

A FINITE ELEMENT-MULTIBODY DYNAMICS CO-SIMULATION
METHODOLOGY APPLIED TO FAST

A Thesis

by

VISHVAS SAMUEL SURYAKUMAR

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Approved by:

Chair of Committee, Thomas W. Strganac
Committee Members, Jonathan D. Rogers
Alan B. Palazzolo
Head of Department, Rodney Bowersox

May 2013

Major Subject: Aerospace Engineering

Copyright 2013 Vishvas Samuel Suryakumar

ABSTRACT

A co-simulation methodology is explored whereby a finite element code and a multi-body dynamics code featuring flexible cantilevered beams can be coupled and interactively executed. The floating frame of reference formulation is used to develop the equations of motion. The floating frame is fixed at the blade root. Such a formulation results in ordinary differential equations without added algebraic constraints. A variety of loose coupling and tight coupling schemes are examined for this problem. To synchronize the coupling variables, a Gauss-Seidel type iterative algorithm is used. The resulting fixed-point iterations are accelerated using Aitken's adaptive relaxation technique. The methodology is evaluated for FAST, a wind turbine aeroelastic simulation code developed by NREL. As with FAST, many multi-body codes which can model flexibility employ modal methods. A proposed addition for FAST to simulate flexible effects using a finite element method module offers a potential to include a variety of non-linearities and also provides possibilities for using a high-fidelity aerodynamics module. The coupling schemes are compared and their applicability and limitations for different scenarios are pointed out. Results validating the approach are provided.

To my Father, Mother and Sister.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Thomas W. Strganac for the opportunity to pursue the graduate program at Texas A&M, for allowing me a great amount of freedom to work on my ideas and continually increasing my understanding of aeroelasticity. I also thank my committee members Dr. Jonathan D. Rogers and Dr. Alan B. Palazzolo for participating in the defense process and for their insightful questions and comments. Thanks to Dr. James D. Turner, whose course on computational multibody dynamics provided the starting point for this work.

I extend my gratitude to my colleagues at the Aero and Fluid Dynamics Lab, especially Yogesh Babbar for providing help in many areas and for helping me get started in the program. Thanks to Robert Ehrmann, Shalom Johnson and Andrew Beckett for their technical assistance in many projects. I would like to thank Arun Mangalam of Tao Systems, Inc for introducing me to industry-standard programming practices in MATLAB and LabView which helped me write concise efficient programs for this thesis. I also thank Dr. Jason M. Jonkman of NREL for providing technical support in the use of FAST.

Various funding sources that supported my master's program - Teaching/Research assistantships, the Regents graduate fellowship and the support from Tao Systems/NASA are gratefully acknowledged.

NOMENCLATURE

\mathbf{R}	Position of floating reference frame origin expressed in the inertial reference frame
θ	Orientation parameters for the floating reference frame.
\mathbf{q}_r	Vector describing the floating reference frame position and orientation
\mathbf{q}_f	Vector containing nodal displacements
\mathbf{q}_0	Vector containing undeformed nodal coordinates
\mathbf{q}	Vector containing nodal displacements
\mathbf{A}	Transformation matrix between floating and inertial reference frames
\mathbf{N}	Element shape function matrix
ρ	Mass density of beam
m_i	Mass of i th finite element
l_i	Length of i th finite element
a_i	Cross-sectional area of i th finite element
$\bar{\omega}$	Angular velocity of the floating reference expressed in its frame
$\bar{\mathbf{u}}$	Position vector to an arbitrary point in the flexible body
$\tilde{\bar{\mathbf{u}}}$	Skew-symmetric matrix formulation of the position vector
$\tilde{\bar{\mathbf{I}}}$	Skew-symmetric matrix formulation of the 3x3 identity matrix
\mathbf{J}	Inertia of the turbine with the blades
$\bar{\mathbf{J}}$	Inertia of the turbine without the blades
<i>Subscript</i>	
\mathbf{l}	Iteration count
\mathbf{n}	Time step
\mathbf{i}	Element number
\mathbf{j}	Blade number

TABLE OF CONTENTS

	Page
ABSTRACT	ii
DEDICATION	iii
ACKNOWLEDGEMENTS	iv
NOMENCLATURE	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	viii
LIST OF TABLES	x
1. INTRODUCTION	1
1.1 Multiphysics Simulations	1
1.1.1 Capability to Model Complex Physics	2
1.1.2 Modularity	2
1.1.3 Rapid Growth in Processing Power	3
1.2 FAST	7
1.2.1 System Model	8
1.2.2 FAST Modularization	9
1.3 Background	11
1.3.1 Floating Frame of Reference Approach	11
1.3.2 Co-simulation Schemes	13
1.4 Current Work	18
2. THEORY	21
2.1 Equations of Motion	21
2.2 Constraint Forces	26
2.3 Coupling Methods	28
2.3.1 Loose Coupling	29
2.3.2 Tight Coupling	32
2.3.3 Aitken's Acceleration	33
2.4 Software Implementation	37
3. RESULTS AND DISCUSSION	42
3.1 Turbine Model	42
3.2 Results	46
3.2.1 Full Tight Coupling	47
3.2.2 Partial Loose Coupling	51
3.2.3 Explicit Loose Coupling	54

	Page
3.2.4 Implicit Loose Coupling	61
3.2.5 Partial Tight Coupling	68
3.3 Summary	72
4. CONCLUSIONS	75
4.1 Concluding Remarks	75
4.2 Future Work	76
REFERENCES	79
APPENDIX A. MATRIX EXPRESSIONS FOR THE PLANAR ROTATION OF THE TURBINE	84
APPENDIX B. FAST DATA INPUTS	91

LIST OF FIGURES

FIGURE	Page
1.1 Moore’s Law - CPU Transistor Counts [1]	4
1.2 FAST Modules [2]	10
1.3 Flexible Multibody Formulations [3]	11
1.4 Jacobi Coupling	16
1.5 Gauss-Seidel Coupling	16
1.6 FAST-FEM Coupling Schemes Investigated	20
2.1 Coordinate Frames	22
2.2 Residual Behavior for FPI Without Acceleration	35
2.3 Residual Behavior for FPI with Aitken’s Acceleration	37
2.4 FAST-FEM Co-simulation Data Flow	41
3.1 AOC 15/50 Wind Turbine [4]	44
3.2 Natural Frequency Convergence with Number of Finite Elements	46
3.3 Full Tight Coupling with External Loads - Blade 1 Response	48
3.4 Full Tight Coupling with External Loads - Blade 2 Response	49
3.5 Full Tight Coupling with External Loads - Blade 3 Response	50
3.6 Full Tight Coupling Without External Loads - Blade 1 Response	51
3.7 Partial Loose Coupling - Blade 1 Response	52
3.8 Partial Loose Coupling - Blade 2 Response	53
3.9 Partial Loose Coupling - Blade 3 Response	54
3.10 Explicit Loose Coupling - Blade 1 Response - Hub Inertia = 9 Kgm^2	55
3.11 Explicit Loose Coupling - Blade 2 Response - Hub Inertia = 9 Kgm^2	56
3.12 Explicit Loose Coupling - Blade 3 Response - Hub Inertia = 9 Kgm^2	57

FIGURE	Page
3.13 Explicit Loose Coupling - Blade 1 Response - Hub Inertia = 9000 Kgm^2	58
3.14 Explicit Loose Coupling - Blade 2 Response - Hub Inertia = 9000 Kgm^2	59
3.15 Explicit Loose Coupling - Blade 3 Response - Hub Inertia = 9000 Kgm^2	60
3.16 Explicit Loose Coupling - Fixed Point Iteration Counts Comparison .	61
3.17 Implicit Loose Coupling - Blade 1 Response - Time Step = 0.001 secs	62
3.18 Implicit Loose Coupling - Blade 2 Response - Time Step = 0.001 secs	63
3.19 Implicit Loose Coupling - Blade 3 Response - Time Step = 0.001 secs	64
3.20 Implicit Loose Coupling - Blade 1 Response - Time Step = 0.01 secs .	65
3.21 Implicit Loose Coupling - Blade 2 Response - Time Step = 0.01 secs .	66
3.22 Implicit Loose Coupling - Blade 3 Response - Time Step = 0.01 secs .	67
3.23 Implicit Loose Coupling - Fixed Point Iteration Counts Comparison .	68
3.24 Partial Tight Coupling - Blade 1 Response	69
3.25 Partial Tight Coupling - Blade 2 Response	70
3.26 Partial Tight Coupling - Blade 3 Response	71
3.27 Partial Tight Coupling - Fixed Point Iteration Counts	72
A.1 Beam Element Coordinates	84

LIST OF TABLES

TABLE	Page
3.1 Turbine System Properties Summary - Modified AOC 15/50	43
3.2 Modes Output	45
3.3 Comparison of Coupling Schemes	73

1. INTRODUCTION

1.1 Multiphysics Simulations

With the advent of increasing computational resources and processing power, research in multiphysics simulations has received significant attention in recent decades. As can be observed in the relevant literature, much of this effort has been focused on fluid-structure interaction problems. Aeroelasticity among other multidisciplinary fields of research requires capabilities for simulation models which can accurately capture potential couplings between the different types of forces involved [5]. Most simulations involving aeroelasticity focus on the coupling between FEM(Finite Element Method) and CFD (Computational Fluid Dynamics)codes. The research described herein addresses a coupling between a third module, MBD(MultiBody Dynamics) with FEM and CFD in order to provide a comprehensive multidisciplinary simulation environment for the three disciplines concerned: multibody dynamics, structural dynamics and aerodynamics. Such an environment is envisioned for the wind turbine aeroelastic simulation environment FAST.

Multiphysics simulations are concerned with the simulation of multiple physical phenomena represented by different physical models. A partitioned form of such an analysis appears attractive when a physical model is too complex to be constructed in a monolithic form or when extensively validated modules handling physical models already exist. Several industries such as the wind turbine industry use computer-aided engineering (CAE) tools to a great extent for design and analysis. In wind energy as larger turbine designs are envisioned, the accuracy of such tools becomes critical along with a corresponding growth in importance of multiphysics simulations [2]. There are many motivating reasons why partitioned multiphysics simulations

have either become commonplace, or may become the norm in the near future, for CAE based industries [6]:

1.1.1 Capability to Model Complex Physics

Perhaps, the greatest advantage of partitioned simulations is their proven capabilities to handle complex physics without the need to construct a unified theoretical model accounting for the complex interactions between different physical models. An important example to cite is computational aeroelasticity, where a complex Navier-Stokes CFD module is linked with a high-fidelity computational structural dynamics (CSD) module.

1.1.2 Modularity

Since most multi-physics analysis are beyond the reach of simple analytical methods, two strategies for computational analysis are typically employed in the industry. One approach is to develop new monolithic simulation codes in-house. This strategy, frequently adopted by the rotorcraft industry leads to codes which are developed for specific configurations and lack flexibility [7]. Such an approach also renders these codes dysfunctional if an analysis for novel configurations is required or more advanced high fidelity analysis methods are pursued.

An alternative approach seeking to enforce modularity combines existing codes representing the different disciplines in an integrated computational environment. This strategy exploits the advantage that many of these codes have been extensively validated already. Modules representing the same physics of varying fidelity may be used as needed depending on available computational resources and accuracy. For instance, to simulate structural dynamics a non-linear finite element method module can be replaced easily with an equivalent modal method module if a faster, but low order model, is required without the need to rewrite the entire code. An advantage

related to the protection of intellectual property rights is offered in that modules can be executed in a black-box approach, or executed remotely across a network, thus ensuring data encapsulation [8]. Another advantage that modularization offers with respect to integration schemes is that individual modules can be executed at different rates and with different discretization schemes as needed. Modularization maintains many of the advantages of Object Oriented Programming concepts and mixed-language programming and also offers programmers the ease of developing, maintaining and sharing code. For instance, in the current study which seeks to co-simulate a FEM module with FAST, the FEM module code is written in MATLAB whereas FAST is written in FORTRAN.

1.1.3 Rapid Growth in Processing Power

Given the rapid development in processing power, high-fidelity analysis is increasingly less expensive. Partitioned simulations often require more computational resources than a corresponding high-fidelity monolithic simulation due to the requirement of subiterations if the coupling is numerically unstable. According to Moore's law [9] which models the development in processing power in past decades, the number of transistors on integrated circuits doubles approximately every two years as illustrated in Fig. 1.1.

In spite of the many advantages that modularization offers, several issues are of concern particularly those related to stability, convergence and accuracy. It is very possible that modules with observed stability in their integration schemes turn unstable when coupled to each other. In such a scenario, as also is the case for the current study, additional stabilization schemes will need to be employed. Such stabilization schemes typically involve many sub-iterations at each time-step to enforce synchronization of the coupling variables. The sub-iterations may prove costly if one

actual physical subsystems as in aeroelasticity, but can represent artificial subsystems (provide examples) for the sake of computational convenience and improved modularization. In this thesis, the interaction of rigid-body dynamics with flexible-body dynamics interaction is partitioned into separate rigid-body dynamics and flexible body dynamics modules. Then, such a partition would represent, based on the aforementioned definitions a artificial subsystem-bulk coupled simulation.

A multibody system is defined as a system of interconnected rigid or flexible bodies. The bodies may undergo large displacements and rotations. The bodies may be connected with joints or by force elements such as springs or dampers [11]. The form of equations that are typically obtained for this class of systems for an arbitrary set of generalized coordinates is a set of ordinary differential equations (assuming the flexible structure has been discretized using either modal methods or finite elements) along with a set of constraint equations coupled using a set of Lagrangian multipliers. If a minimum set of independent coordinates are obtained that can describe the system dynamics, then the set of ordinary differential equations would suffice to describe the system dynamics without the need to explicitly identify the constraints. Several methods can be used to arrive at the equations, the most common being the Lagrangian or Kane's method of dynamics. Depending on the requirements, multibody systems can be modeled with varying fidelity, ranging from from all bodies considered as rigid to including flexibility effects such as geometric and material non-linearities [12].

Several reasons are identified for the usefulness of including a finite element module within an aeroelastic simulation environment such as FAST. Firstly, if a more advanced high fidelity aerodynamics module is coupled to FAST, as pointed out in [7] the accuracy and fidelity of the aerodynamic module should match that of the structural dynamics module. Coupling a modal method with a high-fidelity Navier-

Stokes CFD module or coupling a finite element method with a simplified aerodynamics tool based on the Blade Element Momentum(BEM)theory, would result in a mismatch. Secondly, the Finite Element Method(FEM) provides an opportunity to include a variety of non-linearities apart from the already accounted for inertial nonlinearities (centrifugal and coriolis forces). However, the method developed in this research does not account for the non-linearities due to the large deformation of blades, though it offers a potential to account for these effects [8] . Thirdly, as argued by Bauchau et al [7] , given today’s rapid developments in computer hardware technology, enough processing power is available to simulate complex systems using FEM without resorting to the lower order modal methods. However, it is pointed out that partitioning the flexible dynamics into MBD(Multibody dynamics) and FEM modules is advantageous only if a few flexible bodies need to be simulated with FEM. These include wind turbine and rotorcraft blades. Partitioning provides the capability of using varying fidelity as needed. For example, a situation may be envisioned for a multibody system for a rotorcraft or wind turbine where only the blades are co-simulated with FEM, and all other flexible components are analyzed with simple modal methods or spring-mass-damper models.

As described in detail by Shabana et al [13], three methods are commonly used to integrate finite element and multibody systems: co-simulation or gluing algorithms (GAs), the finite element based direct integration method(FEBDI), and the multibody system based direct integration method (MSBDI). In this study, since modularization is a major requirement, the first method, namely gluing algorithms is pursued. GAs are classified as T-T,X-T and X-X, where X represents kinematic quantities and T represents forces [8,13,14]. For the current problem, both the MBD and FEM modules are capable of taking forces as inputs and providing kinematic quantities as outputs, hence the T-T type algorithm is implemented. In such an al-

gorithm, kinematic quantities are fed to a coordinator module which, in turn returns the forces that must be fed into each module.

A variety of aeroelastic simulation codes exist that are routinely used by the wind turbine industry for design and analysis. Some of the more widely used codes are FAST, Flex, and GH Bladed, as reviewed in Ref. [15]. The US wind industry uses mainly two codes: FAST and MSC.ADAMS-A2AD-Aerodyn. Many of these codes employ low order structural models such as modal methods and low order aerodynamic models such as BEM methods.

1.2 FAST

Fatigue, Aerodynamics, Structures and Turbulence (FAST) is an industry standard wind turbine aeroelastic simulation tool developed by NREL for two or three bladed horizontal axis wind turbines (HAWTs). FAST has been extensively validated and compared with experimental results [16]. It was also recently certified by Germanischer Lloyd (GL) Wind Energie GmbH, a leading wind turbine certifying agency, for on-shore wind turbine certification. The source code is written in FORTRAN and is available open-source for further modifications by developers. Apart from wind turbine aeroelastic simulation, FAST also comes with other capabilities such as linearization, controls, interfacing with Simulink and preprocessing FAST turbine data to enable direct import to the commercial multibody dynamics code MSC.ADAMS. In the linearization mode, FAST is able to provide a linear representation of the system dynamics about an operating point to aid in control system design and analysis. FAST allows the following control inputs: Blade pitch, Generator Torque, High Speed Shaft brake, Tip brakes and Nacelle Yaw. The control laws can be programmed in the provided user-defined subroutines. FAST also allows interfacing with Simulink/MATLAB using S-Functions. For this study, this functionality

is fully used in developing the interface code between FAST and a representative FEM code written in MATLAB.

1.2.1 System Model

FAST consists of a flexible multibody solver at its core. An aerodynamics module (AeroDyn) is co-simulated with the multibody dynamics solver. Currently, the interaction between these components is loosely coupled. A brief description of the structural dynamics and aerodynamics modules is given as follows. More details can be found in the FAST user manual [17].

1.2.1.1 Structural Dynamics Module

FAST models the turbine as a collection of rigid bodies and flexible bodies. Kane's method of dynamics [18] is used to derive the equations of motion. The flexible bodies include the turbine blades, tower and drive shaft. The flexibility of the blades and the tower are modeled using mode shapes. The tower permits two modes each in the longitudinal and transverse directions. The flexibility in the drive shaft is modeled using a spring-damper system. The nine rigid bodies include the earth, support platform, base plate, nacelle, armature, gears, hub, tail and structure that furls with the rotor. Although the modal approach for the blades assumes small deformations, the axial extensional effects due to bending are taken into account. FAST currently does not provide an option to include torsional modes. FAST also comes with a pre-processor program to compute mode shapes for a given rotor rpm. FAST provides the option to enable/disable degrees of freedom (DOF) associated with any flexible or rigid body. For the current analysis, the planar rotation of a simple three bladed turbine is investigated. Only the drivetrain DOF is enabled, with the flexibility in the blades modeled by an attached FEM module.

1.2.1.2 Aerodynamics Module

To model the aerodynamics, the rotor-aerodynamics module AeroDyn supplies the aerodynamic forces that are applied to the multibody model in a loosely-coupled manner. Despite the fact that the aerodynamic sources associated with wind turbines are extremely complicated, to aid in the design and analysis, the blade-element-momentum theory is commonly used. BEM is a simplified model based on two-dimensional quasi-steady theory that has found great use in the wind turbine industry [18]. The BEM theory seeks to unify two aerodynamics models:

- (1) - A blade element model wherein the forces acting on the blade are determined using the two dimensional lift and drag characteristics of the sectional airfoil.
- (2) - A momentum theory model where the forces acting on the blade are computed using mass and momentum conservation principles.

AeroDyn also includes the alternative Generalized Dynamic Wake (GDW) theory. AeroDyn has capabilities to model dynamic stall effects using the Beddoes-Leishman model.

Once the kinematic quantities, such as the rotor rotational and blade section displacements and velocities, are known they are fed to AeroDyn. AeroDyn successively iterates between both aerodynamic models (1) and (2) until convergence is achieved. The updated aerodynamic forces are then returned to FAST for integration towards the next time step, where the scheme is repeated. More details on the theory and programming interface can be found in the AeroDyn user guide [19].

1.2.2 FAST Modularization

The developers of FAST at NREL have recently investigated possibilities to improve the modularity of FAST. A summary of their research efforts are provided in Refs. [2, 20]. A means to achieving this goal is the development of a new FAST

modularization framework through which different modules as illustrated in Fig. 1.2 can be interconnected. The FAST modules are envisioned to be executed in either tight or loosely coupled schemes and with continuous or discrete states.

A major purpose of this thesis is to further modularize the structural dynamics module of FAST with two new modules. The new multibody dynamics module will handle the rigid body blade dynamics and dynamics of other rigid and/or flexible bodies. FAST with the blade flexibility disabled is considered here as the new MBD module. The new FEM module will handle exclusively the flexible-body dynamics of the blades. In keeping with the new framework and objectives discussed in Refs. [2,20] both loose and tight coupling strategies are investigated in this thesis.

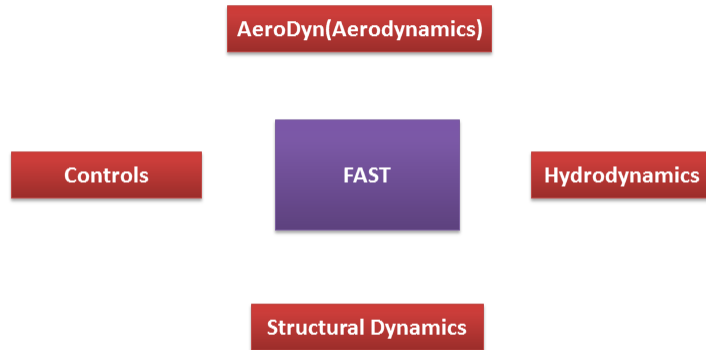


Fig. 1.2: FAST Modules [2]

1.3 Background

In this section, the background to the theory for the flexible multibody dynamics formulation and the co-simulation strategies further developed in the next chapter is introduced.

1.3.1 Floating Frame of Reference Approach

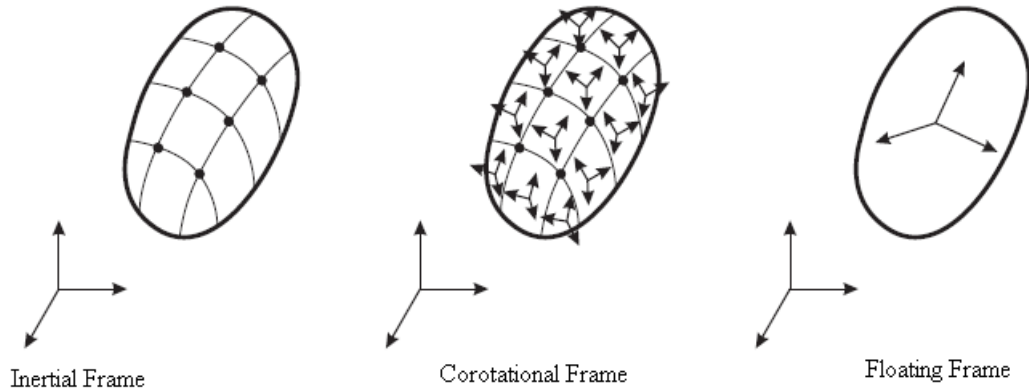


Fig. 1.3: Flexible Multibody Formulations [3]

To model the flexible multi-body behavior using finite elements, three approaches are commonly used as described by Brulls et al [3]. These are illustrated in Fig. 1.3. The inertial frame (IF) approach uses absolute nodal co-ordinates defined in an absolute reference frame. There is no distinction between the rigid and elastic co-ordinates. The co-rotational frame (CRF) approach separates the motion into the gross motion and small deformation motion of the finite element using an intermediate frame which follows the gross motion of the element. The floating frame of reference (FFR) approach uses a reference frame which follows the gross motion of

the flexible body. This results in a separation of the gross motion and the small deformation motion of the individual finite elements in the flexible body. The FFR formulation offers the most straightforward means for integrating a finite-element based structural dynamics module with a multi-body dynamics module such as FAST. FFR will be used to develop the equations of motion in this study. A critical requirement for the use of the FFR formulation is the need for the shape functions to account for rigid-body modes. This condition is naturally fulfilled by isoparametric elements. The shape functions used for beam elements, which can describe arbitrary rigid body translations, cannot describe arbitrary rigid-body rotations since infinitesimal rotations are used for the nodal rotational coordinates. However, the beam element can be assumed to be isoparametric if small deformations are considered, such that the rotations of the element described in the FFR can be considered infinitesimal. Herein, the assumption of small deformations for the nodal rotational displacements is equivalent to linearizing the terms involving these displacements in the kinematic equations. It is pointed out that the concept of small deformations is not to be confused with small strains. Large deformations with small strains is a possibility. For instance, a highly flexible beam can be appropriately discretized using finite elements such that local strains are small but the deformations with respect to the floating frame can still be large. The deformations in this context refer to the nodal displacements relative to the floating frame.

As stated, the FFR offers a method to separate the gross motion of the flexible body from the small deformation motion of the finite element. Once the finite element motion is described in the floating frame as is detailed in the next section, it is necessary to eliminate the rigid-body modes as this is being already accounted for by the FFR. For a general spatial motion, six conditions relating to the motion of the finite element nodal coordinates defined in the FFR need to be identified. Commonly

used reference frame conditions include the nodal-fixed axes and mean-axes [21]. The nodal-fixed axes provides a unique advantage to the objective of MBD-FEM co-simulation. If the nodal coordinates at the root are fixed at zero, i.e the floating frame is fixed at the blade root it is seen that the constraint equations enforcing the attachment of the flexible blade to the hub do not appear in the equations of motion. This leads to a system of ordinary differential equations which can be partitioned and solved separately using a MBD and a FEM solver.

1.3.2 Co-simulation Schemes

Several co-simulation schemes can be envisioned for the MBD-FEM interaction for FAST. For clarity, they can be classified based on a variety of criteria as briefly explained below. Schemes which require sub-iterations for this case are often slow. To accelerate the iterations Aitken's method is briefly discussed. Finally, performance parameters for the stabilization schemes are defined.

1.3.2.1 Multiphysics Architectures

In a monolithic architecture, a unified model of the system is represented in terms of a single set of differential equations and is advanced in time using a single integrator. In contrast, for a partitioned approach the different physical models encapsulated in software modules comprising the system are coupled in a variety of schemes ranging from loose coupling to tight coupling.

1.3.2.2 Solver Schemes

Several definitions exist in the literature concerning what constitutes loose or tight coupling. To conform to the FAST modularization framework, the definitions provided in Refs [2, 20] will be followed. Tight coupling is defined as the scheme where each module sets up the equations pertaining to its respective physics, but

the states are advanced in time using a solver common to all modules. For loose coupling, each module sets up the equations for the pertinent physics and also integrates the equations. The modules exchange coupling variables at pre-defined time intervals. As observed widely in literature, loose coupling schemes without any type of subiterations taking place at each time step are subject to numerical instability problems. In fluid-structure interaction problems, such schemes are found to fail if there are significant added-mass effects [22].

1.3.2.3 Integration Schemes

Time integration schemes are generally classified into explicit and implicit types [20]. With reference to the loose coupling scheme, an explicit scheme is defined as one in which the coupling variables at the current time step are used when advancing to the next time step. For example, consider two systems represented by \mathbf{x} and \mathbf{y} . Herein, an explicit scheme is represented by:

$$\mathbf{x}^{t_{n+1}} = \mathbf{F}(\mathbf{x}^{t_n}, \mathbf{y}^{t_n}) \quad (1.1)$$

$$\mathbf{y}^{t_{n+1}} = \mathbf{G}(\mathbf{y}^{t_n}, \mathbf{x}^{t_n}) \quad (1.2)$$

For an implicit coupling scheme, the coupling variables from the next time step are used in the current time step. Since coupling variables at the next time step are not known a priori, sub-iterations are typically used to facilitate the use of such a scheme.

The implicit coupling scheme can be represented generically by systems \mathbf{x} and \mathbf{y} :

$$\mathbf{x}^{t_{n+1}} = \mathbf{F}(\mathbf{x}^{t_n}, \mathbf{y}^{t_{n+1}}) \quad (1.3)$$

$$\mathbf{y}^{t_{n+1}} = \mathbf{G}(\mathbf{y}^{t_n}, \mathbf{x}^{t_{n+1}}) \quad (1.4)$$

In addition, explicit and implicit schemes could be defined in a similar manner for variables native to a module. As is observed widely in the literature and found in the current study, explicit schemes though computationally less expensive are also much less stable than implicit schemes.

1.3.2.4 Iteration Schemes

For stable solutions, sub-iterations are found to be necessary in all the schemes considered here which involve obtaining a partitioned solution from multiple solvers. Two widely used methods are the Jacobi and Gauss-Seidel approaches. In the Jacobi-type coupling, variables from the previous iteration are used directly for the current iteration in both modules as illustrated in Fig. 1.4. On the other hand, iterations carry forward in a staggered sense in the Gauss-Seidel approach. For instance, as illustrated in Fig. 1.5, the FEM module uses the latest update from the MBD module at the current iteration step. The Gauss-Seidel approach is faster in convergence since variables from the previous time step are used only once per iteration. The Jacobi method uses only the variables from the previous time step to advance the iteration without utilizing any current information. However, the Jacobi method is faster in execution time if both modules can be made to run in parallel [22]. In the research herein, the Gauss-Seidel approach is employed since a sequential programming approach is followed.

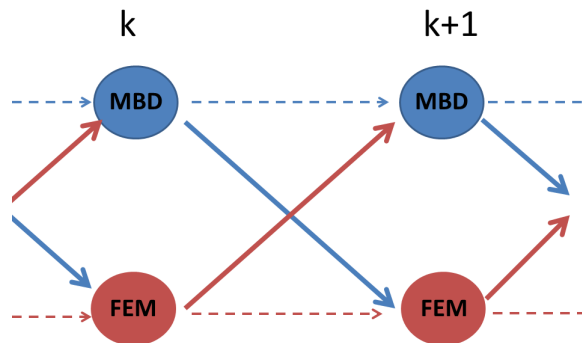


Fig. 1.4: Jacobi Coupling

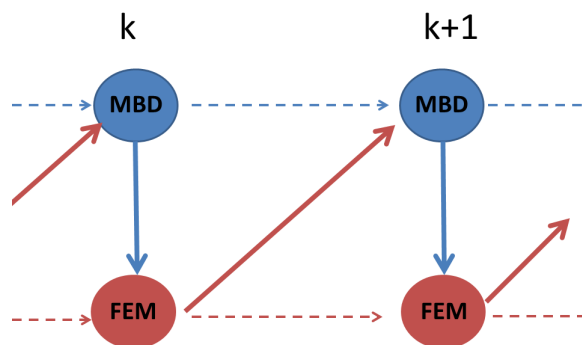


Fig. 1.5: Gauss-Seidel Coupling

1.3.2.5 Acceleration Schemes

An observation that can be made from Fig. 1.5 is that the iterations can be represented in as a fixed-point iteration (FPI) of the form:

$$\mathbf{w}^k = \mathbf{f}(\mathbf{w}^{k+1}) \tag{1.5}$$

where, \mathbf{w} represents the coupling variable for the MBD module that is native to the FEM module and k represents the iteration count. For the current MBD-FEM coupling problem, it is observed that fixed point iterations as represented in Eq.(1.5) are stable and convergent but require many sub-iterations to satisfy the tolerance limits. Therefore, an acceleration scheme known as the Aitken's relaxation method to accelerate the sub-iterations is explored for the current application. A detailed description and implementation of the algorithm is provided in the next section. The superior performance of this method compared with the fixed point iteration without acceleration is demonstrated.

1.3.2.6 Coupling Performance Parameters

The coupling schemes can be analyzed, evaluated and compared using the following performance parameters [22]:

Stability

Stability of the coupling scheme is defined as the property for which the residual, i.e., the error between successive estimates, remains bounded as the iterations advance.

Convergence

Convergence is defined as the tendency of the iteration to approach a fixed point as the iterations advance.

Accuracy

Accuracy is defined as the chosen limit in error in the solution obtained for the fixed-point iteration as described in Eq.(1.5). Tighter tolerances will require more sub-iterations.

1.4 Current Work

A method is developed for co-simulating a multi-body dynamics (MBD) solver such as FAST with a FEM solver using the FFR formulation. Such a formulation allows for the separation of the gross motion from the small deformation motion of the flexible body. The formulation however results in off-diagonal body reference-flexible coordinate coupling terms in the global mass matrix. This poses a challenge to co-simulate MBD and FEM codes in a partitioned manner due to numerical instability issues.

Due to this difficulty among others, for small deformation problems one approach is to neglect the coupling term involving the effect of elastic deformation on the rigid body motion. This procedure is referred to as the Linear Theory of Elastodynamics [11]. However, this approach may not be accurate for cases where the coupling between the rigid-body motion and elastic deformation is significant as in high speed, highly flexible or lightweight systems. Using the mean-axis reference conditions for the FFR, the coupling term does not appear in the equations of motion. Ryu et al [8, 23] exploit this fact in developing the Partition Iteration Method (PIM) to co-simulate MBD and FEM codes. However, a drawback of this method is that an additional MBD solver is needed for each flexible component. Also, enforcing the mean-axis reference conditions requires the simultaneous solution of the related constraint equations. A method is proposed herein for FAST which uses a blade-root nodal fixed floating frame of reference. This formulation allows the possibility of using an existing MBD code to solve for the rigid-body part of the flexible blade dynamics, while an FEM code can be used to solve for the small-deformation motion of the flexible blade. However, this formulation results in a system of equations that cannot be directly solved due to the off-diagonal reference-flexible coordinate

coupling terms in the mass matrix. More specifically, the partitioned equations employed by the MBD code requires nodal accelerations and the FEM code requires rigid-body accelerations (since the equations of motion are second order in time, only displacements and velocities are available at every time step).

A solution to this problem is proposed using an accelerated iterative algorithm. At every time-step the MBD and the FEM codes are iteratively executed until the residual chosen is reduced below a specified tolerance. This approach may be considered for integrating a general FEM code to an aeroelastic multibody dynamics code such as FAST in a multidisciplinary computational environment. It is pointed out that the formulation is valid only for multi-body systems which feature flexible cantilevered beams such as wind turbine or rotorcraft blades.

Figure 1.6 shows the organization of the different coupling schemes, with a similar philosophy followed in Ref. [20], that is implemented and tested in order to validate the formulation. The blocks in the chart are arranged bottom-to-top showing the progressive increase in the degree of coupling and fidelity. Accordingly, the results are compared with a full tight coupling case which is designated as the scheme with the most accurate results, though not necessarily the most computationally convenient from a modularization perspective.

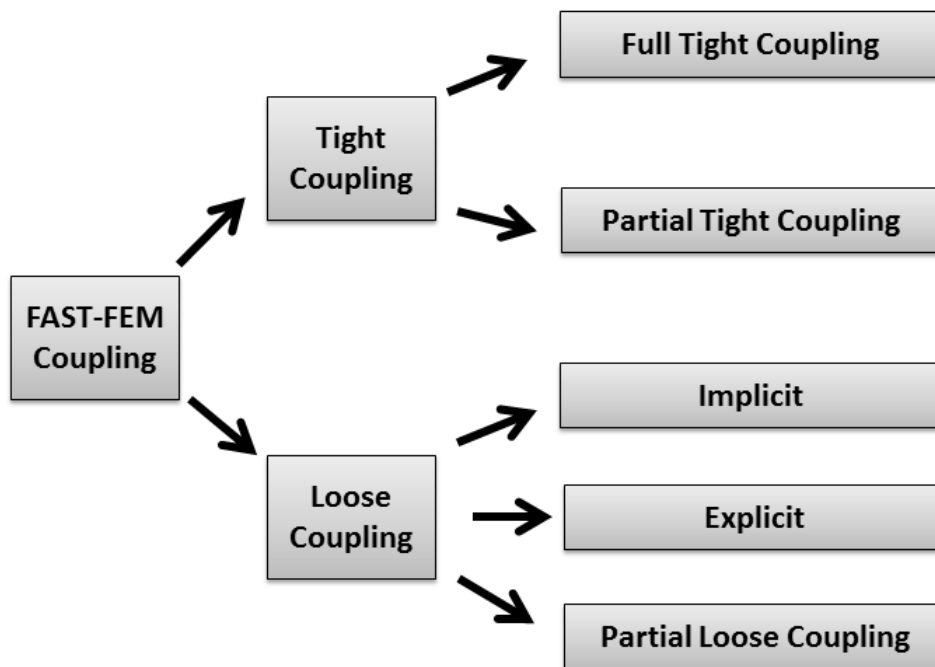


Fig. 1.6: FAST-FEM Coupling Schemes Investigated

2. THEORY

In this section the associated theory for the flexible-multibody dynamics problem using the FFR approach is elucidated and relevant equations outlined. With the equations known, the terms contributing to the coupling between rigid body motion and small deformation motion become apparent. Subsequently, coupling methods to account for these coupling effects are considered.

2.1 Equations of Motion

Using the FFR approach, the motion of the deformable body can be represented by two sets of coordinates: reference and elastic coordinates. The reference coordinates identify the position and orientation of the floating reference origin. The elastic coordinates identify the nodal displacements of the finite elements comprising the deformable body defined with respect to the body reference.

The derivation of the equations of motion is explained in detail by Shabana [24]. A brief outline is provided below: The configuration of the reference frame can be identified by the vector \mathbf{q}_r

$$\mathbf{q}_r = [\mathbf{R}^T \ \theta^T]^T \quad (2.1)$$

As illustrated in Fig. 2.1, the vector \mathbf{R} describes the position of the reference frame($X1',X2',X3'$) origin with respect to the inertial reference($X1,X2,X3$). The vector θ represents the orientation parameters defining the orientation of the body reference with respect to the inertial reference. The position vector \mathbf{r} describes an arbitrary point on the deformable body with respect to the inertial origin. The vector \mathbf{r} is expressed in the inertial reference frame and is given by:

$$\mathbf{r} = \mathbf{R} + \mathbf{A}\bar{\mathbf{u}} \quad (2.2)$$

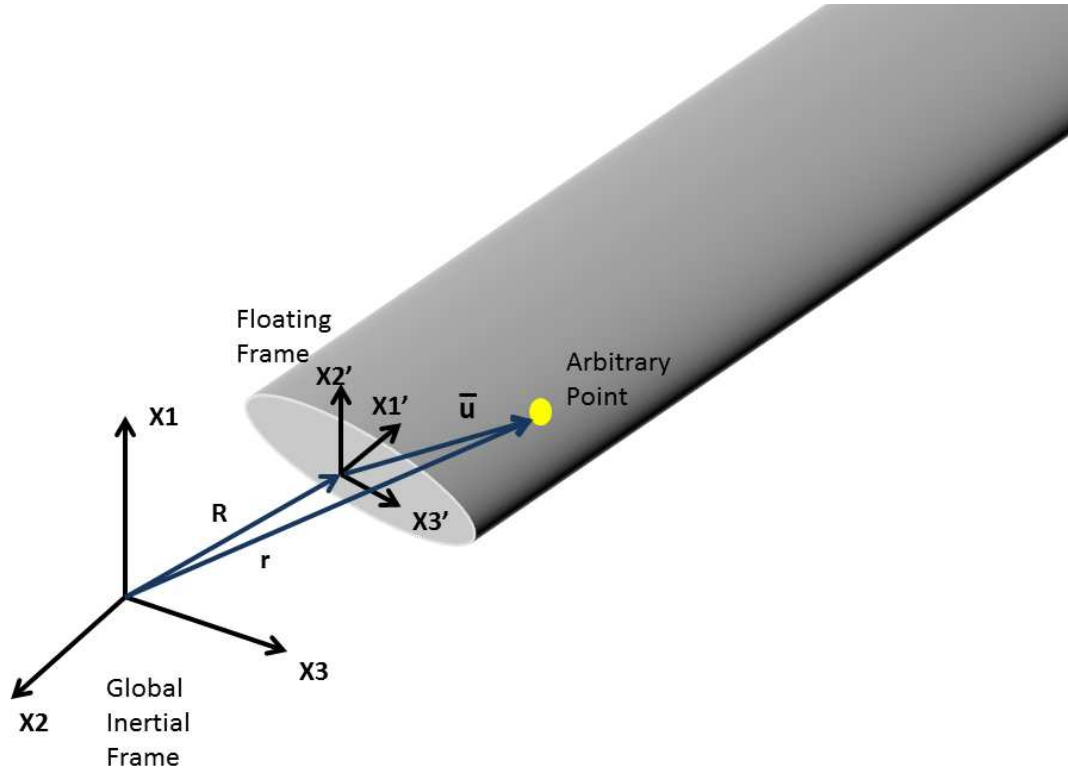


Fig. 2.1: Coordinate Frames

where $\bar{\mathbf{u}}$ is the position vector of the arbitrary point with respect to the reference frame origin and \mathbf{A} is the transformation matrix between the floating and inertial reference frame. The vector \mathbf{r} may be defined using the finite element shape functions as:

$$\mathbf{r} = \mathbf{R} + \mathbf{AN}(\mathbf{q}_0 + \mathbf{q}_f) \quad (2.3)$$

where \mathbf{N} is the shape function matrix for the finite element, \mathbf{q}_0 represents the undeformed nodal coordinates, and \mathbf{q}_f is the vector of nodal deformations.

The time derivative of the position vector \mathbf{r} is given as:

$$\dot{\mathbf{r}} = \dot{\mathbf{R}} - \mathbf{A}\tilde{\mathbf{u}}\bar{\mathbf{G}}\dot{\theta} + \mathbf{A}\mathbf{N}\dot{\mathbf{q}}_{\mathbf{f}} \quad (2.4)$$

where $\bar{\mathbf{G}}$ depends on the choice of the orientation parameters and is obtained from the relation:

$$\bar{\boldsymbol{\omega}} = \bar{\mathbf{G}}\dot{\theta} \quad (2.5)$$

Developing expressions for the kinetic energy and using virtual work principles by employing the above equations, the mass matrix, the quadratic velocity vector and the generalized forces for the elastic and external forces can be derived. The derivations are found in Refs. [8, 24]. Assembling the element level matrices results in the following equation:

$$\begin{aligned} & \begin{bmatrix} \mathbf{M}_{\mathbf{RR}} & \mathbf{M}_{\mathbf{R}\theta} & \mathbf{M}_{\mathbf{R}\mathbf{R}\mathbf{f}} \\ & \mathbf{M}_{\theta\theta} & \mathbf{M}_{\theta\mathbf{f}} \\ \text{symmetric} & & \mathbf{M}_{\mathbf{ff}} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{R}} \\ \ddot{\theta} \\ \ddot{\mathbf{q}}_{\mathbf{f}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & \mathbf{0} & \mathbf{0} \\ \text{symmetric} & & \mathbf{K}_{\mathbf{ff}} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \theta \\ \mathbf{q}_{\mathbf{f}} \end{bmatrix} + \begin{bmatrix} \mathbf{F}_{\mathbf{C}\mathbf{R}} \\ \mathbf{F}_{\mathbf{C}\theta} \\ \mathbf{0} \end{bmatrix} \\ & = \begin{bmatrix} (\mathbf{Q}_{\mathbf{e}})_{\mathbf{R}} \\ (\mathbf{Q}_{\mathbf{e}})_{\theta} \\ (\mathbf{Q}_{\mathbf{e}})_{\mathbf{f}} \end{bmatrix} + \begin{bmatrix} (\mathbf{Q}_{\mathbf{v}})_{\mathbf{R}} \\ (\mathbf{Q}_{\mathbf{v}})_{\theta} \\ (\mathbf{Q}_{\mathbf{v}})_{\mathbf{f}} \end{bmatrix} \end{aligned} \quad (2.7)$$

The assembly is accomplished using a Boolean transformation matrix [24] which

identifies the connectivity conditions between the elements. The mass and stiffness matrix is given by the first and second matrix respectively. The $\mathbf{F}_{\mathbf{C}}$ vector consists of the constraint forces. It is observed that $\mathbf{F}_{\mathbf{C}_f} = \mathbf{0}$ when the concept of kinematically admissible virtual displacements [25] is considered ($\delta \mathbf{q}_{\mathbf{f}_{\text{blade-root}}} = \mathbf{0}$). \mathbf{Q}_e is the vector containing generalized external forces and \mathbf{Q}_v is the quadratic velocity vector. The detailed expression for the terms in the matrices is given by Shabana [24].

For the studies herein, the turbine rotation confined to the rotor plane is assumed. Also, the blades are assumed to be connected to the turbine directly at the point of rotation, i.e. the blades are joined without a hub. Enforcing these assumptions, the representation Eq.(2.7) for the single blade can be simplified and written along with the FAST equations of motion for the three-bladed turbine system as follows:

$$\bar{\mathbf{J}}\ddot{\theta} = (\mathbf{Q}_e)_{\text{FAST}} - \sum_{\mathbf{j}} \mathbf{F}_{\mathbf{C}_{\theta_j}} \quad (2.8)$$

$$\mathbf{M}_{\theta\theta_j}\ddot{\theta} + \mathbf{M}_{\theta\mathbf{f}_j}\ddot{\mathbf{q}}_{\mathbf{f}_j} = \mathbf{F}_{\mathbf{C}_{\theta_j}} + (\mathbf{Q}_e)_{\theta_j} + (\mathbf{Q}_v)_{\theta_j} \quad (2.9)$$

$$\mathbf{M}_{\mathbf{f}_j\mathbf{f}_j}\ddot{\mathbf{q}}_{\mathbf{f}_j} + \mathbf{M}_{\mathbf{f}_j\theta}\ddot{\theta} + \mathbf{K}_{\mathbf{f}_j}\mathbf{q}_{\mathbf{f}_j} = (\mathbf{Q}_e)_{\mathbf{f}_j} + (\mathbf{Q}_v)_{\mathbf{f}_j} \quad (2.10)$$

$$j=1..3(\text{Number of blades})$$

where θ represents the reference coordinates which in this case is simply the rotor azimuth angle and $\mathbf{q}_{\mathbf{f}}$ represents the nodal deformation coordinates. Note that if the full three dimensional motion of the floating frame is considered, the equations are more involved since the kinematic relations between the chosen orientation parameters and angular velocities will need to be considered. Equation (2.8) represents the equations of motion for the portion of the turbine without the blades. The effect of the blades is communicated through the constraint forces $\sum_{\mathbf{j}} \mathbf{F}_{\mathbf{C}_{\theta_j}}$. Accordingly, $\bar{\mathbf{J}}$

represents the inertia of the turbine components without the blades. $(\mathbf{Q}_e)_{\text{FAST}}$ represents the external forces acting on the turbine drive-shaft. As discussed in the next chapter, these forces include the drive-shaft stiffness and damping related forces. It is observed that when equations 2.8 and 2.9 are added, the constraint forces cancel out since these are internal reaction forces. The following set of equations result, written in matrix form:

$$\begin{aligned}
 & \begin{bmatrix} \mathbf{J} & \mathbf{M}_{\theta f_1} & \mathbf{M}_{\theta f_2} & \mathbf{M}_{\theta f_3} \\ & \mathbf{M}_{ff_1} & 0 & 0 \\ & & \mathbf{M}_{ff_2} & 0 \\ & & & \mathbf{M}_{ff_3} \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\mathbf{q}}_{f_1} \\ \ddot{\mathbf{q}}_{f_2} \\ \ddot{\mathbf{q}}_{f_3} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & \mathbf{K}_{ff_1} & \mathbf{0} & \mathbf{0} \\ & & \mathbf{K}_{ff_2} & \mathbf{0} \\ & & & \mathbf{K}_{ff_3} \end{bmatrix} \begin{bmatrix} \theta \\ \mathbf{q}_{f_1} \\ \mathbf{q}_{f_2} \\ \mathbf{q}_{f_3} \end{bmatrix} \\
 & \begin{bmatrix} \textit{symmetric} & & & \\ & & & \\ & & & \\ & & & \textit{symmetric} \end{bmatrix} \\
 & = \begin{bmatrix} (\mathbf{Q}_e)_\theta \\ (\mathbf{Q}_e)_{f_1} \\ (\mathbf{Q}_e)_{f_2} \\ (\mathbf{Q}_e)_{f_3} \end{bmatrix} + \begin{bmatrix} (\mathbf{Q}_v)_\theta \\ (\mathbf{Q}_v)_{f_1} \\ (\mathbf{Q}_v)_{f_2} \\ (\mathbf{Q}_v)_{f_3} \end{bmatrix} + \begin{bmatrix} (\mathbf{Q}_e)_{\text{FAST}} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \\
 & \tag{2.12}
 \end{aligned}$$

The above equation offers a possibility to partition the motion of the flexible multibody. The flexibility effects of the blade can now be modeled using the finite element method. The remaining portion of the dynamics including the rigid body dynamics of the blade can be modeled by FAST. Therefore, $\mathbf{J} = \bar{\mathbf{J}} + \sum_j \mathbf{M}_{\theta\theta_j}$ now represents the inertia of the turbine including the blades. \mathbf{J} is time varying due to the time-dependent blade deflections. This is done by implementing each row of

Eq.(2.12) following the first equation in a finite element code. The first equation row is implemented in FAST. It is observed that the resulting equations from Eq.(2.12) form a set of ordinary differential equations which are much easier to handle than the differential algebraic equations (DAEs) which are obtained for other approaches such as the mean-axis floating frame of reference [23] [8]. The expressions for the terms in the matrices are provided in Appendix A.

It is also seen that the equations related to FAST and FEM are coupled by terms in the mass matrix. In addition, coupling also exists in the external force vector, \mathbf{Q}_e , which in this case is populated with aerodynamics forces and the quadratic velocity vector \mathbf{Q}_v .

2.2 Constraint Forces

In this section, it is demonstrated that the constraint forces do not appear in the finite element equations if the floating frame is fixed to the blade root. Virtual work principles are used. The generalized external forces are also derived [24]. The virtual work of a given external force vector \mathbf{F} acting at a point in the flexible body and associated with the arbitrary virtual displacement [25] is given by:

$$\delta\mathbf{W} = \mathbf{F} \cdot \delta\mathbf{r} \quad (2.13)$$

From Eq.(2.3), and using the shape function matrix \mathbf{N} (defined in Appendix A), a variation in \mathbf{r} is given by:

$$\delta\mathbf{r} = \delta\mathbf{R} + \frac{\partial\mathbf{A}\bar{\mathbf{u}}}{\partial\theta}\delta\theta + \mathbf{A}\mathbf{N}\delta\mathbf{q}_f \quad (2.14)$$

The generalized forces is obtained as :

$$\mathbf{F}_R = \mathbf{F} \quad (2.15)$$

$$\mathbf{F}_\theta = \mathbf{F}^T \frac{\partial \mathbf{A} \bar{\mathbf{u}}}{\partial \theta} \quad (2.16)$$

$$\mathbf{F}_{\mathbf{q}_f} = \mathbf{F}^T \mathbf{A} \mathbf{N} \quad (2.17)$$

If the floating frame is fixed at the root of the blade, i.e. nodal deflections at the root are forced to be zero ($\delta \mathbf{q}_{\mathbf{fblade-root}} = \mathbf{0}$), then the constraint forces do not appear in the finite element equations since the contribution to the virtual work is zero. As discussed in Ref. [25], by using the concept of kinematically admissible virtual displacements in contrast to arbitrary virtual displacements, the constraint forces do not appear in the finite element equations of motion. It is also noted from Eq.(2.17), that the generalized forces that are applied to the FEM module are the external forces expressed in the floating frame. $\bar{\mathbf{u}}$ in Eq.(2.17)(Generalized forces in FAST) includes the effect of deformations, which are obtained from the FEM module.

For the planar rotation problem, the transformation matrix \mathbf{A} is given as:

$$\mathbf{A} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

and therefore, $\mathbf{d}/\mathbf{d}\theta(\mathbf{A}\bar{\mathbf{u}}) = \mathbf{A}_\theta \bar{\mathbf{u}} \dot{\theta}$, with \mathbf{A}_θ given as:

$$\mathbf{A}_\theta = \begin{bmatrix} -\sin(\theta) & -\cos(\theta) \\ \cos(\theta) & -\sin(\theta) \end{bmatrix}$$

Similarly, it can be shown that the constraint moments do not appear in the equations of motion for the nodal coordinates.

Consider the virtual work due to moments M_θ due to an arbitrary infinitesimal virtual displacement θ_T acting on a finite element node:

$$\delta\mathbf{W} = \mathbf{M}_\theta \cdot \delta\theta_{\mathbf{T}} \quad (2.20)$$

where

$$\delta\theta_{\mathbf{T}} = \delta\theta + \delta(\theta_{\mathbf{f}}) \quad (2.21)$$

$\theta_{\mathbf{f}}$ is the nodal rotations expressed in the inertial frame. Equation(2.21) is valid since infinitesimal changes in angles are considered, hence the vector sum of rotations are allowed. If nodal deflections at the blade root are forced to be zero, then the contribution of the nodal deflections to the virtual work is also zero. Therefore, the constraint moments do not appear in the finite element equations of motion.

2.3 Coupling Methods

Several coupling strategies are explored to integrate FAST and the FEM solver interactively ranging from loosely coupled to full tight coupling methods. Equation (2.12) cannot be solved directly by either FAST or the finite element code since the equations are second order in time and $\ddot{\mathbf{q}}_{\mathbf{f}}$ and $\ddot{\theta}$ are not known a priori. Only the velocities and displacements are available at every time step. In this section, coupling methods are listed in increasing order of fidelity and complexity. FAST and the FEM

module interact in a Gauss-Seidel manner while executing iterations. In the schemes considered, with the exception of the full tight coupling scheme, the $\mathbf{M}_{\theta f_1} \ddot{\mathbf{q}}_{f_1}$, $\mathbf{M}_{\theta f_2} \ddot{\mathbf{q}}_{f_2}$ and $\mathbf{M}_{\theta f_3} \ddot{\mathbf{q}}_{f_3}$ terms in Eq.(2.12) are taken to right side of the equation in both the FAST and FEM modules and treated as pseudo-loads acting on the beam.

2.3.1 Loose Coupling

In the loose coupling scheme, the FAST and FEM modules set the state equations and integrates the equations with separate solvers. Coupling variables are exchanged between the modules at fixed time intervals. As mentioned earlier, Eq.(2.12) cannot be directly integrated in time. One possible option is to use the unknown accelerations from the previous time step. But, based on numerical experiments, even if a very small time step is used, an unstable response is observed indicating the scheme is unstable. Certain modifications are suggested as detailed below to enable the use of the loose coupling approach.

2.3.1.1 Partial Loose Coupling

The partial loose coupled method referred to as the Linear Theory of Elastodynamics [24] neglects the $\mathbf{M}_{\theta f_1} \ddot{\mathbf{q}}_{f_1}$, $\mathbf{M}_{\theta f_2} \ddot{\mathbf{q}}_{f_2}$ and $\mathbf{M}_{\theta f_3} \ddot{\mathbf{q}}_{f_3}$ terms in Eq.(2.12). At every time step the FAST solver is executed first and the rigid body motion information, $\ddot{\theta}$ and $\dot{\theta}$, is then fed into the FEM solver. Such an approach would be satisfactory if the flexible-rigid body motion coupling is small. This method offers the least effort in terms of integrating FAST and the FEM module although its applicability and accuracy is limited due to the reduction of the two-way coupling to a one-way data flow for the terms involving acceleration.

2.3.1.2 Explicit Loose Coupling

As discussed in the Introduction chapter, in the explicit loose coupling scheme the information from the next time step is not used when integrating in the current time step. As mentioned earlier an explicit scheme in which the unknown accelerations are used from the previous time step is unstable. However, improved numerical stability can be achieved if the unknown accelerations in Eq.(2.12) are found for at every time step. With the accelerations known, the equations are integrated in time explicitly. Let a guess value for $\ddot{\theta}$ be substituted in the last three rows of Eq.(2.12), i.e. the guess value is given as input to the FEM modules handling the blades. Using the FEM module, $\ddot{\mathbf{q}}_{\mathbf{f}}$ can be obtained and is substituted in the first row. Using FAST, an updated value for $\ddot{\theta}$ is obtained. Thus, it is seen that this procedure can be represented as a fixed point iterating scheme of the type:

$$\ddot{\theta}^{l+1} = \mathbf{f}(\ddot{\theta}^l) \quad (2.22)$$

It is noted that it is more advantageous to use the rigid body acceleration vector $\ddot{\mathbf{q}}_{\mathbf{r}}$ as the variable in the scheme instead of the alternative $\ddot{\mathbf{q}}_{\mathbf{f}}$ since the vector size of the former will usually be much smaller than the size of the latter. The guess value is obtained from the previous time step. For the first time step, the guess value is obtained from FAST enabled with modal methods. It is determined however from numerical experiments that this scheme as is described in Eq.(2.22) is very slow. To accelerate the convergence, Aitken's method as described in the next sub-section is used. It is noted that though the sub-iterations are performed at every time step, the equations are still integrated explicitly in time. Thus, above an associated critical time step, the scheme will be numerically unstable.

2.3.1.3 Implicit Loose Coupling

As previously stated, the explicit integration scheme has numerical instability issues. For numerical schemes which are unconditionally stable or at least stable for large time steps, implicit coupling schemes are suggested. A fundamental difference of this scheme with that of loose coupling is that the equations are integrated implicitly in time. Here, the interface variables from the next time step is used in the evaluation at the current time step. Using the Gauss-Siedel iteration approach, the following implicit scheme is suggested:

$$[\mathbf{q}_f^{l+1}, \dot{\mathbf{q}}_f^{l+1}, \ddot{\mathbf{q}}_f^{l+1}] = \mathbf{FEM}(\dot{\theta}^l, \ddot{\theta}^l, \mathbf{q}_f^n, \dot{\mathbf{q}}_f^n) \quad (2.23)$$

$$[\theta^{l+1}, \dot{\theta}^{l+1}, \ddot{\theta}^{l+1}] = \mathbf{MBD}(\theta^n, \dot{\theta}^n, \mathbf{q}_f^{l+1}, \dot{\mathbf{q}}_f^{l+1}, \ddot{\mathbf{q}}_f^{l+1}) \quad (2.24)$$

A fixed point iterative scheme can be written for the above equations as:

$$\ddot{\theta}^{l+1} = \mathbf{MBD}(\mathbf{FEM}(\ddot{\theta}^l)) \quad (2.25)$$

The scheme is iteratively solved and stabilized using the methods described in the next section until convergence is reached. A tight coupling scheme allows the possibility of using larger time steps with the scheme retaining numerical stability. It is pointed out that the sub-iterations are carried out only for $\ddot{\theta}$ as numerical experiments show that the coupling involving the acceleration terms are found to be unstable. The partial loose coupling scheme which does not include the coupling between the acceleration terms is found to be stable (Refer the Partial Loose coupling section of the Results chapter) indicating the coupling terms involving acceleration are unstable. As with the explicit coupling scheme, the sub-iterations are accelerated

using Aitken’s method.

2.3.2 Tight Coupling

In the tight coupling scheme, the FAST and FEM modules develop the state equations, and these equations are integrated in time using a common solver. Two schemes can be considered for this case based on the level of software intrusion as listed below:

2.3.2.1 Partial Tight Coupling

In this scheme, in a similar manner as the explicit loose coupling scheme, the unknown accelerations are determined in each time step using a fixed-point iterating scheme accelerated using Aitken’s method. Then, the known accelerations are integrated in time using one common solver.

2.3.2.2 Full Tight Coupling

For the full tight coupling scheme, to overcome the problem of the unknown accelerations at each time step, the full system mass matrix is constructed. This method, though the most accurate and computationally least expensive, is also the most intrusive in the sense that the turbine system inertial properties must be known to construct the mass matrix. The forces acting on the rigid-blade turbine system may be directly obtained from FAST using a software data link or may also be obtained from the FAST mass matrix and acceleration data. The forces acting on the flexible blade can be similarly be obtained from the FEM module. Then, the system accelerations are obtained by inverting the mass matrix in Eq.(2.12). It is pointed out that the full tight coupling case is similar to a monolithic simulation with the difference being that the equations are developed in separate modules. Therefore, unlike the previously discussed schemes, iterations are not performed at

every time-step.

2.3.3 Aitken's Acceleration

As observed with many coupled simulation problems, the interactions between the modules - FAST and FEM - can be represented as a fixed point iteration problem. These iterations would occur at every time-step, in order to synchronize the coupling variables. The iterations by themselves may or may not converge. Even if they do converge, the number of iterations needed to satisfy a tolerance limit may be prohibitively expensive. Therefore, acceleration techniques such as the Aitken's acceleration method are used to improve the convergence rates. A brief outline of the derivation is provided as follows as developed in Ref. [26] and [27].

As an aid to understanding the acceleration technique, consider first a representative fixed point iteration problem, where \mathbf{w} represents the interface variable. With respect to the coupling schemes described in the previous section, \mathbf{w} represents the reference frame motion variable vector (accelerations):

$$\mathbf{w}^{l+1} = \mathbf{F}(\mathbf{w}^l) \quad (2.26)$$

An improvement in stability and convergence characteristics can be obtained if some amount of relaxation is used in the form of a linear interpolation of the updated value from the Fixed-point iteration and the value from the previous iteration as shown below. α in Eq.(2.28) is commonly called the relaxation factor.

$$\tilde{\mathbf{w}}^{l+1} = \mathbf{F}(\mathbf{w}^l) \quad (2.27)$$

$$\mathbf{w}^{l+1} = \alpha \tilde{\mathbf{w}}^{l+1} + (1 - \alpha) \mathbf{w}^l \quad (2.28)$$

As can be observed, a small α results in the value of the previous iteration being dominant and a large α results in the updated value obtained from the Fixed-point iteration being more dominant. In case the iterations are unstable, under-relaxation relaxation is suggested. i.e. $0 < \alpha < 1$. This would result in a more stable scheme with an increase in the number of iterations, since only a small fraction of the update from the Fixed-point iteration is used. On the other hand, if the iterations appear stable, over-relaxation ($\alpha > 1$) can be considered to accelerate the convergence. Even though this method appears straight-forward and easy to implement, an obvious drawback is that the relaxation factor is chosen on a trail and error basis.

To circumvent these issues, Aitken's acceleration method [27] [22] which adaptively adjusts the relaxation factor based on information from previous iterations is explored for the current problem. A dramatic improvement in performance is observed when compared with the nominal fixed point iteration without acceleration.

For the current application, since the planar rotation of the turbine is considered, the interface variable is a scalar ($w = \ddot{\theta}$). Let the residual at iteration step $l + 1$ be defined for w as:

$$rs^{l+1} = F(w^l) - w^l \quad (2.29)$$

Figure 2.2 shows the variation of the residual with the interface variable for the sub-iterations in the first time step of the partial tight coupling case without external loads. The variation of the residual as the iterations proceed is also shown. The initial guess value is provided by FAST enabled with two modes.

It is observed that more than 10000 iterations are needed to satisfy a tolerance of 1E-6 on the residual. It is also observed that the iterations show linear convergence. For fixed point iterations with linear convergence, Aitken's acceleration method is

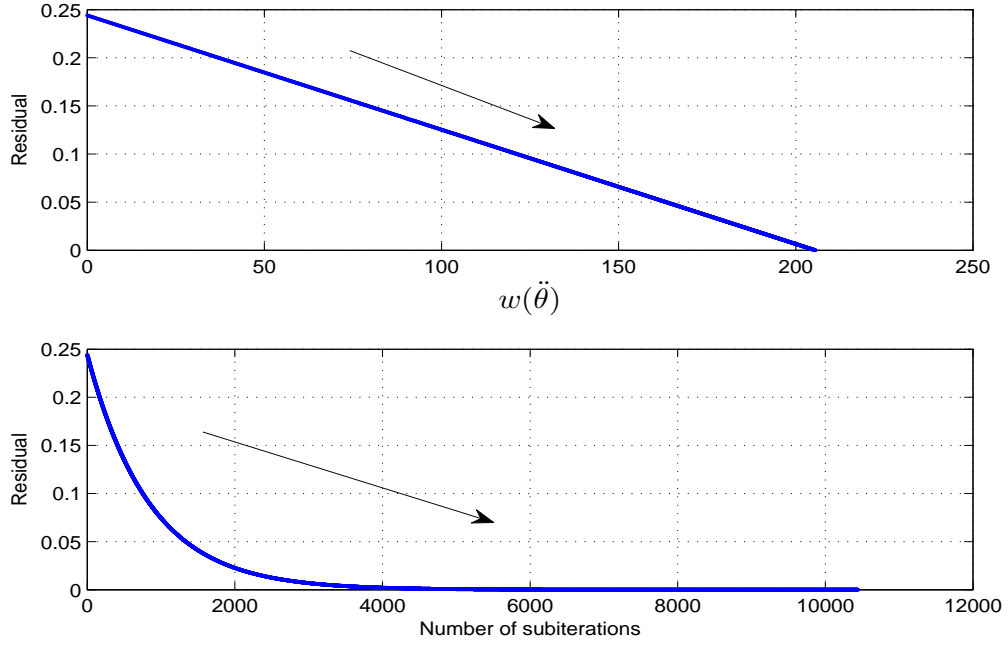


Fig. 2.2: Residual Behavior for FPI Without Acceleration

known to improve the performance with quadratic convergence rates.

Aitken's acceleration method similar to the method of secants can be derived in a straightforward manner as shown below:

If the residuals rs^l and rs^{l+1} from two previous iterations are available, then from Fig. 2.2, the x-intercept for a straight line through the points $(w^{l-1}, rs^l), (w^l, rs^{l+1})$ is given as:

$$w^{l+1} = w^l - rs^{l+1} \frac{(w^l - w^{l-1})}{rs^{l+1} - rs^l} \quad (2.30)$$

Here, w^{l+1} would represent the interface variable for which the corresponding residual is zero if the convergence is exactly linear away from the solution. The fixed point iteration with the relaxation factor α can also be equivalently be represented

by:

$$w^{l+1} = w^l + \alpha r s^{l+1} \quad (2.31)$$

Rewriting Eq.(2.32) in the form of Eq.(2.31), we have:

$$w^{l+1} = w^l + \alpha^{l+1} r s^{l+1} \quad (2.32)$$

where,

$$\alpha^{l+1} = \frac{w^l - w^{l-1}}{r s^{l+1} - r s^l} = -\alpha^l \frac{r s^l}{r s^{l+1} - r s^l} \quad (2.33)$$

Thus, it is seen that Aitken's method adaptively adjusts the relaxation factor as the iterations proceed. The above formulation is valid for the scalar interface variable w . A recursive formulation for a vector interface variable is given as:

$$\alpha^{l+1} = -\alpha^l \mathbf{w}^{l\top} \frac{\mathbf{r} \mathbf{s}^{l+1} - \mathbf{r} \mathbf{s}^l}{|\mathbf{r} \mathbf{s}^{l+1} - \mathbf{r} \mathbf{s}^l|^2} \quad (2.34)$$

To avoid the situation where a division by zero occurs in Eq.(2.33) if $r s^l = r s^{l-1}$ the iterations revert to the fixed relaxation case using the relaxation factor estimated at the previous iteration. To begin the iterations, a fixed relaxation factor is chosen for the first iteration. Fig. 2.3 demonstrates the superior performance of this method. For the same tolerance on the residual, just two iterations are sufficient to reach convergence. This is true, since as 2.2 shows, the the convergence is linear even away from the solution. A major benefit of this method is that not only is the scheme simple to implement, there is also no need to evaluate sensitivities or the Jacobian matrix.

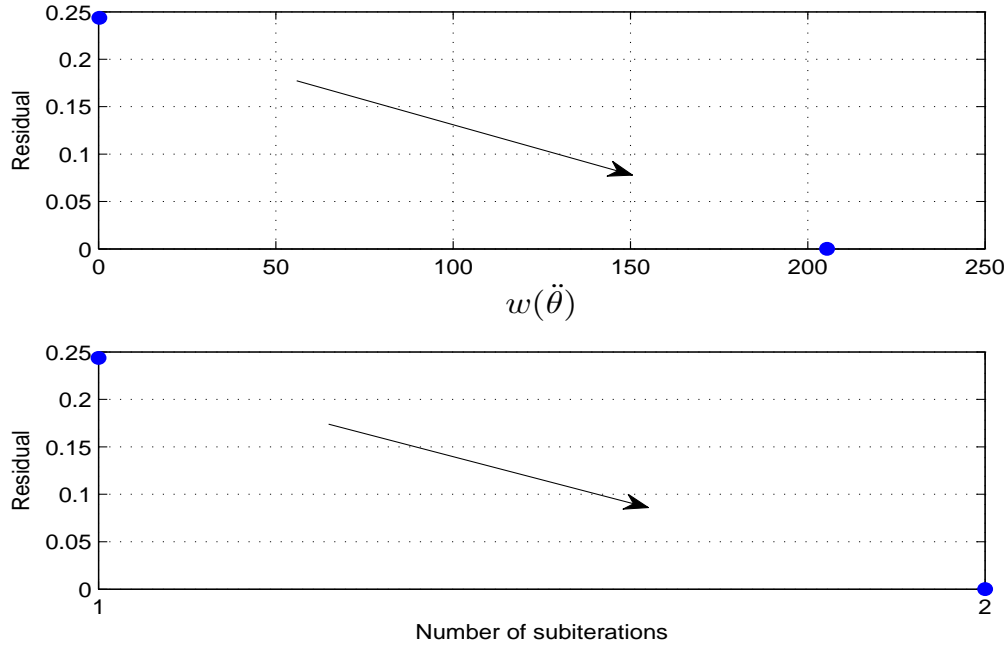


Fig. 2.3: Residual Behavior for FPI with Aitken's Acceleration

2.4 Software Implementation

In this section, the details regarding the programming interfaces between the FAST and FEM modules are provided.

FAST is written in FORTRAN. To interface with MATLAB and to connect with the FEM module, an S-Function that calls FAST's FORTRAN routines is used. The S-Function routines are available open-source from NREL. The S-Function provides the essential data links between FAST and the FEM module. The S-Function has been modified to accept from the FEM module nodal deformations, rotor inertia, and additional loads from the nodal acceleration or velocity terms such as $\mathbf{M}_{\theta_{\mathbf{f}_i}} \ddot{\mathbf{q}}_{\mathbf{f}_i}$. The S-Function communicates with FAST and returns with the rotor accelerations and aerodynamic loads. Since the blade deformations are handled by the FEM module,

blade flexibility is disabled in FAST.

A representative finite element code is written in MATLAB with the mass matrix, stiffness matrix, and shape function matrix definitions provided in Appendix A. Since the finite element matrices (mass, stiffness, and shape function) are independent of time, a pre-processor program is executed which load these matrices into the memory as global variables. The aerodynamic loads from the AeroDyn module connected to FAST are transferred to the FEM module using the S-Function. The uniformly distributed aerodynamic loads acting on the blade elements are converted to an equivalent load system acting on the finite element nodal points using the concept of work equivalence for arbitrary nodal displacements [28]. Note that for the present case, the equivalent loads are calculated only for the in-plane forces. Since planar rotation of the turbine is assumed, there are no aerodynamic moments (axis parallel to rotor axis) acting on the blade. The FEM module returns with the nodal accelerations of the blade.

Once the accelerations are obtained from the FAST module and the FEM module, they are integrated using available variable time-step ODE solvers in MATLAB. The explicit fourth order Runge-Kutta integrator - ode45 is used for the current application. It is pointed out that the explicit and implicit integration scheme in this context is defined with respect to variables native to a module. Implicit schemes are suggested if the explicit scheme appears stiff, i.e. the adaptive time-step chosen by the explicit ODE solver is very small due to the high frequency content in the system response.

Since a three-blade turbine is considered to validate the approach, three FEM modules are used to interconnect with FAST. For a given rotor rpm, angular acceleration and nodal aerodynamic loads are obtained from the FAST S-Function. The FEM module returns with nodal accelerations. The updated rotor inertia as well as

the generalized loads that are applied to the rigid rotor are computed. This data is fed back into the S-Function which eventually is input to FAST. Figure 2.4 shows the flow of data through the FEM-AeroDyn-FAST cycle. Other schemes can be envisioned, but since AeroDyn is called from within FAST, the scheme illustrated in Fig. 2.4 is seen to be appropriate for the current purpose.

Figure 2.4 illustrates the flow of data for the co-simulation of the FEM and FAST modules. For both the explicit loose coupling and the partial tight coupling approaches, initial guess values for the rotor angular acceleration, velocity, and aerodynamic loads are assumed. For the first iteration, the angular acceleration is provided from FAST with the modal methods enabled; the initial rotor angular velocity given to FAST is chosen as the guess value for the angular velocity; and the aerodynamic loads are assumed to be all zero. For subsequent iterations, information from the previous time step is used for the guess values at the current time step. The FEM modules return with the nodal displacements and accelerations. The updated rotor inertia is calculated from the nodal displacements, as also are the coupling loads arising out of the flexible-rigid body motion. Next, at step 2, the FAST state variables and the nodal displacements are fed to FAST using the S-Function. FAST first calls AeroDyn which returns with the aerodynamic forces considering the structural displacements and velocities. At step 3, the updated rotor inertia, flexible-rigid body motion coupling loads, and the aerodynamic loads are then applied to the FAST multibody system. FAST then returns with the rotor accelerations, returning the data flow back to step 1. It is clear that, starting with an initial guess value for the rotor acceleration $\ddot{\theta}$ at step 1, we arrive with an updated rotor acceleration after cycling through steps 1-2-3. Thus, we provide the rationale behind the fixed-point iterating scheme. As previously discussed, the iterating scheme based on $\ddot{\theta}$ is stable but very slow. Acceleration methods will need to be used.

The inertia computed from the FEM modules is provided to FAST even though FAST has the capability to compute the inertia of the deforming blades. This is because FAST computes the inertia using the lumped mass approach whereas the FEM modules compute the inertia using the consistent mass approach. For the fixed point iterating scheme to arrive at a solution, it is important that the value of inertia used in Eq.(2.8) and modeled by FAST is identical to the value used by the FEM modules. This fulfills the integrity of the floating frame of reference approach. It is determined from numerical experiments that an error in calculating the inertia term will result in unstable iterations. However, stability can be improved if a large number of elements are used since both the lumped mass and consistent mass approach converge to the same value. The FEM module also provides FAST with the nodal displacements and velocities. However, since FAST takes in inputs at the midpoints of the elements rather than the nodes at either end of a beam element, the displacements and velocities are evaluated at the midpoint using the shape function definitions given in Appendix A.

A similar procedure is followed for the implicit loose coupling scheme, except that in this case, with the nodal accelerations obtained at step 2, the FEM module equations are integrated in time. The coupling variables(nodal displacements,velocities and accelerations) at the new time step are then fed to AeroDyn and FAST. FAST is then integrated from its state at the previous time step. The coupling variables (rotor velocities and acclerations) at this new time step are fed back into the FEM modules. It is noted that stabilizing for only $\ddot{\theta}$ resulted in stable solutions even though many coupling variables are used.

A purpose of the framework just described is to demonstrate the validity of the approach described herein. The framework may be extended to a more advanced multiphysics framework facilitating connections between commercial multibody codes

such as ADAMS or DADS and FEM softwares such as ABAQUS or FEAP. Many of these codes are equipped with user-defined routines similar in functionality to the S-Functions which can be used for co-simulation.

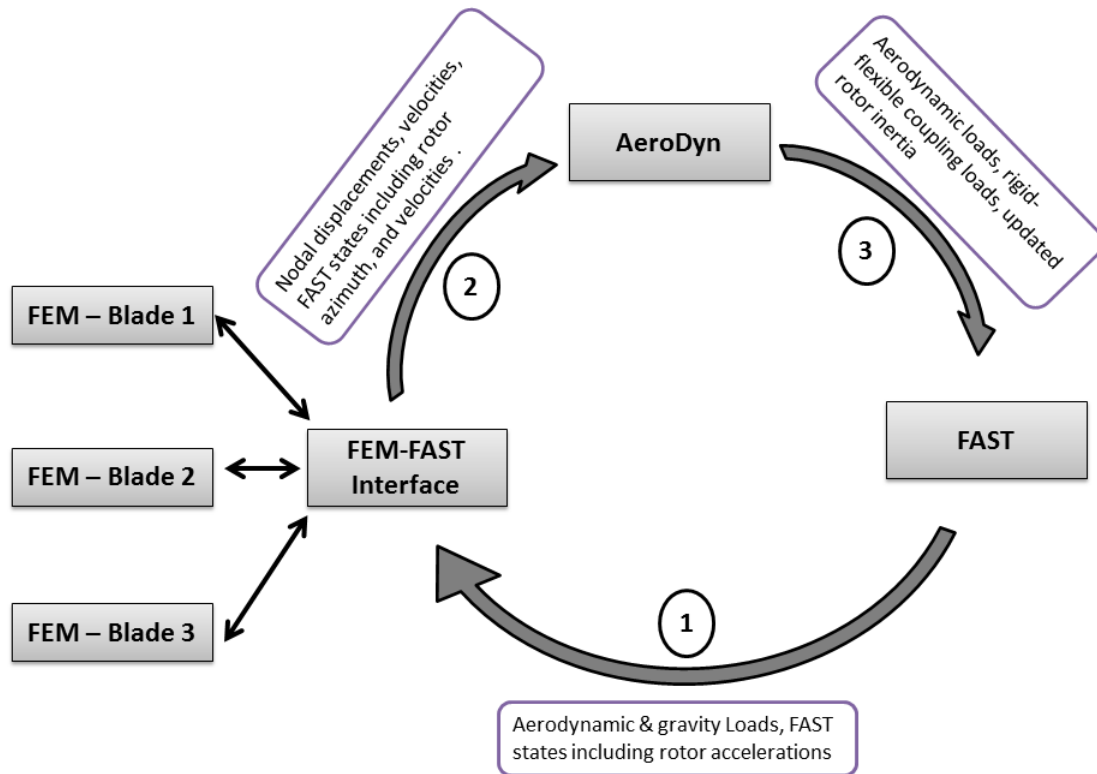


Fig. 2.4: FAST-FEM Co-simulation Data Flow

3. RESULTS AND DISCUSSION

In this section the results validating the approach detailed in this thesis are provided. A brief description of the turbine system used is also discussed.

3.1 Turbine Model

The planar case of a 3-bladed turbine is considered, i.e only the rotational degree of freedom of the drive train and the edge-wise deformation (rotor in-plane) of the blades are allowed. A modified version of one of the FAST certification test files (Test #08, AOC 15/50 FAST v.7.01.00) is used as the test case for validation. Figure 3.1 shows a photo of an AOC turbine installed at NREL's test facility. Simplified aerodynamics is considered i.e. blade element momentum theory without the effects of dynamic stall is used. Uniform structural and aerodynamic beam properties are used. The uniform properties are obtained by averaging span-wise varying properties of the original model. A summary of the turbine system properties is provided in Table 3.1. The beam is discretized using five finite elements. It is noted that the extension of the approach to a full three dimensional turbine system with other complexities is straightforward in principle, though not in terms of software implementation. The purpose here is to provide a proof of concept for the approach detailed in this thesis.

Table 3.1: Turbine System Properties Summary - Modified AOC 15/50

Parameter	Value
Turbine Make	Atlantic Orient Corporation
Rotation Axis	Horizontal
Orientation	Downwind
Rotor Radius(m)	7.490
Rotor Speed(RPM)	64.14
Number of Blades	3
Nacelle Yaw(deg)	-15
Hub Inertia(Kgm^2)	9
Beam Mass per unit length(Kg/m)	20.179
Beam Bending Stiffness - EI(Nm^2)	6.14E6
No. of FEM, AeroDyn beam elements	5
Airfoil Chord Length(m)	0.6019
Wind Velocity(m/s)	12
Drivetrain torsional stiffness(Nm/rad)	6.0E5
Drivetrain torsional damping($Nm/(rad/s)$)	1.0E3



Fig. 3.1: AOC 15/50 Wind Turbine [4]

The FAST modal pre-processor program, MODES is used to obtain the modal shapes for the chosen rotor rotational rate of 64.14 rpm. It is noted that FAST accepts only one mode for the edgewise motion and two modes for the flapwise motion. The blades are therefore given a structural twist of 90 deg to use two modes for the in-plane rotor motion. The modes obtained are used for the comparison for the co-simulation approach with the modal approach. It is noted that the formulation presented includes axial modes. FAST does not include axial modes but considers the axial deflection due to bending. Therefore, for both models to be as consistent,

the axial modes are suppressed in the finite element code by using a very large value for the axial stiffness. However, this tends to make the equations of motion stiff for time-integration.

For spatial integration needed for the modal method capability in FAST, numerical integration is performed across a number of stations along the blade. These stations which also serve as collocation points for the aerodynamic forces are defined in the AeroDyn input file. It is determined that a large number of integration stations is needed for the modal approach to match the results from the finite element method.

The following table shows the output from MODES in terms of the modal frequencies. The 6th order polynomial form of the mode shape as normalized with the tip displacement are shown:

Table 3.2: Modes Output

	Frequency(Hz)	x^2	x^3	x^4	x^5	x^6
Mode 1	5.7349	1.6006	-0.0307	-1.4453	1.2027	-0.3273
Mode 2	35.2671	-9.7180	9.9859	20.0685	-29.9356	10.5992
Mode 3	102.0608	35.9689	141.7970	173.7561	-67.5438	0.6158
Mode 4	210.2372	-83.6436	529.3278	-1161.0175	1064.1913	-347.8582
Mode 5	872.1095	29.9830	-231.3566	614.3611	-678.3636	266.3761

It is pointed out that only the first two modes in Table 3.2 are used in FAST. Figure 3.2 shows the convergence of natural frequencies as the number of finite elements increases. For five elements, the natural frequencies converge for the first two modes.

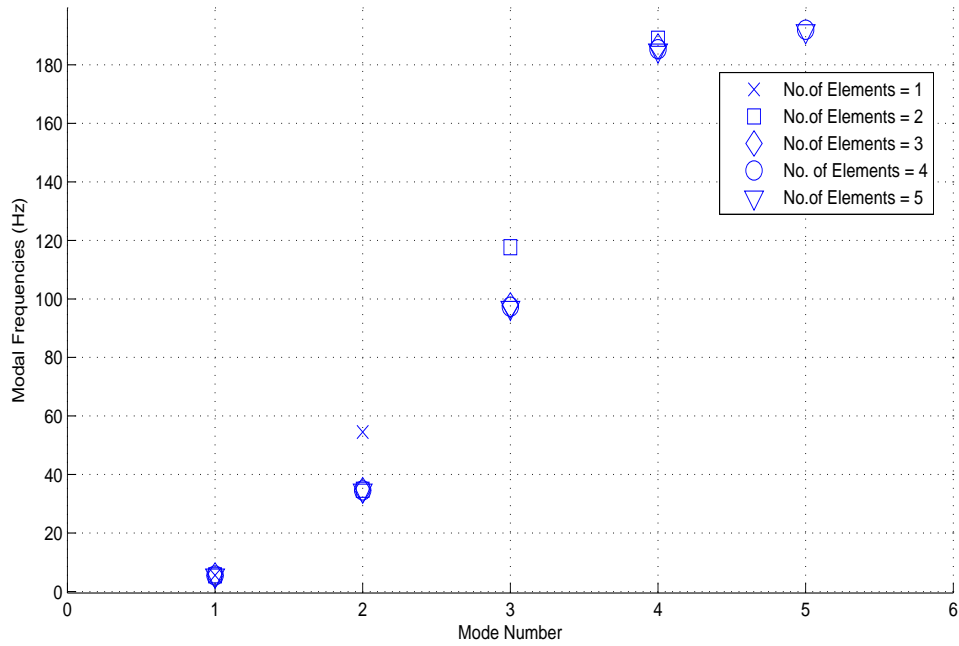


Fig. 3.2: Natural Frequency Convergence with Number of Finite Elements

Since direct load transfer is used from the aerodynamics module (AeroDyn) to the FEM module, both modules have the same number of elements. It is determined from the results shown in this section, that five elements are sufficient to model both the structural dynamics and aerodynamics for the specified parameters.

3.2 Results

Results are provided for the different schemes investigated detailed in the previous section. Since the full tight coupling case has the highest accuracy among the schemes considered, the accuracy of other coupling schemes will be compared to the data obtained for this case. For all cases considered, an initial tip displacement of 1cm is set for the three blades. The plots presented show the response for one full revolution of the turbine system. It is noted that the time step used in this section refers to the

time interval at which the coupling variables are exchanged. This is different from the variable time-step used by the ODE solver to integrate the equations. However, in the case of the partial tight coupling scheme, both definitions of the time-step are equivalent since the coupling variables are exchanged at the variable time-step chosen by the ODE solver.

3.2.1 Full Tight Coupling

To validate the full tight coupling case, data obtained from modal methods are used for comparison. For linear material behavior and small deformations, both the modal method and the finite element method should show a close match given the same initial conditions. Since FAST allows a maximum of two modes, the system parameters are chosen such that the resulting dynamics may be adequately represented by only a few modes.

FAST takes into account the axial deflection due to bending and the formulation presented herein does not take this effect into account. Therefore, the FAST source code was modified to disable this effect. Figures 3.3,3.4 and 3.5 show the tip displacements of the blade using the full tight coupling scheme. Displacements are compared with a version of FAST which does not include axial deflection due to bending and the original version which takes into account this effect. FAST is configured in this case to include both gravity and aerodynamic loads. As observed, the FAST-FEM co-simulation approach shows an excellent agreement with the modal method when the axial deflection is not taken into account. However, a time-varying phase-difference is seen when compared with the modal method with axial deflection taken into account. This may be due to a minor change in the modal frequency response when the axial deflection due to bending is neglected.

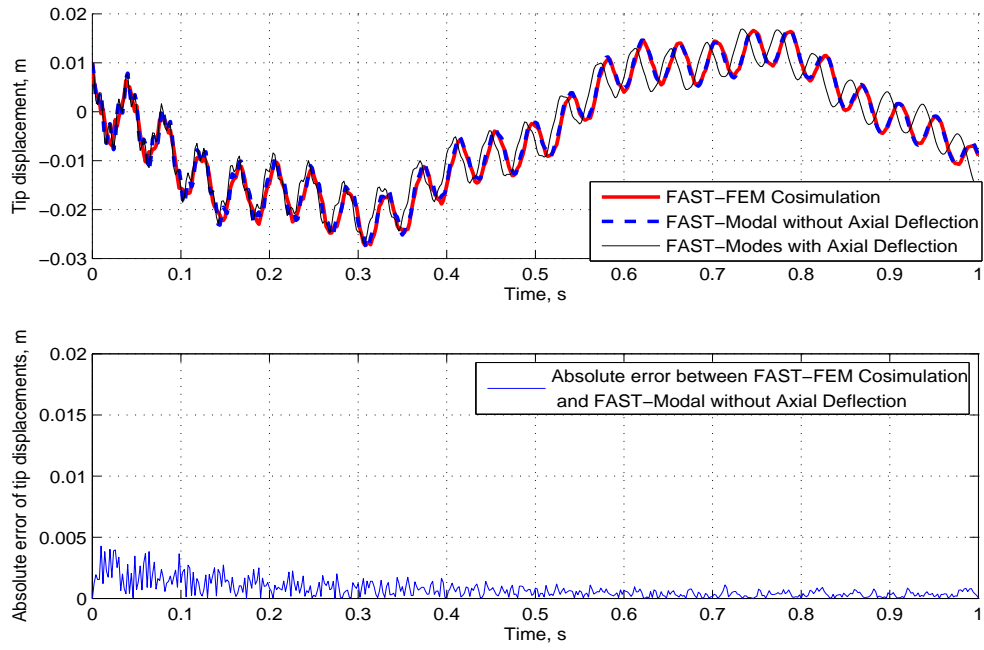


Fig. 3.3: Full Tight Coupling with External Loads - Blade 1 Response

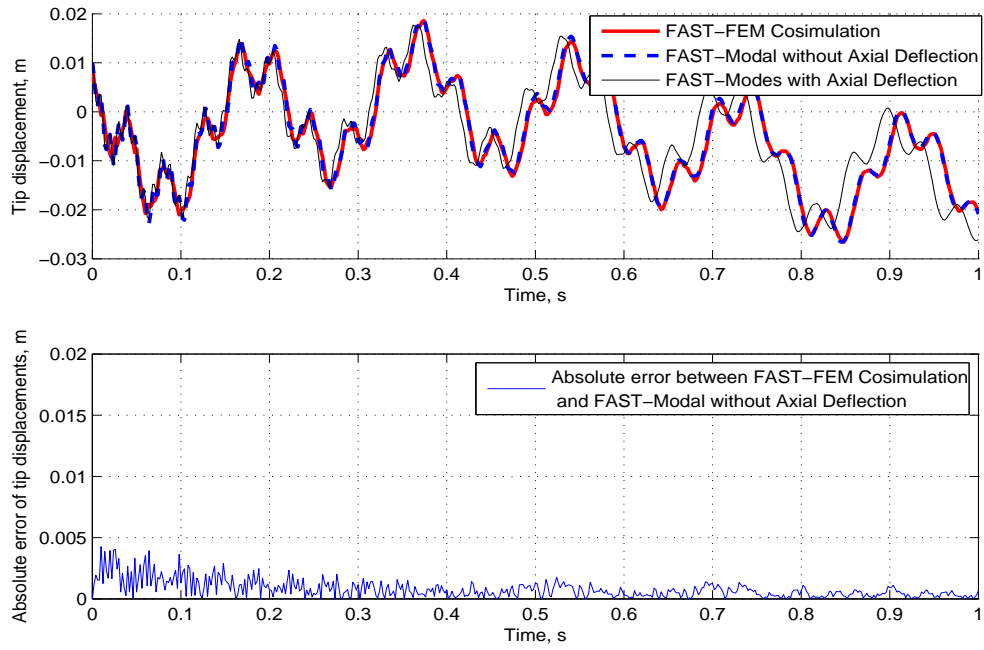


Fig. 3.4: Full Tight Coupling with External Loads - Blade 2 Response

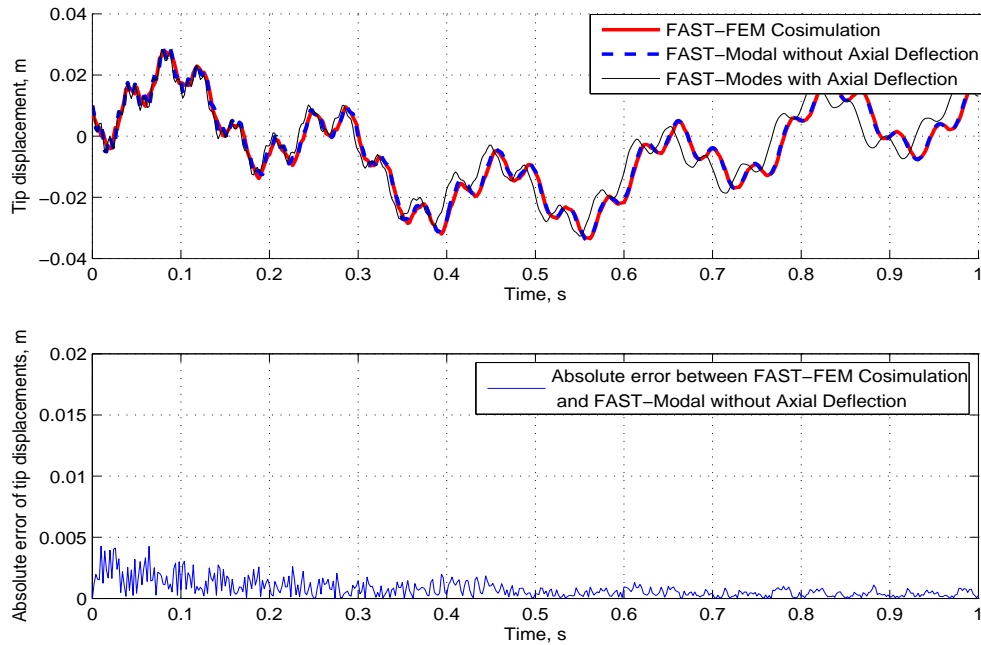


Fig. 3.5: Full Tight Coupling with External Loads - Blade 3 Response

It is observed that gravity and aerodynamic loads influence the behavior to a large extent. Therefore to analyze the influence of the rigid-body-flexible motion alone on the blade response, results are presented (Fig. 3.6) which show the blade response without external loading. The negligible errors shown in this case arise purely out of the inconsistencies between the modal method and finite element method in modeling the flexible-multibody motion. A major source of this error is due to FAST providing only two modes. The response of only one blade is shown since the the flexible dynamics of all three blades due to symmetry will be exactly the same.

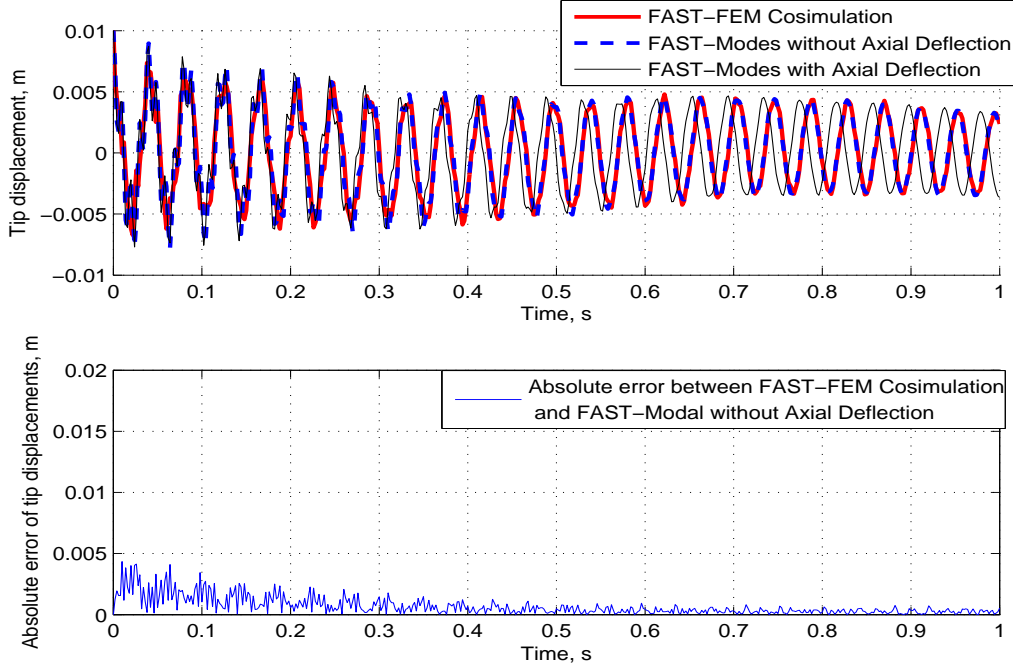


Fig. 3.6: Full Tight Coupling Without External Loads - Blade 1 Response

3.2.2 Partial Loose Coupling

Figures 3.7, 3.8 and 3.9 show the comparison of the partial loose coupling (Linear Theory of elastodynamics) with the full tight coupling case with external loading. As expected, the accuracy is low due to previously discussed one-way coupling. The marked discrepancy between both the modal and co-simulation approaches may be interpreted as a measure of the flexible-rigid body motion coupling that exists for the given system parameters. A time step of 0.0001 secs is used. An improvement in accuracy may be observed if the coupling between the rigid-body and flexible-body dynamics is weak.

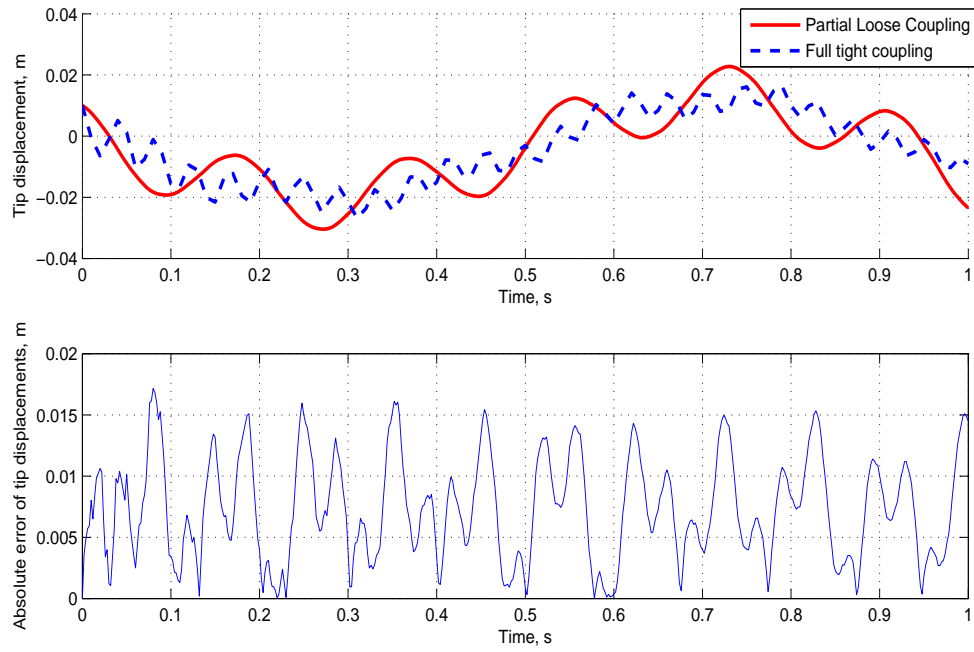


Fig. 3.7: Partial Loose Coupling - Blade 1 Response

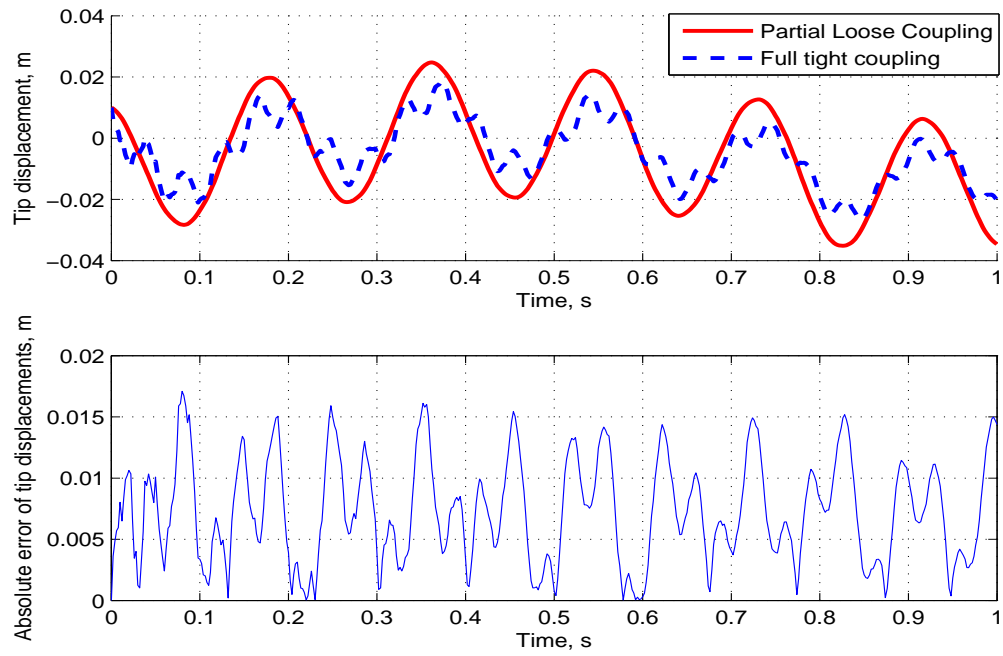


Fig. 3.8: Partial Loose Coupling - Blade 2 Response

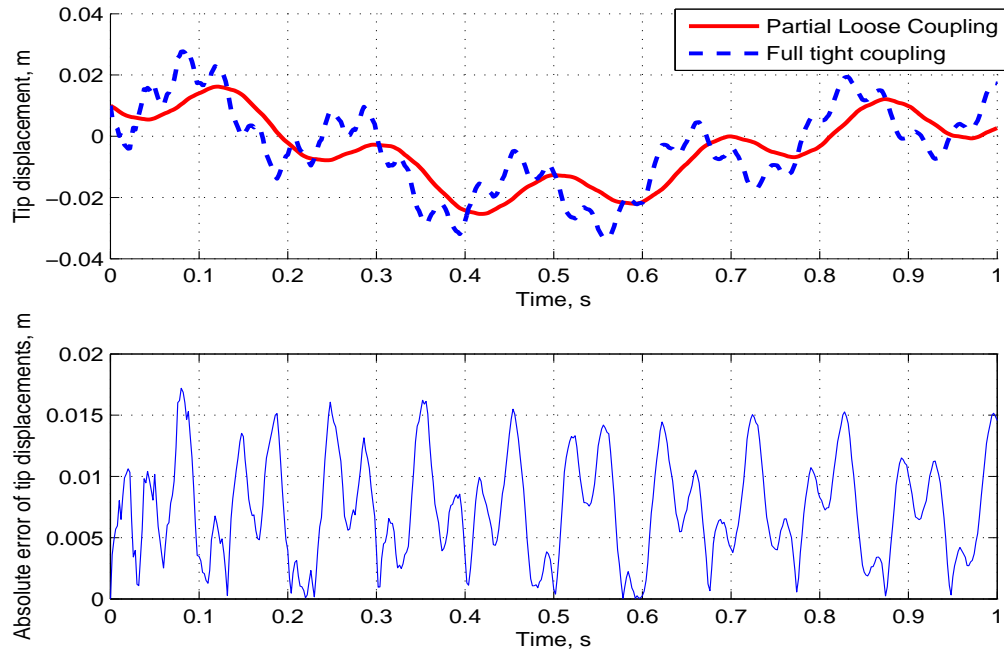


Fig. 3.9: Partial Loose Coupling - Blade 3 Response

3.2.3 Explicit Loose Coupling

Figures. 3.10, 3.11 and 3.12 show the comparison the explicit loose coupling scheme with the full tight coupling data. External loading is considered. As mentioned earlier, the explicit nature of the scheme results in numerical instability issues. A reduced time-step of 0.00001 secs is used to obtain stable results.

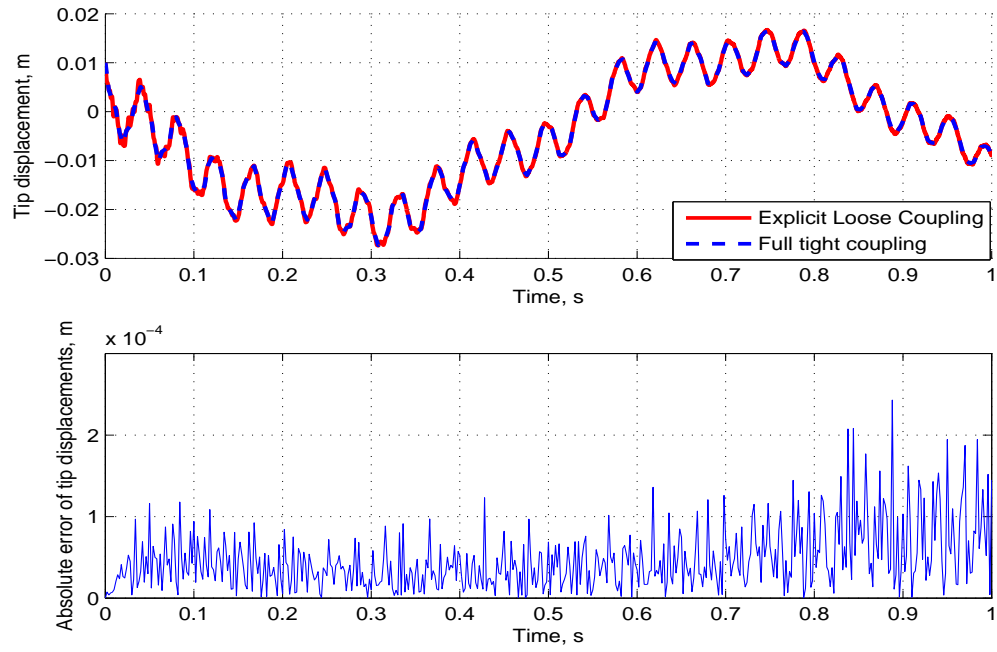


Fig. 3.10: Explicit Loose Coupling - Blade 1 Response - Hub Inertia = $9 \text{ Kg}m^2$

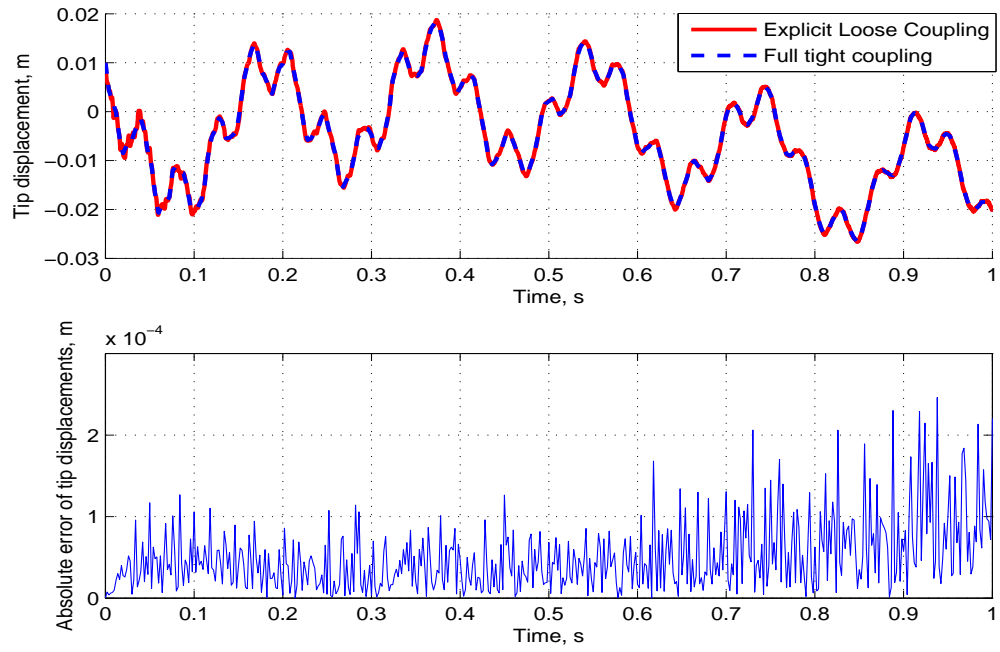


Fig. 3.11: Explicit Loose Coupling - Blade 2 Response - Hub Inertia = $9 \text{ Kg}m^2$

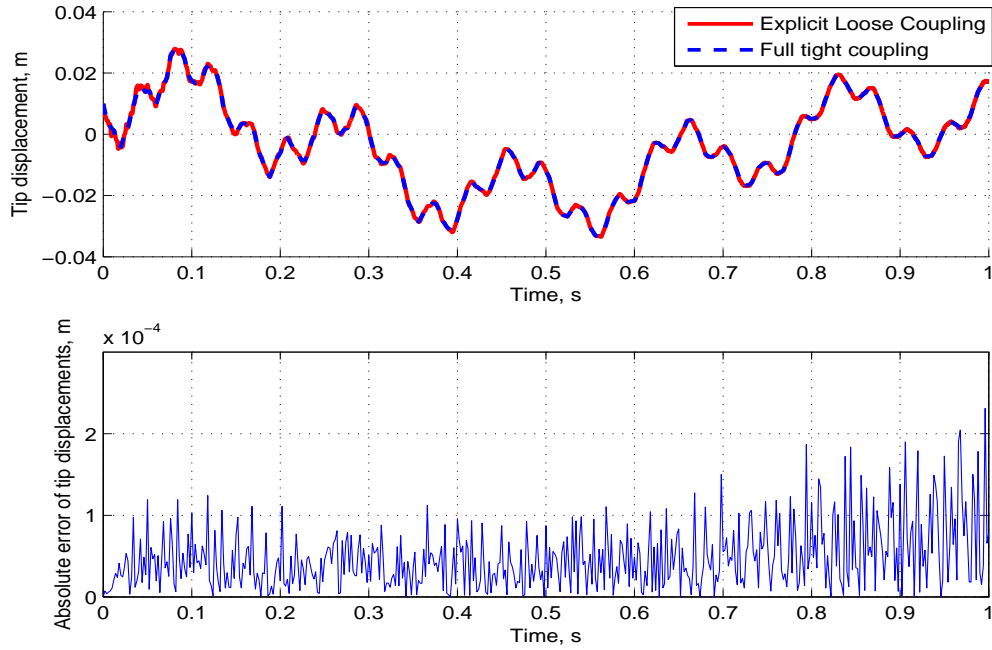


Fig. 3.12: Explicit Loose Coupling - Blade 3 Response - Hub Inertia = $9 \text{ Kg}m^2$

The use of a small time-step for a stable solution is to adequately capture the high frequency content seen in Figs. 3.10, 3.11 and 3.12. A larger time step may be used if the response is dominated by relatively low frequencies. If the Hub inertia is increased to $9000 \text{ Kg} - m^2$, a time-step of 0.0001 secs is found to provide stable results. Figures. 3.13, 3.14 and 3.15 show the blade response for this case. Figure 3.16 shows the number of FPI cycles needed to reach convergence. A small number of FSI cycles, an average of 3, is needed since the time-steps used are small.

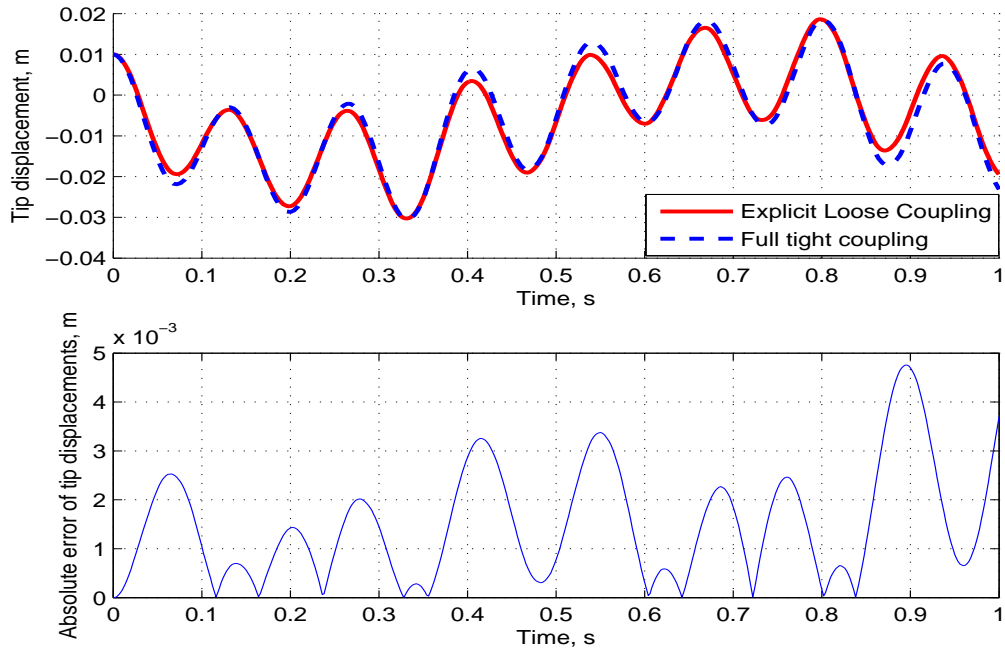


Fig. 3.13: Explicit Loose Coupling - Blade 1 Response - Hub Inertia = $9000 \text{ Kg}m^2$

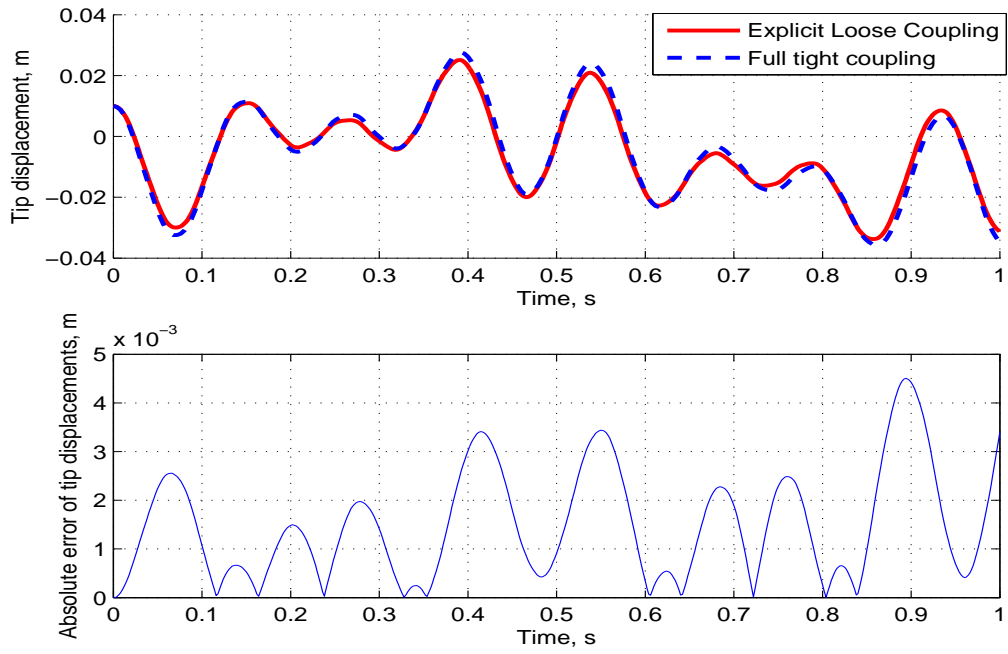


Fig. 3.14: Explicit Loose Coupling - Blade 2 Response - Hub Inertia = $9000 \text{ Kg}m^2$

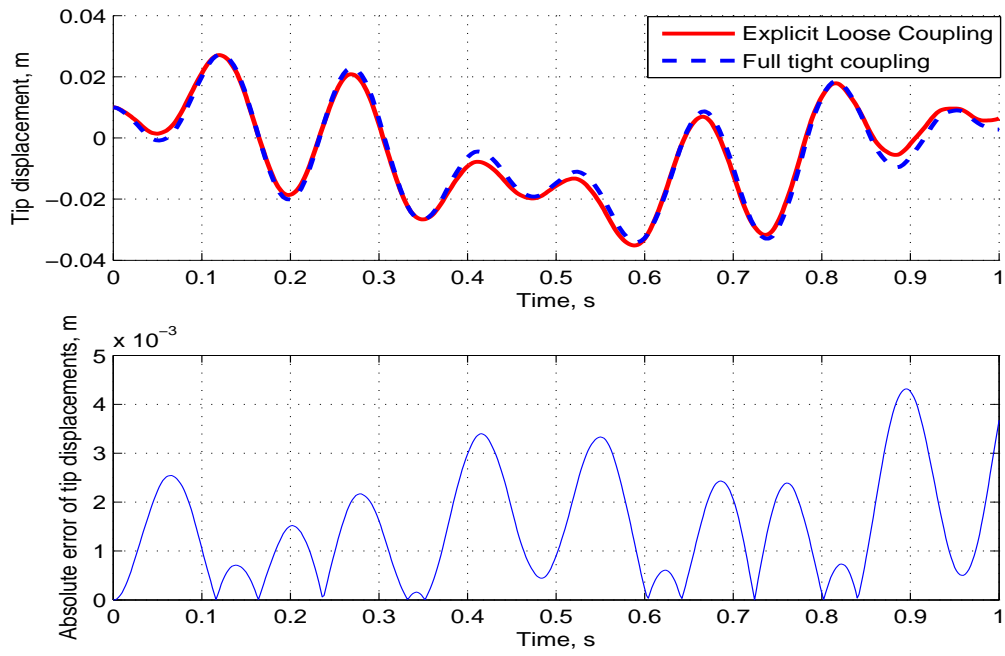


Fig. 3.15: Explicit Loose Coupling - Blade 3 Response - Hub Inertia = $9000 \text{ Kg}m^2$

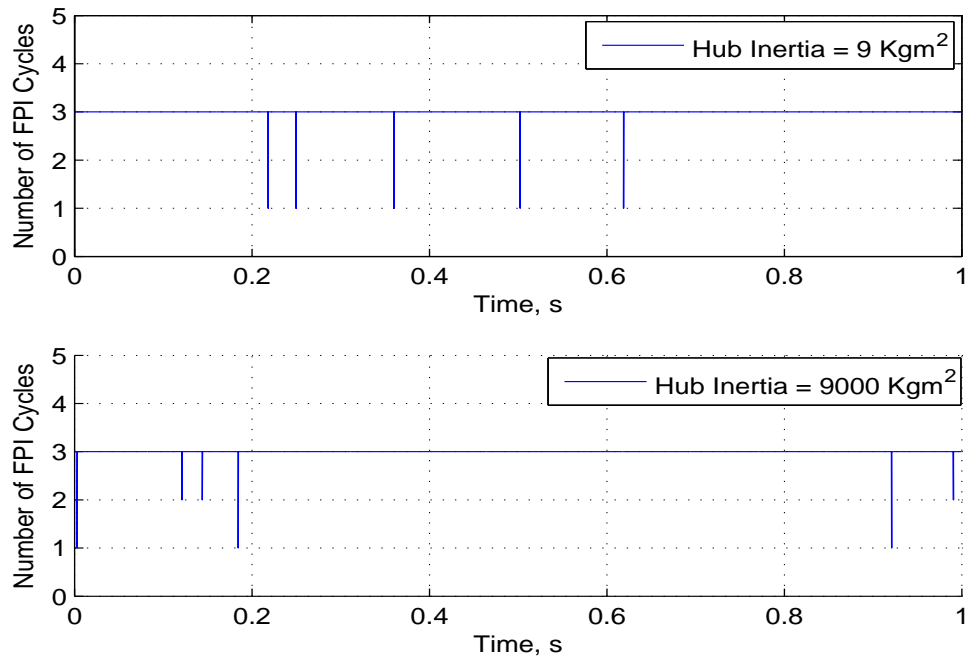


Fig. 3.16: Explicit Loose Coupling - Fixed Point Iteration Counts Comparison

3.2.4 Implicit Loose Coupling

A significant increase in numerical stability can be achieved using the implicit scheme. The results, Figs. 3.17, 3.18 and 3.19 show good agreement for this case. A time-step of 0.001 secs, two orders of magnitude greater than the loose coupling scheme is used. External loading is considered.

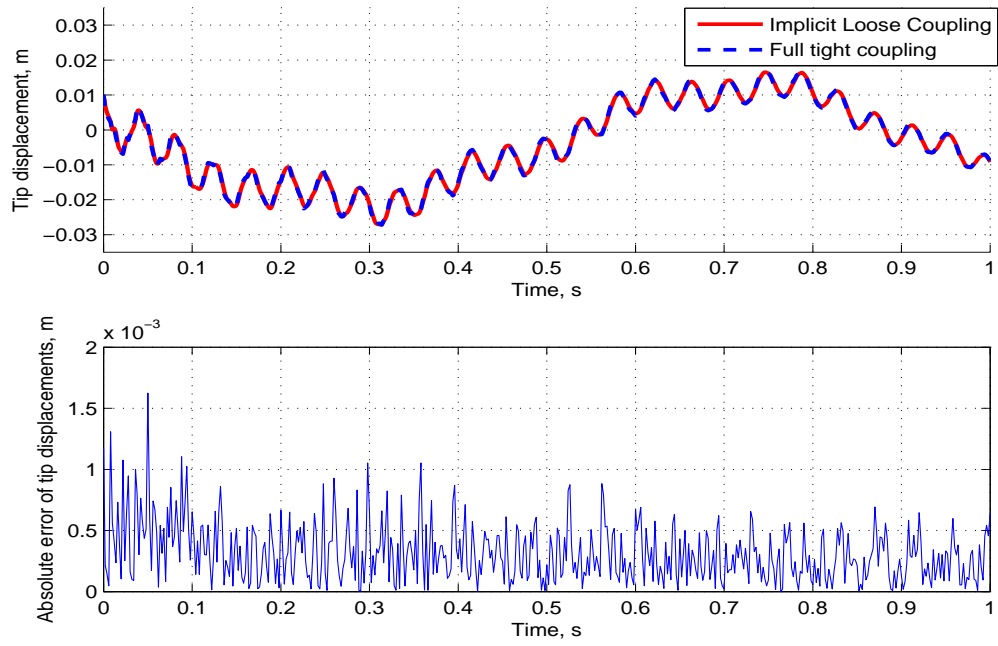


Fig. 3.17: Implicit Loose Coupling - Blade 1 Response - Time Step = 0.001 secs

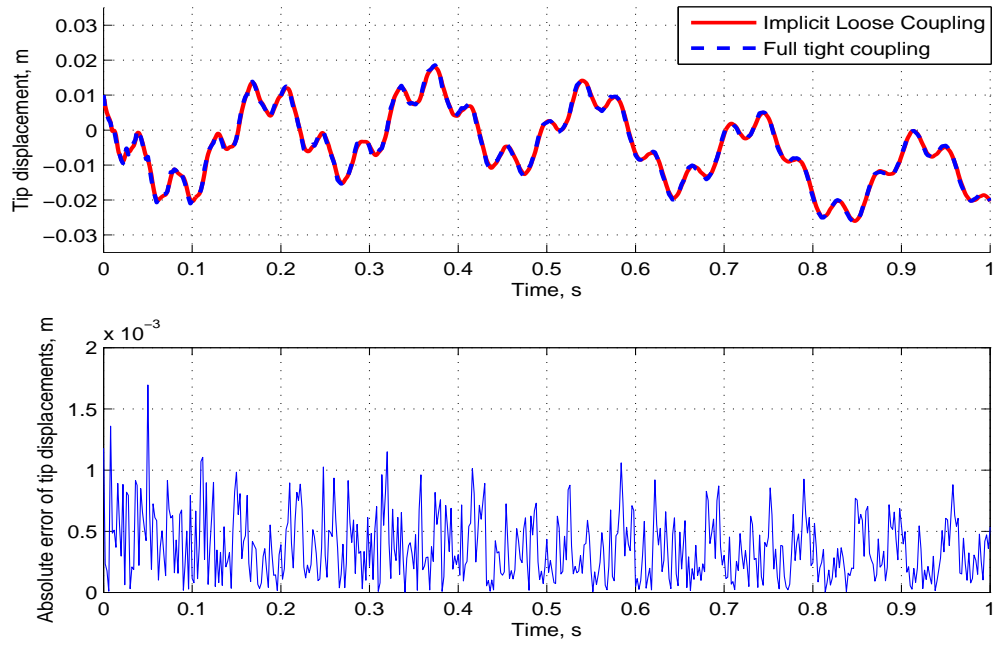


Fig. 3.18: Implicit Loose Coupling - Blade 2 Response - Time Step = 0.001 secs

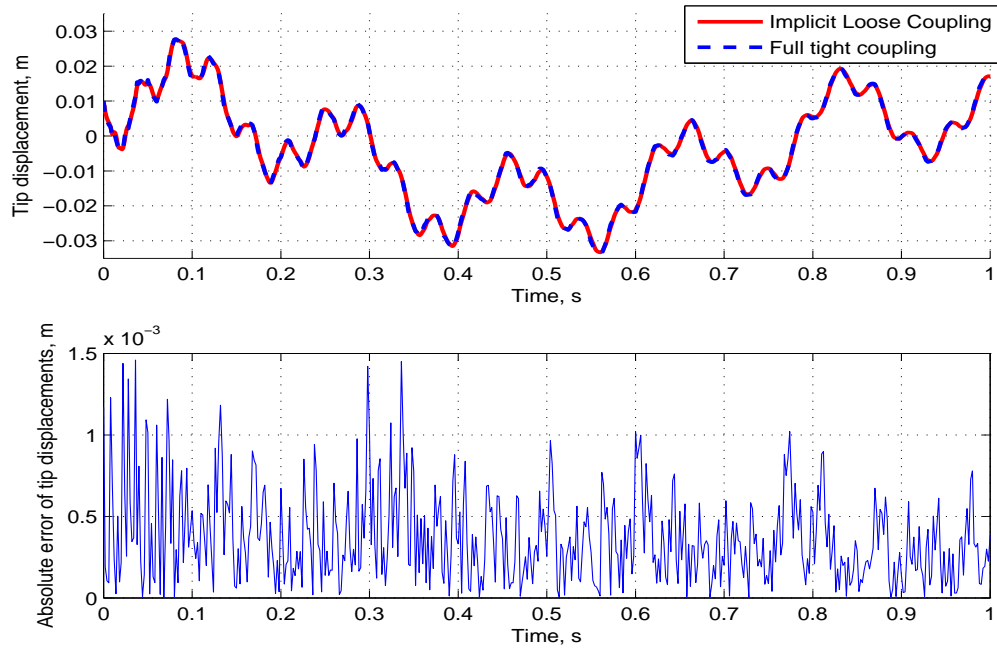


Fig. 3.19: Implicit Loose Coupling - Blade 3 Response - Time Step = 0.001 secs

Stable solutions are obtained even at 0.01 secs. However, a lower accuracy is observed. This is due to the larger coupling time-steps used. Figures 3.20, 3.21 and 3.22 show the blade response for this case compared with the full tight coupling data with external loading.

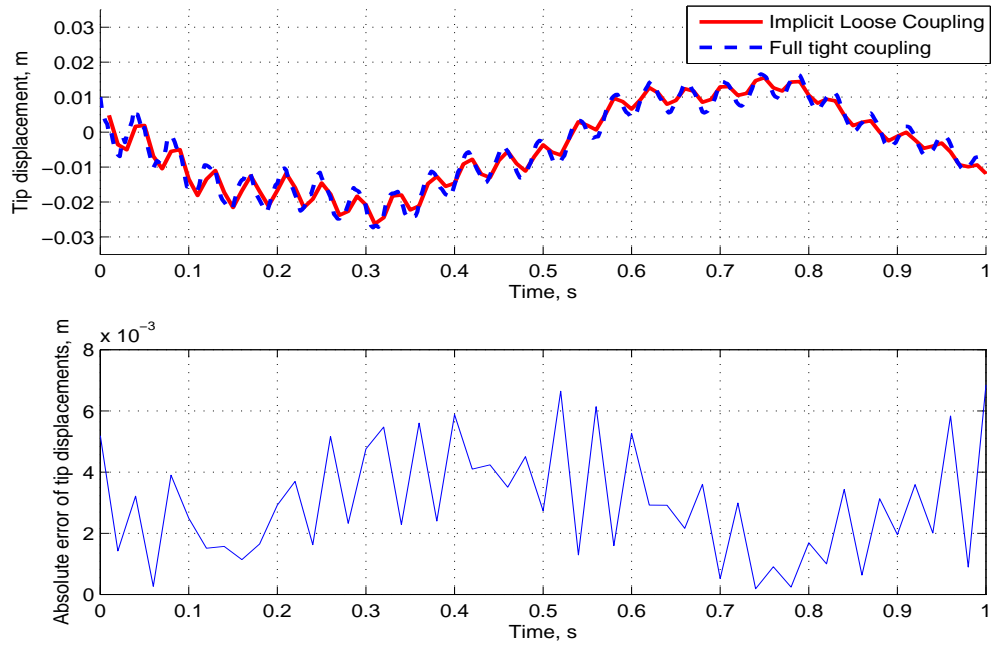


Fig. 3.20: Implicit Loose Coupling - Blade 1 Response - Time Step = 0.01 secs

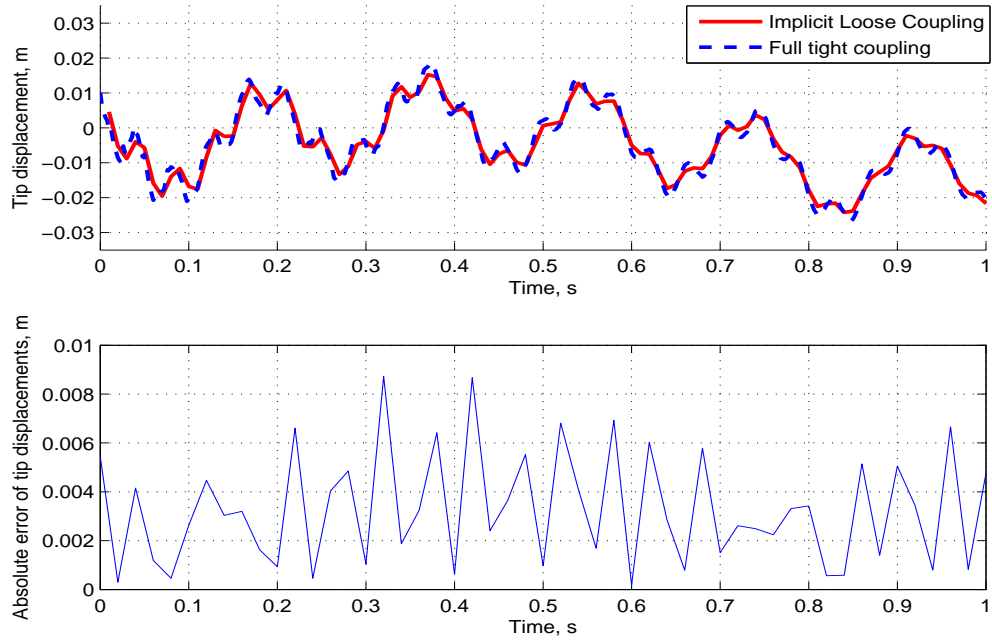


Fig. 3.21: Implicit Loose Coupling - Blade 2 Response - Time Step = 0.01 secs

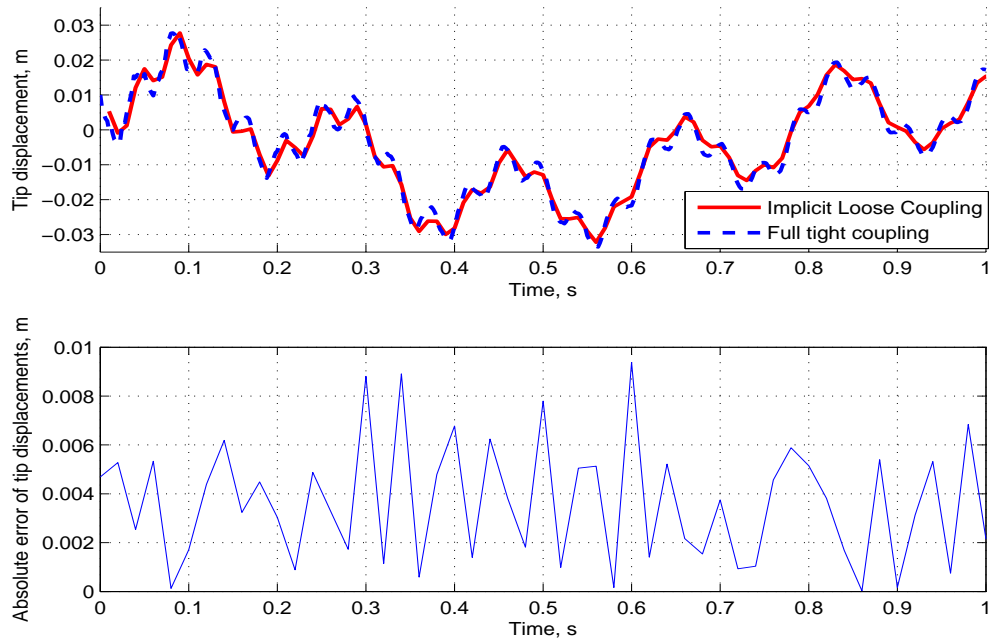


Fig. 3.22: Implicit Loose Coupling - Blade 3 Response - Time Step = 0.01 secs

Figure 3.23 shows the comparison for the number of FPI cycles needed for convergence for both time-steps considered. As expected, the implicit scheme at a time-step of 0.01 secs requires more iterations than the case for which a time step of 0.001 secs is used. This is true since at larger time steps, the initial guess value to start the iteration is less accurate. At a time-step of 0.01 secs, an average of 8.92 iterations are needed. At a time-step of 0.001 secs, an average of 7.39 iterations are needed.

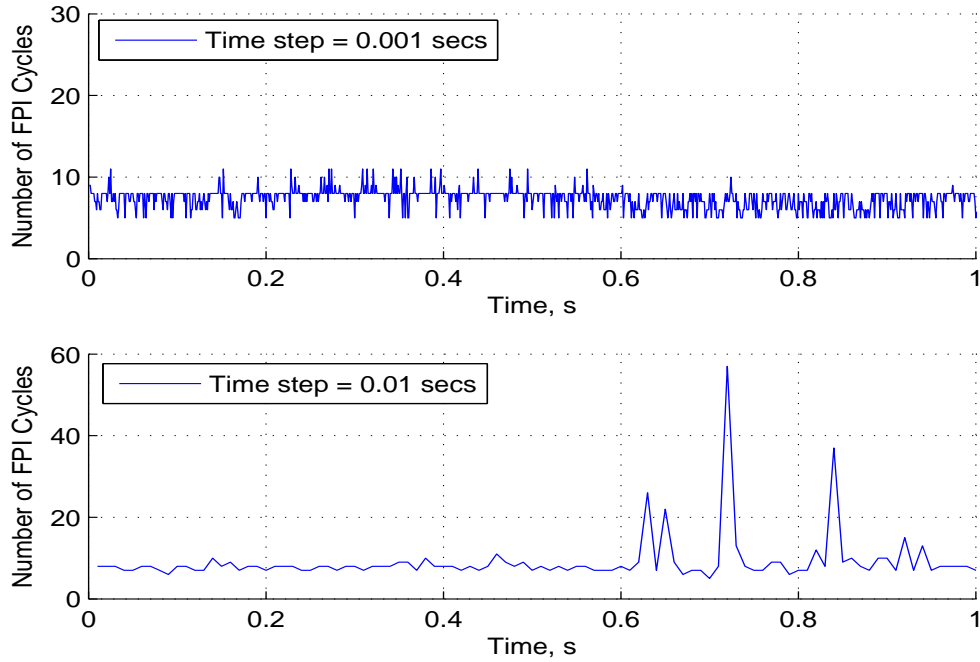


Fig. 3.23: Implicit Loose Coupling - Fixed Point Iteration Counts Comparison

3.2.5 Partial Tight Coupling

Figures 3.24, 3.25 and 3.26 show the blade response for the partial tight coupling compared with the full tight coupling case. External loading is considered. Figure 3.27 shows the number of FPI iterations needed to reach convergence. As stated earlier, since the variable time-step chosen by the ODE solver tends to be small, the initial guess value for the FPI iterations which is estimated from the previous time-step is close to the actual solution. It is observed that this scheme, though in principle equivalent to the explicit loose coupling scheme (with the exception that a common solver is used) is significantly faster.

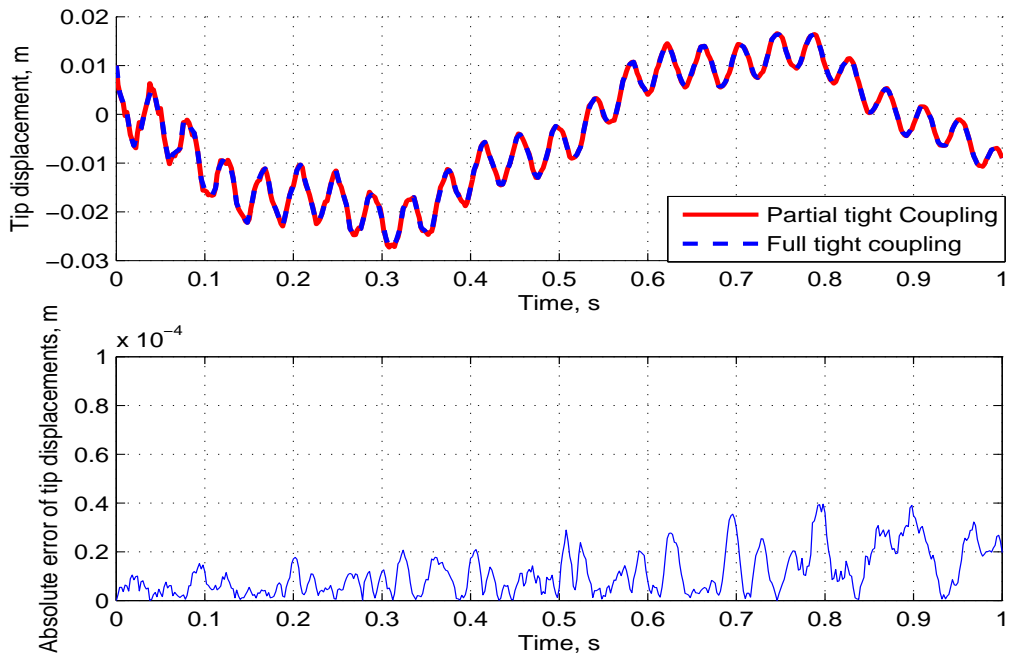


Fig. 3.24: Partial Tight Coupling - Blade 1 Response

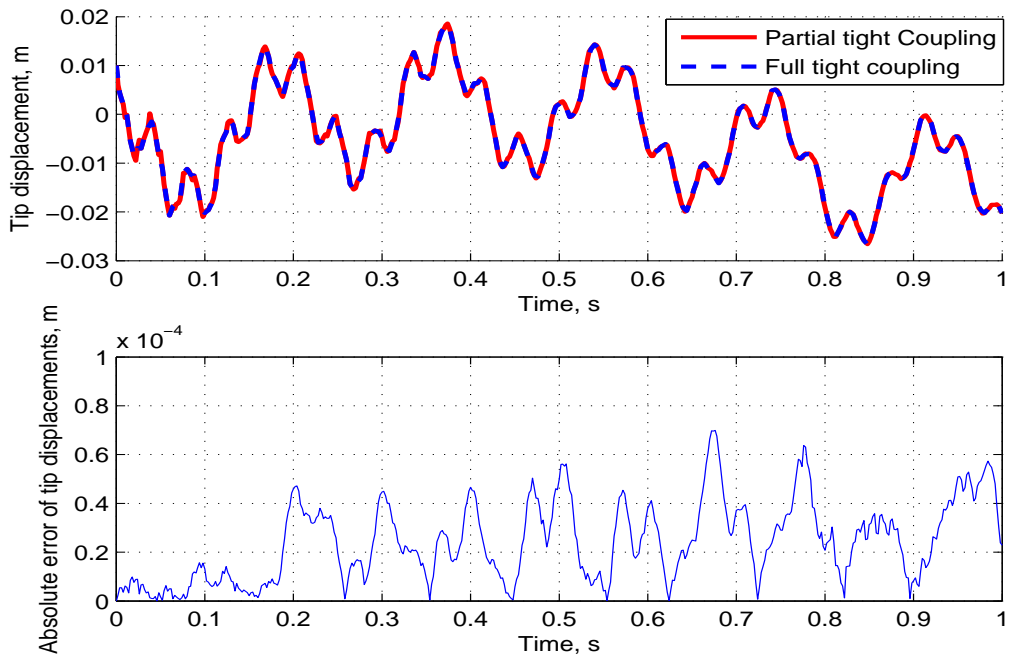


Fig. 3.25: Partial Tight Coupling - Blade 2 Response

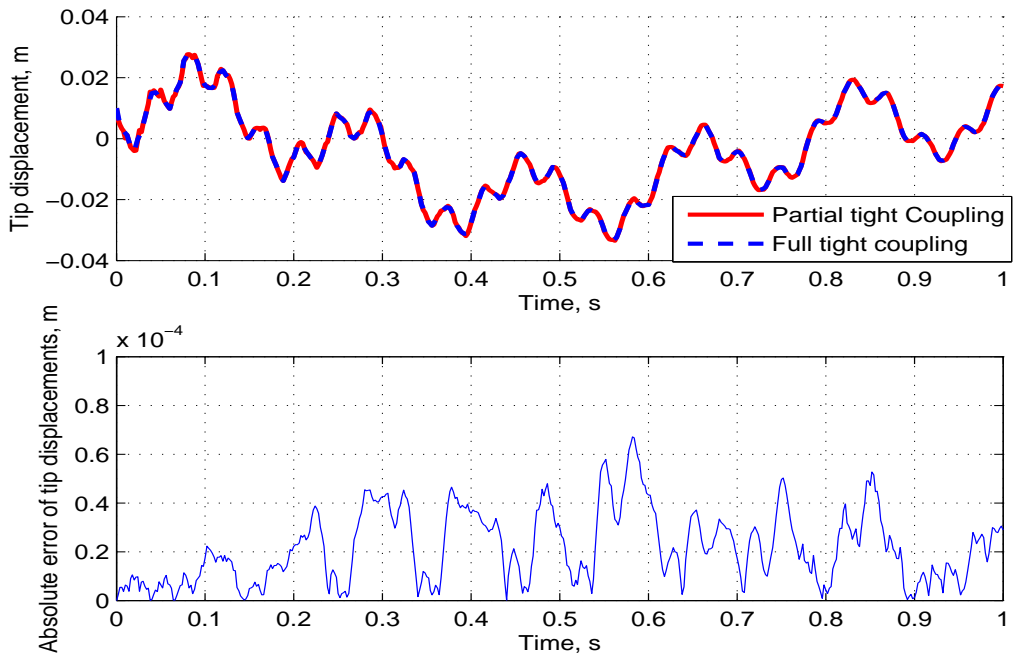


Fig. 3.26: Partial Tight Coupling - Blade 3 Response

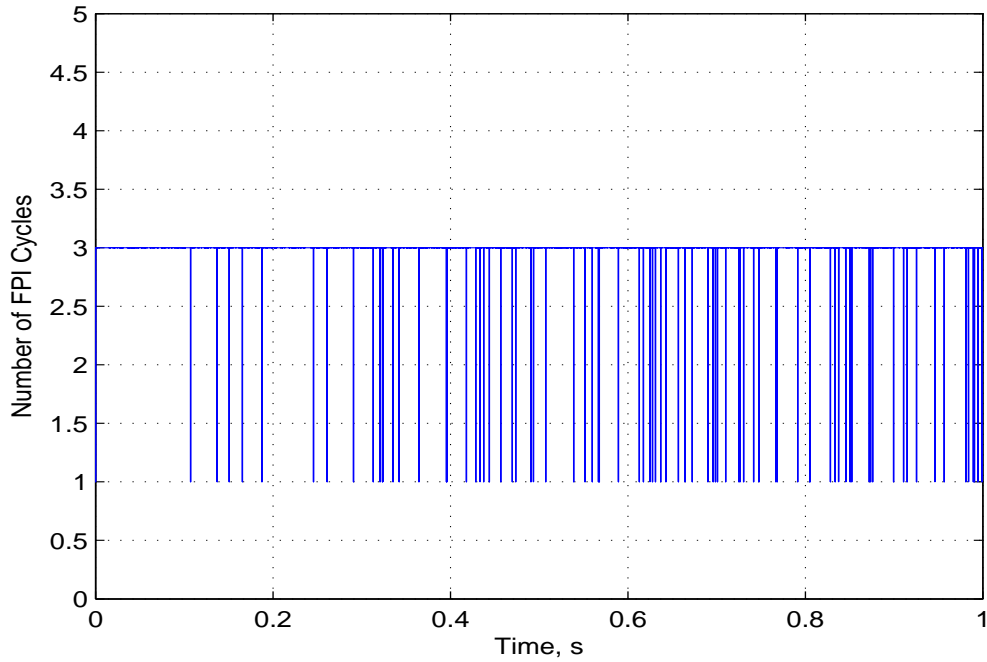


Fig. 3.27: Partial Tight Coupling - Fixed Point Iteration Counts

3.3 Summary

The results for the coupling schemes investigated in this thesis, show good agreement with the validation data with the exception of the partial loose coupling case. It is determined that the accuracy and stability of the loose coupling schemes depends on the chosen system parameters. The applicability of a particular scheme depends on the system parameters, the level of modularity and accuracy desired. Table 3.3 summarizes the coupling schemes explored in terms of Software intrusion, computational expense, stability and accuracy.

Table 3.3: Comparison of Coupling Schemes

	Software Intrusion	Expense	Stability	Accuracy
Partial Loose Coupling	Low	Medium	High	Low
Explicit Loose Coupling	Medium	High	Low	High
Implicit Loose Coupling	Medium	Medium	High	High
Partial Tight Coupling	High	Medium	High	High
Full Tight Coupling	High	Low	High	High

The partial loose coupling case is the least intrusive in terms of software integration, though the least accurate among the schemes considered. If the coupling between the rigid-body dynamics and flexible-body dynamics is weak, this scheme may be considered on a preliminary basis.

The explicit loose coupling scheme, for the chosen system parameters of the AOC turbine needed an extremely small time-step to maintain numerical stability. This resulted in long run times. However, this may not be a disadvantage if a CFD solver is coupled to FAST. In such a case, the CFD solver may determine the time-step. The time-steps used by CFD solvers are typically very small. However, the computational expense of performing a few sub-iterations at each time-step may be expensive. This scheme may also find applicability if the blade response is dominated by a few low-frequency modes, so that larger time-steps may be used. The implicit loose coupling scheme is advantageous if relatively large time-steps can be used. Such is the case, with low-order aerodynamic models such as AeroDyn.

The partial tight coupling case is suitable as a common solver can be used to integrate the states. To link general purpose multibody codes such as ADAMS with

FEM codes such as Abaqus or FEAP, developing a common solver if possible, may require substantial programming effort. However, this scheme is much faster than the equivalent explicit loose coupling scheme. The full tight coupling scheme requires the greatest software intrusion in that the system mass matrix needs to be known. This scheme is in principle equivalent to a monolithic scheme, with the exception that state time-derivatives are computed by separate modules. Therefore, this scheme is considered the most accurate among the schemes considered. As with the partial tight coupling case, using this scheme may require significant programming effort and commonly available codes may not provide such a level of access.

4. CONCLUSIONS

4.1 Concluding Remarks

A method is developed to co-simulate the wind turbine aeroelasticity simulation tool FAST with a FEM module. This is shown to be accomplished using a floating frame of reference formulation which allows a clear separation of rigid-body motion and small deformation motion of the turbine blades. Developing the equations for multibody dynamics usually results in a system of differential algebraic equations since the constraints have to be accounted for in the equations. However as developed in this study, fixing the reference frame at the blade root results in the constraint forces not appearing in the equations of motion. Consequently, the approach results in a system of ordinary differential equations which are integrated in time using conventional numerical techniques.

Various coupling strategies in accordance with the new modularization philosophy developed by NREL [2, 20] are explored ranging from partial loose coupling to full tight coupling schemes. For the explicit loose coupling, implicit loose coupling and partial tight coupling cases, the unknown accelerations at each time step are obtained using an iterative process. To accelerate these iterations, Aitken's method is used. To the author's best knowledge, no previous studies exploring the application of Aitken's adaptive relaxation technique for the FEM-MBD co-simulation problem exist. Aitken's method is used more commonly in applications involving fluid-structure interactions. The results for the full tight coupling case are first compared with the modal approach. The good agreement validates the floating frame of reference approach for the finite element formulation of the flexible blades. Subsequently, results for the other coupling schemes are compared with the full tight

coupling case. Their individual merits and applicability are pointed out.

It is noted that the methodology developed in this thesis may be generally applied to the co-simulation of any finite element method code with a multibody dynamics code. The method is valid only if the flexible beams are cantilevered such as the case for rotorcraft and wind turbines. If additional constraints are added to the beams, such as in a simply supported configuration, gluing algorithms developed in Refs. [14] and [8] may be considered for this purpose.

4.2 Future Work

Several areas for further improvement are identified.

Firstly, the floating frame of reference formulation is valid only for the small deformation motion of the blade. This is a consequence of using infinitesimal nodal rotational displacements, essentially linearizing the kinematics. For large deformations observed in large flexible turbines, the errors induced on account of this linear analysis may be significant. Several solutions are proposed in the literature to account for the large deformation while still exploiting the advantages of using the floating frame of reference formulation. One common solution is to divide the beam into several elements and define a floating frame for each of these elements. The elements are joined using algebraic constraints. [11]. A method proposed by Das [29] uses a co-rotational frame for each finite element including using a floating frame for the overall flexible body in an updated Lagrangian description of motion, i.e. the current configuration is used as the reference. Das develops the formulation for the multibody analysis for plates using the finite element method. Rotorcraft blades and other aerospace structures are constructed using plate and shell elements for which the finite element method becomes an appropriate choice. Another method proposed by Tsai et al [30] uses a combined co-rotational floating frame of reference formulation

but in a total Lagrangian description of motion, i.e the initial configuration is used as the reference. Both of the aforementioned methods, though allowing a variety of nonlinearities to be included in the formulation, are valid for small strains. However, the combined co-rotational floating frame approach may not allow the possibility of using general-purpose FEM codes.

Secondly, the methods developed in this thesis are for the planar rotation of the turbine system. This significantly simplifies the analysis for the purposes of validation and comparison. The development of a full three dimensional formulation is relatively straightforward. However, care must be exercised in the choice of the orientation parameters to avoid encountering singularities. Many similar works in the literature use Euler parameters to avoid singularities.

Thirdly, the inclusion of the bending-axial strain coupling is necessary to avoid the instabilities observed at critical speeds. The floating frame of reference allows for the inclusion of such nonlinear stress-strain relationships in the form of the geometric stiffness matrix. [31], [32], [33].

Fourthly, concerning the coupling schemes and programming interfaces, further research is required to optimize the framework. In this work, rather simplified tools already available in MATLAB are used for time-integration. Specialized integration schemes have been identified in the literature for the optimum performance of the floating frame of reference approach. Reference [33] discusses schemes which take into account the effect of the high frequency axial modes.

Finally, as is true with many multiphysics simulations, the simulations demonstrated herein are computationally much more expensive than the existing low order model which is used by FAST. Also, if a high-fidelity CFD code is also co-simulated with FAST as is appropriate for the FEM module, the involved computational expense may preclude any iterative design process or control system development and

analysis. Therefore, a common approach for such problems is to generate a reduced order model of the system. For the general multiphysics environment envisioned for FAST with interaction of different physics modules such as aerodynamics, hydrodynamics and structural dynamics with varying degrees of fidelity, techniques applied to general complex systems such as detailed in Ref. [34] may be explored for this purpose.

REFERENCES

- [1] Wikipedia, “Moore’s Law,” Available: http://en.wikipedia.org/wiki/Moore's_law. Accessed on March 2013.
- [2] Jonkman, J. M., “The New Modularization Framework for the FAST Wind Turbine CAE Tool,” *51st AIAA Aerospace Sciences Meeting*, Grapevine, Texas, 2013.
- [3] Brüls, O., Cardona, A., and Géradin, M., “Modelling, Simulation and Control of Flexible Multibody Systems,” *Simulation Techniques for Applied Dynamics*, edited by M. Arnold and W. Schiehlen, Vol. 507 of *CISM International Centre for Mechanical Sciences*, Springer Vienna, Vienna, Austria, 2009, pp. 21 – 74.
- [4] Jacobson, R., Meadors, M., Jacobson, E., and Link, H., “Power Performance Test Report AOC 15/50 Wind Turbine, Test B Revision 3,” Tech. rep., National Wind Technology Center, National Renewable Energy Laboratory, Golden, Colorado, 2003.
- [5] Gupta, K. K. and Meek, J. L., *Finite Element Multidisciplinary Analysis*, AIAA, Reston, Virginia, 2003, pp. 1.
- [6] Felippa, C. A., Park, K., and Farhat, C., “Partitioned Analysis of Coupled Mechanical Systems,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, No. 24-25, March 2001, pp. 3247–3270.
- [7] Bauchau, O., Bottasso, C., and Nikishkov, Y., “Modeling Rotorcraft Dynamics with Finite Element Multibody Procedures,” *Mathematical and Computer Modelling*, Vol. 33, No. 10-11, May 2001, pp. 1113–1137.

- [8] Ryu, G., Ma, Z.-D., and Hulbert, G. M., “Integration of Finite Element and Multibody Dynamics Models Using Gluing Algorithm,” *Volume 4: 7th International Conference on Multibody Systems, Nonlinear Dynamics, and Control, Parts A, B and C*, Vol. 2009, ASME, 2009, pp. 103–113.
- [9] Brock, D. C. and Moore, G. E., *Understanding Moore’s Law: Four Decades of Innovation*, Chemical Heritage Foundation, Philadelphia, Pennsylvania, 2006, Chap. 5.
- [10] Keyes, D., McInnes, C. W., Gropp, D., Myra, E., and Pernice, M., “Multiphysics Simulations: Challenges and Opportunities,” Tech. rep., Argonne National Laboratory, Park City, Utah, 2011.
- [11] Shabana, A. A., “Flexible Multibody Dynamics: Review of Past and Recent Developments,” *Multibody System Dynamics*, Vol. 1, No. 2, 1997, pp. 189–222.
- [12] Pan, W. and Edward J. Haug, “Nonlinear Structural Multibody System Simulation Using Structural and Rigid Body Dynamic Analysis Software,” *Computational Aspects of Nonlinear Structural Systems With Large Rigid Body Motion*, IOS Press, Amsterdam, Netherlands, 2001, p. 275.
- [13] Shabana, A. A., Bauchau, O. A., and Hulbert, G. M., “Integration of Large Deformation Finite Element and Multibody System Algorithms,” *Journal of Computational and Nonlinear Dynamics*, Vol. 2, No. 4, 2007, pp. 351.
- [14] Wang, J., Ma, Z.-D., and Hulbert, G. M., “A Distributed Mechanical System Simulation Platform Based on a Gluing Algorithm,” *Journal of Computing and Information Science in Engineering*, Vol. 5, No. 1, 2005, pp. 71.

- [15] Zhang, P. and Huang, S., “Review of Aeroelasticity for Wind Turbine: Current Status, Research Focus and Future Perspectives,” *Frontiers in Energy*, Vol. 5, No. 4, 2011, pp. 419–434.
- [16] Simms, D., Schreck, S., Hand, M., and Fingersh, L. J., “NREL Unsteady Aerodynamics Experiment in the NASA-Ames Wind Tunnel : A Comparison of Predictions to Measurements,” Tech. Rep. June, NREL, Golden, Colorado, 2001.
- [17] Jonkman, J. M. and Buhl Jr, M. L., “FAST User’s Guide,” Tech. rep., NREL, Golden, Colorado, 2005.
- [18] Jonkman, J. M., “Modeling of the UAE Wind Turbine for Refinement of FAST - AD,” Tech. rep., NREL, Golden, Colorado, 2003.
- [19] J.Laino, D. and Hansen, A. C., “User’s Guide to the Wind Turbine Aerodynamics Computer Software,” Tech. rep., Windward Engineering, LC, Salt Lake City, Utah, 2002.
- [20] Gasmi, A., Sprague, M. A., Jonkman, J. M., and Jones, W. B., “Numerical Stability and Accuracy of Temporally Coupled Multi-Physics Modules in,” *51st AIAA Aerospace Sciences Meeting*, Grapevine, Texas, 2013.
- [21] Nikravesh, P. E., “Understanding Mean-Axis Conditions as Floating Reference Frames,” *Advances in Computational Multibody Systems, Computational Methods in Applied Sciences*, Vol. 2, 2005, pp. 185–203.
- [22] Weghs, M., *Efficiency Improvement of Strongly Coupled Fluid-Structure Interaction Simulations*, Master thesis, Faculty of Aerospace Engineering, Delft University of Technology, Delft, Netherlands, 2010.

- [23] Ryu, G. S., *Integration of Heterogenous Simulation Models for Network-Distributed Simulation*, Ph.D. thesis, University of Michigan, Ann Arbor, Michigan, 2009.
- [24] Shabana, A. A., *Dynamics of Multibody Systems*, Cambridge University Press, New York, 2005, Chaps. 5,6.
- [25] Bauchau, O. A. and Craig, J. I., *Structural Analysis: With Applications to Aerospace Structures*, Springer, Berlin, Germany, 2009, pp. 400-407.
- [26] Richter, T., *Numerical Methods for Fluid-Structure Interaction Problems*, Course of Lectures, University of Heidelberg, Heidelberg, Germany, 2010. Available:<http://numerik.uni-hd.de/~richter/SS10/fsi/fsi.pdf>. Accessed on March 2013.
- [27] Küttler, U. and Wall, W. A., “Fixed-point Fluid-Structure Interaction Solvers with Dynamic Relaxation,” *Computational Mechanics*, Vol. 43, No. 1, Feb. 2008, pp. 61–72.
- [28] Logan, D. L., *A First Course in the Finite Element Method*, PWS Publishers, Boston, Massachusetts, 1986, pp. 145-148.
- [29] Das, M., *Aeroelastic Analysis of Rotor Blades Using Three Dimensional Multibody Dynamic Analysis*, Ph.D. thesis, University of Arizona, Tucson, Arizona, 2008.
- [30] Tsai, M. H., Lin, W. Y., Zhou, Y. C., and Hsiao, K. M., “A Corotational Finite Element Method Combined with Floating Frame Method for Large Steady-State Deformation and Free Vibration Analysis of a Rotating-Inclined Beam,” *Mathematical Problems in Engineering*, Vol. 2011, 2011, pp. 1–29.

- [31] Berzeri, M., Shabana, A. A., and Street, W. T., “A Finite Element Study of the Geometric Centrifugal Stiffening Effect,” Tech. rep., Department of Mechanical Engineering, University of Illinois at Chicago, Chicago, Illinois, 1999.
- [32] H.El-Absy and A.A.Shabana, “Geometric Stiffness and Stability of Rigid Body Modes,” *Journal of Sound and Vibration*, Vol. 207, 1997, pp. 465–496.
- [33] Nada, A. A., Hussein, B. A., Megahed, S. M., and Shabana, A. A., “Use of the Floating Frame of Reference Formulation in Large Deformation Analysis: Experimental and Numerical Validation,” *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, Vol. 224, No. 1, Jan. 2010, pp. 45–58.
- [34] Antoulas, A., “An Overview of Approximation Methods for Large-scale Dynamical Systems,” *Annual Reviews in Control*, Vol. 29, No. 2, Jan. 2005, pp. 181–190.

APPENDIX A

MATRIX EXPRESSIONS FOR THE PLANAR ROTATION OF THE TURBINE

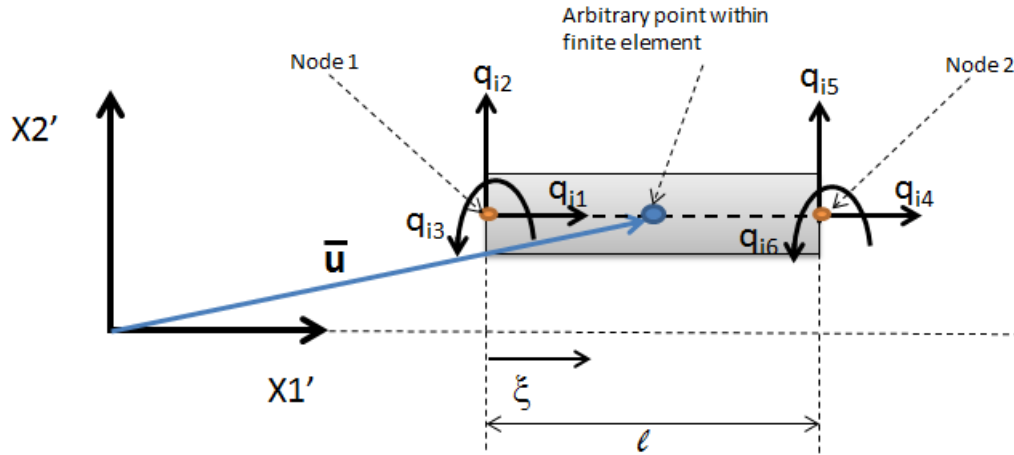


Fig. A.1: Beam Element Coordinates

In this section, the finite element matrix definitions such as the shape function, mass matrix and stiffness matrix are presented as derived by Shabana [13] for the two-dimensional beam element. Three dimensional formulations are found in Refs. [24], [23]. Figure A.1 shows the two node, two dimensional beam line element with three degrees of freedom - axial, transverse, and rotational - defined at the two nodes. $(X1', X2')$ refers to the floating frame attached to the flexible blade at the root. It is noted that since all the beam elements are assumed initially to lie along the $X2'$ axis, there is no need to define the intermediate element coordinate system [11]. However, the intermediate element coordinate system is required for flexible structures with complex shapes. The nodal coordinates shown can be used to define

both the displacements and the position with respect to the (X1',X2') frame of any arbitrary point within the element. This isoparametric nature is shown readily by the translational coordinates $q_{i_1}, q_{i_2}, q_{i_4}, q_{i_5}, q_{i_3}$ and q_{i_6} . The rotational displacements are represented in terms of slopes, hence these coordinates can be used to describe positions accurately as well only if small deformations are assumed. The ability to correctly describe the position of any point in the flexible body is important in order to derive the equations of motion. In the derivations presented, $\mathbf{q}_i = [q_{i_1} \dots q_{i_6}]^T$ refers to the displacements with respect to the (X1',X2') frame and $\mathbf{q}_{f_i} = [q_{f_{i_1}} \dots q_{f_{i_6}}]^T$ refers to nodal the deformations. The (X1',X2') frame provides the coordinate basis for both displacements and deformations.

The position $\bar{\mathbf{u}}_i$ at any point within the element is given by:

$$\bar{\mathbf{u}}_i = \mathbf{N}_i \mathbf{q}_i \quad (\text{A.1})$$

$\frac{d\bar{\mathbf{u}}_i}{d\mathbf{X1}'}$ at any point within the element is given by:

$$\frac{d\bar{\mathbf{u}}_i}{d\mathbf{X1}'} = \frac{d\mathbf{N}_i}{d\xi} \mathbf{q}_i \quad (\text{A.2})$$

The infinitesimal rotational displacements(slopes) from Eq.(A.2) are given by $\left(\frac{d\bar{\mathbf{u}}_i}{d\mathbf{X1}'}\right)_{\mathbf{X2}'}$, the component of the vector along the X2' axis. \mathbf{N}_i the shape function matrix given the element length l is defined as :

$$\mathbf{N}_i = \begin{bmatrix} 1 - \xi & 0 & 0 & \xi & 0 & 0 \\ 0 & 1 - 3\xi^2 - 2\xi^3 & l(\xi - 2\xi^2 + \xi^3) & 0 & 3\xi^2 - 2\xi^3 & l(\xi^3 - \xi^2) \end{bmatrix}$$

where ξ is the coordinate along the X1' axis (also the element axis) normalized with the beam length l and measuring from node 1 to node 2 (See Fig. A.1).

The definitions of the displacement of a point within the element is used to transfer nodal translational displacements, velocities and rotational displacements from the FEM module to FAST (to be used by AeroDyn). Since the nodes defined in AeroDyn are at the mid-section of the beam element, the shape function matrix is evaluated at the $\xi = 0.5$.

Using the shape function matrix and Eq.(2.3), the position vector in the inertial reference frame is obtained. Differentiating this vector as in Eq.(2.4), the velocity vector is obtained, using which an expression for the kinetic energy is formulated. Integrating the kinetic energy expression through the beam element (i.e along the ξ axis), the mass matrix for the element is derived. As stated in previous sections, since the blades are allowed to deform and rotate only in the in-plane directions and since the blades are assumed to be connected at the point of rotation, the terms in the mass matrix associated with rotation and deformation and the coupling between rotation and deformation need to be considered.

The part of the mass matrix related to the nodal deformations is given by:

$$\mathbf{m}_{ff_i} = \int_0^1 \rho a_i l_i \mathbf{N}_i^T \mathbf{N}_i d\xi \quad (\text{A.4})$$

Integrating through the beam the element using the shape function definition in Eq.(A.3), we obtain the following matrix expression assuming uniform beam properties:

$$\mathbf{m}_{\mathbf{f}_i} = 1/3 \begin{bmatrix} 1 & & & & & & \text{symmetric} \\ 0 & 13/35 & & & & & \\ 0 & 11l/210 & l^2/105 & & & & \\ 1/6 & 0 & 0 & 1/3 & & & \\ 0 & 9/70 & 13l/420 & 0 & 13/35 & & \\ 0 & -13l/420 & -l^2/140 & 0 & -11l/210 & l^2/105 & \end{bmatrix}$$

The mass matrix term related to the rotational coordinate θ is given as:

$$\mathbf{m}_{\theta\theta_i} = \int_0^1 \rho a_i l_i \bar{\mathbf{u}}_i^T \bar{\mathbf{u}}_i d\xi = \mathbf{q}_i^T \int_0^1 \rho a_i l_i \mathbf{N}_i^T \mathbf{N}_i d\xi \mathbf{q}_i = \mathbf{q}_i^T \mathbf{m}_{\mathbf{f}_i} \mathbf{q}_i \quad (\text{A.6})$$

Eq.(A.6) represents the inertia of the beam element about the X3' axis. Note that inertia is computed using the consistent mass approach. Also, though $\mathbf{m}_{\mathbf{f}_i}$ is constant, the inertia of the element is time varying since \mathbf{q}_i is time varying.

The part of the mass matrix related to the coupling between rigid and flexible coordinates is given as:

$$\mathbf{m}_{\theta\mathbf{f}_i} = \int_0^1 \rho a_i l_i \bar{\mathbf{u}}_i^T \tilde{\mathbf{I}} \mathbf{N}_i d\xi = \mathbf{q}_i^T \int_0^1 \rho a_i l_i \mathbf{N}_i^T \tilde{\mathbf{I}} \mathbf{N}_i d\xi \mathbf{q}_i = \mathbf{q}_i^T \tilde{\mathbf{N}}_i \quad (\text{A.7})$$

where $\tilde{\mathbf{N}}_i$ is given as:

$$\tilde{\mathbf{N}}_{\mathbf{i}} = m_i/60 \begin{bmatrix} 0 & 2l & 3l & 0 & 9 & -2l \\ -2l & 0 & 0 & -9 & 0 & 0 \\ -3l & 0 & 0 & -2l & 0 & 0 \\ 0 & 9 & 2l & 0 & 2l & -3l \\ -9 & 0 & 0 & -2l & 0 & 0 \\ 2l & 0 & 0 & 3l & 0 & 0 \end{bmatrix}$$

The stiffness matrix is obtained by using the strain energy expression for the Euler-Bernoulli beam model.

$$\mathbf{k}_{\mathbf{ff}_i} = EI_{x_i}/l \begin{bmatrix} a/l & & & & & & \textit{symmetric} \\ 0 & 12/l^2 & & & & & \\ 0 & 6/l & 4 & & & & \\ -a/l & 0 & 0 & a/l & & & \\ 0 & -12/l^2 & -6/l & 0 & 12/l^2 & & \\ 0 & 6/l & 2 & 0 & -6/l & 4 & \end{bmatrix}$$

It can be seen that the axial deformation is decoupled from the transverse deformation. Higher order beam theories which include such coupling do not suffer from instability problems at critical rotor speeds [31].

The quadratic velocity vector expressions are listed as follows adapted from Ref. [24]

$$\mathbf{Q}_{\mathbf{v}_{\theta_i}} = -2\dot{\theta}\mathbf{q}_{\mathbf{f}_i}^T(\mathbf{m}_{\mathbf{ff}_i}\mathbf{q}_{\mathbf{f}_i} + \bar{\mathbf{I}}_0) \quad (\text{A.10})$$

$$\mathbf{Q}_{\mathbf{v}_{\mathbf{f}_i}} = \dot{\theta}^2(\mathbf{m}_{\mathbf{ff}_i}\mathbf{q}_{\mathbf{f}_i} + \bar{\mathbf{I}}_0) + 2\dot{\theta}\tilde{\mathbf{N}}_i\dot{\mathbf{q}}_{\mathbf{f}_i} \quad (\text{A.11})$$

where

$$\bar{\mathbf{I}}_0 = \int_0^1 \rho a_i l_i \mathbf{N}_i^T \bar{\mathbf{u}}_{0_i} d\xi \quad (\text{A.12})$$

and $\bar{\mathbf{u}}_{0_i}$ represents an arbitrary point in the undeformed configuration.

Let \mathbf{q}_{0_i} represent the undeformed nodal coordinates, then $\bar{\mathbf{u}}_{0_i} = \mathbf{N}_i\mathbf{q}_{0_i}$. Since nodal displacements represent the superposition of deformations over the undeformed configuration, we have:

$$\mathbf{q}_i = \mathbf{q}_{0_i} + \mathbf{q}_{\mathbf{f}_i} \quad (\text{A.13})$$

Substituting Eq.(A.13) in Eqs.(A.10) and (A.11) and using the definition for $\mathbf{m}_{\mathbf{ff}_i}$ from Eq.(A.5), we have:

$$\mathbf{Q}_{\mathbf{v}_{\theta_i}} = -2\dot{\theta}\dot{\mathbf{q}}_{\mathbf{f}_i}^T \mathbf{m}_{\mathbf{ff}_i} \mathbf{q}_i \quad (\text{A.14})$$

$$\mathbf{Q}_{\mathbf{v}_{\mathbf{f}_i}} = \dot{\theta}^2 \mathbf{m}_{\mathbf{ff}_i} \mathbf{q}_i + 2\dot{\theta}\tilde{\mathbf{N}}_i\dot{\mathbf{q}}_{\mathbf{f}_i} \quad (\text{A.15})$$

The matrices and quadratic velocity vector expressions are presented above for the element i . To accomplish the assembly of the finite elements, a Boolean transformation connecting two adjacent elements is employed. For example, for n_{nodes} nodes let $\mathbf{q} = [q_{1_1} \dots q_{n_{nodes}_3}]^T$ represent the nodal displacements for all nodes in one blade with the finite elements assembled. Then, displacements for the first element, $\mathbf{q}_1 = [q_{1_1} \dots q_{1_6}]^T$, used to derive the expressions above is selected from \mathbf{q} by using the constant Boolean transformation matrix \mathbf{B}_1 ,

$$\mathbf{q}_1 = \mathbf{B}_1 \mathbf{q} \quad (\text{A.16})$$

where \mathbf{B}_1 is a square matrix of $n_{nodes} \times 6$, populated with zeros except for the

first six elements along the diagonal which are unity. Note, that with reference to Fig. A.1, each node has 3 degrees of freedom and each element has with 6 degrees of freedom.

Substituting $\mathbf{q}_i = \mathbf{B}_i \mathbf{q}$ for \mathbf{q}_i in the equations used to derive the mass and stiffness matrices and the quadratic velocity vector expressions and summing through the elements as shown below for instance for the mass sub-matrix associated with the deformation coordinates, the finite element assembly is accomplished.

$$\mathbf{M}_{ff} = \sum_i \mathbf{B}_i^T \mathbf{m}_{ff_i} \mathbf{B}_i \quad (\text{A.17})$$

APPENDIX B

FAST DATA INPUTS

FAST INPUT

----- FAST INPUT FILE -----
FAST certification Test #08(Modified): AOC 15/50 with only Drive Train DOF, with fixed yaw error and steady wind.
Many parameters are pure fiction. Compatible with FAST v7.00.00.
----- SIMULATION CONTROL -----
False Echo - Echo input data to "echo.out" (flag)
1 ADAMSPrep - ADAMS preprocessor mode {1: Run FAST, 2: use FAST as a preprocessor to create an ADAMS model, 3: do both} (switch)
1 AnalMode - Analysis mode {1: Run a time-marching simulation, 2: create a periodic linearized model} (switch)
3 NumBl - Number of blades (-)
1.0 TMax - Total run time (s)
0.001 DT - Integration time step (s)
----- TURBINE CONTROL -----
0 YCMode - Yaw control mode {0: none, 1: user-defined from routine UserYawCont, 2: user-defined from Simulink} (switch)
9999.9 TYCon - Time to enable active yaw control (s) [unused when YCMode=0]
0 PCMode - Pitch control mode {0: none, 1: user-defined from routine PitchCntrl, 2: user-defined from Simulink} (switch)
9999.9 TPCOn - Time to enable active pitch control (s) [unused when PCMode=0]
0 VSContrl - Variable-speed control mode {0: none, 1: simple VS, 2: user-defined from routine UserVSCont, 3: user-defined from Simulink} (switch)
9999.9 VS_RtGnSp - Rated generator speed for simple variable-speed generator control (HSS side) (rpm) [used only when VSContrl=1]
9999.9 VS_RtTq - Rated generator torque/constant generator torque in Region 3 for simple variable-speed generator control (HSS side) (N-m) [used only when VSContrl=1]
9999.9 VS_Rgn2K - Generator torque constant in Region 2 for simple variable-speed generator control (HSS side) (N-m/rpm²) [used only when VSContrl=1]
9999.9 VS_SIPc - Rated generator slip percentage in Region 2 1/2 for simple variable-speed generator control (%) [used only when VSContrl=1]

1 GenModel - Generator model {1: simple, 2: Thevenin, 3: user-defined from routine UserGen} (switch) [used only when VSContrl=0]

True GenTiStr - Method to start the generator {T: timed using TimGenOn, F: generator speed using SpdGenOn} (flag)

True GenTiStp - Method to stop the generator {T: timed using TimGenOf, F: when generator power = 0} (flag)

9999.9 SpdGenOn - Generator speed to turn on the generator for a startup (HSS speed) (rpm) [used only when GenTiStr=False]

0.0 TimGenOn - Time to turn on the generator for a startup (s) [used only when GenTiStr=True]

9999.9 TimGenOf - Time to turn off the generator (s) [used only when GenTiStp=True]

1 HSSBrMode - HSS brake model {1: simple, 2: user-defined from routine UserHSSBr} (switch)

9999.9 THSSBrDp - Time to initiate deployment of the HSS brake (s)

9999.9 TiDynBrk - Time to initiate deployment of the dynamic generator brake [CURRENTLY IGNORED] (s)

9999.9 TTPBrDp(1) - Time to initiate deployment of tip brake 1 (s)

9999.9 TTPBrDp(2) - Time to initiate deployment of tip brake 2 (s)

9999.9 TTPBrDp(3) - Time to initiate deployment of tip brake 3 (s) [unused for 2 blades]

9999.9 TBDepISp(1) - Deployment-initiation speed for the tip brake on blade 1 (rpm)

9999.9 TBDepISp(2) - Deployment-initiation speed for the tip brake on blade 2 (rpm)

9999.9 TBDepISp(3) - Deployment-initiation speed for the tip brake on blade 3 (rpm) [unused for 2 blades]

9999.9 TYawManS - Time to start override yaw maneuver and end standard yaw control (s)

9999.9 TYawManE - Time at which override yaw maneuver reaches final yaw angle (s)

0.0 NacYawF - Final yaw angle for yaw maneuvers (degrees)

9999.9 TPitManS(1) - Time to start override pitch maneuver for blade 1 and end standard pitch control (s)

9999.9 TPitManS(2) - Time to start override pitch maneuver for blade 2 and end standard pitch control (s)

9999.9 TPitManS(3) - Time to start override pitch maneuver for blade 3 and end standard pitch control (s) [unused for 2 blades]

9999.9 TPitManE(1) - Time at which override pitch maneuver for blade 1 reaches final pitch (s)

9999.9 TPitManE(2) - Time at which override pitch maneuver for blade 2 reaches final pitch (s)

9999.9 TPitManE(3) - Time at which override pitch maneuver for blade 3 reaches final pitch (s) [unused for 2 blades]

0.0 BIPitch(1) - Blade 1 initial pitch (degrees)

0.0 BIPitch(2) - Blade 2 initial pitch (degrees)

0.0 BIPitch(3) - Blade 3 initial pitch (degrees) [unused for 2 blades]

0.0 BIPitchF(1) - Blade 1 final pitch for pitch maneuvers (degrees)

0.0 BIPitchF(2) - Blade 2 final pitch for pitch maneuvers (degrees)

0.0 BIPitchF(3) - Blade 3 final pitch for pitch maneuvers (degrees) [unused for 2 blades]

----- ENVIRONMENTAL CONDITIONS -----

9.8 Gravity - Gravitational acceleration (m/s²)

----- FEATURE FLAGS -----

False FlapDOF1 - First flapwise blade mode DOF (flag)
False FlapDOF2 - Second flapwise blade mode DOF (flag)
False EdgeDOF - First edgewise blade mode DOF (flag)
False TeetDOF - Rotor-teeter DOF (flag) [unused for 3 blades]
True DrTrDOF - Drivetrain rotational-flexibility DOF (flag)
False GenDOF - Generator DOF (flag)
False YawDOF - Yaw DOF (flag)
False TwFADOF1 - First fore-aft tower bending-mode DOF (flag)
False TwFADOF2 - Second fore-aft tower bending-mode DOF (flag)
False TwSSDOF1 - First side-to-side tower bending-mode DOF (flag)
False TwSSDOF2 - Second side-to-side tower bending-mode DOF (flag)
True CompAero - Compute aerodynamic forces (flag)
False CompNoise - Compute aerodynamic noise (flag)

----- INITIAL CONDITIONS -----

0.0 OoPDefl - Initial out-of-plane blade-tip displacement (meters)
0.0 IPDefl - Initial in-plane blade-tip deflection (meters)
0.0 TeetDefl - Initial or fixed teeter angle (degrees) [unused for 3 blades]
0.0 Azimuth - Initial azimuth angle for blade 1 (degrees)
64.14 RotSpeed - Initial or fixed rotor speed (rpm)
-15.0 NacYaw - Initial or fixed nacelle-yaw angle (degrees)
0.0 TTDspFA - Initial fore-aft tower-top displacement (meters)
0.0 TTDspSS - Initial side-to-side tower-top displacement (meters)

----- TURBINE CONFIGURATION -----

7.490 TipRad - The distance from the rotor apex to the blade tip (meters)
0 HubRad - The distance from the rotor apex to the blade root (meters)
1 PSpnEIN - Number of the innermost blade element which is still part of the pitchable portion of the blade for partial-span pitch control [1 to BldNodes] [CURRENTLY IGNORED] (-)
0.0 UndSling - Undersling length [distance from teeter pin to the rotor apex] (meters) [unused for 3 blades]

0.0 HubCM - Distance from rotor apex to hub mass [positive downwind] (meters)

1.341 OverHang - Distance from yaw axis to rotor apex [3 blades] or teeter pin [2 blades] (meters)

0.0 NacCMxn - Downwind distance from the tower-top to the nacelle CM (meters)

0.0 NacCMyn - Lateral distance from the tower-top to the nacelle CM (meters)

0.6 NacCMzn - Vertical distance from the tower-top to the nacelle CM (meters)

24.4 TowerHt - Height of tower above ground level [onshore] or MSL [offshore] (meters)

0.6 Twr2Shft - Vertical distance from the tower-top to the rotor shaft (meters)

0.0 TwrRBHt - Tower rigid base height (meters)

0.0 ShftTilt - Rotor shaft tilt angle (degrees)

0.0 Delta3 - Delta-3 angle for teetering rotors (degrees) [unused for 3 blades]

0.0 PreCone(1) - Blade 1 cone angle (degrees)

0.0 PreCone(2) - Blade 2 cone angle (degrees)

0.0 PreCone(3) - Blade 3 cone angle (degrees) [unused for 2 blades]

0.0 AzimB1Up - Azimuth value to use for I/O when blade 1 points up (degrees)

----- MASS AND INERTIA -----

0.0 YawBrMass - Yaw bearing mass (kg)

1747.0 NacMass - Nacelle mass (kg)

247.3 HubMass - Hub mass (kg)

0.0 TipMass(1) - Tip-brake mass, blade 1 (kg)

0.0 TipMass(2) - Tip-brake mass, blade 2 (kg)

0.0 TipMass(3) - Tip-brake mass, blade 3 (kg) [unused for 2 blades]

976.3 NacYIner - Nacelle inertia about yaw axis (kg m²)

10.0 GenIner - Generator inertia about HSS (kg m²)

9.0 HubIner - Hub inertia about rotor axis [3 blades] or teeter axis [2 blades] (kg m²)

----- DRIVETRAIN -----

100.0 GBoxEff - Gearbox efficiency (%)

89.4 GenEff - Generator efficiency [ignored by the Thevenin and user-defined generator models] (%)

28.25 GBRatio - Gearbox ratio (-)

False GBRevers - Gearbox reversal {T: if rotor and generator rotate in opposite directions} (flag)

9999.9 HSSBrTqF - Fully deployed HSS-brake torque (N-m)

9999.9 HSSBrDt - Time for HSS-brake to reach full deployment once initiated (sec) [used only when HSSBrMode=1]

"" DynBrkFi - File containing a mech-gen-torque vs HSS-speed curve for a dynamic brake [CURRENTLY IGNORED]
(quoted string)

6.0E5 DTTorSpr - Drivetrain torsional spring (N-m/rad)

1.0E3 DTTorDmp - Drivetrain torsional damper (N-m/(rad/s))

----- SIMPLE INDUCTION GENERATOR ----- Crude approximation of torque/speed curve.

2.222 SIG_SIPc - Rated generator slip percentage (%) [used only when VSContrl=0 and GenModel=1]

1800.0 SIG_SySp - Synchronous (zero-torque) generator speed (rpm) [used only when VSContrl=0 and GenModel=1]

314.3 SIG_RiTq - Rated torque (N-m) [used only when VSContrl=0 and GenModel=1]

1.75 SIG_PORt - Pull-out ratio (Tpullout/Trated) (-) [used only when VSContrl=0 and GenModel=1]

----- THEVENIN-EQUIVALENT INDUCTION GENERATOR -----

60.0 TEC_Freq - Line frequency [50 or 60] (Hz) [used only when VSContrl=0 and GenModel=2]

4 TEC_NPol - Number of poles [even integer > 0] (-) [used only when VSContrl=0 and GenModel=2]

4.92E-02 TEC_SRes - Stator resistance (ohms) [used only when VSContrl=0 and GenModel=2]

5.34E-04 TEC_RRes - Rotor resistance (ohms) [used only when VSContrl=0 and GenModel=2]

480.0 TEC_VLL - Line-to-line RMS voltage (volts) [used only when VSContrl=0 and GenModel=2]

1.00E-04 TEC_SLR - Stator leakage reactance (ohms) [used only when VSContrl=0 and GenModel=2]

1.00E-04 TEC_RLR - Rotor leakage reactance (ohms) [used only when VSContrl=0 and GenModel=2]

4.49E-03 TEC_MR - Magnetizing reactance (ohms) [used only when VSContrl=0 and GenModel=2]

----- PLATFORM -----

0 PtfmModel - Platform model {0: none, 1: onshore, 2: fixed bottom offshore, 3: floating offshore} (switch)

"" PtfmFile - Name of file containing platform properties (quoted string) [unused when PtfmModel=0]

----- TOWER -----

11 TwrNodes - Number of tower nodes used for analysis (-)

"AOC_Tower.dat" TwrFile - Name of file containing tower properties (quoted string)

----- NACELLE-YAW -----

0.0 YawSpr - Nacelle-yaw spring constant (N-m/rad)

0.0 YawDamp - Nacelle-yaw damping constant (N-m/(rad/s))

0.0 YawNeut - Neutral yaw position--yaw spring force is zero at this yaw (degrees)

----- FURLING -----

False Furling - Read in additional model properties for furling turbine (flag)

"" FurlFile - Name of file containing furling properties (quoted string) [unused when Furling=False]

----- ROTOR-TEETER -----

0 TeetMod - Rotor-teeter spring/damper model {0: none, 1: standard, 2: user-defined from routine UserTeet} (switch)
[unused for 3 blades]

0.0 TeetDmpP - Rotor-teeter damper position (degrees) [used only for 2 blades and when TeetMod=1]

0.0 TeetDmp - Rotor-teeter damping constant (N-m/(rad/s)) [used only for 2 blades and when TeetMod=1]

0.0 TeetCDmp - Rotor-teeter rate-independent Coulomb-damping moment (N-m) [used only for 2 blades and when TeetMod=1]

0.0 TeetSStP - Rotor-teeter soft-stop position (degrees) [used only for 2 blades and when TeetMod=1]

0.0 TeetHStP - Rotor-teeter hard-stop position (degrees) [used only for 2 blades and when TeetMod=1]

0.0 TeetSSSp - Rotor-teeter soft-stop linear-spring constant (N-m/rad) [used only for 2 blades and when TeetMod=1]

0.0 TeetHSSp - Rotor-teeter hard-stop linear-spring constant (N-m/rad) [used only for 2 blades and when TeetMod=1]

----- TIP-BRAKE -----

0.0 TBDrConN - Tip-brake drag constant during normal operation, Cd*Area (m²)

0.0 TBDrConD - Tip-brake drag constant during fully-deployed operation, Cd*Area (m²)

0.0 TpBrDT - Time for tip-brake to reach full deployment once released (sec)

----- BLADE -----

"AOC_Blade.dat" BldFile(1) - Name of file containing properties for blade 1 (quoted string)

"AOC_Blade.dat" BldFile(2) - Name of file containing properties for blade 2 (quoted string)

"AOC_Blade.dat" BldFile(3) - Name of file containing properties for blade 3 (quoted string) [unused for 2 blades]

----- AERODYN -----

"Test08_FEM_AD.ipt" ADFile - Name of file containing AeroDyn input parameters (quoted string)

----- NOISE -----

"" NoiseFile - Name of file containing aerodynamic noise input parameters (quoted string) [used only when CompNoise=True]

----- ADAMS -----

"AOC_ADAMS.dat" ADAMSFile - Name of file containing ADAMS-specific input parameters (quoted string) [unused when ADAMSPrep=1]

----- LINEARIZATION CONTROL -----

"AOC_Linear.dat" LinFile - Name of file containing FAST linearization parameters (quoted string) [unused when AnalMode=1]

----- OUTPUT -----

True SumPrint - Print summary data to "<RootName>.fsm" (flag)

True TabDelim - Generate a tab-delimited tabular output file. (flag)
 "ES10.3E2" OutFmt - Format used for tabular output except time. Resulting field should be 10 characters. (quoted string) [not checked for validity!]
 0.0 TStart - Time to begin tabular output (s)
 10 DecFact - Decimation factor for tabular output {1: output every time step} (-)
 5.0 SttsTime - Amount of time between screen status messages (sec)
 0.0 NcIMUxn - Downwind distance from the tower-top to the nacelle IMU (meters)
 0.0 NcIMUyn - Lateral distance from the tower-top to the nacelle IMU (meters)
 0.0 NcIMUzn - Vertical distance from the tower-top to the nacelle IMU (meters)
 0.5 ShftGagL - Distance from rotor apex [3 blades] or teeter pin [2 blades] to shaft strain gages [positive for upwind rotors] (meters)
 0 NTwGages - Number of tower nodes that have strain gages for output [0 to 9] (-)
 0 TwrGagNd - List of tower nodes that have strain gages [1 to TwrNodes] (-) [unused if NTwGages=0]
 0 NBIGages - Number of blade nodes that have strain gages for output [0 to 9] (-)
 2.6 BldGagNd - List of blade nodes that have strain gages [1 to BldNodes] (-) [unused if NBIGages=0]
 OutList - The next line(s) contains a list of output parameters. See OutList.txt for a listing of available output channels, (-)
 "RootMxb1,Azimuth,RotSpeed,RotAccel" - Blade 2 flapwise and edgewise tip deflections RootMxb1,TipDyb1
 END of FAST input file (the word "END" must appear in the first 3 columns of this last line).

AERODYN INPUT

AOC aerodynamic parameters for FAST Certification Test #8.

SI SysUnits - System of units for used for input and output [must be SI for FAST] (unquoted string)
 STEADY StallMod - Dynamic stall included [BEDDOES or STEADY] (unquoted string)
 NO_CM UseCm - Use aerodynamic pitching moment model? [USE_CM or NO_CM] (unquoted string)
 EQUIL InfModel - Inflow model [DYNIN or EQUIL] (unquoted string)
 SWIRL IndModel - Induction-factor model [NONE or WAKE or SWIRL] (unquoted string)
 0.005 AToler - Induction-factor tolerance (convergence criteria) (-)
 PRANDtl TLModel - Tip-loss model (EQUIL only) [PRANDtl, GTECH, or NONE] (unquoted string)
 NONE HLMModel - Hub-loss model (EQUIL only) [PRANDtl or NONE] (unquoted string)

```

"Wind\AOC\Shr11_30.wnd"      WindFile - Name of file containing wind data (quoted string)

25.0          HH      - Wind reference (hub) height [TowerHt+Twr2Shft+OverHang*SIN(ShftTilt)] (m)

0.3           TwrShad - Tower-shadow velocity deficit (-)

0.2           ShadHWid - Tower-shadow half width (m)

1.341        T_Shad_Refpt - Tower-shadow reference point (m)

0.9526       AirDens - Air density (kg/m^3)

1.4639e-5    KinVisc - Kinematic air viscosity (m^2/sec)

0.005        DTAero - Time interval for aerodynamic calculations (sec)

5            NumFoil - Number of airfoil files (-)

"AeroData\AOC\S814_1.DAT "    FoilNm - Names of the airfoil files [NumFoil lines] (quoted strings)

"AeroData\AOC\S814_15.DAT"

"AeroData\AOC\S812_15.DAT"

"AeroData\AOC\S812_2.DAT "

"AeroData\AOC\S813_15.DAT"

5            BldNodes - Number of blade nodes used for analysis (-)

RNodes  AeroTwst DRNodes Chord NFoil PrnElm

0.749   3.419   1.498   0.6019   2      NOPRINT

2.247   3.419   1.498   0.6019   2      NOPRINT

3.745   3.419   1.498   0.6019   2      NOPRINT

5.243   3.419   1.498   0.6019   2      NOPRINT

6.741   3.419   1.498   0.6019   2      NOPRINT

```

WIND INPUT

! Wind file for sheared 18 m/s wind with 30 degree direction.

! Time	Wind	Wind	Vert.	Horiz.	Vert.	LinV	Gust			
!			Speed	Dir	Speed	Shear		Shear	Shear	Speed
0.0		11.0	30.0	0.0	0.0		0.2	0.0	0.0	
0.1		11.0	30.0	0.0	0.0		0.2	0.0	0.0	
999.9		11.0	30.0	0.0	0.0		0.2	0.0	0.0	

BLADE INPUT

```

-----
----- FAST INDIVIDUAL BLADE FILE -----
AOC 15/50 blade file. 9999 represent dummy values. Edge mode shapes not used. Distributed properties word-wrapped to fit in
page
----- BLADE PARAMETERS -----
2    NBInpSt  - Number of blade input stations (-)
False CalcBMode - Calculate blade mode shapes internally {T: ignore mode shapes from below, F: use mode shapes from
below} [CURRENTLY IGNORED] (flag)
0.0  BldFIDmp(1) - Blade flap mode #1 structural damping in percent of critical (%)
0.0  BldFIDmp(2) - Blade flap mode #2 structural damping in percent of critical (%)
0.0  BldEdDmp(1) - Blade edge mode #1 structural damping in percent of critical (%)
----- BLADE ADJUSTMENT FACTORS -----
1.0  F1StTunr(1) - Blade flapwise modal stiffness tuner, 1st mode (-)
1.0  F1StTunr(2) - Blade flapwise modal stiffness tuner, 2nd mode (-)
1.0  AdjBIMs   - Factor to adjust blade mass density (-)
1.0  AdjF1St   - Factor to adjust blade flap stiffness (-)
1.0  AdjEdSt   - Factor to adjust blade edge stiffness (-)
----- DISTRIBUTED BLADE PROPERTIES -----
BIFract AeroCent StrcTwst BMassDen FlpStff EdgStff GJStff EASStff Alpha FlpIner EdgIner PrecrvRef PreswpRef
FlpcgOf  EdgcgOf  FlpEAOf  EdgEAOf
(-) (-) (deg) (kg/m) (Nm^2) (Nm^2) (Nm^2) (N) (-) (kg m) (kg m) (m) (m) (m) (m)
(m) (m)
0      0.25  -90    20.17909091    6144424.364    9999  9999  9999  9999  9999
      9999  9999  9999  9999  9999  9999  9999
1      0.25  -90    20.17909091    6144424.364    9999  9999  9999  9999  9999
      9999  9999  9999  9999  9999  9999  9999
----- BLADE MODE SHAPES -----
1.6006 BldF11Sh(2) - Flap mode 1, coeff of x^2

```

-0.0307 BldF11Sh(3) - , coeff of x^3
-1.4453 BldF11Sh(4) - , coeff of x^4
1.2027 BldF11Sh(5) - , coeff of x^5
-0.3273 BldF11Sh(6) - , coeff of x^6
-9.7180 BldF12Sh(2) - Flap mode 2, coeff of x^2
9.9859 BldF12Sh(3) - , coeff of x^3
20.0685 BldF12Sh(4) - , coeff of x^4
-29.9356 BldF12Sh(5) - , coeff of x^5
10.5992 BldF12Sh(6) - , coeff of x^6
1.6006 BldEdgSh(2) - Edge mode 1, coeff of x^2
-0.0307 BldEdgSh(3) - , coeff of x^3
-1.4453 BldEdgSh(4) - , coeff of x^4
1.2027 BldEdgSh(5) - , coeff of x^5
-0.3273 BldEdgSh(6) - , coeff of x^6