Partitioned nonlinear structural analysis of wind turbines using BeamDyn

Qi Wang[*], Michael A. Sprague[†], Jason Jonkman[‡], and Bonnie Jonkman[§]

National Renewable Energy Laboratory, Golden, CO 80401

We present the numerical implementation of BeamDyn, a finite element beam solver based on geometrically exact beam theory (GEBT), and its coupling to different modules in the FAST modularization framework. After reviewing the underlying BeamDyn theory and implementation, we describe the coupling algorithm and numerical integration scheme specifically designed for wind turbine analysis. The loose coupling method between BeamDyn and other modules is used where the nonlinear input-output coupling relations are solved by the Newton-Raphson method. Finally, numerical examples are provided to verify both BeamDyn and the coupling algorithm.

I. Introduction

Recently, researchers at the National Renewable Energy Laboratory (NREL) developed BeamDyn, a nonlinear structural dynamics module for composite wind turbine blade analysis, in the FAST modularization framework. BeamDyn, which is founded on geometrically exact beam theory (GEBT), was created to acurately simulate the large, nonlinear deflections of modern wind turbine blades that are designed with aero-elastic tailoring and complicated composite structures. GEBT was first proposed by Reissner[1] and then extended to three-dimensional (3D) beams by Simo[2] and Simo and Vu-Quoc[3]. Readers are referred to Hodges[4], in which comprehensive derivations and discussions on nonlinear composite-beam theories can be found. In the BeamDyn implementation, the GEBT equations are discrretized spatially with Legendre spectral finite elements (LSFEs), which are p-type high order elements that combine the accuracy of global spectral methods with the geometric modeling flexibility of low-order h-type elements. More details on BeamDyn can be found in Wang et al.[5].

FAST is NREL's flagship multi-physics engineering tool for analyzing both land-based and offshore wind turbines under realistic operating conditions. The established FAST blade model included in the module Elastodyn, is incabable of predictive analysis of highly flexible, composite wind turbine blades. FAST has been reformulated under a new modularized framework that provides a rigorous means by which various mathematical systems are implemented in distinct modules and coupled to other modules. The restructuring of FAST greatly enhanced flexibility and expandability to enable further developments of functionality without the need to recode established modules. These modules are interconnected to solve for the globally coupled dynamic responses of wind turbines and wind plants[6,7]. In previous work, this framework is extended to handle the non-matching spatial grids at interfaces and non-matching temporal meshes that allow module solutions to advance with different time increments and different time integrators.[8]

In this paper, the formulation of the beam theory is first reviewed, and then the coupling algorithm of BeamDyn with other modules is presented. Numerical examples are provided to verify the proposed beam solver and its coupling algorithm when it is running in a coupled-to-FAST mode.

II. Formulation and Implementation

A. Geometrically Exact Beam Theory

This section briefly reviews the geometrically exact beam theory. Further details on the content of this section can be found in many other papers[9] and textbooks[4,10]. Figure 1 shows a beam in its initial undeformed and deformed states. A reference frame $\mathbf{b}_i$ is introduced along the beam axis for the undeformed state and a frame $\mathbf{B}_i$ is introduced along each point of the deformed beam axis. The curvilinear coordinate x_1 defines the intrinsic parameterization of the reference line. In this paper, matrix notation is used to denote vectorial or vectorial-like quantities. For example, an underline denotes a vector $\underline{u}$, a bar denotes unit vector $\bar{n}$, and a double underline denotes a tensor $\underline{\underline{\Delta}}$. Note that

[*] Research Engineer, National Wind Technology Center, AIAA Member. Email: Qi.Wang2@nrel.gov

[†] Senior Research Scientist, Computational Science Center.

[‡] Senior Engineer, National Wind Technology Center, AIAA Professional Member.

[§] Senior Engineer, National Wind Technology Center.

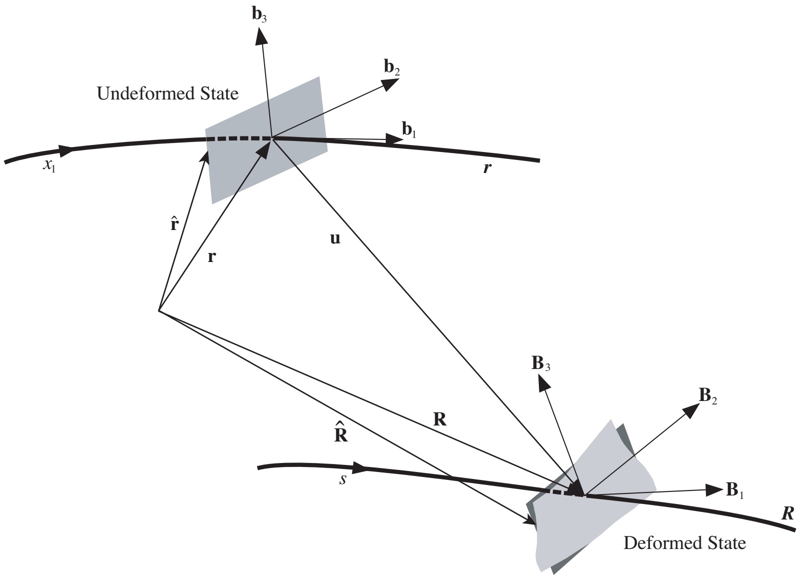


Figure 1: A beam deformation schematic.

sometimes the underlines only denote the dimension of the corresponding matrix. The governing equations of motion for geometrically exact beam theory can be written as[10]

$$\underline{\dot{h}} - \underline{F}' = \underline{f} \tag{1}$$

$$\underline{\dot{g}} + \dot{\tilde{u}}\underline{h} - \underline{M}' + (\tilde{x}_0' + \tilde{u}')^T \underline{F} = \underline{m} \tag{2}$$

where $\underline{h}$ and $\underline{g}$ are the linear and angular momenta resolved in the inertial coordinate system, respectively; $\underline{F}$ and $\underline{M}$ are the beam's sectional force and moment resultants, respectively; $\underline{u}$ is the one-dimensional (1D) displacement of a point on the reference line; $\underline{x}_0$ is the position vector of a point along the beam's reference line; and $\underline{f}$ and $\underline{m}$ are the distributed force and moment applied to the beam structure. The notation $(\bullet)'$ indicates a derivative with respect to beam axis x_1 and $(\dot{\bullet})$ indicates a derivative with respect to time. The tilde operator $(\widetilde{\bullet})$ defines a skew-symmetric tensor corresponding to the given vector. In the literature, it is also termed as "cross-product matrix". For example,

$$\widetilde{n} = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix}$$

The constitutive equations relate the velocities to the momenta and the 1D strain measures to the sectional resultants as

$$\begin{Bmatrix} \underline{h} \\ \underline{g} \end{Bmatrix} = \underline{\underline{\mathcal{M}}} \begin{Bmatrix} \underline{\dot{u}} \\ \underline{\omega} \end{Bmatrix} \tag{3}$$

$$\begin{Bmatrix} \underline{F} \\ \underline{M} \end{Bmatrix} = \underline{\underline{\mathcal{C}}} \begin{Bmatrix} \underline{\epsilon} \\ \underline{\kappa} \end{Bmatrix} \tag{4}$$

where $\underline{\underline{\mathcal{M}}}$ and $\underline{\underline{\mathcal{C}}}$ are the 6×6 sectional mass and stiffness matrices, respectively (note that they are not really tensors); $\underline{\epsilon}$ and $\underline{\kappa}$ are the 1D strains and curvatures, respectively; and, $\underline{\omega}$ is the angular velocity vector that is defined by the rotation tensor $\underline{\underline{R}}$ as $\underline{\omega} = \text{axial}(\underline{\underline{\dot{R}}}\;\underline{\underline{R}}^T)$. The axial vector $\underline{a}$ associated with a second-order tensor $\underline{\underline{A}}$ is denoted $\underline{a} = \text{axial}(\underline{\underline{A}})$ and its components are defined as

$$\underline{a} = \text{axial}(\underline{\underline{A}}) = \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \frac{1}{2} \begin{Bmatrix} A_{32} - A_{23} \\ A_{13} - A_{31} \\ A_{21} - A_{12} \end{Bmatrix} \tag{5}$$

The 1D strain measures are defined as

$$\begin{Bmatrix} \underline{\epsilon} \\ \underline{\kappa} \end{Bmatrix} = \begin{Bmatrix} \underline{x}_0' + \underline{u}' - (\underline{\underline{R}}\;\underline{\underline{R}}_0)\bar{\imath}_1 \\ \underline{k} \end{Bmatrix} \tag{6}$$

where $\underline{k} = \text{axial}[(\underline{\underline{RR_0}})'(\underline{\underline{RR_0}})^T]$ is the sectional curvature vector resolved in the inertial basis and $\bar{\imath}_1$ is the unit vector along x_1 direction in the inertial basis. Note that these three sets of equations, including equations of motion

Eq. (1) and (2), constitutive equations Eq. (3) and (4), and kinematical equations Eq. (6), provide a full mathematical description of elasticity problems.

For a displacement-based finite-element implementation, there are six degrees of freedom at each node: three displacement components and three rotation components. Here, $\underline{q}$ denotes the elemental displacement array as $\underline{q}^T = \left[\underline{u}^T \;\; \underline{p}^T\right]$ where $\underline{u}$ is the displacement and $\underline{p}$ is the rotation-parameter vector. The acceleration array can thus be defined as $\underline{a}^T = \left[\underline{\ddot{u}}^T \;\; \underline{\dot{\omega}}^T\right]$. For nonlinear finite-element analysis, the discretized forms of displacement, velocity, and acceleration are written as

$$\underline{q}(x_1) = \underline{\underline{N}}\,\underline{\hat{q}} \qquad \underline{q}^T = \left[\underline{u}^T \;\; \underline{p}^T\right] \tag{7}$$

$$\underline{v}(x_1) = \underline{\underline{N}}\,\underline{\hat{v}} \qquad \underline{v}^T = \left[\underline{\dot{u}}^T \;\; \underline{\omega}^T\right] \tag{8}$$

$$\underline{a}(x_1) = \underline{\underline{N}}\,\underline{\hat{a}} \qquad \underline{a}^T = \left[\underline{\ddot{u}}^T \;\; \underline{\dot{\omega}}^T\right] \tag{9}$$

where $\underline{\underline{N}}$ is the shape function matrix and $(\hat{\cdot})$ denotes a column matrix of nodal values.

B. Spatial and Temporal Discretization

As described above, the GEBT model is discretized in space with Legendre Spectral Finite Elements[11] (LSFEs), which are a high-order p-type finite element (FE). Elements have $p + 1$ nodes located at the Gauss-Lobatto-Legendre points, where p is the polynomial order of the Lagrangian-interpolant basis functions. For a given number of nodes, we have shown[5,12] that LSFEs for GEBT-based models can be dramatically more accurate than low-order elements. As a tool in BeamDyn, an LSFE evaluated with an appropriate quadrature scheme provides the option to model a modern, highly flexibly turbine blade with a single element. The choice of numerical quadrature is described in the next section. The GEBT-LSFE equations are time-integrated with a second-order-accurate generalized-alpha algorithm that is equipped with user-defined numerical damping (see Wang et al.[5] for details).

C. Finite-Element Quadrature

We describe here BeamDyn options for numerical integration (quadrature) of the finite-element inner products over an element domain. Typically, the quadrature rule employed in a finite-element implementation is Gauss-Legendre (GL), for which the number of quadrature points is chosen based on the polynomial order of the underlying FE basis functions. In the case where material properties vary significantly over an element domain, the accuracy of the quadrature is degraded, which can affect the overall accuracy of the solution. If the number of quadrature points is fixed to the FE basis-function order, accuracy is increased by either increasing number of elements (h-refinement) or the order of the elements (p-refinement). However, if the quadrature order is chosen for accurate evaluation of FE inner products, then the choice in FE resolution can be chosen based on overall solution accuracy.

The material sectional properties are defined discretely at n_s stations along the beam axis. BeamDyn is equipped with two quadrature options: Gauss-Legendre quadrature and trapezoidal-rule (TR) quadrature. For GL quadrature, $n_q = p + 1$, where n_q is the number of quadrature points and p is the order of the LSFE. Material properties are linearly interpolated from the nearest-neighbor discrete stations. Depending on the nature of the material properties, an increase in the element order p could instigate a dramatically different response, simply because the quadrature points captured different material properties. For TR quadrature, $n_q = jn_s$ is user specified, where j is a positive integer. TR quadrature enables a user to model a modern turbine blade defined by many cross-sectional-proper stations with few node points (i.e., $p \ll n_s$), while capturing all of the provided material properties. For example, the widely used NREL 5-MW reference wind turbine blade is defined by 49 stations along the blade axis. If one were using first-order FEs with a fixed quadrature scheme, at least 48 elements would be required to accurately capture the material data in the FE inner products. BeamDyn, with the GEBT model and LSFE p-type discretization, is equipped to model a wind turbine blade with a *single* element, or very few elements. LSFE discretization with TR quadrature is an effective modeling approach when the beam deformation can be described accurately with relatively few points, despite the large number of material-property stations. However, for a given element order and $n_q \gg p$, solutions will be significantly more expensive than if $n_q \approx p$ because inner products are evaluated at least once per time time step.

D. Module-Coupling Algorithm

The FAST modularization framework was created to loosely couple multi-physics modules for time domain simulation. Modules interact through input-output relationships, where inputs and outputs are defined on spatial meshes , and a predictor-corrector algorithm is employed to improve stability and accuracy of the coupled system. The modularization environment provides utilities for coupling non-matching meshes in space and time. A detailed description of the coupling algorithm can be found in Sprague et al.[8].

We describe here the inputs and outputs of the BeamDyn module. In the context of wind turbine blade modeling, BeamDyn inputs are root motion (including displacements/rotations, linear and angular velocities, and linear and angular accelerations), point loads (forces and moments), and distrbuted loads (e.g., those from the AeroDyn module). BeamDyn outputs are root reaction forces and moments, and beam motions. The BeamDyn mesh defining distributed-load inputs is a contiguous number of two-node line elements with nodes located at the BeamDyn quadrature points and the beam end points. The mesh definining point-load inputs is composed of points collocated with the BeamDyn nodes. The root-motion input is a point mesh at the beam root.

The following is incomplete, but I think we can delete

The inputs and outputs can be written as

$$\mathbf{u}_{BD} = \begin{bmatrix} \underline{q} & \underline{v} & \underline{a} \end{bmatrix}^T \tag{10}$$

$$\mathbf{y}_{BD} = \begin{bmatrix} \underline{f} & \underline{m} \end{bmatrix}^T \tag{11}$$

The inputs and outputs of the module that being coupled to BeamDyn have the form

$$\mathbf{u}_{O} = \begin{bmatrix} \underline{f} & \underline{m} \end{bmatrix}^T \tag{12}$$

$$\mathbf{y}_{O} = \begin{bmatrix} \underline{q} & \underline{v} & \underline{a} \end{bmatrix}^T \tag{13}$$

The input-output equations can be written as

$$\mathbf{U}_1 : \quad \mathbf{u}_{BD} - \mathbf{y}_O(\mathbf{u}_O, t) = 0 \tag{14}$$

$$\mathbf{U}_2 : \quad \mathbf{u}_{O} - \mathbf{y}_{BD}(\mathbf{u}_{BD}, t) = 0 \tag{15}$$

Newton-Raphson method is adopted here to solve this nonlinear system:

$$\begin{bmatrix} \frac{\partial \mathbf{U}_1}{\partial \mathbf{u}_{BD}} & \frac{\partial \mathbf{U}_1}{\partial \mathbf{u}_{O}} \\ \frac{\partial \mathbf{U}_2}{\partial \mathbf{u}_{BD}} & \frac{\partial \mathbf{U}_2}{\partial \mathbf{u}_{O}} \end{bmatrix} \begin{Bmatrix} \Delta \mathbf{u}_{BD} \\ \Delta \mathbf{u}_{O} \end{Bmatrix} = - \begin{Bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{Bmatrix} \tag{16}$$

The Jacobian matrix $\mathbf{J}$ on the left-hand-side is computed numerically using the forward differences formulae

$$\mathbf{J}_{ij} = \frac{1}{\epsilon_j} \left[\mathbf{U}_i(\mathbf{u} + \epsilon_j \mathbf{e}_j) - \mathbf{U}_i(\mathbf{u}) \right] \tag{17}$$

where $\mathbf{e}_j$ is the unit vector in the $\mathbf{u}_j$ direction and ϵ_j represents a small increment.

III. Numerical Examples

A. Example 1: Partitioned Analysis

This example verifies our numerical implementation and examines the accuracy and numerical behavior of the coupling algorithm. A beam is attached at its root to the mass in a spring-mass-damper system, see Figure 2.

The beam is modeled by BeamDyn and the time integrator for the spring-damper-mass system is Adams-Bashforth-Moulton (ABM4). The material properties, coordinate system, and geometric parameters can be found in Figure 2. The dimensions of the cross-section of the beam is 0.1 m × 0.1 m. The natural frequency of the uncoupled mass-spring-damper system is 6.28 rad/s and the first five natural frequencies for the uncoupled beam (in a clamped/cantilevered configuration) are $0.26, 1.72, 5.78, 22.62$, and 24.21 rad/s, respectively, as determined by a refined ANSYS modal analysis. The first five natural frequencies of the coupled system, obtained by ANSYS modal analysis, are $0.25, 0.85, 1.80, 4.76$, and 9.34 rad/s, respectively.

In the present study, the system has quiescent initial conditions with an initial displacement of 0.1 m in the positive Z direction (see Figure 2). We examine first the stability of the proposed coupling algorithm, for which the beam is discretized by a single fifth-order LSFE. Figure 3 shows the maximum stable time increments obtained by numerical experiment against the number of correction iterations in the predictor-coupling scheme. For this system, increasing the number of correction iterations increases the allowable time increment for stability. However, the savings offered by a larger time increment must be considered against the additional computational cost of the correction steps. For example, adding a correction step makes each time step about twice as expensive. Moreover, by introducing numerical damping in the BeamDyn module, the maximum stable time increments also increase for most cases.

For verification, we analyzed this case in ANSYS using 20 BEAM188 elements (and a point mass with spring and damper) and the time increment was 10^{-5} s. With this well-refined mesh in both time and space, the ANSYS solution is used as the benchmark in the following comparisons. The root and tip displacements of the beam can be found in Figure 4. For the BeamDyn results, the beam was discretized in by a single 5^{th}-order element, and the time increment was The time increment was 2×10^{-5} s. Good agreements can be observed between the benchmark solution and the BeamDyn results.

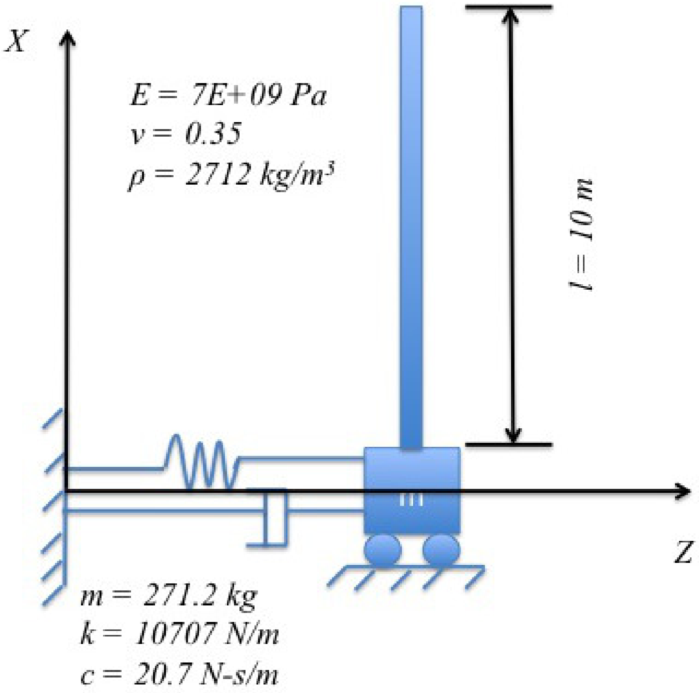


Figure 2: Schematic showing a 0.1 m × 0.1 m beam attached to a spring-mass-damper system with system properties and dimensions.

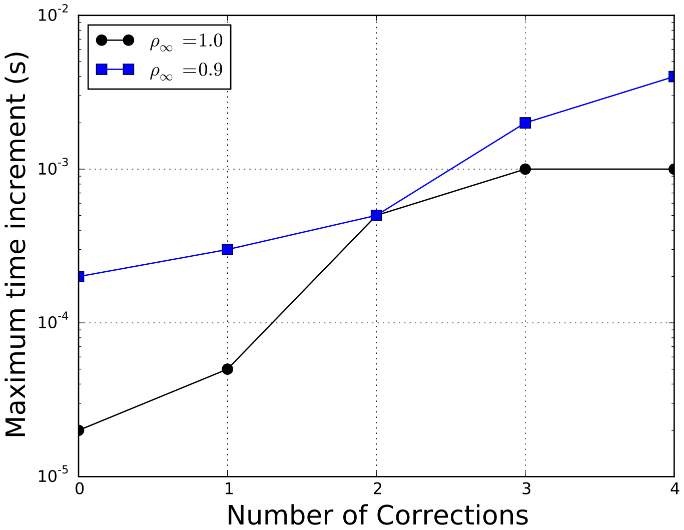
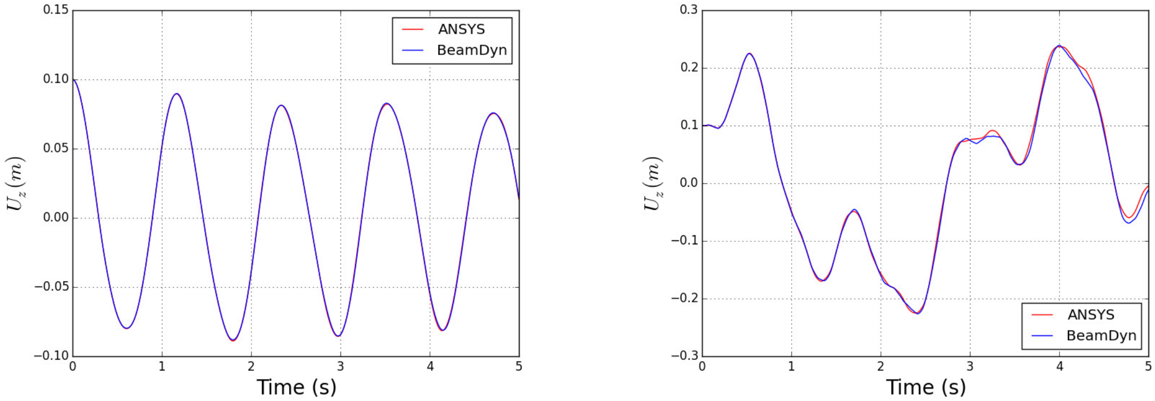


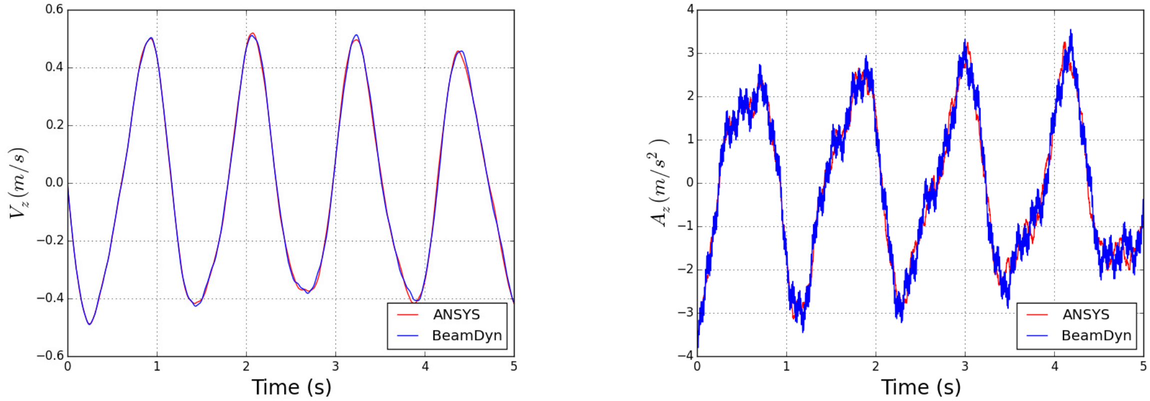
Figure 3: Maximum stable time increments as a function of the number of correction iterations in the predictor-corrector coupling algorithm for the beam-spring-mass-damper system. Results are shown with ($\rho_\infty = 0.9$) and without ($\rho_\infty = 1.0$) numerical damping in the BeamDyn module.



(a) Root Displacement

(b) Tip Displacement

Figure 4: Beam root and tip displacement histories calculated with ANSYS BEAM188 elements and BeamDyn.



(a) Root Velocity

(b) Root Acceleration

Figure 5: Beam root velocity and acceleration histories calculated with ANSYS BEAM188 elements and BeamDyn.

The root velocity and accelerations are also compared between the current model and the ANSYS benchmark solution in Figure 5. Again, good agreements are found in those quantities. It is noted that high-frequency fluctuations can be observed in the acceleration plot of BeamDyn results. Introducing numerical damping into the coupling algorithm could be a solution to further reduce such high frequency behavior, which is a subject for future work.

Next we studied the numerical performance of BeamDyn. The accuracy and efficiency of the present results of the coupled system were examined by the root-mean-square (RMS) errors, which aggregates the magnitudes of the errors in predictions for various times into a single measure of predictive performance. The error was calculated using

$$\varepsilon_{RMS} = \sqrt{\frac{\sum_{k=0}^{n_{max}} [U_z^k - U_b(t^k)]^2}{\sum_{k=0}^{n_{max}} [U_b(t^k)]^2}} \tag{18}$$

where $U_b(t)$ is the benchmark solution given by ANSYS. Figure 6 shows the RMS error of the tip displacements along Z direction as a function of total number of nodes for three different time increment sizes and number of predictor-correction times. The time increments for Case 1, Case 2, and Case 3 are $2E-5$, $5E-4$, and $1E-3$, respectively; and the number of corrections for the three cases are 0 (no correction), 2, and 3, respectively. It can be concluded that although other factors have been introduced into the simulation, exponential convergence rate which is a feature of LSFEs in space discritization can be observed. 5^{th}-order element (six nodes) being used in this case is a reasonable choice for a converged analysis. Moreover, results obtained by corrections are more accurate than those with smaller time increments but no correction.

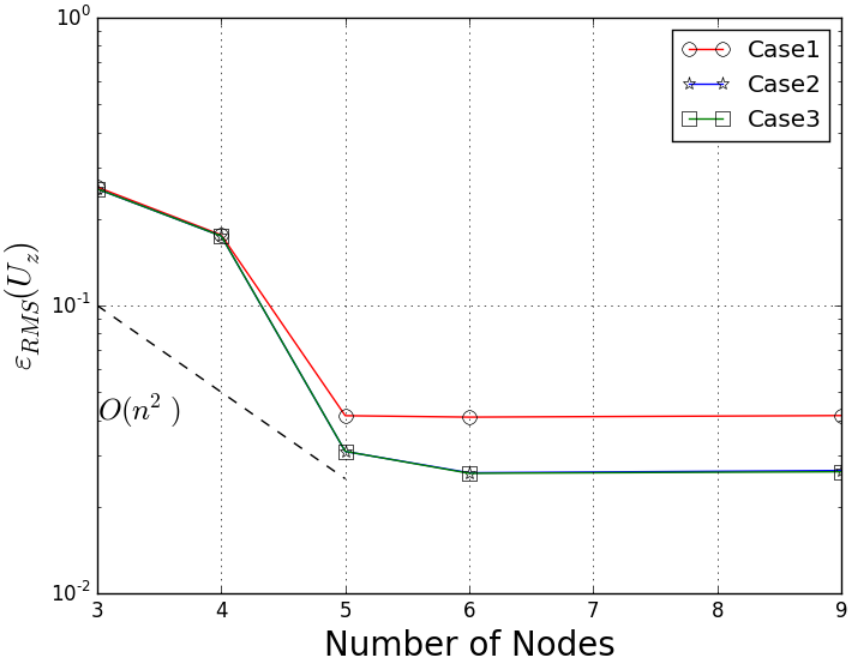


Figure 6: Normalized RMS error of U_z histories as a function of total number of nodes for three different time increments and number of corrections. The dashed line shows ideal second-order convergence.

B. Example 2: NREL 5-MW Wind Turbine

The second example is a coupled analysis of NREL 5-MW reference wind turbine system. The three blades are modeled by BeamDyn while the scheme discussed in this paper were implemented to couple BeamDyn to FAST.

Firstly we examines the numerical performance of two different quadrature methods, Gauss and trapezoidal quadratures, on this realistic blade analysis. A static cantilevered blade under a uniformly distributed force of magnitude $1E+04$ along the flap direction were analyzed. Figure 7 shows the results. Monotonous convergence can be observed for the trapezoidal quadrature results with increasing number of nodes; while the convergence curve for the Gauss quadrature was very jagged, depending on which stations being chosen and how these data being interpolated to the Gauss point. This can be even clearly observed from the mass calculation, see Figure 8: the constant mass by trapezoidal quadrature which includes all the cross-sectional data regardless of the finite element discretization; and the mass calculated by Gauss quadrature is more like a serendipity.

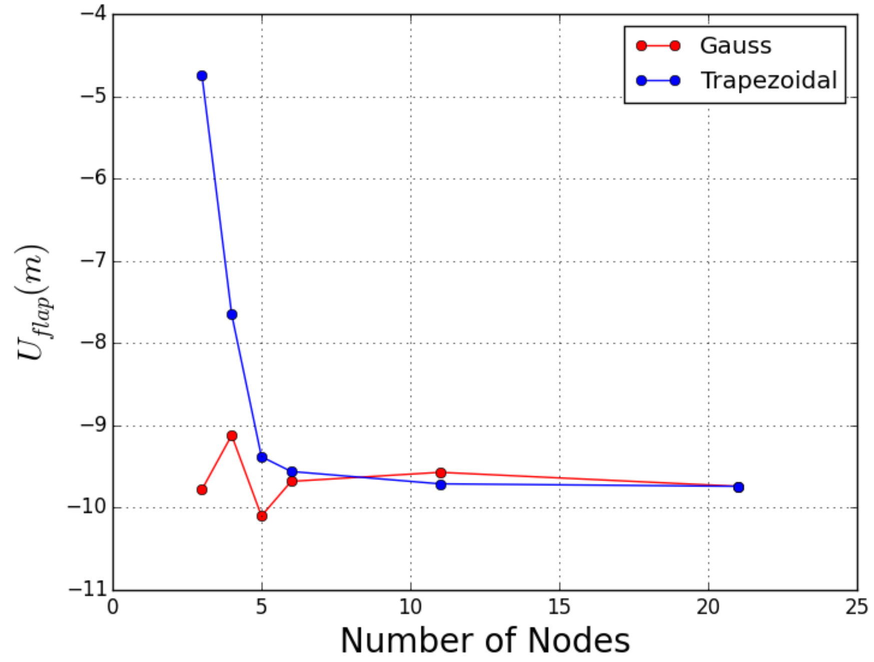


Figure 7: Tip deflections of a cantilevered NREL 5-MW blade under uniformly distributed load by Gauss and trapezoidal quadratures.

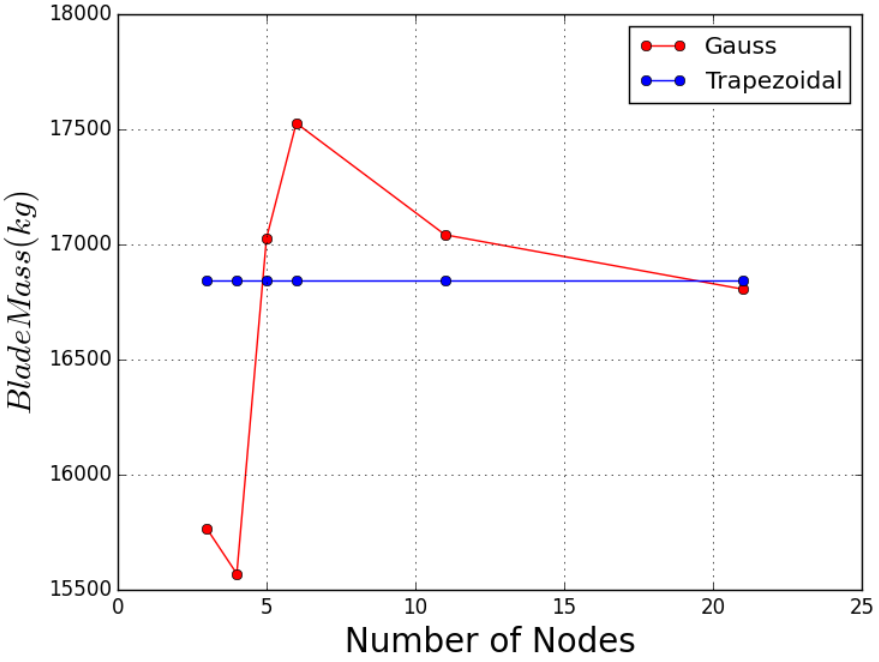


Figure 8: Total blade mass of a NREL 5-MW reference blade calculated by Gauss and trapezoidal quadratures.

Next the stabilities of BeamDyn in stand-alone and couple-to-FAST modes are studied. Figure 9 shows the maximum time step size versus the number of nodes. In the stand-alone mode, we are using FAST as the driver but turning off all the coupling options so that the blade is free rotating under gravity. For the coupled-to-FAST case, we conducted a fully-coupled wind turbine analysis under mean wind speed of $12\ m/s^2$ with turbulence, which is a certification test case in FAST package.

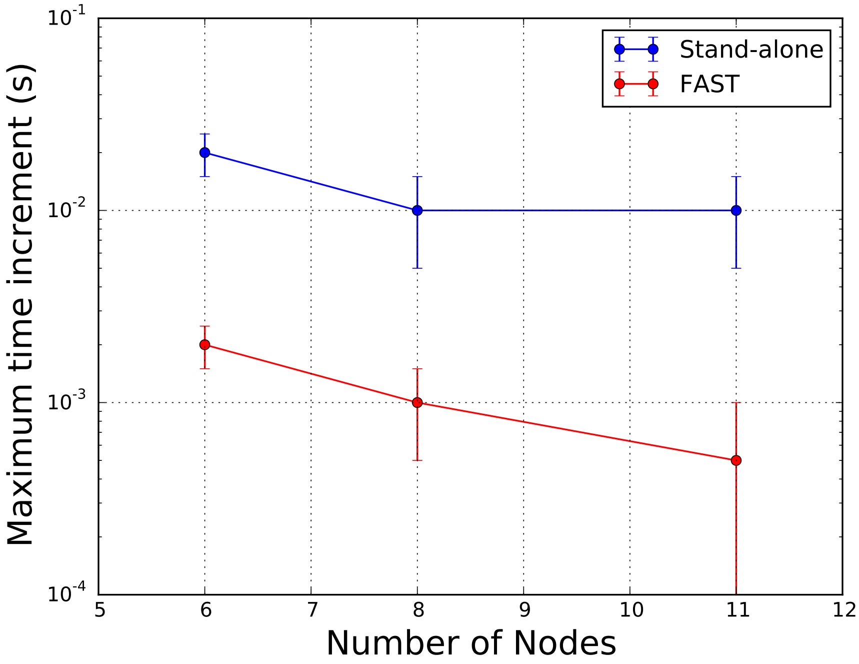


Figure 9: Maximum stable time increments vs number of FE nodes.

The convergence of the time integrator is also evaluated using the RMS error defined in Eq. (18). Again, a free spinning case which uses FAST as a driver code is studied for the convergence, see Figure 10. The blade is discretized by a single fifth-order element and the benchmark solution is obtained with a time step size of 0.001 seconds. Second-order convergence rate can be observed in the plot.

Finally we studied the performance of BeamDyn in the fully coupled FAST analysis. Figures 11 and **??** show the tip flap displacement histories under different time and space discretizations. Note that all the quantities studied here are resolved in the body-attached blade reference coordinate system following the IEC standard, where X direction towards the suction side of the airfoil, Y direction towards the trailing edge, and Z direction towards the blade tip from root. It can be observed that for this case, reasonable results can be obtained by the $2E-3$ s time increment and a single fifth-order element.

The tip displacements of blade 1 are shown in Figure 13.

We compared the results obtained by BeamDyn with those obtained by ElastoDyn for this case. Results are plotted from fifth seconds to remove the initial effects. Good agreements can be observed. It is noted that due to the Trapeze effect and centrifugal force considered in BeamDyn, the mean value the the axial displacement in Figure 13c by BeamDyn is different from ElastoDyn. Figure 14 shows the root reaction forces calculated by BeamDyn and ElastoDyn. Again, reasonable agreement can be found in these quantities.

IV. Conclusion

In this paper, we examined the beam theory and the coupling algorithm for partitioned mechanical system analysis. The full set of governing equations for BeamDyn are reviewed and the coupling schemes between BeamDyn and other mechanical modules are derived. For wind turbine analysis which features a large number of cross-sectional stations along blade axis, we implemented a trapezoidal numerical integration so that all the cross-sectional data, including inertial and stiffness matrices, can be used in the analysis. The users do not need to pick certain stations in a large set of data neither numerically interpolates them, which usually introduces errors. The new integration method, coupling algorithm as well as fully coupled mode have been validated in the numerical examples. The results obtained by trapezoidal quadrature were monotonously converged while those obtained by conventional Gauss integration were randomly converged depending on the stations chosen and interpolation method. In the coupled analysis, exponential

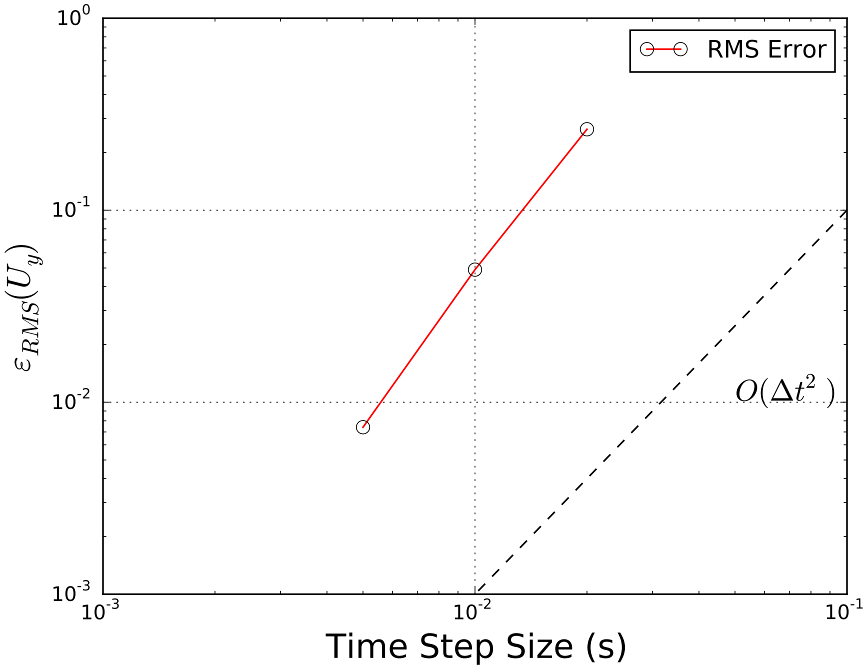


Figure 10: RMS error of tip edge displacement vs. time step size.

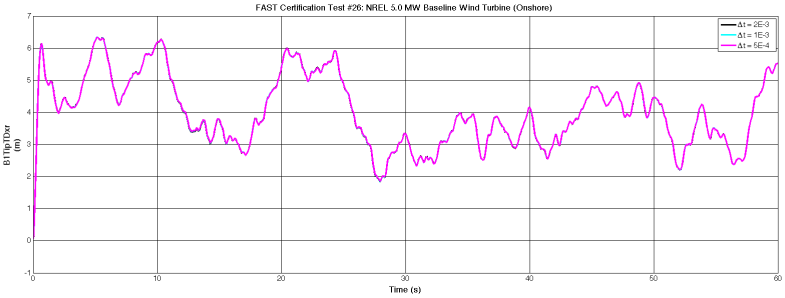


Figure 11: Blade tip deflection histories along flap direction obtained using different time increments.

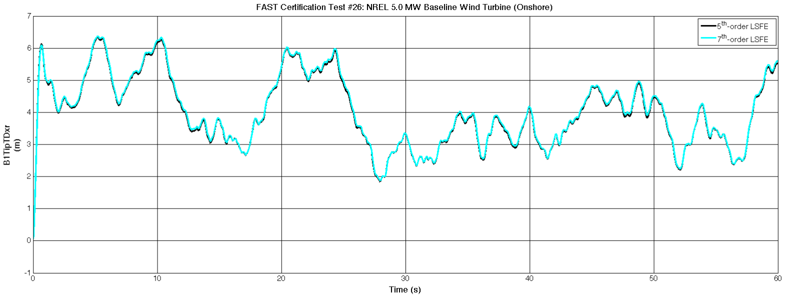
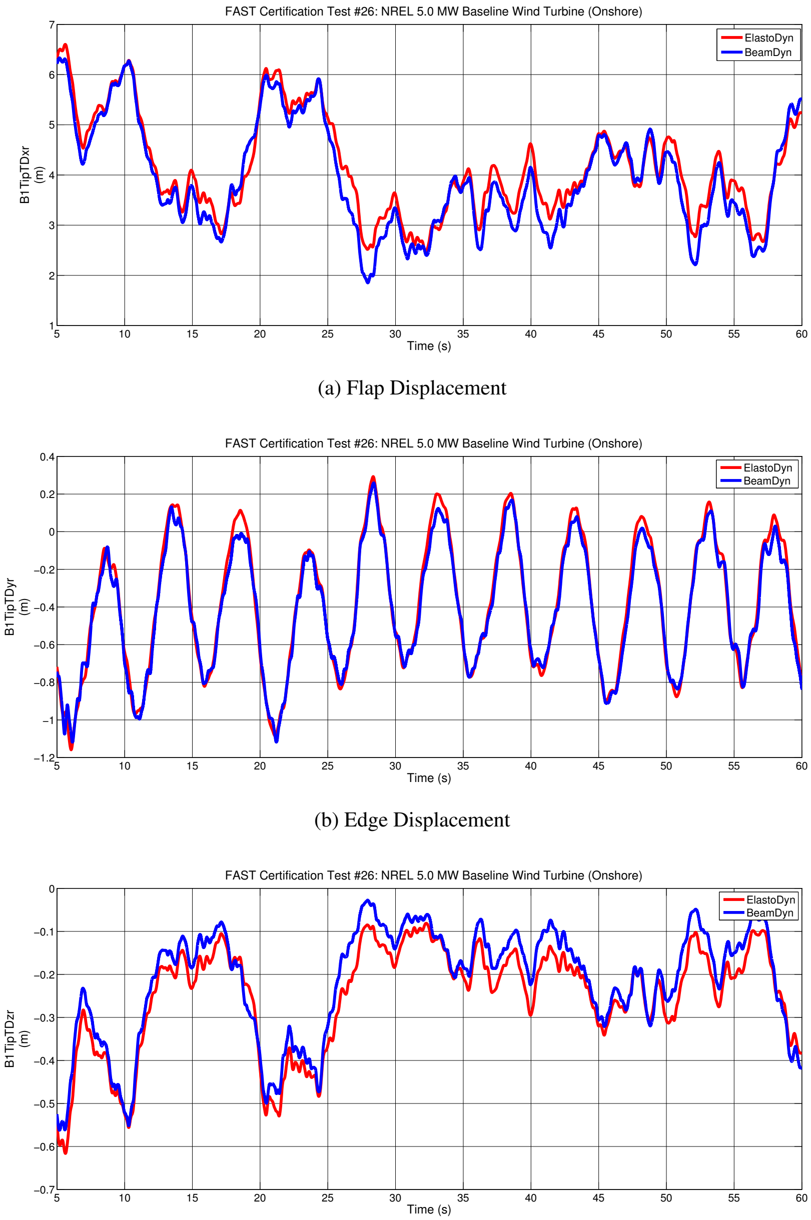


Figure 12: Blade tip deflection histories along flap direction obtained using different elements.

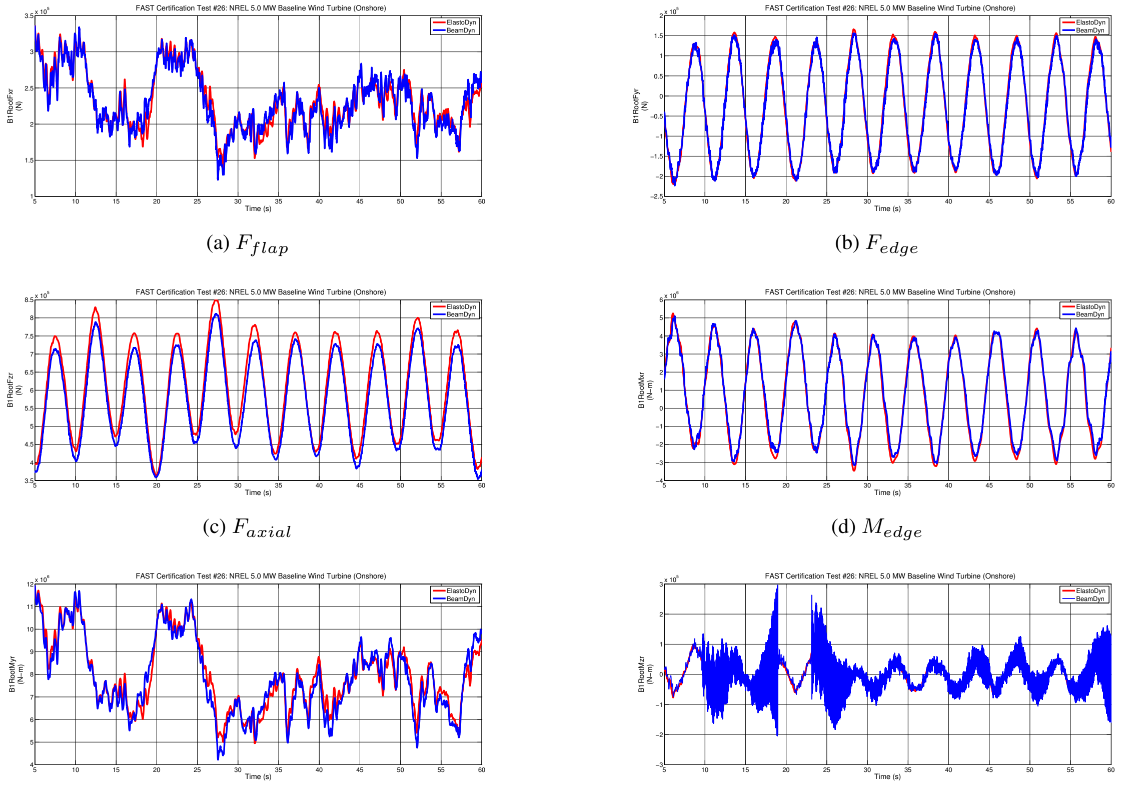


(a) Flap Displacement

(b) Edge Displacement

(c) Axial Displacement

Figure 13: Comparisons of blade 1 tip displacements between ElastoDyn and BeamDyn results



(a) F_{flap}

(b) F_{edge}

(c) F_{axial}

(d) M_{edge}

(e) M_{falp}

(f) $Torque$

Figure 14: Comparisons of root reaction loads between ElastoDyn and BeamDyn results

convergence rate can be seen in the BeamDyn results, which is a prominent feature of the LSFEs. In the full turbine analysis, reasonable agreement between BeamDyn and ElastoDyn results is found. More verification and validation results can be found in WAITING TO CITE SRI'S PAPER

Acknowledgments

This work was supported by the U.S. Department of Energy under Contract No. DE-AC36-08GO28308 with the National Renewable Energy Laboratory. Funding for the work was provided by the DOE Office of Energy Efficiency and Renewable Energy, Wind and Water Power Technologies Office.

References

[1] Reissner, E., "On one-dimensional large-displacement finite-strain beam theory," *Studies in Applied Mathematics LII*, 1973, pp. 87–95.

[2] Simo, J. C., "A finite strain beam formulation. The three-dimensional dynamic problem. Part I," *Computer Methods in Applied Mechanics and Engineering*, Vol. 49, 1985, pp. 55–70.

[3] Simo, J. C. and Vu-Quoc, L., "A three-dimensional finite-strain rod model. Part II," *Computer Methods in Applied Mechanics and Engineering*, Vol. 58, 1986, pp. 79–116.

[4] Hodges, D. H., *Nonlinear Composite Beam Theory*, AIAA, 2006.

[5] Wang, Q., Sprague, M. A., Jonkman, J., and Johnson, N., "Nonlinear Legendre spectral finite elements for wind turbine blade dynamics," *Proceedings of the 32nd ASME Wind Energy Symposium*, National Harbor, Maryland, January 2014.

[6] Jonkman, J. M., "The new modularization framework for the FAST wind turbine CAE tool," *Proceedings of the 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Grapevine, Texas, January 2013.

[7] Jonkman, J. and Jonkman, B., "FAST v8," `https://nwtc.nrel.gov/FAST8`, October 2013, [Online; accessed 29-OCTOBER-2014].

[8] Sprague, M., Jonkman, J., and Jonkman, B., "FAST modular wind turbine CAE tool: non matching spatial and temporal meshes," *Proceedings of the 32nd ASME Wind Energy Symposium*, National Harbor, Maryland, January 2014.

[9] Yu, W. and Blair, M., "GEBT: A general-purpose nonlinear analysis tool for composite beams," *Composite Structures*, Vol. 94, 2012, pp. 2677–2689.

[10] Bauchau, O. A., *Flexible Multibody Dynamics*, Springer, 2010.

[11] Ronquist, E. M. and Patera, A. T., "A Legendre Spectral Element Method for the Stefan Problem," *International Journal for Numerical Methods in Engineering*, Vol. 24, 1987, pp. 2273–2299.

[12] Wang, Q. and Sprague, M. A., "A Legendre spectral finite element implementation of geometrically exact beam theory," *Proceedings of the 54th Structures, Structural Dynamics, and Materials Conference*, Boston, Massachusetts, April 2013.