis approach,” *Journal of Sound and Vibration*, vol. 180, no. 2, pp. 185–203, 1995.
- [46] P. Fisette, D. Johnson, and J.-C. Samin, “A fully symbolic generation of the equations of motion of multibody systems containing flexible beams,” *Computer Methods in Applied Mechanics and Engineering*, vol. 142, no. 1, pp. 123–152, 1997.
- [47] J. M. Lee, *MSC-Nastran Linear Static Analysis User’s Guide*. MSC Software, 1997.
- [48] G. Ferretti, F. Schiavo, and L. Vigano, “Object-oriented modelling and simulation of flexible multibody thin beams in modelica with the finite element method,” in *4th Modelica Conference, Hamburg-Harburg, Germany, 2005*.
- [49] P. Fritzson and V. Engelson, “Modelicaa unified object-oriented language for system modeling and simulation,” in *European Conference on Object-Oriented Programming*, pp. 67–90, Springer, 1998.
- [50] P. Shi, *Flexible multibody dynamics: A new approach using virtual work and graph theory*. Phd thesis, Systems Design Engineering, University of Waterloo, 1998.
- [51] K.-J. Bathe, *Finite Element Procedures*. Klaus-Jurgen Bathe, 2006.
- [52] A. Shabana, “Resonance conditions and deformable body coordinate systems,” *Journal of Sound and Vibration*, vol. 192, no. 1, pp. 389–398, 1996.

- [53] J. Banerjee and J. McPhee, “Acausal modelling for dynamic multibody simulation of damped three-dimensional flexible beams,” *Multibody System Dynamics*, 2016.
- [54] P. Shi, J. McPhee, and G. Heppler, *DynaFlex Users Guide*. Systems Design Engineering, University of Waterloo, 2002.

A Appendix

A.1 Inertia Shape Integrals

The mass matrices derived from the conventional formulation in Eq. (3.40) and the geometrically nonlinear formulation in Eq. (3.118) are different from each other, but the derivation steps are similar. The general method is presented in this appendix to derive the mass matrix of the geometrically nonlinear formulation, and, at the end, it will be shown that by disabling a single term before the derivation will yield the mass matrix of the conventional formulation. Using symbolic computation, all expressions derived in this section can be pre-calculated and saved as symbolic equations.

The first submatrix \mathbf{m}_{rrGN}^i in Eq. (3.118) is defined as

$$\mathbf{m}_{rrGN}^i = \int_0^{l^i} \rho^i \mathbf{I} A^i dx = m^i \mathbf{I} \quad (\text{A.1})$$

where m^i is the mass of the i^{th} element and \mathbf{I} is 3×3 identity matrix.

The submatrix $\mathbf{m}_{r\theta GN}^i$, which represents the inertia coupling between the translation and rotation of the body-fixed frame, is defined as

$$\mathbf{m}_{r\theta GN}^i = - \int_0^{l^i} \rho^i \mathbf{A} \tilde{\mathbf{u}}^i A^i dx = -\mathbf{A} \left(\int_0^{l^i} \widetilde{\rho^i \mathbf{u}^i A^i dx} \right) \quad (\text{A.2})$$

where $\tilde{\cdot}$ transfers the entire integration term in the parenthesis to its skew symmetric matrix, and the term inside the parenthesis can be derived, using the definition of \mathbf{u} in Eq. (3.109), as

$$\int_0^{l^i} \rho^i \mathbf{u}^i A^i dx = [\bar{\mathbf{S}}^i](\mathbf{q}_0 + \mathbf{B}_2 \mathbf{q}_f) + [\bar{\mathbf{B}}_H^i] \mathbf{B}_2 \mathbf{q}_f \quad (\text{A.3})$$

where $\bar{\mathbf{S}}^i$ and $\bar{\mathbf{B}}_H^i$ are the inertia shape integrals defined as

$$[\bar{\mathbf{S}}^i] = \int_0^{l^i} \rho^i \mathbf{S}^i \mathbf{B}_1 A^i dx \quad (\text{A.4})$$

$$[\bar{\mathbf{B}}_H^i] = \int_0^{l^i} \rho^i \mathbf{B}_H^i A^i dx \quad (\text{A.5})$$

in which \mathbf{S} and \mathbf{B}_H are defined, respectively, in Eq. (3.17) and Eq. (3.110). These shape integrals are also required for deriving the submatrix $\mathbf{m}_{rf_{GN}}^i$, which represents the inertia coupling between the translation of the body-fixed frame and the elastic deformation of the element. The $\mathbf{m}_{rf_{GN}}^i$ can be written as

$$\mathbf{m}_{rf_{GN}}^i = \int_0^{l^i} \rho^i \mathbf{A} \mathbf{N}_{GN}^i A^i dx = \mathbf{A} \left(\int_0^{l^i} \rho^i (\mathbf{N}^i + 2\mathbf{B}_H^i) A^i dx \right) \quad (\text{A.6})$$

in which \mathbf{N}^i is defined in Eq. 3.111. Using the shape integrals \mathbf{S} and \mathbf{B}_H , the term inside the parenthesis can be written as

$$\int_0^{l^i} \rho^i (\mathbf{N}^i + 2\mathbf{B}_H^i) A^i dx = [\bar{\mathbf{S}}^i] \mathbf{B}_2 + 2 \cdot [\bar{\mathbf{B}}_H^i] \quad (\text{A.7})$$

The central term in the mass matrix, $\mathbf{m}_{\theta\theta_{GN}}^i$, represents the symmetric inertia tensor of the i^{th} element with respect to the body-fixed frame, which is written in a more explicit form as

$$\mathbf{m}_{\theta\theta_{GN}}^i = \int_0^{l^i} \rho^i \tilde{\mathbf{u}}^{iT} \tilde{\mathbf{u}}^i A^i dx = \begin{bmatrix} i_{11} & i_{12} & i_{13} \\ & i_{22} & i_{23} \\ sym & & i_{33} \end{bmatrix} \quad (\text{A.8})$$

where the elements i_{jk} ($j, k = 1, 2, 3$) are defined by carrying out the matrix multiplication of the skew-symmetric matrices, which can be defined as follows:

$$\left. \begin{aligned} i_{11} &= \int_0^{l^i} \rho^i \left[(u_2^i)^2 + (u_3^i)^2 \right] A^i dx, & i_{12} &= - \int_0^{l^i} \rho^i u_2^i u_1^i A^i dx \\ i_{13} &= - \int_0^{l^i} \rho^i u_3^i u_1^i A^i dx, & i_{22} &= \int_0^{l^i} \rho^i \left[(u_3^i)^2 + (u_1^i)^2 \right] A^i dx \\ i_{23} &= - \int_0^{l^i} \rho^i u_3^i u_2^i A^i dx, & i_{33} &= \int_0^{l^i} \rho^i \left[(u_1^i)^2 + (u_2^i)^2 \right] A^i dx \end{aligned} \right\} \quad (\text{A.9})$$

where u_j^i ($j = 1, 2, 3$) are the components of $\tilde{\mathbf{u}}^i$. Therefore, these elements can be written

in terms of shape integrals as

$$\left. \begin{aligned}
i_{11} &= \mathbf{q}_n^T ([\bar{\mathbf{S}}_{2,2}^i] + [\bar{\mathbf{S}}_{3,3}^i]) \mathbf{q}_n \\
i_{12} &= -\mathbf{q}_n^T [\bar{\mathbf{S}}_{1,2}^i]^T \mathbf{q}_n - \mathbf{q}_n^T [\bar{\mathbf{S}}_{2,B_s}^i] \mathbf{B}_2 \mathbf{q}_f \\
i_{13} &= -\mathbf{q}_n^T [\bar{\mathbf{S}}_{1,3}^i]^T \mathbf{q}_n - \mathbf{q}_n^T [\bar{\mathbf{S}}_{3,B_s}^i] \mathbf{B}_2 \mathbf{q}_f \\
i_{22} &= \mathbf{q}_n^T ([\bar{\mathbf{S}}_{1,1}^i] + [\bar{\mathbf{S}}_{3,3}^i]) \mathbf{q}_n + \mathbf{q}_n^T [\bar{\mathbf{S}}_{1,B_s}^i] \mathbf{B}_2 \mathbf{q}_f \\
&\quad + (\mathbf{B}_2 \mathbf{q}_f)^T [\bar{\mathbf{S}}_{1,B_s}^i]^T \mathbf{q}_n + (\mathbf{B}_2 \mathbf{q}_f)^T [\bar{\mathbf{S}}_{B_s,B_s}^i] \mathbf{B}_2 \mathbf{q}_f \\
i_{23} &= -\mathbf{q}_n^T [\bar{\mathbf{S}}_{2,3}^i]^T \mathbf{q}_n \\
i_{33} &= \mathbf{q}_n^T ([\bar{\mathbf{S}}_{1,1}^i] + [\bar{\mathbf{S}}_{2,2}^i]) \mathbf{q}_n + \mathbf{q}_n^T [\bar{\mathbf{S}}_{1,B_s}^i] \mathbf{B}_2 \mathbf{q}_f \\
&\quad + (\mathbf{B}_2 \mathbf{q}_f)^T [\bar{\mathbf{S}}_{1,B_s}^i]^T \mathbf{q}_n + (\mathbf{B}_2 \mathbf{q}_f)^T [\bar{\mathbf{S}}_{B_s,B_s}^i] \mathbf{B}_2 \mathbf{q}_f
\end{aligned} \right\} \quad (\text{A.10})$$

in which the shape integrals are defined as follows:

$$[\bar{\mathbf{S}}_{j,k}^i] = \int_0^{l^i} \rho^i \mathbf{N}_j^{iT} \mathbf{N}_k^i A^i dx, \quad j, k = 1, 2, 3 \quad (\text{A.11})$$

$$[\bar{\mathbf{S}}_{j,B_s}^i] = \int_0^{l^i} \rho^i \mathbf{N}_j^{iT} \mathbf{B}_s^i A^i dx, \quad j = 1, 2, 3 \quad (\text{A.12})$$

$$[\bar{\mathbf{S}}_{B_s,B_s}^i] = \int_0^{l^i} \rho^i \mathbf{B}_s^{iT} \mathbf{B}_s^i A^i dx \quad (\text{A.13})$$

where \mathbf{N}_j^i ($j = 1, 2, 3$) is the j^{th} row of the matrix \mathbf{N}^i , and \mathbf{B}_s^i is defined in Eq. (3.106).

These shape integrals, which depend on the nodal deformations, are also required to evaluate the matrix $\mathbf{m}_{\theta f_{GN}}^i$ that describes the inertia coupling between the rotation of the body-fixed frame and the deformation of the element. This can be demonstrated by writing $\mathbf{m}_{\theta f_{GN}}^i$ in the partitioned form as

$$\mathbf{m}_{\theta f_{GN}}^i = \int_0^{l^i} \rho^i \tilde{\mathbf{u}}^i \mathbf{N}_{GN}^i A^i dx = \int_0^{l^i} \rho^i \begin{bmatrix} u_2^i \mathbf{N}_{GN,3}^i - u_3^i \mathbf{N}_{GN,2}^i \\ u_3^i \mathbf{N}_{GN,1}^i - u_1^i \mathbf{N}_{GN,3}^i \\ u_1^i \mathbf{N}_{GN,2}^i - u_2^i \mathbf{N}_{GN,1}^i \end{bmatrix} A^i dx \quad (\text{A.14})$$

where $\mathbf{N}_{GN,j}^i$ ($j = 1, 2, 3$) is the j^{th} row of the matrix \mathbf{N}_{GN}^i . Substituting the derivations

of the u_j^i ($j = 1, 2, 3$) yields

$$\mathbf{m}_{\theta f_{GN}}^i = \begin{bmatrix} \mathbf{q}_n^T \left([\bar{\mathbf{S}}_{2,3}^i] - [\bar{\mathbf{S}}_{2,3}^i]^T \right) \\ -\mathbf{q}_n^T \left([\bar{\mathbf{S}}_{1,3}^i] - [\bar{\mathbf{S}}_{1,3}^i]^T \right) + 2\mathbf{q}_n^T [\bar{\mathbf{S}}_{3,B_s}^i] - (\mathbf{B}_2 \mathbf{q}_f)^T [\bar{\mathbf{S}}_{3,B_s}^i]^T \\ \mathbf{q}_n^T \left([\bar{\mathbf{S}}_{1,2}^i] - [\bar{\mathbf{S}}_{1,2}^i]^T \right) - 2\mathbf{q}_n^T [\bar{\mathbf{S}}_{2,B_s}^i] + (\mathbf{B}_2 \mathbf{q}_f)^T [\bar{\mathbf{S}}_{2,B_s}^i]^T \end{bmatrix} \quad (\text{A.15})$$

The last term $\mathbf{m}_{ff_{GN}}^i$ is a constant matrix and can be written as

$$\mathbf{m}_{ff_{GN}}^i = \int_0^{l^i} \rho^i \mathbf{N}_{GN}^{iT} \mathbf{N}_{GN}^i A^i dx \quad (\text{A.16})$$

Substituting for \mathbf{N}_{GN}^i , one obtains

$$\mathbf{m}_{ff_{GN}}^i = \mathbf{B}_2^T \left([\bar{\mathbf{S}}_{S,S}^i] + 2 [\bar{\mathbf{S}}_{S,B_H}^i] + 2 [\bar{\mathbf{S}}_{S,B_H}^i]^T + 4 [\bar{\mathbf{S}}_{B_H,B_H}^i] \right) \mathbf{B}_2 \quad (\text{A.17})$$

in which the shape integrals are defined as follows:

$$[\bar{\mathbf{S}}_{S,S}^i] = \int_0^{l^i} \rho^i \mathbf{N}^{iT} \mathbf{N}^i A^i dx \quad (\text{A.18})$$

$$[\bar{\mathbf{S}}_{S,B_H}^i] = \int_0^{l^i} \rho^i \mathbf{N}^{iT} \mathbf{B}_H^i A^i dx \quad (\text{A.19})$$

$$[\bar{\mathbf{S}}_{B_H,B_H}^i] = \int_0^{l^i} \rho^i \mathbf{B}_H^{iT} \mathbf{B}_H^i A^i dx \quad (\text{A.20})$$

where \mathbf{B}_H^i is defined in Eq. (3.110).

Even though the mass matrix depends on the rotation of the body-fixed frame and the elastic deformation of the body, and thus time-variant, it is clear that building all the shape integrals highlighted in the brackets first will make the calculation of the mass matrix become just basic multiplications and summations between the shape integrals and the nodal coordinates. This is saving much effort, and the same manner can be applied when deriving the generalized force vectors as well, especially the quadratic velocity force vector \mathbf{Q}_v and the generalized gravitational force vector \mathbf{F}_g .

One may also notice that the mass matrix derived from the geometrically nonlinear formulation shown above contains the shape integrals related to the foreshortening correction term \mathbf{B}_H (or \mathbf{B}_s). Apparently, it leads to an easy way to generate the mass matrix of the conventional formulation by just disabling all the shape integrals related to the foreshortening correction. In practice, this can be done, for example, by setting \mathbf{B}_s as a null matrix with the same size.

A.2 Jacobian Matrix for the Illustrative Example Problem

The Jacobian matrix associated with the 8 reaction forces and torques in Eq. (4.38) can be written as

$$\mathbf{C}_q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & C_1 & 0 & C_2 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ \mathbf{0}_6^1 & \mathbf{0}_6^1 \\ 0 & 0 & 0 & 0 & 0 & C_3 & 0 & C_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & C_5 & 0 & C_6 \\ \mathbf{0}_3^1 & \mathbf{0}_3^1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}^T \quad (\text{A.21})$$

where the subscript and superscript of the bold zeros indicate, respectively, the row and column dimensions, and C_j ($j = 1..6$) are defined as

$$\left. \begin{aligned} C_1 &= (l + q_{f_7}) \sin \theta_y - q_{f_9} \cos \theta_y \\ C_2 &= (l + q_{f_7}) \cos \theta_y + q_{f_9} \sin \theta_y \\ C_3 &= \cos \theta_y \\ C_4 &= \sin \theta_y \\ C_5 &= -\sin \theta_y \\ C_6 &= \cos \theta_y \end{aligned} \right\} \quad (\text{A.22})$$

where l is the beam length, and q_{f_7} and q_{f_9} are two nodal deformation coordinates.