Development plan and underlying theory for the *ExtPtfm* module

E. Branlard, J.Jonkman

July 25, 2019

Contents

Introduction

This document presents a development plan that outlines the underlying theory for the *ExtPtfm* module. This module is intended to be used within the *OpenFAST* modular framework [1, 2] to model substructures using a Craig-Bampton-reduction approach. This module uses superelement properties (mass, stiffness, damping and excitation forces) that are provided by the user. These properties are not computed internally by the module. The implementation builds upon a previous work where the substructure was modeled using a Guyan-reduction approach, with a module called *ExtPtfm* (for external platform). This past implementation led to an investigation on the accuracy of the reduction for the modelling of wind turbines with monopile or jackets substructures [3]. The current work also benefit from the fact that *OpenFAST* is coupled with the structural module *SubDyn* [4], which can model a substructure using a Timoshenko finite element beam formulation and perform a Craig-Bampton-reduction. The main difference with the module *ExtPtfm* is that in *SubDyn* the hydrodynamics are computed with an external module and provided at each time step to *SubDyn*. The level of complexity of *ExtPtfm* is reduced since the hydrodynamics are assumed to be accounted for by the user model that performed the Craig-Bampton reduction.

The first section of this document presents the development plan of the module, providing an overview of the method, and outlining the equations to be implemented. The second section provides additional details for the practical FORTRAN implementation. The third section describes the file format used, in particular a new module input file is provided for *ExtPtfm*. The appendix provides additional support for the Guyan implementation, and some elements of theory about reduction techniques.

Notations and abbreviations

CB	Craig-Bampton
DOF	Degrees Of Freedom

1 Background and theory for the implementation of *ExtPtfm*

1.1 Introduction

It is common practice in the offshore wind industry to perform sequentially-coupled load analyses for fixed-bottom wind turbines. The substructure and foundation designers perform a dynamic reduction of the system of equations of the substructure. The reduction method used is typically the Craig-Bampton (CB) method, of which the Guyan method is a particular case. The substructure and foundation designers provide reduced mass, damping, and stiffness matrices together with reduced forces (hydrodynamical and gravitational) at the interface with the tower. The wind turbine designers can then analyze aeroelastic loads on the tower and rotor-nacelle-assembly through an aero-servo-elastic code capable of simultaneously processing the CB information at tower base.

In this fashion, the respective intellectual properties (for the turbine and substructure) do not need to be shared, while still accounting for all the actions on the system (aero-hydro-servo-elastic). The CB reduced set of matrices and loads is an expedient way of reducing the large number of DOFs ($\approx 10^3$) associated with a standard finite-element analysis of a typical multi-member structure, thus leading to an increase in computational efficiency during wind turbine system dynamic simulations. The level of fidelity in the overall system response can be maintained by guaranteeing that the substructure reduced-DOF model retains the fundamental low-frequency response modes of the original finite-element model.

1.2 Overall workflow

The overall workflow is described below:

- A time-domain simulation of the substructure is done with the high-fidelity tool. The time series of loads are stored.
- The high fidelity model of the substructure is reduced using the Craig-Bampton technique described in subsection B.3, where the master degrees of freedom are selected as the ones at the substructure interface node.
- The results from the reduction are written to a file, containing the constant system matrices $\boldsymbol{M}_r$, $\boldsymbol{C}_r$, $\boldsymbol{K}_r$ and the time series of loads $\boldsymbol{f}_r$.
- The file is imported in *OpenFAST* and used in the *ExtPtfm* module to perform a time-domain simulation of the full-turbine where the substructure is represented with the reduced model.
- Within the time simulation, the *ExtPtfm* modules takes as input the displacement, velocity and acceleration of the interface point and returns the loads at this point.
- *OpenFAST* exports the times series of loads and displacements at the interface. These loads and displacements are then used as boundary conditions in another time-domain simulation of the sub-structure with the high-fidelity tool.

The different steps are detailed in the following paragraphs.

1.3 Notations

The background theory of the Craig-Bampton reduction is presented in Appendix B. The notations from subsection B.3 are adapted in this chapter. The label "1" is used for the master

DOFs at the interface point. The label "2" is used for the CB DOFs. The label "0" is introduced for the other DOFs (typically the structure above the interface point).

1.4 Craig-Bampton reduction

The underlying theory behind the Craig-Bampton reduction is described in subsection B.3. It is the responsibility of the substructure designer to provide the reduced mass, damping and stiffness matrices ($\boldsymbol{M}_r$, $\boldsymbol{D}_r$, $\boldsymbol{K}_r$) and the time series of the reduced loads on the master DOFs and the CB DOFs, $\boldsymbol{f}_r(t)$. The definitions of these quantities are given in Equation 29. The output file format from the Craig-Bampton reduction is given in subsection 3.3. This output file from the reduction is the one that will be given as input to *ExtPtfm*. Six degrees of freedom are retained as "master" DOF, corresponding to the three displacements and three rotations of the interface point between the substructure and the rest of the structure (e.g. the tower). In *OpenFAST*, this point is located using the variable `PtfmRefzt` at the coordinate $(0, 0, \texttt{PrfmRefzt})$. The coordinate system used is given in the sketch of Figure 1. The six master

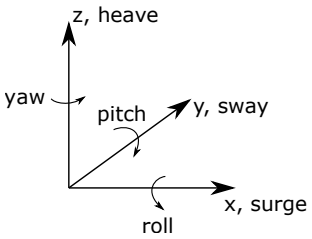

Figure 1: Coordinate systems used for the six degrees of freedom at the interface point

DOFs, stored in the vector $\boldsymbol{x}_1$, are listed below:

- DOF 1 ($\boldsymbol{x}_1(1)$): displacement of interface point along x (surge)
- DOF 2 ($\boldsymbol{x}_1(2)$): displacement of interface point along y (sway)
- DOF 3 ($\boldsymbol{x}_1(3)$): displacement of interface point along z (heave)
- DOF 4 ($\boldsymbol{x}_1(4)$): rotation of interface point around x (roll)
- DOF 5 ($\boldsymbol{x}_1(5)$): rotation of interface point around y (pitch)
- DOF 6 ($\boldsymbol{x}_1(6)$): rotation of interface point around z (yaw)

The Craig-Bampton modes can be selected in different ways, and this selection is left open for further investigation. It is common to select the ones with the lowest frequencies. In the report from DNV-GL [5], a set of high frequency and low frequency modes are retained in the reduction. The number of modes selected is further written n_{CB}.

1.5 *OpenFAST* system of equations using the reduced system

This section describes how the reduced matrices and loads from the substructure are integrated in the equation of motions of *OpenFAST*. Such integration is similar to the integration already done with the Guyan reduction. The main differences lays in the fact that the former requires the resolution of the Craig-Bampton modes as internal states. The equations of motions are presented in this section for the full system using a monolithic approach first and then using a modular approach. The damping terms are omitted to simplify the equations. Their inclusion is straightforward since they are introduced using a matrix similar to the stiffness matrix.

Monolithic approach The monolithic approach consists in solving the full system using a combined set of equations, taking all the degrees of freedom into one space vector. The system of equations is obtained by assembling the different mass and stiffness matrices of the different subsystems. Typically, the system $0-1$ consists of the tower and rotor nacelle assembly, and the system $1-2$ consists of the substructure, where 1 is the top of the transition piece. Using Equation 29, the equation of motions of the full system written in a monolithic form is:

$$\begin{bmatrix} \boldsymbol{M}_{00} & \boldsymbol{M}_{01} & \boldsymbol{0} \\ & \boldsymbol{M}_{11}+\boldsymbol{M}_{r11} & \boldsymbol{M}_{r12} \\ \text{sym} & & \boldsymbol{M}_{r22} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_0 \\ \ddot{\boldsymbol{x}}_1 \\ \ddot{\boldsymbol{x}}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{00} & \boldsymbol{K}_{01} & \boldsymbol{0} \\ & \boldsymbol{K}_{11}+\boldsymbol{K}_{r11} & \boldsymbol{0} \\ \text{sym} & & \boldsymbol{K}_{r22} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_0 \\ \boldsymbol{f}_1+\boldsymbol{f}_{r1} \\ \boldsymbol{f}_{r2} \end{bmatrix} \quad (1)$$

The vectors labelled "1" have dimension 6, and are associated to the degrees of the freedom of the interface point. The vectors labelled "2" have dimension n_{CB} and are associated to the Craig-Bampton modes. Accordingly, the matrices labelled "11" have dimension 6×6, the matrices "12" have dimension $6 \times n_{\text{CB}}$ and the matrices "22" have dimension $n_{\text{CB}} \times n_{\text{CB}}$. The reduced matrices $\boldsymbol{M}_r$ (consisting of the block matrices $\boldsymbol{M}_{r11}$, $\boldsymbol{M}_{r12}$, $\boldsymbol{M}_{r12}^t$ and $\boldsymbol{M}_{r22}$) and $\boldsymbol{K}_r$ are constant and read at initialization from the output file of the CB reduction. The values of $\boldsymbol{f}_r(t)$ are also obtained by interpolation of the discrete values contained in this file.

Modular approach In the modular approach (see [1]), the equations of motion are written for each subsystem. Couplings with other subsystems are introduced using external forces and constraints. In the current case, this is done via the coupling forces at 1 between the two systems, further written $\boldsymbol{f}_C$. The equations of motion for system $0-1$ are:

$$\begin{bmatrix} \boldsymbol{M}_{00} & \boldsymbol{M}_{01} \\ \text{sym} & \boldsymbol{M}_{11} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_0 \\ \ddot{\boldsymbol{x}}_1 \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{00} & \boldsymbol{K}_{01} \\ \text{sym} & \boldsymbol{K}_{11} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_0 \\ \boldsymbol{f}_1 \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{f}_C \end{bmatrix} \quad (2)$$

System $1-2$ receives the opposite forces $\boldsymbol{f}_C$ from system $0-1$ leading to the following set of equations for system $1-2$:

$$\begin{bmatrix} \boldsymbol{M}_{r11} & \boldsymbol{M}_{r12} \\ \text{sym} & \boldsymbol{M}_{r22} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_1 \\ \ddot{\boldsymbol{x}}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}_{r11} & \boldsymbol{C}_{r12} \\ \text{sym} & \boldsymbol{C}_{r22} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{r11} & \boldsymbol{0} \\ \text{sym} & \boldsymbol{K}_{r22} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_{r1} \\ \boldsymbol{f}_{r2} \end{bmatrix} - \begin{bmatrix} \boldsymbol{f}_C \\ \boldsymbol{0} \end{bmatrix} \quad (3)$$

The damping matrix was re-introduced for completeness. The damping matrix may be full if for instance Rayleigh damping is used. The module *ExtPtfm* is responsible for solving Equation 3.

1.6 State space representation of the module *ExtPtfm*

Notations According to the modularization framework adopted in *OpenFAST* [1, 2], the system of equation of the module is restructured into states and outputs equations. In the case of *ExtPtfm*, the equations take the following linear form:

$$\dot{\boldsymbol{x}} = \boldsymbol{X}(\boldsymbol{x}, \boldsymbol{u}, t) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{f}_x, \qquad \boldsymbol{y} = \boldsymbol{Y}(\boldsymbol{x}, \boldsymbol{u}, t) = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} + \boldsymbol{f}_y \quad (4)$$

where $\boldsymbol{x}$ is the state vector, $\boldsymbol{u}$ the input vector, $\boldsymbol{y}$ the output vector. The matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$ and the vectors $\boldsymbol{f}_x$, $\boldsymbol{f}_y$ will be identified below. The inputs are a set of values supplied to *ExtPtfm* that, along with the states, are needed to calculate future states and the system's output. The outputs $\boldsymbol{y}$ are a set of values calculated and returned by the module that depend on the states, inputs, and/or parameters through output equations. The states $\boldsymbol{x}$ are a set of internal values of the module that are influenced by the inputs and used to calculate future state values and the output. Only continuous states are employed, meaning that the states are differentiable in time and characterized by continuous time differential equations. The parameters p are a

set of internal system values that are independent of the states and inputs. The parameters can be fully defined at initialization and characterize the system's state equations and output equations.

The linearization of the state equations given in Equation 4 is straightforward since the equations are in a linear form. The Jacobian of the state and output equation with respect to the states and inputs are:

$$\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{u}} = \boldsymbol{D},\ \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{u}} = \boldsymbol{B}, \qquad \frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}} = \boldsymbol{C},\ \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{x}} = \boldsymbol{A} \tag{5}$$

Application to *ExtPtfm* The main objective of the module *ExtPtfm* is to provide the coupling force at the interface, $\boldsymbol{f}_C$, given the motions of the interface node: $\boldsymbol{x}_1$, $\dot{\boldsymbol{x}}_1$, $\ddot{\boldsymbol{x}}_1$. The coupling force is obtained by resolution of Equation 3. The state of *ExtPtfm* consists of the motions and velocities of the Craig-Bampton modes, $\boldsymbol{x}_2$ and $\dot{\boldsymbol{x}}_2$. The inputs are the motion of the interface node $\boldsymbol{x}_1, \dot{\boldsymbol{x}}_1, \ddot{\boldsymbol{x}}_1$. The output is the coupling force at the interface node $\boldsymbol{f}_C$. The main vectors of Equation 4 are then identified as:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_2 \\ \dot{\boldsymbol{x}}_2 \end{bmatrix}, \qquad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{x}_1 \\ \dot{\boldsymbol{x}}_1 \\ \ddot{\boldsymbol{x}}_1 \end{bmatrix}, \qquad \boldsymbol{y} = \begin{bmatrix} \boldsymbol{f}_C \end{bmatrix} \tag{6}$$

where the vectors have the following dimensions: $\boldsymbol{x}(1 \times 2n_{\text{CB}})$, $\boldsymbol{u}(1 \times 18)$, $\boldsymbol{y}(1 \times 6)$. The second block-row of Equation 3 is developed to isolate $\ddot{\boldsymbol{x}}_2$. Using $\boldsymbol{M}_{r22} = \boldsymbol{I}$ and re-introducing the damping matrix for completeness, gives:

$$\ddot{\boldsymbol{x}}_2 = \boldsymbol{f}_{r2} - \boldsymbol{M}_{r12}^t \ddot{\boldsymbol{x}}_1 - \boldsymbol{K}_{r22}\boldsymbol{x}_2 - \boldsymbol{C}_{r12}^t \dot{\boldsymbol{x}}_1 - \boldsymbol{C}_{r22}\dot{\boldsymbol{x}}_2 \tag{7}$$

The matrices of the state space relation from Equation 4 are then directly identified as:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{K}_{r22} & -\boldsymbol{C}_{r22} \end{bmatrix}, \qquad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{C}_{r12}^t & -\boldsymbol{M}_{r12}^t \end{bmatrix}, \qquad \boldsymbol{f}_x = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{f}_{r2} \end{bmatrix} \tag{8}$$

Isolating $\boldsymbol{f}_C$ from the first block-row of Equation 3, and using the expression of $\ddot{\boldsymbol{x}}_2$ from Equation 7 leads to:

$$\begin{aligned} \boldsymbol{f}_C =& \boldsymbol{f}_{r1} - \boldsymbol{M}_{r11}\ddot{\boldsymbol{x}}_1 - \boldsymbol{C}_{r11}\dot{\boldsymbol{x}}_1 - \boldsymbol{C}_{r12}\dot{\boldsymbol{x}}_2 - \boldsymbol{K}_{r11}\boldsymbol{x}_1 - \boldsymbol{M}_{r12}\ddot{\boldsymbol{x}}_2 && (9) \\ =& \boldsymbol{f}_{r1} - \boldsymbol{M}_{r11}\ddot{\boldsymbol{x}}_1 - \boldsymbol{C}_{r11}\dot{\boldsymbol{x}}_1 - \boldsymbol{C}_{r12}\dot{\boldsymbol{x}}_2 - \boldsymbol{K}_{r11}\boldsymbol{x}_1 \\ & - \boldsymbol{M}_{r12}(\boldsymbol{f}_{r2} - \boldsymbol{M}_{r12}^t\ddot{\boldsymbol{x}}_1 - \boldsymbol{C}_{r12}^t\dot{\boldsymbol{x}}_1 - \boldsymbol{C}_{r22}\dot{\boldsymbol{x}}_2 - \boldsymbol{K}_{r22}\boldsymbol{x}_2) && (10) \end{aligned}$$

The matrices of Equation 4 for the output $\boldsymbol{y}$ are then identified as follows:

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{M}_{r12}\boldsymbol{K}_{r22} & \boldsymbol{M}_{r12}\boldsymbol{C}_{r22} - \boldsymbol{C}_{r12} \end{bmatrix}, \qquad \boldsymbol{f}_y = \begin{bmatrix} \boldsymbol{f}_{r1} - \boldsymbol{M}_{r12}\boldsymbol{f}_{r2} \end{bmatrix} \tag{11}$$

$$\boldsymbol{D} = \begin{bmatrix} -\boldsymbol{K}_{r11} & -\boldsymbol{C}_{r11} + \boldsymbol{M}_{r12}\boldsymbol{C}_{r12}^t & -\boldsymbol{M}_{r11} + \boldsymbol{M}_{r12}\boldsymbol{M}_{r12}^t \end{bmatrix} \tag{12}$$

All the block matrices and vectors labelled with "r" are given from the input file. It is recalled that $\boldsymbol{K}_{r22} = \boldsymbol{N}^2$ is a diagonal matrix containing the square of the eigenvalues originating from the CB reduction (see Equation 26).

Link to *SubDyn*'s formulation The *SubDyn* module also performs a CB reduction, yet, the process has several major differences. First, in *SubDyn*, the hydrodynamic loads at the interface node and the internal nodes ($\boldsymbol{F}_L$ and $\boldsymbol{F}_{HDR}$) are provided by an external module, so these loads have to be projected onto the original nodes of the structure and then reduced onto the CB

modes. These forces are inherently included in $\boldsymbol{f}_r$ here. Second, *SubDyn* accounts for potential rigid connections between the interface node and other nodes of the substructure. This rigid transformation is performed using the matrix $\boldsymbol{T}_I$. Last, some assumptions are done about the damping in *SubDyn*: the terms $\boldsymbol{C}_{r11}$ and $\boldsymbol{C}_{r12}$ are set to $\boldsymbol{0}$ and $\boldsymbol{C}_{r22}$ is assumed diagonal, where each coefficient is effectively taken as the modal damping $2\zeta_i\nu_i$, where ζ_i and ν_i are the i^{th} damping ratio and eigenvalue of the constrained eigenvalue problem given in Equation 26. The link between the notations used in *SubDyn* and the one of *ExtPtfm* are given in Table 1. As mentioned above, the assumptions $\boldsymbol{C}_{r11} \equiv \boldsymbol{0}$, $\boldsymbol{C}_{r12} \equiv \boldsymbol{0}$ and $\boldsymbol{C}_{r22} \equiv 2\boldsymbol{\zeta\Omega}_m$ are done. The

Table 1: Link between current notations and *SubDyn* documentation

Current	$\leftrightarrow$	**SubDyn**
$\boldsymbol{x}_1$		$\boldsymbol{U}_{\mathrm{TP}}$
$\boldsymbol{x}_2$		$\boldsymbol{q}_m$
$\boldsymbol{f}_C$		$-\boldsymbol{F}_{\mathrm{TP}}$
$\boldsymbol{\Phi}_1$		$\boldsymbol{\Phi}_R$
$\boldsymbol{\Phi}_2$		$\boldsymbol{\Phi}_m$
$\boldsymbol{M}_{r11}$		$\tilde{\boldsymbol{M}}_{BB}$
$\boldsymbol{M}_{r12}$		$\tilde{\boldsymbol{M}}_{Bm}$
$\boldsymbol{K}_{r11}$		$\tilde{\boldsymbol{K}}_{BB}$
$\boldsymbol{K}_{r22} = \boldsymbol{N}^2$		$\boldsymbol{\Omega}_m^2$

tilde notation indicates that the matrices were transformed using the rigid body transformation matrix $\boldsymbol{T}_I$. The values of the different matrices used in *SubDyn* can be found in [4].

Time Integration At time $t = 0$, the initial states are all assumed to be zero in *ExtPtfm*. During each subsequent time step, the inputs and states are known values, with the inputs $\boldsymbol{u}(t)$ coming from *ElastoDyn*, and the states $\boldsymbol{x}(t)$ known from the previous time-step integration. At any time step, the values of $\boldsymbol{f}_r(t)$ are obtained by linear interpolation of the discrete values given in the CB reduction file. With known $\boldsymbol{u}(t)$ and $\boldsymbol{x}(t)$, $\dot{\boldsymbol{x}}(t)$ and $\boldsymbol{y}$ can be calculated using the state equations Equation 4. The next time-step states $\boldsymbol{x}(t+\Delta t)$ are obtained by integration:

$$[\boldsymbol{u}(t), \dot{\boldsymbol{x}}(t), \boldsymbol{x}(t)] \xrightarrow{\text{Integrate}} \boldsymbol{x}(t + \Delta t) \tag{13}$$

For loose coupling, *ExtPtfm* uses its own integrator, whereas for tight coupling, the states from all the modules will be integrated simultaneously using an integrator in the glue-code. The following time integrator are implemented for loose coupling:

- Fourth-order explicit Runge-Kutta
- Fourth-order explicit Adams-Bashforth predictor
- Fourth-order explicit Adams-Bashforth-Moulton predictor-corrector
- Implicit second-order Adams-Moulton.

Details on these numerical schemes can be found e.g. in [6].

2 *OpenFAST* implementation of *ExtPtfm*

2.1 Introduction

The baseline module for the Guyan-reduced model is described in Appendix A. This section focuses on describing the changes to be made to this module in order to use the Craig-Bampton reduction input files. The main difference lays in the fact that the state variable x containing the Craig-Bampton modes needs to be introduced, and solved for. Since these implementations are similar, it is intended to use this module both for the Guyan and Craig-Bampton method, without creating a dedicated module.

The output file format from the *OpenFAST* simulation is given in subsection 3.4.

2.2 Description of the individual routines

- `ExtPtfm_Init` This routine reads the new module input file, which specifications are given in subsection 3.1. This input file contains in particular options for the time integration, the number of degrees of freedom to retain, and a list of outputs to be generated for the user and the name of the input file from the Craig-Bampton reduction. This input file is the output file from the Craig-Bampton reduction, which format specifications are given in subsection 3.3. If the file format is "Guyan-only" or if the number of degrees of freedom is 6 then the module will have no Craig-Bampton modes. If a list of active DOF is provided, `ActiveCBDOF`, then the matrices and load time series provided in the input file are truncated to include only the DOF requested. In this routine, the state variable $\boldsymbol{x}$ is initialized to the length of the Craig-Bampton modes, and set to $\mathbf{0}$. The interface point is defined as a mesh point of initial global coordinates $(0, 0,$ `PrfmRefzt`$)$.

- `ExtPtfm_UpdateStates` This function now performs the time integration of the internal states. Its implementation can be directly taken from `SubDyn`. It simply calls one of the numerical scheme routine based on the numerical scheme: `ExtPtfm_RK4`, `ExtPtfm_AB4`, `ExtPtfm_ABM4`, `ExtPtfm_AM2`. All of these functions can be directly copied from the `SubDyn` module, they all depend on the `ExtPtfm_CalcContStateDeriv`, which implementation is described further below.

- `ExtPtfm_CalcOutput` Similar to the existing implementation but it needs to be ensured that the matrix-vector products account for the increased dimension of $n_{\text{CB}} + 6$ instead of 6. The output are the 6 loads at the interface. The loads on the internal degrees of freedom may be stored for internal use in the module. Additional outputs to be written in the output files will be added into the vector `y%WriteOutput`. The name and definitions of these outputs channels are given in subsection 3.4

- `ExtPtfm_CalcContStateDeriv` This function computes $\dot{\boldsymbol{x}}$ using Equation 4. The function can be inspired from `SubDyn`'s implementation.

- `ExtPtfm_UpdateDiscState`: empty, no discrete states

- `ExtPtfm_CalcConstrStateResidual`: empty, no constraints

- `ExtPtfm_JacobianPInput`: When no write outputs are present, this routine simply returns $\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{u}} = \boldsymbol{D}$ (6×18), and $\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{u}} = \boldsymbol{B}$ $(2n_{\text{CB}} \times 18)$. If the module has some n_{out} write outputs, the matrix $\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{u}}$ has then $6 + n_{\text{out}}$ rows. These rows are filled as follows:

- If the write output $i \in 1..n_{\text{out}}$ is the interface load $f_{C,j}$, $j \in [1..6]$, then the row of the Jacobian is the corresponding row of the matrix $\boldsymbol{D}$:

$$\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{u}}[6+i,:] = \boldsymbol{D}[j,:] \tag{14}$$

- Otherwise, the row is zero (the other write outputs are independent of the inputs)

- `ExtPtfm_JacobianPContState`: When no write outputs are present, this routine simply returns $\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}} = \boldsymbol{C}$ $(6 \times 2n_{\text{CB}})$ and $\frac{\partial \dot{\boldsymbol{X}}}{\partial \boldsymbol{x}} = \boldsymbol{A}$ $(2n_{\text{CB}} \times 2n_{\text{CB}})$. If the module has some n_{out} write outputs, the matrix $\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}}$ has then $6 + n_{\text{out}}$ rows. These rows are filled as follows:

  - If the write output is $i \in 1..n_{\text{out}}$ is the interface load $f_{C,j}$, $j \in [1..6]$, then:

$$\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}}[6+i,:] = \boldsymbol{C}[j,:] \tag{15}$$

  - If the write output is $i \in 1..n_{\text{out}}$ the position of mode $j \in 1..n_{\text{CB}}$, then

$$\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}}[6+i,\tilde{j}] = 1 \tag{16}$$

    (The notation $\tilde{j}$ accounts for a potential change of indexing of the modes, see 3.1)

  - If the write output is $i \in 1..n_{\text{out}}$ the velocity of mode $j \in 1..n_{\text{CB}}$, then

$$\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}}[6+i,\tilde{j}+n_{\text{CB}}] = 1 \tag{17}$$

  - Otherwise, the row is zero (the other write outputs are independent of the states)

- `ExtPtfm_JacobianPDiscState`: empty, no discrete states

- `ExtPtfm_JacobianPConstrState`: empty, no constraints

- `ExtPtfm_GetOP`: Returns $\boldsymbol{u}_{\text{op}}$ and $\boldsymbol{x}_{\text{op}}$, which are simply taken from the definitions of $\boldsymbol{x}$ and $\boldsymbol{u}$. Internally the module may store $\boldsymbol{x}$ and $\boldsymbol{u}$ as Fortran types, but the variables $\boldsymbol{x}_{\text{op}}$ and $\boldsymbol{u}_{\text{op}}$ are returned as arrays of double.

2.3 Using the module

The `ExtPtfm` module is activated by setting the flat `CompSub` to 2 in the main OpenFAST input file. The variable `SubFile` in this same file needs to be set to a valid input file for the module (see section 3 for the specifications).

```
---  [...]
  2  CompSub - Compute sub-structural dynamics (switch) {0=None; 1=SubDyn; 2=ExtPtfm}
---  [...]
"Turbine_ExtPtfm.dat" SubFile - Sub-structure input file (SubDyn or ExtPtfm)
```

2.4 Future work

- Perform linearization at the glue-code level

- Perform EVA of Craig-Bampton reduction to show frequencies of substructure and give hint on simulation time

- Static equilibrium. The static equilibrium is given by $\dot{\boldsymbol{x}} = 0$, that is: $\boldsymbol{x} = -\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{u}$. Given the form of $\boldsymbol{B}$, only velocities and acceleration of the interface results in static positions of the CB modes.

3 File format specifications

The file formats supported are described in this section. Unfortunately, these file formats are not generic and require special treatments. A generic file format, and binary file formats will be considered in the future. This file format should also allow the user to turn degrees of freedom on or off.

3.1 *ExtPtfm* module input file

An example of input file for the *ExtPtfm* module is provided below:

```
---------------------- EXTPTFM INPUT FILE -------------------------------------
Comment describing the model
---------------------- SIMULATION CONTROL --------------------------------------
True           Echo          - Write input data to <RootName>.ech (flag)
"default"      DT            - Time step for numerical integration (s) (or "default")
   3           IntMethod     - Integration Method {1:RK4; 2:AB4, 3:ABM4} (switch)
---------------------- REDUCTION INPUTS ----------------------------------------
   1           FileFormat       - File Format {0:GuyanASCII; 1:FlexASCII} (switch)
"Redfile.dat"  Red_FileName     - Path of file containing Guyan/Craig-Bampton inputs (-)
"NA"           RedCst_FileName  - Path of file containing Guyan/Craig-Bampton constant
         ↪inputs (-) (currently unused)
   -1          NActiveCBDOF     - Number of active CB mode listed in ActiveCBDOF, use -1
         ↪for all modes (integer)
  [ , ]        ActiveCBDOF      - List of CB modes indices that are active, [unused if
         ↪NActiveCBDOF<=0]
    0          NInitPosList     - Number of initial positions listed in InitPosList, using
         ↪0 implies all DOF initialized to 0  (integer)
  [ , ]        InitPosList      - List of initial positions for the CB modes  [unused if
         ↪NInitPosList<=0 or EquilStart=True]
    0          NInitVelList     - Number of initial positions listed in InitVelList, using
         ↪0 implies all DOF initialized to 0  (integer)
  [ , ]        InitVelList      - List of initial velocities for the CB modes  [unused if
         ↪NInitVelPosList<=0 or EquilStart=True]

---------------------- OUTPUT --------------------------------------------------
True           SumPrint      - Print summary data to <RootName>.sum (flag)
         1     OutFile       - Switch to determine where output will be placed: {1: in
         ↪module output file only; 2: in glue code output file only; 3: both}
         ↪(currently unused)
True           TabDelim      - Use tab delimiters in text tabular output file? (flag)
         ↪(currently unused)
"ES10.3E2"     OutFmt        - Format used for text tabular output (except time).
         ↪Resulting field should be 10 characters. (quoted string) (currently unused)
          0    TStart        - Time to begin tabular output (s) (currently unused)
               OutList       - The next line(s) contains a list of output parameters (-)
"IntrfFx"      - Platform interface force  - Directed along the x-direction  (N)
"IntrfFy"      - Platform interface force  - Directed along the y-direction  (N)
"IntrfFz"      - Platform interface force  - Directed along the z-direction  (N)
"IntrfMx"      - Platform interface moment - Directed along the x-direction  (Nm)
"IntrfMy"      - Platform interface moment - Directed along the y-direction  (Nm)
"IntrfMz"      - Platform interface moment - Directed along the z-direction  (Nm)
"CBQ_001"      - Modal displacement of internal Craig-Bampton mode number XXX (-)
"CBQD_001"     - Modal velocity     of internal Craig-Bampton mode number XXX (-)
"CBF_001"      - Modal force        on internal Craig-Bampton mode number XXX (-)
"WavElev"      - Wave elevation                                               (m)
END of input file (the word "END" must appear in the first 3 columns of this last
         ↪OutList line)
---------------------------------------------------------------------------------
```

The format is similar to other *OpenFAST* module. Most parameters should be self explanatory to *OpenFAST* users. Of importance to the module are the following parameters:

- `DT`: Time step for numerical integration (s). The user may specify a time step here or use "default" to rely on the glue-code time-step.

- `IntMethod`: numerical method for the time integration. The Runge-Kutta, Adams–Bashforth and Adams–Bashforth-Moutlon methods are available.
- `FileFormat`: file format used for the reduction inputs. Available formats are `GuyanASCII` (see subsection 3.2) and `FlexASCII` (see subsection 3.3)
- `Red_Filename`: path of file containing the Guyan/Craig-Bampton inputs.
- `RedCst_Filename`: path of file containing the Guyan/Craig-Bampton inputs that are constant. This input is not used yet but may be introduced in the future to accommodate for reduction file formats that use two files: one that contains the constants that are structure dependent (static loads from e.g. gravity and matrices), and one that contain the time-varying signals that are simulation dependent (e.g. loads (on top of the constants loads) and wave elevation) !!!Not implemented!!!
- `ActiveCBDOF`: list of size `NActiveCBDOF` containing the CB modes indices that are active. This list is not read if `NActiveCBDOF<=0`. All matrices are reshaped as $\boldsymbol{M}_{\text{new}} = \boldsymbol{M}(I, I)$ where I is the list of indices, potentially unsorted and non contiguous. Setting `NActiveCBDOF=0` is equivalent to a Guyan reduction. Setting `NActiveCBDOF` $= -1$ uses all the CB DOF provided in the input file.
- `InitPosList`: list of size `NInitPosList` containing the initial positions for the CB modes. This list is not read if `NInitPosList<=0`, in which case all the CB DOF positions are initialized to 0.
- `InitVelList`: list of size `NInitVelList` containing the initial positions for the CB modes. This list is not read if `NInitVelList<=0`, in which case all the CB DOF velocities are initialized to 0.
- `OutList`: Specifies the list of outputs that the user requests. These outputs are described in subsection 3.4.

3.2 Former file format for the Guyan reduction (`GuyanASCII`

ASCII file format

Example An example of ASCII file that was used for Guyan-Reduced sub-structure is given below, where numerical values are implied instead of `M11`, `t1`, `Fx1` etc.

```
Comment
#Mass
  M11 ... M16
     [...]
  M61 ... M66
#Damping
  [6 x 6 matrix]
#Stiffness
  [6 x 6 matrix]
# time-varying force
# Time   Fx   Fy   Fz    Mx    My     Mz
# s     (N)  (N)  (N)  (N-m)  (N-m)  (N-m)
  t1    Fx1  Fy1  Fz1   Mx1    My1    Mz1
                 [...]
  tN    FxN  FyN  FzN   MxN    MyN    MzN
```

Specifications Based on this example, it appears that the format is a fixed form format where the line number are assumed. The format specifications are defined below:

- ASCII file
- Line 1 is an arbitrary comment
- Line 2 must contain '`#mass`' (case insensitive). If not, the file format is invalidated. This requirement is important to differentiate between this format and other ASCII formats.
- Lines: 9 and 16 are comment lines that are ignored
- Lines 3-8, 10-15, 17-22 contain six float values, forming the elements of the mass, damping and stiffness matrices respectively. These values corresponds to $\boldsymbol{M}_r$, $\boldsymbol{K}_r$ and $\boldsymbol{D}_r$.
- Lines 23-25 are comment lines and are ignored
- The remaining lines of the files contain 7 float values, corresponding to the values: t, $[F_x(t),\ F_y(t),\ F_z(t),\ M_x(t),\ M_y(t),\ M_z(t)] = \boldsymbol{f}_{r1}(t)$. The number of time steps is here noted N but it is not specified in the file.

This fixed form format is kept for legacy purposes, but should be replaced by a keyword based format. In particular the same file format should be used for Guyan and Craig-Bampton reduced substructures. The following sections define formats that can serve for both purposes.

3.3 File formats for the Craig-Bampton reduction

SES - ASCII file format - `FlexASCII`

The extension SES (for Super Element Structure) is usually used for input files dedicated to `Flex 5`. Unfortunately there is no official `Flex 5` implementation and different manufacturers use different formats.

Example An example of SES ASCII file is given below, where numerical values are implied for: `n`, `dt`, `t`, `M11`, `F1` etc.

```
!Comment
!Comment Flex 5 Format
!Dimension:                          n
!Time increment in simulation:       dt
!Total simulation time in file:      T
!Mass Matrix (Units (kg,m))
!Dimension:                          n
  M11 ... M1n
    [...]
  Mn1 ... Mnn
!Stiffness Matrix (Units (N,m))
!Dimension:                          n
  [n x n matrix]
!Damping Matrix (Units (N,m,kg))
!Dimension:                          n
  [n x n matrix]
!Loading and Wave Elevation (Units (N,m))
!Dimension: 1 time column -  n force columns - 1 wave elevation column
  t1   F11 ... F1n eta1
         [...]
  tN   F1N ... FnN etaN
```

Specifications Based on the above example, the following specifications are provided:

- ASCII file
- Line 1: arbitrary comment that needs to start with an exclamation mark '`!`'
- Line 2: comment which must contain the string '`Flex 5 format`' (case insensitive)
- The following lines are header lines that should start with an exclamation mark.
- The header lines are either comments or lines containing keyword/value pairs
- The following (case insensitive) keywords are currently supported for the header:
  - '`!dimension:`' followed by the integer `n`$= 6 + n_{CB}$
  - '`!time increment in simulation:`': followed by the time step dt
  - '`!total simulation time in file:`': following by the simulation length T
- The remaining lines consists of the following special (case insensitive) keywords:
  - '`!mass matrix`': followed by some text. The next line provide a dimension, but it is ignored. The dimension line is then followed by n lines each containing n float values These values corresponds to $\boldsymbol{M}_r$.
  - '`!stiffness matrix`': similar to the mass matrix, the values corresponds to $\boldsymbol{K}_r$.
  - '`!damping matrix`': similar to the mass matrix, the values corresponds to $\boldsymbol{D}_r$.
  - '`!loading`': followed by some text. The next line contains the dimensions but is ignored. The remaining lines of the file after this keyword should each contain `n`+2 values, corresponding to the time t, the loads $\boldsymbol{f}_r(t)$ and the wave elevation $\eta(t)$. The number of lines N should be consistent with the definition of dt and T from the header.
- For now, the units information and the dimension information after the keywords are ignored. Only the dimension provided in the header is read and should be respected throughout the file. The reason for discarding these information is that at the time of writing, there is no guarantee that this information is always provided, and the format specifications of the units and dimension were not specified.

Additional superelement file formats Additional file formats may be considered in a revised version of this module.

3.4 File format outputted from *OpenFAST*

The time series of loads and displacements at the interface are exported by *OpenFAST* via the generic simulation output file: '.out' or '.outb' file. Additional "write outputs" may be implemented in *ExtPtfm* to export time series of the state-variables. A list of supported output channels for the module is given in Table 2.

Table 2: Output channels for the *ExtPtfm* module

Channel name	**Description**	**Symbol**	**Units**
`IntrfFx`	Platform interface force - Directed along the x-direction	$f_C[1]$	(N)
`IntrfFy`	Platform interface force - Directed along the y-direction	$f_C[2]$	(N)
`IntrfFz`	Platform interface force - Directed along the z-direction	$f_C[3]$	(N)
`IntrfMx`	Platform interface moment - Directed along the x-direction	$f_C[4]$	(Nm)
`IntrfMy`	Platform interface moment - Directed along the y-direction	$f_C[5]$	(Nm)
`IntrfMz`	Platform interface moment - Directed along the z-direction	$f_C[6]$	(Nm)
`InpF_Fx`	Reduced input force at interface point - Directed along the x-direction	$f_{r1}[1]$	(N)
`InpF_Fy`	Reduced input force at interface point - Directed along the y-direction	$f_{r1}[2]$	(N)
`InpF_Fz`	Reduced input force at interface point - Directed along the z-direction	$f_{r1}[3]$	(N)
`InpF_Mx`	Reduced input moment at interface point - Directed along the x-direction	$f_{r1}[4]$	(Nm)
`InpF_My`	Reduced input moment at interface point - Directed along the y-direction	$f_{r1}[5]$	(Nm)
`InpF_Mz`	Reduced input moment at interface point - Directed along the z-direction	$f_{r1}[6]$	(Nm)
`CBQ_XXX`	Displacement of CB DOF number XXX	$\boldsymbol{x}_2[XXX]$	(-)
`CBQD_XXX`	Velocity of CB DOF number XXX	$\dot{\boldsymbol{x}}_2[XXX]$	(-)
`CBQD2_XXX`	Acceleration of CB DOF number XXX	$\ddot{\boldsymbol{x}}_2[XXX]$	(-)
`CBF_XXX`	Reduced input modal force in CB DOF number XXX	$\boldsymbol{f}_{r2}[XXX]$	(-)
`WaveElevExt`	Wave elevation provided in the external file!!!Not implemented!!!	η	(m)

A Module description of the previous Guyan implementation

This section describes the state of the module `ExtPtfm` which was used to model the substructure using a Guyan reduction performed externally. The main module file is `MODULE ExtPtfm_MCKF` in `modules/extpfm/ExtPtfm_MCKF.f90`.
Main variables

- Variables for (6x6) matrices: `p%PtfmAM` (added mass), `p%Damp`, `p%Stiff`
- Variable for loads times series: `p%PtfmFt`, at the time values `p%PtfmFt_t`

Routines:

- `ExtPtfm_Init` initialize p, u, y, but x, xd and z are not set.
  - Inputs u: has translational&rotational motions, velocities and accelerations at the interface point.
  - Outputs y: has Forces and Moments at the interface point.
  - Number of outputs to be written `NumOuts` is 0.
  - `ReadPrimaryFile` reads input file. Sets up some p variables
  - `Init_meshes` initialize mesh. Sets up u and y meshes, one node, same meshes
- `ExtPtfm_UpdateStates`: does nothing
- `ExtPtfm_CalcOutput`
  - Takes the displacements, velocities and acceleration of the interface point from the inputs u, stores in the misc variable m. The motion of the 6 DOF at the interface is noted $\boldsymbol{q}_m$.
  - Computes the platform forces without added mass by interpolation of the Guyan times series (`p%PtfmFt`) (using `InterpStpMat` from `NWTC_NUM`), stores it in `m%PtfmFt`.
  - The damping and restoring forces are added to `m%PtfmFt` based on the damping and stiffness matrix and the positions and velocities of the interface point $\boldsymbol{q}_m$ and $\dot{\boldsymbol{q}}_m$, $-\boldsymbol{K}\boldsymbol{q}_m - \boldsymbol{D}\dot{\boldsymbol{q}}_m$.
  - The contribution from the added mass $-\boldsymbol{M}_a\ddot{\boldsymbol{q}}_m$ is also added to the total loads.
  - The total loads at the interface are returned in y: `y%PtfmMesh%Force` and `y%PtfmMesh%Moment`
- `ExtPtfm_CalcContStateDeriv`: does nothing
- `ExtPtfm_UpdateDiscState`: does nothing
- `ExtPtfm_CalcConstrStateResidual`: does nothing
- `ExtPtfm_JacobianPInput`: does nothing
- `ExtPtfm_JacobianPContState`: does nothing
- `ExtPtfm_JacobianPDiscState`: does nothing
- `ExtPtfm_JacobianPConstrState`: does nothing
- `ExtPtfm_GetOP`: does nothing

B Background on reduction techniques

B.1 Introduction

Reduction, also called condensation, consists in partitioning the DOF in masters and slaves and retaining only the master DOFs in the system of equation.

$$\begin{bmatrix} \boldsymbol{M}_{mm} & \boldsymbol{M}_{ms} \\ \boldsymbol{M}_{ms}^t & \boldsymbol{M}_{ss} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_m \\ \ddot{\boldsymbol{x}}_s \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{mm} & \boldsymbol{K}_{ms} \\ \boldsymbol{K}_{ms}^t & \boldsymbol{K}_{ss} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_m \\ \boldsymbol{x}_s \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_m \\ \boldsymbol{f}_s \end{bmatrix} \tag{18}$$

The associated eigenvalue problem is:

$$\left(\begin{bmatrix} \boldsymbol{K}_{mm} & \boldsymbol{K}_{ms} \\ \boldsymbol{K}_{ms}^t & \boldsymbol{K}_{ss} \end{bmatrix} - \omega^2 \begin{bmatrix} \boldsymbol{M}_{mm} & \boldsymbol{M}_{ms} \\ \boldsymbol{M}_{ms}^t & \boldsymbol{M}_{ss} \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{\phi}_m \\ \boldsymbol{\phi}_s \end{bmatrix} = 0 \tag{19}$$

B.2 Guyan reduction

The assumption of the Guyan reduction is that inertia forces on slave DOFs are negligible in comparison with elastic forces transmitted to slaves by the motion of the master DOFs.

Assumed master/slave relationship In the case where $\boldsymbol{M}_{ms} = \boldsymbol{M}_{ss} = f_s = 0$, then the second row of Equation 18 leads to

$$\boldsymbol{x}_s = -\boldsymbol{K}_{ss}^{-1}\boldsymbol{K}_{sm}\boldsymbol{x}_m, \quad \text{i.e.} \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_m \\ \boldsymbol{x}_s \end{bmatrix} = \boldsymbol{T}\boldsymbol{x}_m \quad \text{with} \quad \boldsymbol{T} = \begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{K}_{ss}^{-1}\boldsymbol{K}_{sm} \end{bmatrix} \tag{20}$$

Guyan's assumption is that the above relation holds for all cases, i.e. where $\boldsymbol{M}_{ms}$ $\boldsymbol{M}_{ss}$ and $\boldsymbol{f}_s$ are non zero.

Reduced equations of motions

$$\boldsymbol{M}_r\ddot{\boldsymbol{x}}_m + \boldsymbol{C}_r\dot{\boldsymbol{x}}_m + \boldsymbol{K}_r\boldsymbol{x}_m = \boldsymbol{f}_r \tag{21}$$

with

$$\boldsymbol{M}_r = \boldsymbol{T}^t\boldsymbol{M}\boldsymbol{T}, \qquad \boldsymbol{C}_r = \boldsymbol{T}^t\boldsymbol{C}\boldsymbol{T}, \qquad \boldsymbol{K}_r = \boldsymbol{T}^t\boldsymbol{K}\boldsymbol{T}, \qquad \boldsymbol{f}_r = \boldsymbol{T}^t\boldsymbol{f} \tag{22}$$

Equation 22 is expanded as follows:

$$\boldsymbol{M}_r = \boldsymbol{M}_{mm} - \boldsymbol{K}_{ms}\boldsymbol{K}_{ss}^{-1}\boldsymbol{M}_{sm} - \boldsymbol{M}_{ms}\boldsymbol{K}_{ss}^{-1}\boldsymbol{K}_{sm} + \boldsymbol{K}_{ms}\boldsymbol{K}_{ss}^{-1}\boldsymbol{M}_{ss}\boldsymbol{K}_{ss}^{-1}\boldsymbol{K}_{sm} \tag{23}$$

$$\boldsymbol{K}_r = \boldsymbol{K}_{mm} - \boldsymbol{K}_{ms}\boldsymbol{K}_{ss}^{-1}\boldsymbol{K}_{sm} \tag{24}$$

$$\boldsymbol{f}_r = \boldsymbol{f}_m - \boldsymbol{K}_{ms}\boldsymbol{K}_{ss}^{-1}\boldsymbol{f}_s \tag{25}$$

B.3 Craig-Bampton reduction

The Craig-Bampton reduction uses a similar static relationship between master and slave coordinates as the Guyan reduction but supplemented by a set of constrained mode shapes. The eigenvalue problem for the constrained system is:

$$\boldsymbol{M}_{ss}\ddot{\boldsymbol{x}}_s + \boldsymbol{K}_{ss}\boldsymbol{x}_s = 0, \qquad \left(\boldsymbol{K}_{ss} - \nu_i^2\boldsymbol{M}_{ss}\right)\boldsymbol{\phi}_i = 0, \qquad |\boldsymbol{K}_{ss} - \nu_i^2\boldsymbol{M}_{ss}| = 0 \tag{26}$$

where ν_i and $\boldsymbol{\phi}_i$ are the i^{th} angular frequency and mode shape respectively. A selection of n_{CB} mode shapes are gathered as column vectors into a matrix noted $\boldsymbol{\Phi}_2$. These mode shapes can be selected as the ones with the lowest frequency or a mix of low and high frequency mode

shapes. The scaling of the modes is such that $\boldsymbol{\Phi}_2^t \boldsymbol{M}_{ss} \boldsymbol{\Phi}_2 = \boldsymbol{I}$ and hence $\boldsymbol{\Phi}_2^t \boldsymbol{K}_{ss} \boldsymbol{\Phi}_2 \triangleq \boldsymbol{N}^2$ where $\boldsymbol{N}^2$ is the diagonal matrix containing the values ν_i^2.

The following change of coordinates is introduced (to be compared with the Guyan relation Equation 20):

$$\boldsymbol{x} = \boldsymbol{T}\boldsymbol{x}_r \qquad \Leftrightarrow \qquad \begin{bmatrix} \boldsymbol{x}_m \\ \boldsymbol{x}_s \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{K}_{ss}^{-1}\boldsymbol{K}_{sm} & \boldsymbol{\Phi}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{r1} \\ \boldsymbol{x}_{r2} \end{bmatrix} \tag{27}$$

with

$$\boldsymbol{x}_r \triangleq \begin{bmatrix} \boldsymbol{x}_{r1} \\ \boldsymbol{x}_{r2} \end{bmatrix}, \qquad \boldsymbol{T} \triangleq \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{\Phi}_1 & \boldsymbol{\Phi}_2 \end{bmatrix}, \qquad \boldsymbol{\Phi}_1 \triangleq -\boldsymbol{K}_{ss}^{-1}\boldsymbol{K}_{sm} \tag{28}$$

where $\boldsymbol{x}_{r2}$ is the vector containing the n_{CB} modal coordinates associated with the modes $\boldsymbol{\Phi}_2$. The equations of motion are adapted to these coordinates by the transformation: $\boldsymbol{M}_r \triangleq \boldsymbol{T}^t\boldsymbol{M}\boldsymbol{T}$, $\boldsymbol{K}_r \triangleq \boldsymbol{T}^t\boldsymbol{K}\boldsymbol{T}$, $\boldsymbol{f}_r \triangleq \boldsymbol{T}^t\boldsymbol{f}$, leading to $\boldsymbol{M}_r\ddot{\boldsymbol{x}}_r + \boldsymbol{K}_r\boldsymbol{x}_r = \boldsymbol{f}_r$, which is written in a developed form below:

$$\begin{bmatrix} \boldsymbol{M}_{r11} & \boldsymbol{M}_{r12} \\ \boldsymbol{M}_{r12}^t & \boldsymbol{M}_{r22} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_{r1} \\ \ddot{\boldsymbol{x}}_{r2} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}_{r11} & \boldsymbol{C}_{r12} \\ \boldsymbol{C}_{r12}^t & \boldsymbol{C}_{r22} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}}_{r1} \\ \dot{\boldsymbol{x}}_{r2} \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{r11} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_{r22} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{r1} \\ \boldsymbol{x}_{r2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_{r1} \\ \boldsymbol{f}_{r2} \end{bmatrix} \tag{29}$$

with

$$\boldsymbol{M}_{r22} = \boldsymbol{\Phi}_2^t\boldsymbol{M}_{ss}\boldsymbol{\Phi}_2 = \boldsymbol{I}, \qquad \boldsymbol{K}_{r22} = \boldsymbol{\Phi}_2^t\boldsymbol{K}_{ss}\boldsymbol{\Phi}_2 \triangleq \boldsymbol{N}^2 \tag{30}$$

$$\boldsymbol{M}_{r11} = \boldsymbol{M}_{mm} + \boldsymbol{\Phi}_1^t\boldsymbol{M}_{sm} + \boldsymbol{M}_{ms}\boldsymbol{\Phi}_1 + \boldsymbol{\Phi}_1^t\boldsymbol{M}_{ss}\boldsymbol{\Phi}_1 \tag{31}$$

$$\boldsymbol{K}_{r11} = \boldsymbol{K}_{mm} + \boldsymbol{K}_{ms}\boldsymbol{\Phi}_1 \tag{32}$$

$$\boldsymbol{M}_{r12} = \left(\boldsymbol{M}_{ms} + \boldsymbol{\Phi}_1^t\boldsymbol{M}_{ss}\right)\boldsymbol{\Phi}_2 \tag{33}$$

$$\boldsymbol{f}_{r2} = \boldsymbol{\Phi}_2^t\boldsymbol{f}_s, \qquad \boldsymbol{f}_{r1} = \boldsymbol{f}_m + \boldsymbol{\Phi}_1^t\boldsymbol{f}_s \tag{34}$$

The values of $\boldsymbol{M}_{r11}$ and $\boldsymbol{M}_{r12}$ are identical to the ones given in Equation 25. The accuracy of the method is improved as the number of selected modes n_{CB} is increased. If a damping matrix $\boldsymbol{C}$ is known, it is similarly reduced as $\boldsymbol{C}_r = \boldsymbol{T}^t\boldsymbol{C}\boldsymbol{T}$.

C Validation

C.1 Validation case of the numerical procedure

Independent Craig-Bampton modes - Forced harmonic vibrations In this test, the matrices corresponding to the interface node ("11"-matrices) and to the couplings with the CB-modes ("12","21"-matrices) are set to zero while the matrices of the CB-modes are set with diagonal coefficients. This allows for the testing of the time integration of the CB-modes. Each mode is set using a given mass and stiffness m, k, providing an angular frequency $\omega_0 = \sqrt{k/m}$, and the damping coefficient is then set to $c = 2m\omega_0\zeta$, with $\zeta = 0.1$. The force applied to each mode is $F = k\sin(\Omega t)$ with $\Omega = 0.95\omega_0$. The time response of a given mode is then compared to the analytical solution for forced harmonic vibrations of a second order system:

$$x = H_0 \sin(\Omega t - \phi) + Ae^{-\zeta\omega_0 t}\sin(\omega_0 t + \psi) \tag{35}$$

with

$$H_0 = \frac{F_0/k}{\sqrt{\left(1-\tilde{\Omega}^2\right)^2 + \left(2\zeta\tilde{\Omega}\right)^2}}, \qquad \phi = \operatorname{atan}\left(\frac{2\zeta\tilde{\Omega}}{1-\tilde{\Omega}^2}\right), \qquad \tilde{\Omega} = \frac{\Omega}{\omega_0} \tag{36}$$

where with the initial conditions $x(0) = \dot{\boldsymbol{x}}(0) = 0$ gives:

$$\psi = \operatorname{atan}\frac{\sqrt{1-\zeta^2}}{\zeta - \tilde{\Omega}\frac{\cos\phi}{\sin\phi}}, \qquad A = H_0\frac{\sin\phi}{\sin\psi} \tag{37}$$

The comparison of the dispalcements of one of the CB modes with the theoretical results is shown in Figure 2.

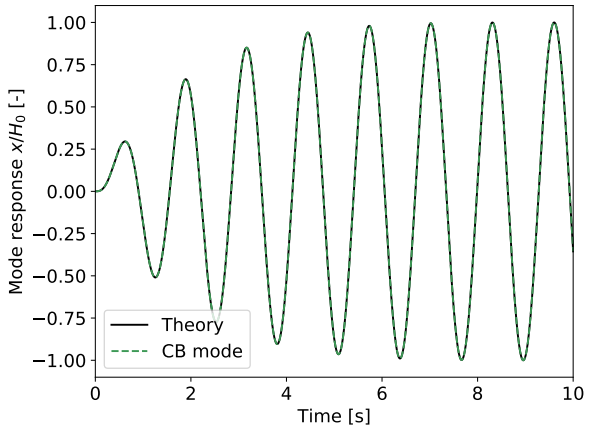

Figure 2: Time integration of an independent CB DOF

C.2 Validation case for a monopile in vacuum

Monopile considered A monopile of uniform properties is considered and it is modelled using a Euler-Bernoulli beam formulation clamped at the bottom. No hydrodynamics effects are considered and the simulations are done as if the monopile was in vacuum. The following

values are used for the beam: $L = 100$m, $D = 8$m, $t = 45$mm, $\rho = 7850$kg/m^3, $E = 210$GN/m^2, leading to $EI = 1.8682 \times 10^{12}$Nm^2 and $m = 8.8282 \times 10^3$kg/m

Analytical results for the Euler-Bernoulli The analytical mode shapes for the uniform Euler-Bernoulli beam are:

$$\phi_n = \cosh(\beta_n x) - \cos(\beta_n x) - \sigma_n \left[\sinh(\beta_n x) - \sin(\beta_n x)\right]; \tag{38}$$

where the β_n are obtained from the solution of

$$\cosh \beta_n \cos \beta_n + 1 = 0 \tag{39}$$

and the σ_n are obtained as:

$$\sigma_n = \frac{\sinh \beta_n - \sin \beta_n}{\cosh \beta_n + \cos \beta_n} \tag{40}$$

The frequencies of the different modes are given by:

$$f_n = \left(\frac{\beta_n}{L}\right)^2 \frac{1}{2\pi} \sqrt{\frac{EI}{\rho A}} \tag{41}$$

The first mode has the frequency $f_1 = 0.814$Hz and a slope at it extremity of 0.7887deg/m when the tip-displacement is normalized to unity.

Generation of the Guyan/Craig-Bampton files A finite element method (FEM) based on 3D beam Euler-Bernoulli beam elements ("3D frame" elements) was used to obtain the mass and stiffness matrix of the full system. The FEM formulation is found e.g. in the book of Amirouche [7, p.531]. The full system matrices were reduced and the input files for *ExtPtfm* generated. The FEM and Craig-Bampton reduction were implemented in Matlab by one of the authors[1]. The damping matrix using a Rayleigh-Damping assumption $\boldsymbol{C} = \alpha \boldsymbol{M} + \beta \boldsymbol{K}$, where the following values were used, $\alpha = 0.00609$, $\beta = 0.00150$, leading to a damping ratio $\zeta = 0.00442$. A low damping was used to help reveal any numerical issues. In this section, the loads are either prescribed at the top of the beam or at the individual CB degrees of freedom, so that the time series of loads can be directly written to the *ExtPtfm* input file without having to perform a reduction of the loads (e.g. applying Equation 34).

Free decay A free-decay test is applied in this section. The structure is initialized with a top displacement of 1m (`PtfmSurge`) and an top-angle of 0.7887deg (`PtfmPitch`) corresponding to the angle expected for the mode shape. No loading is applied to the structure. The results from the Guyan and Craig-Bampton models of the monopile are shown in Figure 3 and compared with the modal response from ElastoDyn (using an equivalent tower model), and the theoretical results from a 1DOF system. Results for 4 and 12 CB modes are shown. Using several CB modes improve the results compared to the simple Guyan reduction. The results would be further improved if the CB modes were initialized to the static equilibrium states corresponding to a unit displacement of the beam top. The response of the structure is consistent with the expected theoretical results, showing that the mass stiffness and damping matrices are properly implemented.

Response to a ramp and step of loading The structure initially at rest is loaded with a ramp and step load, as illustrated with a dot-dashed line in Figure 4. A reference FEM simulation is used for comparison. The response of the CB reduced structure is seen to be closer to the reference simulation than the Guyan reduced structure.

[1] E.Branlard

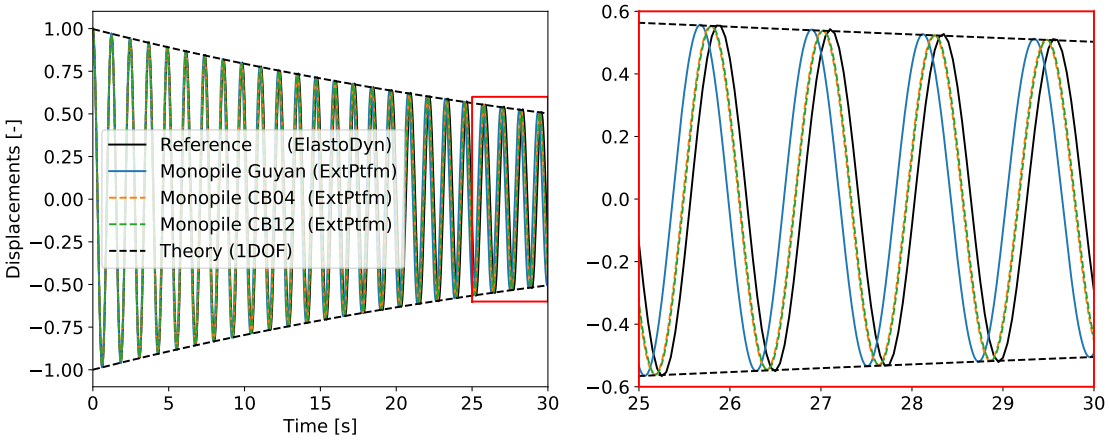

Figure 3: Free decay of a beam using different models

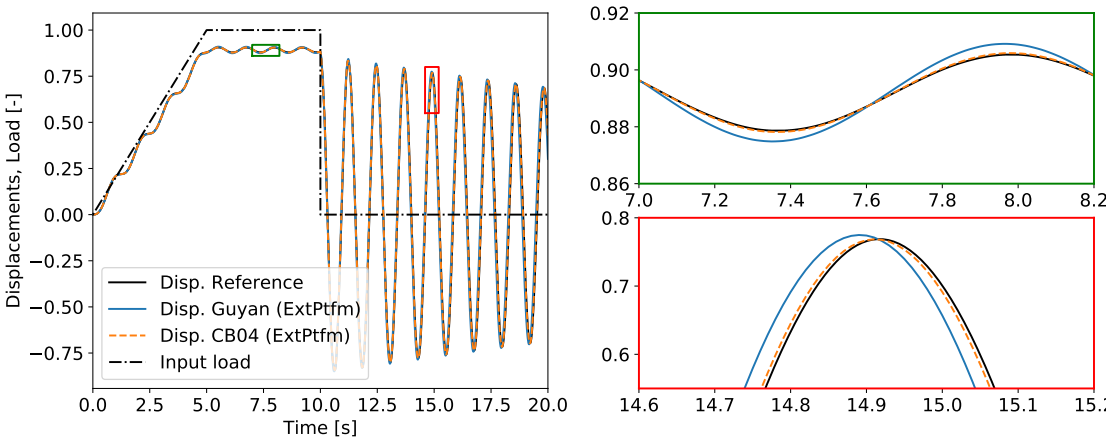

Figure 4: Response to a ramp and step of load

References

[1] J. Jonkman. The new modularization framework for the fast wind turbine cae tool. Technical Report NREL/CP-5000-57228, National Renewable Energy Laboratory, 2013.

[2] Jason M. Jonkman Michael A. Sprague and Bonnie J. Jonkman. Fast modular framework for wind turbine simulation: New algorithms and numerical examples. Technical Report NREL/CP-2C00-63203, National Renewable Energy Laboratory, 2015.

[3] R. Damiani, F. Wendt, W. Musial, Z. Finucane, L. Hulliger, S. Chilka, D. Dolan, J. Cushing, D. O'Connell, and S. Falk. Fully vs. sequentially coupled loads analysis of offshore wind turbines. Technical Report 68368, NREL, 2013.

[4] J. Jonkman R. Damiani and G. Hayman. Subdyn user's guide and theory manual. Technical Report NREL/TP-5000-63062, National Renewable Energy Laboratory, 2015.

[5] DNV GL. Implementing an interface between bladed and sesam - verification report of sesam's bladed interface. Technical Report 2016-0866, Rev. 1, Risø-DTU, 2017.

[6] S.C. Chapra and R.P. Canale. *Numerical Methods for Engineers*. McGraw-Hill Science/Engineering/Mat, 2010.

[7] Farid Amirouche. *Fundamentals of multibody dynamics: Theory and applications*. Birkhauser Boston, 2006.