

FAST v8 and the transition to OpenFAST

This document describes the transition from FAST v8, a computer-aided engineering tool for simulating the coupled dynamic response of wind turbines, to OpenFAST. OpenFAST was established by researchers at the National Renewable Energy Laboratory (NREL) in 2017, who were supported by the U.S. Department of Energy Wind Energy Technology Office (DOE-WETO).

FAST v8

FAST v8 is a computer-aided engineering tool for simulating the coupled dynamic response of wind turbines. FAST joins aerodynamics models, hydrodynamics models for offshore structures, control and electrical system (servo) dynamics models, and structural (elastic) dynamics models to enable coupled nonlinear aero-hydro-servo-elastic simulation in the time domain. The FAST tool enables the analysis of a range of wind turbine configurations, including two- or three-blade horizontal-axis rotor, pitch or stall regulation, rigid or teetering hub, upwind or downwind rotor, and lattice or tubular tower. The wind turbine can be modeled on land or offshore on fixed-bottom or floating substructures. FAST is based on advanced engineering models derived from fundamental laws, but with appropriate simplifications and assumptions, and supplemented where applicable with computational solutions and test data.

The aerodynamic models use wind-inflow data and solve for the rotor-wake effects and blade-element aerodynamic loads, including dynamic stall. The hydrodynamics models simulate the regular or irregular incident waves and currents and solve for the hydrostatic, radiation, diffraction, and viscous loads on the offshore substructure. The control and electrical system models simulate the controller logic, sensors, and actuators of the blade-pitch, generator-torque, nacelle-yaw, and other control devices, as well as the generator and power-converter components of the electrical drive. The structural-dynamics models apply the control and electrical system reactions, apply the aerodynamic and hydrodynamic loads, adds gravitational loads, and simulate the elasticity of the rotor, drivetrain, and support structure. Coupling between all models is achieved through a modular interface and coupler.

Transition to OpenFAST

The release of OpenFAST represents a transition to better support an open-source developer community across research laboratories, industry, and academia around FAST-based aero-hydro-servo-elastic engineering models of wind-turbines and wind-plants. OpenFAST aims to provide a solid software-engineering framework for FAST development including well documented source code, extensive automated regression and unit testing, and a robust multi-platform and compiler build system.

OpenFAST includes the following organizational changes relative to FAST v8.16:

- A new GitHub organization has been established at <https://github.com/openfast>
- The OpenFAST glue codes, modules, module drivers, and compiling tools are contained within a single repository: <https://github.com/openfast/openfast>
- The FAST program has been renamed OpenFAST (starting from OpenFAST v1.0.0)
- Version numbering has been updated for OpenFAST (starting from OpenFAST v1.0.0), e.g., OpenFAST-v1.0.0-123-gabcd1234-dirty, where:

- v1.0.0 is the major-minor-bugfix numbering system and corresponds to a tagged commit made by NREL on GitHub
- 123-g is the number of additional commits after the most recent tag for a build [the '-g' is for 'git']
- abcd1234 is the first 8 characters of the current commit hash
- dirty denotes that local changes have been made but not committed
- Because all modules are contained in the same repository, the version numbers of each module have been eliminated and now use the OpenFAST version number (starting from OpenFAST v1.0.0) though old documentation may still refer to old version numbers
- The OpenFAST regression test baseline solutions (formerly the Certification Tests or CertTest) reside in a standalone repository: <https://github.com/openfast/r-test> (starting from OpenFAST v1.0.0)
- Unit testing has been introduced at the subroutine level (starting with BeamDyn from OpenFAST v1.0.0).
- An online documentation system has been established to replace existing documentation of FAST v8: <http://openfast.readthedocs.io/>; during the transition to OpenFAST, most user-related documentation is still provided through the NWTTC Information Portal, <https://nwtc.nrel.gov>
- Cross platform compiling is accomplished with CMake on Mac, Linux, and Cygwin (Windows) systems
- Visual Studio Projects (VS-Build) are provided for compiling OpenFAST on Windows (starting from OpenFAST v1.0.0), but the development team is working to automate the generation of Visual Studio build files via CMake in a future release
- **GitHub Issues** has been made the primary platform for developers to report and track bugs, request feature enhancements, and to ask questions related to the source code, compiling, and regression/unit testing; general user-related questions on OpenFAST theory and usage should still be handled through the forum at <https://wind.nrel.gov/forum/wind>
- A new API has been added that provides a high level interface to run OpenFAST through a C++ driver code helping to interface OpenFAST with external programs like CFD solvers written in C++ (starting in OpenFAST v1.0.0)

Release Notes for OpenFAST

This section outlines significant modifications to OpenFAST made with each tagged release.

v0.1.0 (April 2017)

Algorithmically, OpenFAST v0.1.0 is the release most closely related to FAST v8.16.

- Organizational changes:
 - A new GitHub organization has been established at <https://github.com/openfast>
 - The OpenFAST glue codes, modules, module drivers, and compiling tools are contained within a single repository: <https://github.com/openfast/openfast>
 - Cross platform compiling is accomplished with CMake on Mac, Linux, and Cygwin (Windows) systems
 - An online documentation system has been established to replace existing documentation of FAST v8: <http://openfast.readthedocs.io/>
 - **GitHub Issues** has been made the primary platform for developers to report and track bugs, request feature enhancements, and to ask questions related to the source code, compiling, and regression/unit testing; general user-related questions on OpenFAST theory and usage should still be handled through the forum at <https://wind.nrel.gov/forum/wind>

- The AeroDyn v15 aerodynamics module has been significantly updated. The blade-element/momentum theory (BEMT) solution algorithm has been improved as follows:
 - BEMT now functions for the case where the undisturbed velocity along the x-direction of the local blade coordinate system (V_x) is less than zero
 - BEMT no longer aborts when a valid value of the inflow angle (ϕ) cannot be found; in this case, the inflow angle is computed geometrically (without induction)
 - The inflow angle (ϕ) is now initialized on the first call instead of defaulting to using $\phi = 0$, giving better results during simulation start up
 - When hub- and/or tip-loss are enabled (HubLoss = True and/or TipLoss = True), tangential induction (a') is set to 0 instead of -1 at the root and/or tip, respectively (axial induction (a) is still set to 1 at the root and/or tip)
 - The BEMT solution has been made more efficient
 - In addition, several bugs in AeroDyn v15 have been fixed, including:
 - Fixed a bug whereby when hub- and/or tip-loss are enabled (HubLoss = True and/or TipLoss = True) along with the Pitt/Peters skewed-wake correction (SkewMod = 2), BEMT no longer modifies the induction factors at the hub and/or tip, respectively
 - Fixed a bug whereby the time series was affected after the linearization analysis with AeroDyn coupled to OpenFAST when frozen wake is enabled (FrozenWake = True)
- The BeamDyn finite-element blade structural-dynamics model has undergone an extensive cleanup of the source code. A bug in an off-diagonal term in the structural damping-induced stiffness (i.e., representing a change in the damping force with beam displacement) has been corrected.
- A new module for user-specified platform loading (ExtPtfm) has been introduced. ExtPtfm allows the user to specify 6x6 added mass, damping, and stiffness matrices, as well as a 6x1 load vector to define loads to be applied to ElastoDyn's tower base/platform, e.g., to support the modeling of substructures or foundations through a super-element representation (with super-element derived from external software). ExtPtfm also provides the user with a module to customize with more advanced platform applied loads. Module ExtPtfm can be enabled by setting CompSub to 2 in the FAST primary input file (a new option) and setting SubFile to the name of the file containing the platform matrices and load time history, but setting CompSub to 2 requires one to disable hydrodynamics (by setting CompHydro to 0). Please note that the introduction of option 2 for CompSub represents a minor input file change (the only input file change in OpenFAST v0.1.0), but the MATLAB conversion scripts have not yet been updated.
- In the ServoDyn control and electrical-system module, the units and sign of output parameter YawMom have been corrected
- In the InflowWind wind-inflow module, the ability to use TurbSim-generated tower wind data files in Bladed-style format was corrected
- Minor fixes were made to the error checking in ElastoDyn

v1.0.0 (September 2017)

- Organizational changes:
 - The FAST program has been renamed OpenFAST
 - Version numbering has been updated for OpenFAST (see Section 4.3.2 for details)
 - The OpenFAST regression test baseline solutions (formerly the Certification Tests or CertTest) reside in a standalone repository: <https://github.com/openfast/r-test>
 - Unit testing has been introduced at the subroutine level (starting with BeamDyn)

- The online documentation (<http://openfast.readthedocs.io/en/latest/index.html>) has been extensively updated with additions for installation, testing, user (AeroDyn BeamDyn, transition from FAST v8, release notes), and developer guides, etc
- The scripts for compiling OpenFAST using CMake on Mac, Linux, and Cygwin (Windows) systems have been updated, including the ability to compile in single precision and building with Spack
- Visual Studio Projects (VS-Build) are provided for compiling OpenFAST on Windows
- TurbSim has been included in the OpenFAST repository
- The AeroDyn aerodynamics module has been updated:
- Added a cavitation check for marine hydrokinetic (MHK) turbines. This includes the additions of new input parameters CavitCheck, Patm, Pvap, and FluidDepth in the AeroDyn primary input file, the addition of the Cpmin to the airfoil data files (required when CavitCheck = True), and new output channels for the minimum pressure coefficient, critical cavitation, and local cavitation numbers at the blade nodes. Please note that this input file changes represent the only input file change in OpenFAST v1.0.0, but the MATLAB conversion scripts have not yet been updated.
- Fixed a bug in the calculation of wind loads on the tower whereby the tower displacement was used in place of the tower velocity
- Tower strikes detected by the models to calculate the influence of the tower on the wind local to the blade are now treated as fatal errors instead of severe errors
- Fixed minor bugs in the unsteady airfoil aerodynamics model
- The BeamDyn finite-element blade structural-dynamics module has undergone additional changes:
- The source-code has further undergone clean up, including changing the internal coordinate system to match IEC (with the local z axis along the pitch axis)
- Trapezoidal points are now correctly defined by blade stations instead of key points
- The tip rotation outputs were corrected as per GitHub issue #10 (<https://github.com/OpenFAST/openfast/issues/10>)
- The BeamDyn driver has been fixed for cases involving spinning blades
- BeamDyn no longer produces numerical “spikes” in single precision, so, it is no longer necessary to compile OpenFAST in double precision when using BeamDyn
- The ElastoDyn structural-dynamics model was slightly updated:
- The precision on some module-level outputs used as input to the BeamDyn module were increased from single to double to avoid numerical “spikes” when running BeamDyn in single precision
- Minor fixes were made to the error checking
- The ServoDyn control and electrical system module was slightly updated:
- Fixed the values of the generator torque and electrical power sent from ServoDyn to Bladed-style DLL controllers as per GitHub issue # 40 (<https://github.com/OpenFAST/openfast/issues/40>)
- Minor fixes were made to the error checking
- The OpenFAST driver/glue code has been updated:
- Correction steps have been added to the OpenFAST driver during the first few time steps to address initialization problems with BeamDyn (even with NumCrctn = 0)
- Fixed a bug in the Line2-to-Point mapping of loads as per GitHub issue #8 (<https://github.com/OpenFAST/openfast/issues/8>). Previously, the augmented mesh was being formed using an incorrect projection, thus caus-

ing strange transfer of loads in certain cases. This could cause issues in the coupling between ElastoDyn and AeroDyn and/or in the coupling between HydroDyn and SubDyn

- Added an otherwise undocumented feature for writing binary output without compression to support the new regression testing. The new format is available by setting `OutFileFmt` to 0 in the FAST primary input file.
- A new API has been added that provides a high level interface to run OpenFAST through a C++ driver code. The primary purpose of the C++ API is to help interface OpenFAST to external programs like CFD solvers that are typically written in C++.
- The TurbSim wind-inflow turbulence preprocessor was updated:
- The API spectra was corrected
- Several minor bugs were fixed.

OpenFAST: Looking forward

Our goal is to continually improve OpenFAST documentation and to increase the coverage of automated unit and regression testing. In order to increase testing coverage and to maintain robust software, we will require that

- new modules be equipped by the module developer(s) with sufficient module-specific unit and regression testing along with appropriate OpenFAST regression tests;
- bug fixes include appropriate unit tests;
- new features/capabilities include appropriate unit and regression tests. We are in the process of better instrumenting the BeamDyn module with extensive testing as a demonstration of requirements for new modules.

For unit testing, we will employ the pFUnit framework (<https://sourceforge.net/projects/pfunit>).

For the time being OpenFAST provides project and solution files to support users developing and compiling using Visual Studio. However, the team is continually working to automate the generation of Visual Studio build files via CMake in future releases.

Please contact Michael.A.Sprague@NREL.gov with questions regarding the OpenFAST development plan.