Overview of the New Modularization Framework for FAST

NREL/DOE Workshop on the New Modularization Framework for FAST

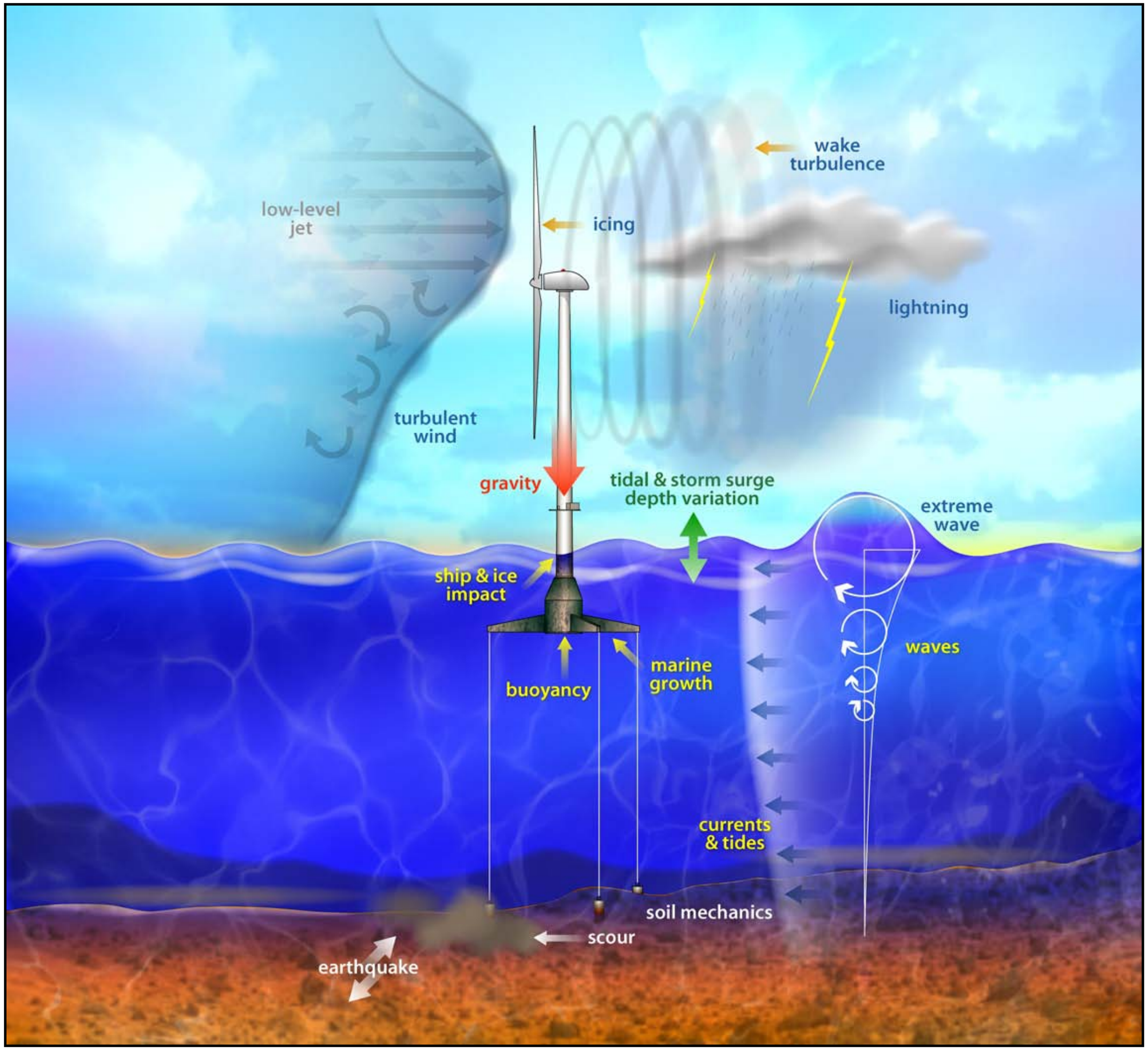
October 8, 2012
Virginia Beach, VA

Jason Jonkman, Ph.D.
Senior Engineer, NREL

Outline

- Introduction & Background:
  - Modeling Requirements
  - Existing FAST-AeroDyn-HydroDyn Coupling
  - Why is a New Framework Needed?
  - Benefits of the New Modularization Framework
- Features of the New Framework:
  - Inputs, Outputs, States, & Parameters
  - Loose & Tight Coupling
  - Module Form
  - Input-Output Transformations
  - Independent Time & Spatial Discretizations
  - Time Marching, OP Determination, & Linearization
  - Modular Interface & Coupling (AKA “Glue Code”)
  - Data Encapsulation & Dynamic Allocation
  - Other Features
  - Summary
- Current & Planned Work:
  - Progress to Date & Timeframe for Completion
  - New Modules Under Development

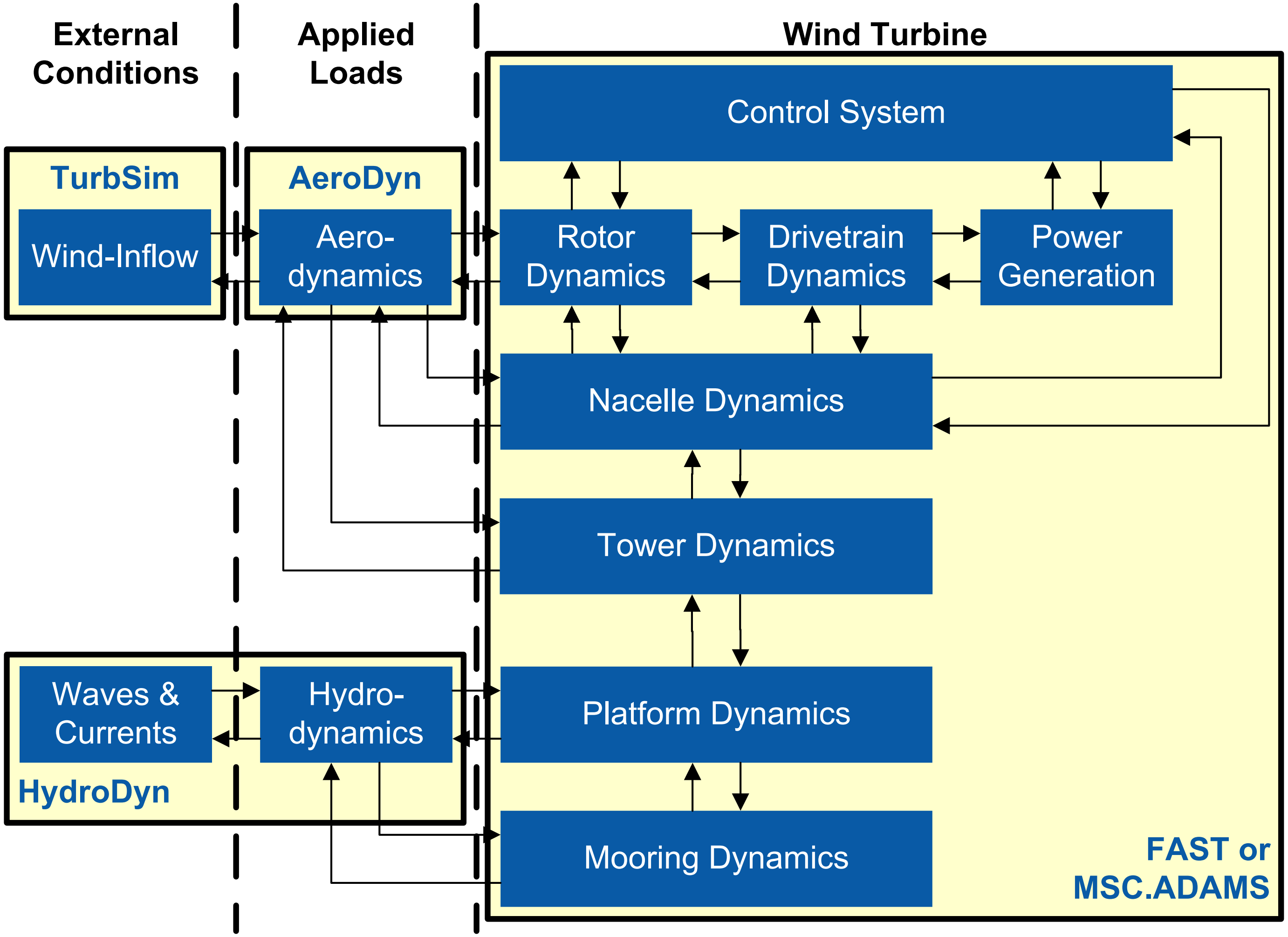
Introduction & Background
Modeling Requirements



- Coupled aero-hydro-servo-elastic interaction
- Wind-inflow:
  - –Discrete events
  - –Turbulence
- Waves:
  - –Regular
  - –Irregular
- Aerodynamics:
  - –Induction
  - –Rotational augmentation
  - –Skewed wake
  - –Dynamic stall
- Hydrodynamics:
  - –Diffraction
  - –Radiation
  - –Hydrostatics
- Structural dynamics:
  - –Gravity / inertia
  - –Elasticity
  - –Foundations / moorings
- Control system:
  - –Yaw, torque, pitch

Introduction & Background

Existing FAST-AeroDyn-HydroDyn Coupling



Introduction & Background

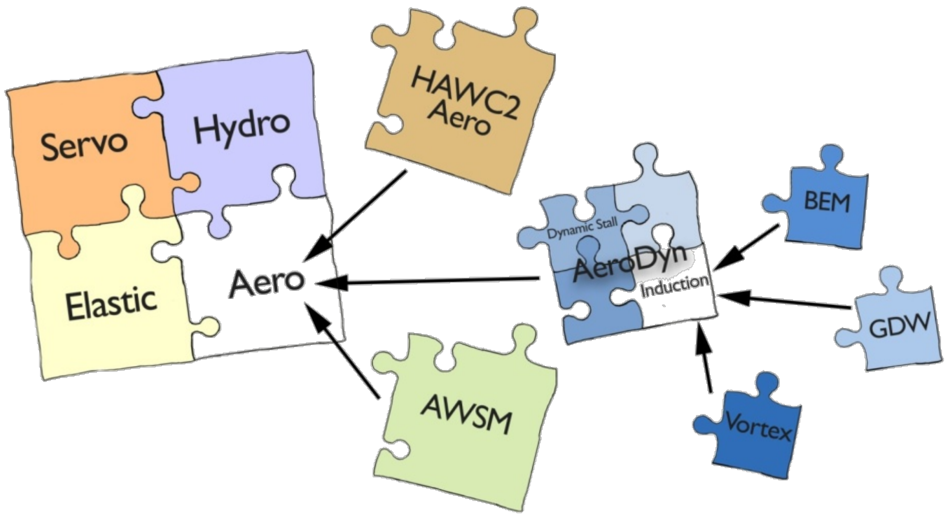
Why is a New Framework Needed?

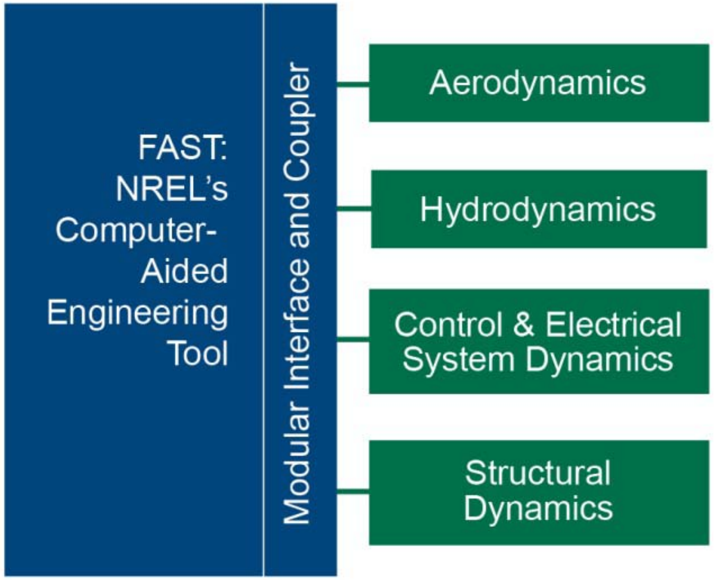
Problem	Solution
• Limited range of modeling fidelity	• Framework allowing modules to be exchanged • Development of new modules of higher fidelity
• Solution driven by structural solver	• Separate module interface & coupler
• Inability to isolate a given model	• Modules that can be called by separate driver programs or interfaced together to form a coupled solution
• Dependent spatial discretizations & time steps across modules	• Library of spatial elements & mesh-to-mesh mapping • Data transfer with interpolation/extrapolation in time
• Inability to linearize all system equations	• Tight coupling with options for operating-point determination & linearization
• Focus on single turbine	• Dynamic allocation of modules for wind-plant simulation
• "Spaghetti code" due to unclear data transfer & global data	• Modularization with data encapsulation
• Limited number of developers due to code size & complexity	• Modularization of code into separate components • Programmer's handbook explaining code development requirements & best practices
• Potentially poor numerical accuracy & stability	• Multiple coupling schemes & integration/solver options

Introduction & Background

Benefits of the New Modularization Framework

- Improve the ability to read, implement, & maintain source code
- Increase module sharing & shared code development across the wind community
- Improve numerical performance & robustness
- Enhance flexibility & expandability to enable further developments of functionality without the need to recode established modules





Features of the New Framework

Inputs, Outputs, States, & Parameters

- *Inputs* (*u*) – Values supplied to a module
- *Outputs* (*y*) – Values calculated by & returned from a module
- *States* – Internal values influenced by inputs &/or time:
  - *Continuous States* (*x*) – Values differentiable in time & characterized by differential equations
  - *Discrete States* (x^d) – Values at discrete steps in time & characterized by difference (recurrence) equations
  - *Constraint States* (*z*) – Algebraic values characterized by algebraic constraint equations
- *Parameters* (*p*) – Internal values fully definable at initialization:
  - Possibly with time dependence fully prescribed at initialization
  - Coefficients of a module's state & output equations
- All are module-independent (defined by the module developer)
- *u* must be algebraically derivable from *y* of available modules

Features of the New Framework

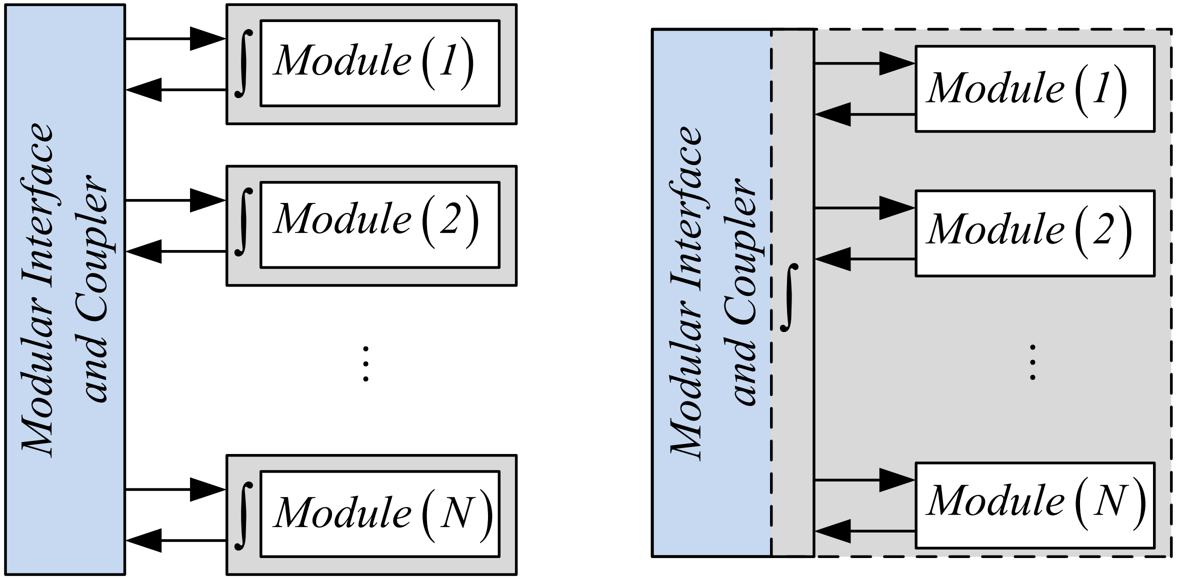
u, *y*, *x*, x^d, *z*, *p* Examples

Variable	Aerodynamics	Hydrodynamics	Controller	Structural Dynamics
u	• Turbine displacements • Turbine velocities	• Substructure displ. • Substructure velocities	• Structural accelerations • Reaction loads	• Aerodynamic loads • Hydrodynamic loads • Controller commands
y	• Aerodynamic loads	• Hydrodynamic loads	• Controller Commands	• Displacements • Velocities • Accelerations • Reaction loads
x	• Induction in GDW	• State-space-based radiation “memory”	• Analog control signals	• Displacements • Velocities
x^d	• B-L dynamic-stall states	• Convolution-based radiation “memory”	• Digital control signals	
z	• Induction in BEM			• Constraint loads at joints • Quasi-static mooring
p	• Geometry • Static airfoil data • Undisturbed wind inflow	• Geometry • Hydro coefficients • Undisturbed incident waves	• Controller gains • Control limits	• Geometry • Mass/inertia • Stiffness

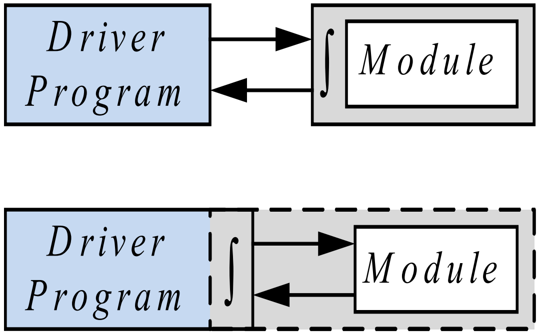
Features of the New Framework

Loose & Tight Coupling

- *Loose Coupling* – *u*/*y* exchanged between modules at each step, but each module tracks its own states & integrates its equations with its own solver
- *Tight Coupling* – Each module sets up its own equations, but the states are tracked & integrated by a solver common to all modules
- *Uncoupled* – Module called individually by a separate driver program



Loose- (left) & tight- (right) coupling schemes



Uncoupled solution of a module intended for loose (top) & tight (bottom) coupling

Features of the New Framework
Module Form

- General (need-not-be linear) state-space formulation
- Outputs permit direct feedthrough of input
- No constraints → ODEs
- With constraints → DAEs
- Tight coupling restricted to semi-explicit index-1 DAE
- No restriction on formulation in loose coupling, but solution is up to the developer
- A module can be set up for both loose & tight coupling

$$Given: p, x(0), x^d[0], \& \ u$$

$$\dot{x} = X\left(x, x^d, z, u, t\right)$$

$$x^d[n+1] = X^d\left[x|_{t=n\Delta t}, x^d[n], z|_{t=n\Delta t}, u|_{t=n\Delta t}, t|_{t=n\Delta t}\right]$$

$$0 = Z\left(x, x^d, z, u, t\right)$$

$$y = Y\left(x, x^d, z, u, t\right)$$

$$Note: Index-1\ DAE\ implies \left|\frac{\partial Z}{\partial z}\right| \neq 0,$$

$$thus \left[\frac{\partial Z}{\partial z}\right]^{-1} exists$$

$$Note: If\ x^d\ exists, x^d[n]\ is\ applied\ over$$

$$n\Delta t <= t < (n+1)\Delta t\ in\ all\ equations$$

Features of the New Framework

Input-Output Transformations

- *u* algebraically derived from *y*:
  - Customized for each grouping of modules
- *u*/*y* transformations (*U*) form additional constraint equations

$$Given: N \text{ coupled systems}$$

$$u = \begin{Bmatrix} u^{(1)} \\ u^{(2)} \\ \vdots \\ u^{(N)} \end{Bmatrix}, y = \begin{Bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{Bmatrix}, etc...$$

$$0 = U(u,y) \; or$$

$$0 = U(\tilde{u},y) \; with \left|\frac{\partial U}{\partial \tilde{u}}\right| \neq 0$$

$$\Rightarrow \quad 0 = U\left(u, Y\left(x, x^d, z, u, t\right)\right)$$

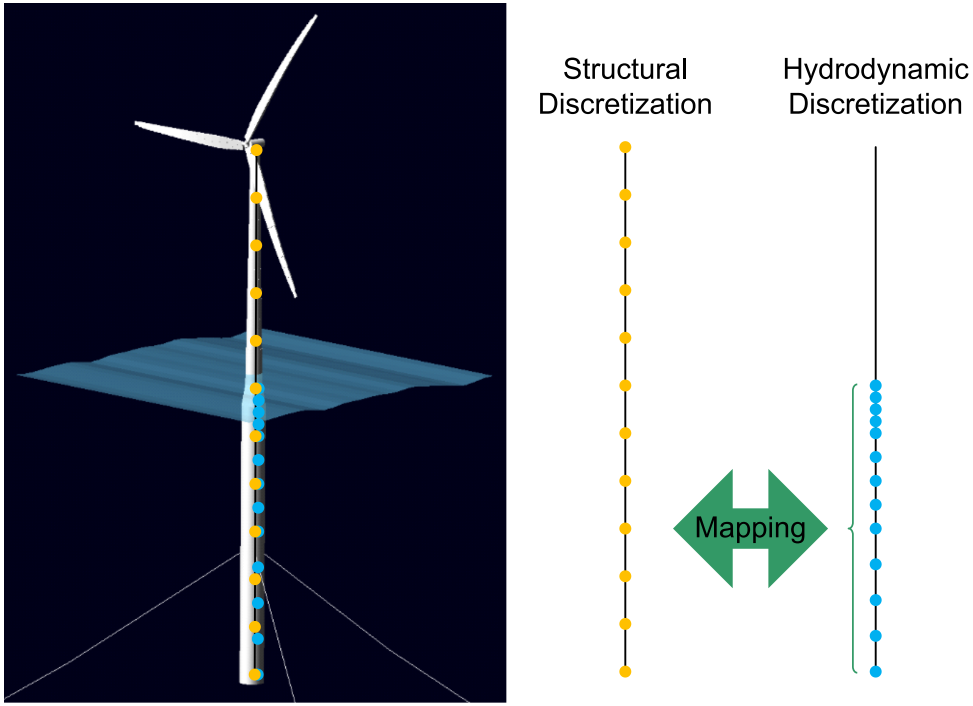
$$0 = U\left(x, x^d, z, u, t\right)$$

$$\Rightarrow \quad \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} Z\left(x, x^d, z, u, t\right) \\ U\left(x, x^d, z, u, t\right) \end{Bmatrix}$$

$$\left| \frac{\partial U}{\partial \tilde{u}} + \frac{\partial U}{\partial y} \begin{bmatrix} \left[\frac{\partial Y^{(1)}}{\partial u^{(1)}} - \frac{\partial Y^{(1)}}{\partial z^{(1)}}\left[\frac{\partial Z^{(1)}}{\partial z^{(1)}}\right]^{-1}\frac{\partial Z^{(1)}}{\partial u^{(1)}}\right] & 0 & \cdots & 0 \\ 0 & \left[\frac{\partial Y^{(2)}}{\partial u^{(2)}} - \frac{\partial Y^{(2)}}{\partial z^{(2)}}\left[\frac{\partial Z^{(2)}}{\partial z^{(2)}}\right]^{-1}\frac{\partial Z^{(2)}}{\partial u^{(2)}}\right] & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & & \left[\frac{\partial Y^{(N)}}{\partial u^{(N)}} - \frac{\partial Y^{(N)}}{\partial z^{(N)}}\left[\frac{\partial Z^{(N)}}{\partial z^{(N)}}\right]^{-1}\frac{\partial Z^{(N)}}{\partial u^{(N)}}\right] \end{bmatrix} \right| \neq 0$$

Features of the New Framework
Independent Time & Spatial Discretizations

- *u*/*y* may reside on a spatial boundary
- Library of spatial elements:
  - Points (0D – rigid bodies / concentrated forces)
  - Lines (1D – beams / forces per unit length)
  - Surfaces (2D – shells / pressure forces)
  - Volumes (3D – solids / body forces)
- Mesh-to-mesh mapping:
  - Relative motion or follower meshes
- Interpolation/extrapolation in time for loose coupling



Mapping Independent Structural & Hydrodynamic Discretizations

Features of the New Framework

Time Marching, OP Determination & Linearization

- Time marching about initial conditions (ICs) (loose & tight)
- Operating-point (OP) determination (tight only):
  - Static equilibrium
  - Steady state
  - Periodic steady state
  - With or without trim of inputs

$$\Delta\dot{x} = A\Delta x + A^{d}\Delta x^{d} + B\Delta u$$

$$\Delta x^{d}\left[n+1\right] = A^{dx}\,\Delta x\big|_{t=n\Delta t} + A^{dd}\Delta x^{d}\left[n\right] + B^{du}\,\Delta u\big|_{t=n\Delta t}$$

$$\Delta y = C\Delta x + C^{d}\Delta x^{d} + D\Delta u$$

- Linearization (tight only):
  - Formed from Jacobians
  - Linearization about given ICs, given time, or OP
  - Useful for modal analysis, controls design, & stability analysis

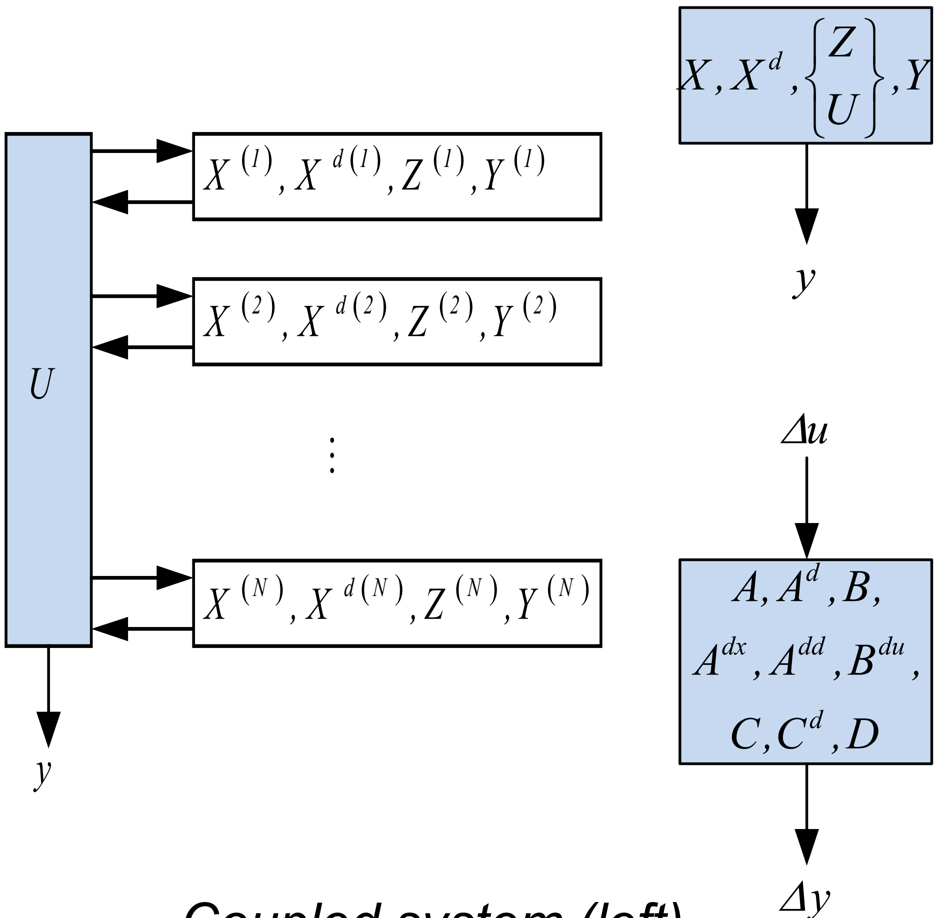
$$A = \left[\frac{\partial X}{\partial x} - \frac{\partial X}{\partial z}\left[\frac{\partial Z}{\partial z}\right]^{-1}\frac{\partial Z}{\partial x}\right]\Bigg|_{op}, etc...$$

- OP & linearization not available in loose coupling due to more general formulation allowed

Features of the New Framework

Module Interface & Coupler (AKA "Glue Code")

- Loose coupling:
  - Interconnects individual modules
  - Derives *u* from *y* (algebraically):
    - Including mesh-to-mesh mapping
    - Including interpolation/extrapolation in time
  - Drives overall solution forward
- Tight coupling (all above, plus):
  - Integrates the coupled system equations using one of its own solvers
  - Drives the OP & linearization calculations
- Modules link through the glue code
- Submodules can be linked with each other if & only if they collectively act as one module

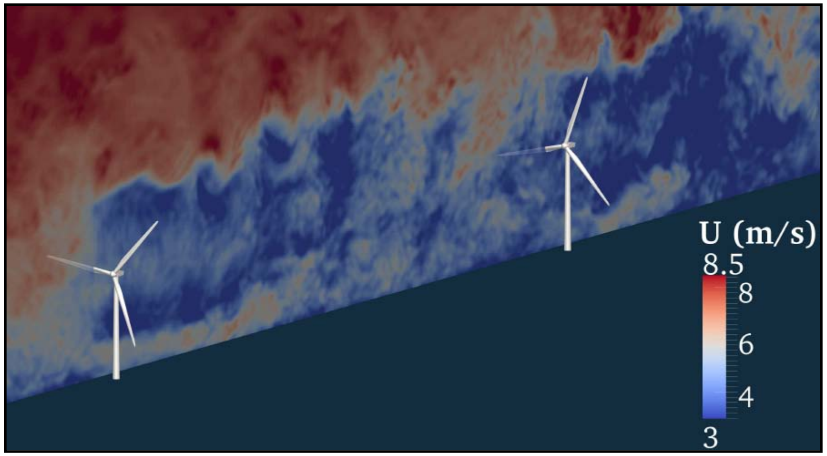


Coupled system (left), effective tightly coupled system (top right), & effective coupled linear system (bottom right)

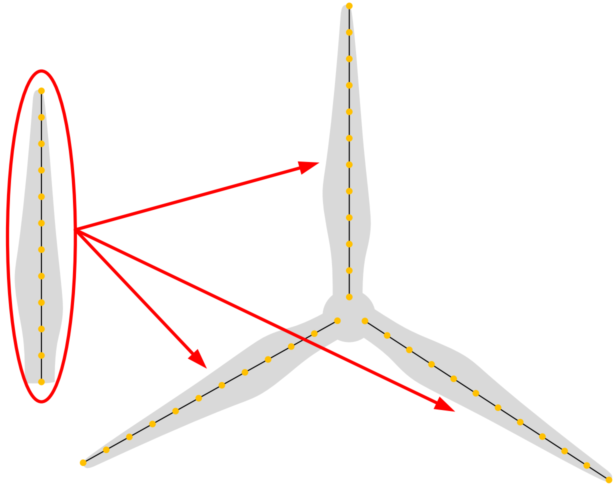
Features of the New Framework

Data Encapsulation & Dynamic Allocation

- Data encapsulation:
  - No global data
  - Data structures for *u*, *y*, *x*, x^d, *z*, & *p* defined in a module
  - Data stored in the driver program & passed to the module through subroutine arguments
- Consequences:
  - No module can access the states or parameters of another module:
    - Clear data transfer
    - *u* can only be derived from *y*
  - Dynamic allocation:
    - Multiple instances of a module can exist simultaneously



Example SOWFA Simulation

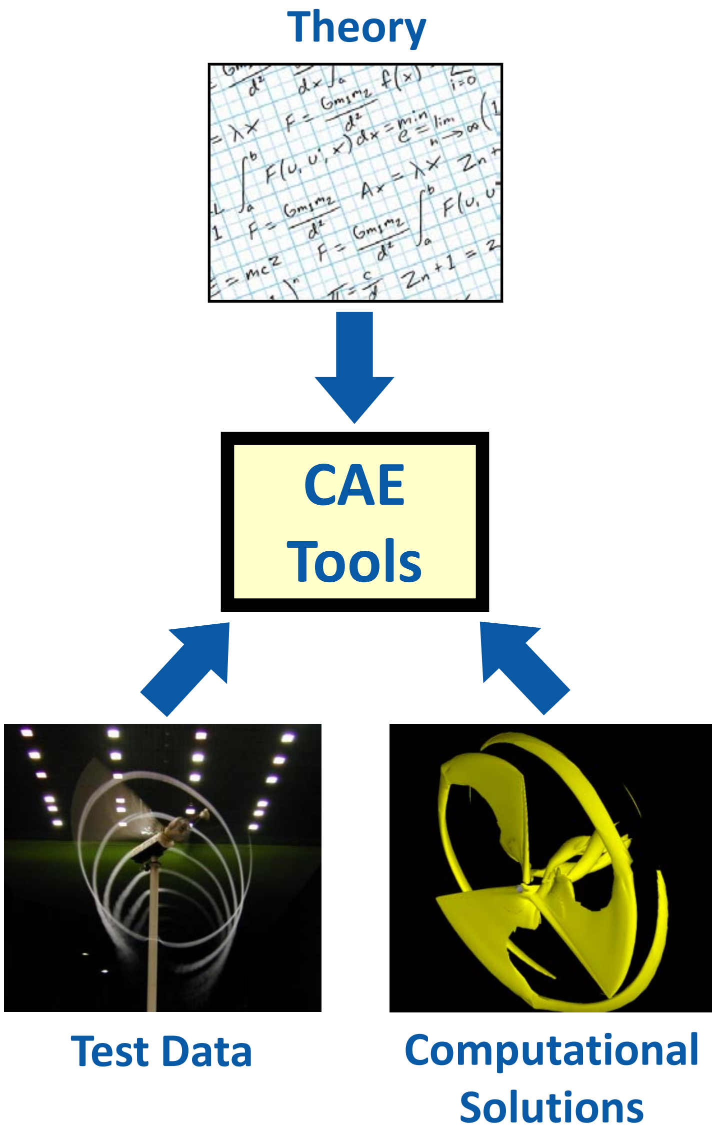


From One Blade to an Entire Rotor

Features of the New Framework

Other Features

- Save/retrieve capability:
  - Save all data, write to a file, retrieve later, & continue simulating from that point
- The new framework also establishes a basis for:
  - Mixed-language programming
  - Multicore processing
  - Co-simulation across a network
  - Hiding the details of individual model components to protect intellectual property (IP)
  - Hardware-in-the-loop (HIL) simulation



Features of the New Framework

Summary

- Module-independent inputs, outputs, states, & parameters
- States in continuous-time, discrete-time, & in constraint form
- Independent time & spatial discretizations
- Loose & tight coupling
- Time marching, operating-point determination, & linearization
- Data encapsulation & dynamic allocation of modules
- Save/restart capability

Features	**Loose**	**Tight**
Module-Independent Variables		
• Inputs	✓	✓
• Outputs	✓	✓
• Parameters	✓	✓
• Continuous states	✓	✓
• Discrete states	✓	✓
• Constraint states	✓	✓
System Formulation		
• Explicit continuous-time ODEs	✓	✓
• Explicit discrete-time updates	✓	✓
• Constraint equations of index 1	✓	✓
• Output equations with direct feedthrough	✓	✓
• Semi-explicit DAEs of index 1	✓	✓
• Systems of any form	✓	
Independent Spatial Discretizations		
• Available	✓	✓
Operating-Point Determination		
• Static equilibrium		✓
• Steady state		✓
• Periodic steady state		✓
• With trim of inputs		✓
Linearization		
• About given initial conditions		✓
• About given time		✓
• About operating point		✓
Time Marching		
• From given initial conditions	✓	✓
• From operating point		✓
• Independent time steps for continuous states between modules	✓	
• Independent time steps for discrete states between modules	✓	✓
Solution		
• Solver implementation is up to the module developer	✓	
• Solver is selectable from those available in the glue		✓
• Overall solvability, numerical stability, and convergence verifiable		✓
Data Encapsulation and No Global Data		
• Required	✓	✓
Dynamic Allocation of Instances of Modules		
• Available	✓	✓
Save/Retrieve Capability		
• Available	✓	✓

Current & Planned Work

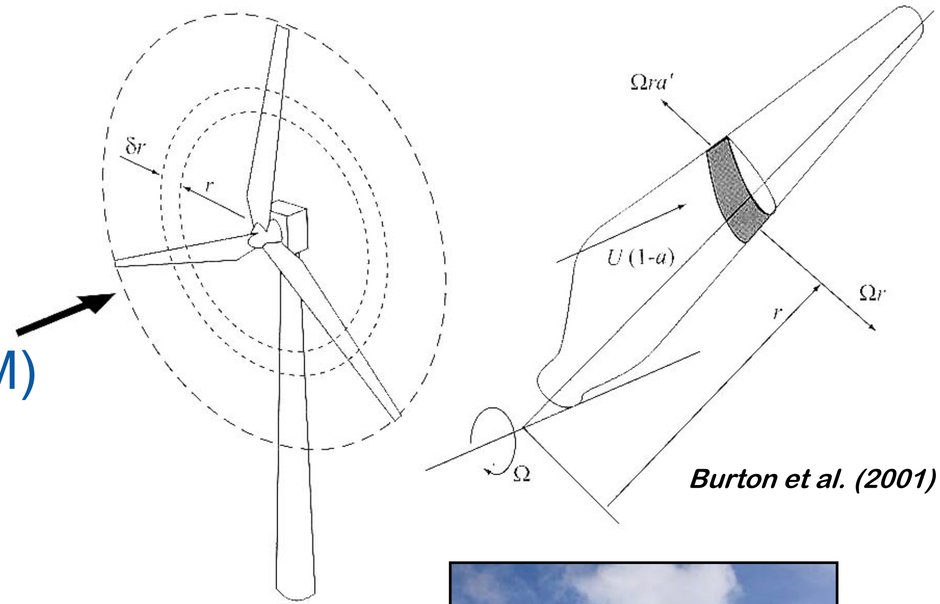
Progress to Date & Timeframe for Completion

- Progress to date:
  - Established key features of the modularization framework
  - Developed a Fortran-based module template
  - Drafted a Programmer's Handbook
- Under development:
  - Finalizing Programmer's Handbook by end of 2012
  - Developing a fully functional mesh library by end of 2012
  - Converting existing modules of FAST to framework by end of 2012
  - Developing a fully functional glue code for loose coupling by end of 2012
  - Assessing loose & tight coupling schemes by end of 2013
  - Developing modules with new functionality (see next slides)
- Future work:
  - Develop glue code for time marching in tight coupling by end of 2013
  - Develop glue code for OP determination & linearization by end of 2014

Current & Planned Work

New Modules Under Development

- Aerodynamics:
  - AeroDyn overhaul
  - FAST-OpenFOAM-WRF coupling (SOWFA) (FOA 1.5)
  - Dynamic wake meandering (DWM) (UMASS) (FOA 1.5)
- Hydrodynamics:
  - Multi-member substructure modeling in HydroDyn
  - 2nd-order hydrodynamics in HydroDyn (FOA 1.1)
  - State-space-based radiation in HydroDyn (FOA 1.1)
  - Nonlinear fluid-impulse theory (MIT) (FOA 1.1)
  - Ice loading (DNV) (FOA 1.2)
  - Ice loading (UMICH) (FOA 1.2)



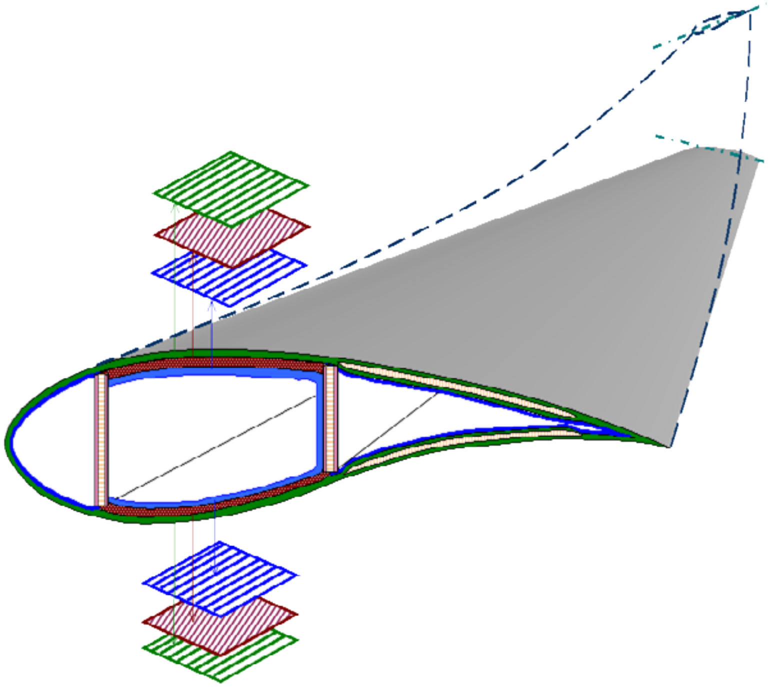
Burton et al. (2001)

Hywind Prototype

Current & Planned Work

New Modules Under Development (Cont.)

- Electrical drive & controls:
  - Generator Types 1-4
  - Gain-scheduled PI controller
  - Wind farm controller (FOA 1.5)
- Structural dynamics:
  - Multi-member support-structure
  - Nonlinear beam FE for blades (GEBT)
  - Improved modal formulation in FAST
  - Mooring Analysis Program (MAP)
  - CHARM3D mooring (FOA 1.3)
- Other?



Blade Twist Induced By Anisotropic Layup

Questions?

Jason Jonkman, Ph.D.
+1 (303) 384 – 7026
jason.jonkman@nrel.gov