



SEMESTER PROJECT

LSMS

Boundary approach for solid dynamics in large translation & rotations



Students

Oisín MORRISON - 343223

Leonhard DRIEVER - 341757

Supervisor

Guillaume ANCIAUX

February 23, 2023

Contents

List of Tables	2
List of Figures	2
List of Symbols	3
1 Introduction	4
2 Theory of Modal Analysis	4
2.1 Concepts of Modal Analysis	4
2.2 Adaptation to Step Functions	6
2.3 Numerical Implementation	6
3 Theory of Rigid Body Motion	7
3.1 Need for Considering RBM Separately	8
3.2 Theory Governing RBM	9
3.3 Interfacing between Modal Analysis and RBM	11
3.4 Numerical Implementation	13
3.4.1 Computing the Inertia Tensor	13
3.4.2 Numerical Time Integration	13
3.4.3 The SimpleRB Algorithm	14
3.4.4 The CoupledRB Algorithm	14
4 Methods for Efficient Simulation	15
4.1 Neglecting Coupling	15
4.2 Boundary Approach	17
4.2.1 Computational Savings	17
4.2.2 Energy Computation for Boundary Approach	18
4.3 Truncation of Modes	19
5 Numerical Verification	19
5.1 Mesh Statistics and Material Properties	21
5.2 Convergence of Step Function Approximations	21
5.3 Validation of Mode Truncation Method	22
5.4 Effects of Decoupling	23
5.5 Energy Considerations	25
5.5.1 Step Function Loading on a Single Mode	25
5.5.2 Constant Rotation	25
5.6 Timestep Considerations	26
5.7 Comparison with Akantu and Analytic Solutions	27
5.7.1 Shear Test for Clamped Beam	27
5.7.2 Pure Rotation with Constant Angular Acceleration	28
5.7.3 Combined Rotation, Translation and Modal Analysis	29
6 Case Study	30
6.1 Comparison with Akantu	31
6.2 Computational Savings of Boundary Approach	32
7 Recommendations for Future Work	33
7.1 Strongly Recommended Topics	33

7.2	Further Areas to Investigate	35
8	Conclusion	37
A	Notes on Using the WOBBLE Package	41

List of Tables

1	Mesh statistics for the considered beam mesh	21
2	Summary of the effective masses for a shear force (left) and a complex forcing (right)	22
3	Computational cost of repeated case study runs	32

List of Figures

1	Modes with eigenvalue zero for a beam	8
2	Rotation of a node by an angle $d\theta$, with the x-axis pointing out of the page	9
3	Relation between the G and B frames of reference	10
4	Bi-directional coupling between rotation and vibration	12
5	The beam mesh used for simulations in this work.	20
6	Types of forcing used for verification	20
7	Numerical solution (red) and analytic solution (blue) for the non-resonant case (top) with $\omega_F = 400$ and $\omega = 1096$ and the resonant case (bottom) with $\omega_F = \omega = 1096$	22
8	Convergence analysis of the displacements at the point A with respect to the number of modes.	23
9	Visualisation of $\alpha - \Omega$ decision boundaries	24
10	Behaviour of the energy due to a step-function loading of a single mode of frequency $\omega = 198$	25
11	Comparison of the energy computed with CoupledRB and SimpleRB in the case of pure rotation with a timestep of $\Delta t = 10^{-5}$	26
12	Convergence analysis w.r.t. the time step for pure rotation modelled with SimpleRB.	26
13	Convergence analysis w.r.t. the time step for complex forcing modelled with SimpleRB.	27
14	Predictions of WOBBLE (green) and Euler-Bernoulli beam theory (red).	28
15	Akantu and WOBBLE results for accelerating rotation about the x-axis.	29
16	Akantu and WOBBLE results for complex forcing.	29
17	RMS error as percentage of Akantu RMS displacement for complex forcing .	30
18	Case study forcing indicating a collision for 10 ms	30
19	Case study simulation results	31
20	Computational savings of the boundary approach	33
21	Illustration of method to simulate more complex bodies using spring connections.	34
22	The key decisions and steps in running a WOBBLE simulation	42

List of Symbols

Symbol	Meaning	Unit
α	Angular acceleration of the body	[s^{-2}]
α_i	Mode vibration amplitude	[$-$]
β	Mode vibration phase shift	[$-$]
ρ_{cm}	Coordinate of body cm in the G frame	[m]
ω	Frequencies of eigenmodes	[s^{-1}]
ω^2	Eigenvalue of structure, square of eigenfrequency	[s^{-2}]
Ω	Angular velocity of the body	[s^{-1}]
θ	Angle of rotation of the body	[$-$]
Λ	Eigenvalue matrix $diag\{\omega^2\}$	[s^{-2}]
A	beam corner node, origin of mesh coordinate frame	[$-$]
b	Coordinate in the B frame	[m]
B	Body reference frame, centered at the body cm	[$-$]
C	Damping matrix	[$N s m^{-1}$]
cm	Centre of mass	[m]
E	Total energy	[J]
E_K	Kinetic energy (vibrational or RBM)	[J]
$E_{P_{vib}}$	Vibrational potential energy	[J]
f_{ext}	External force applied to a node	[N]
f_{ext_G}	Summed external force in the G frame	[N]
f_{fict}	Fictitious forces due to rotating reference frame	[N]
\mathcal{F}	Projected force in nodal space	[$-$]
G	Global inertial reference frame	[$-$]
I	Identity matrix	[$-$]
\hat{I}	Moment of Inertia tensor	[$kg m^2$]
K	Stiffness matrix	[$N m^{-1}$]
m_i	Mass associated with node i	[kg]
m_{tot}	Total mass of the body	[kg]
m_j^{eff}	Effective mass of mode j	[$-$]
M	Mass matrix	[kg]
M'	Number of modes	[$-$]
N	Number of nodes	[$-$]
N_b	Number of boundary nodes	[$-$]
N_t	Number of time steps	[$-$]
P	Matrix of eigenvectors of $M^{-1}K$	[$-$]
r	Coordinate in modal space	[$-$]
$R(\theta)$	Rotation matrix	[$-$]
t	Time	[s]
Δt	Timestep	[s]
T	Torque	[$N m$]
u	Displacement of nodes in body frame	[m]
v	Eigenvectors	[$-$]

For all relevant symbols, single and double dots such as \dot{r} and \ddot{r} represent first and second time derivatives respectively.

1 Introduction

High levels of space debris, created by decades of human activity in space, are becoming an increasing threat to astronauts and satellites alike. To tackle this issue, innovative solutions for cleaning space are called for. One company with a promising concept is Clearspace, who propose the use of satellites with claw-like arms that can be used to catch and de-orbit large pieces of debris. Designing such a satellite is highly complex and requires large financial means. To reduce the dependency on expensive testing during the development phase, simulations can be used as a cheap option for fast prototyping and analysis of ideas. However, conventional simulation tools can be very slow and thus delay or even hinder the creative process. Thus, a new simulation tool is called for that can quickly and efficiently simulate the behaviour of structures in large rotation and translation over long periods of time.

This report presents the theory and results of the WOBBLE code package that has been developed by the authors of this report as a means to provide such efficient simulation capabilities. The package is a supplement to the finite element code Akantu [1] and thus builds on the work of the EPFL LSMS¹. WOBBLE is an acronym for *Waves Of Beams and other Bodies due to Loadings and Excitations* and has been chosen because the package is based on a combination of modal analysis and rigid body mechanics.

The first two sections of this report present the theory of modal analysis and rigid body motion, which form the theoretical basis of the simulation code. Next, Section 4 discusses how the implemented models can be used for efficient simulations. Section 5 then presents the verification of the WOBBLE simulation results using analytical models and the existing Akantu code. A comparative case study for a more realistic pulse forcing is presented in Section 6. Finally, recommendations for future work are given in Section 7 and the report is concluded in Section 8. Additional information on the practical usage of the WOBBLE code can be found in Appendix A.

2 Theory of Modal Analysis

As noted, the WOBBLE code package is largely built upon modal analysis. This section provides an explanation of what modal analysis is and how it is used in the context of structural simulations. First the general concepts of modal analysis are discussed, before it is explained how the theory can be simplified through the consideration of step loadings. Finally, efficient numerical implementation and the algorithm used in the WOBBLE package are discussed.

2.1 Concepts of Modal Analysis

The system considered here are structures represented by a mesh with a finite number of nodes. The mass and stiffness properties of the structure are described using finite element methods in the mass and stiffness matrix. Arising from Newton's second law, the governing equation of the system without damping is as shown in Equation 1 (for comments on how to include damping, consult Section 7). The system is based on linear elastic theory, and thus the displacements in Equation 1 are assumed to be sufficiently small.

$$M\ddot{u} + Ku = f_{ext} \quad (1)$$

¹The Computational Solid Mechanics Laboratory at the École Polytechnique Fédérale de Lausanne

Solving this equation directly is possible but expensive, especially for large structures or refined meshes. The high computational cost arises because the simultaneous differential equations described by Equation 1 are coupled. One option to reduce the computational cost is thus to uncouple the equations, which requires first finding the eigenvectors and eigenvalues of the matrix $M^{-1}K$ and constructing the matrix of eigenvectors as shown in Equation 2. Importantly, although finding the eigenvectors can be expensive, the computation must only be done once as all future simulations for the same structure can simply use the previously computed eigenmodes². As shown in Equation 3, the eigenvectors are orthogonal and normalised with respect to an inner product involving the mass matrix, and the inner product involving the stiffness matrix yields the matrix of eigenvalues [2].

$$P = [v_0 \dots v_{N-1}] = \text{eigenvectors}\{M^{-1}K\} \quad (2)$$

$$P^T M P = I \quad \text{and} \quad P^T K P = \text{diag}\{\omega^2\} = \Lambda \quad (3)$$

The key idea of modal analysis is then that the displacement of the nodes can be described as a weighted sum of the different eigenvectors, as shown in Equation 4. The number of eigenvectors equals the number of mesh nodes, so in theory, using all eigenmodes allows for a perfect representation of any displacement using the coordinates r in modal space. However, most of the time the displacements encountered during structural simulations can be sufficiently described as the superposition of only a few main modes, so the sum in Equation 4 can be approximated to a sufficient degree of accuracy using only a few terms. This approximation technique works because structural behaviour is typically dominated by low-frequency modes [3], and it makes the simulation computations significantly cheaper by reducing the number of involved degrees of freedom. Further remarks on which modes are important can be found in subsection 4.3.

$$u = \sum^? v_i r_i = Pr \quad (4)$$

Inserting the expression of Equation 4 into Equation 1 and left-multiplying all terms by P^T then has the desired effect of transforming the coupled equation into the uncoupled equation shown in Equation 5. Because the equation is uncoupled, each row of the matrix equation can be considered individually, as shown in Equation 6.

$$P^T M P \ddot{r} + P^T K P r = P^T f_{ext} \implies \ddot{r} + \Lambda r = \mathcal{F} \quad (5)$$

$$\ddot{r}_i + \omega_i^2 r_i = \mathcal{F}_i \quad (6)$$

Under the assumption that the eigenvalues ω_i^2 are real and positive, Equation 6 has the analytic solution shown in Equation 7. The assumption of real, non-negative eigenvalues is viable as it is guaranteed by the positive-definiteness of M and the positive semi-definiteness of K . That the eigenvalues are also non-zero is guaranteed by setting sufficient boundary conditions or considering the rigid body modes separately (see Section 3 for more information). From Equation 7 it also becomes clear why the notation ω_i^2 is used for denoting the eigenvalues - the eigenfrequencies of the different eigenmodes are simply the square root of the corresponding eigenvalues.

²Thus, finding the eigenmodes can be classified as an *offline computation*

$$r_i(t) = A_i \sin(\omega_i t + \varphi_i) + \frac{1}{\omega_i} \int_0^t \sin(\omega_i(t - \tau)) \mathcal{F}_i(\tau) d\tau \quad (7)$$

Note that in Equation 7, A and φ are determined by the initial conditions and can be found as shown in Equation 8.

$$A = \sqrt{\left(\frac{\dot{r}_{0i}}{\omega_i}\right)^2 + r_{0i}^2} \quad \text{and} \quad \varphi_i = \sin^{-1}\left(\frac{r_{0i}}{A_i}\right) \quad (8)$$

2.2 Adaptation to Step Functions

The issue with the analytic solution found in Equation 7 is that the computation of the convolution integral can be expensive, which goes against the objective of developing a computationally cheap simulation tool. Thus, an approach had to be found to circumvent this issue. The chosen solution is to discretise the forcing using step functions, thereby assuming that the variation in the applied load is sufficiently small such that the loading can locally be approximated as constant. This expression for the forcing is summarised in Equation 9. As the timely spacing between step functions is not necessarily constant, the advantage of this approximation approach is that many step functions can be used to describe times where the loading changes rapidly, and only a few are used for times where changes are gradual.

$$f_{ext} \approx \sum_j h(t - t^{(j)}) \Delta f_{ext}^{(j)} \implies \mathcal{F}_i \approx \sum_j h(t - t_i^{(j)}) \Delta \mathcal{F}_i^{(j)} \quad (9)$$

Substituting Equation 9 into Equation 7 makes it possible to then analytically solve the convolution integral. The solution is shown in Equation 12 and promises a more efficient simulation program than if the convolution integral had to be solved numerically. Importantly, it can be noted that, even though the use of step functions makes the loading discontinuous, the solution for the modal coordinate is continuous and thus suitable for describing the behaviour of a physical system. One of the key aspects that makes this simulation approach so efficient is that, because an analytic solution is computed, it is simply possible to evaluate the solution at the times of interest without the need to compute the solution at intermediate points.

$$r_i(t) = A_i \sin(\omega_i t + \varphi_i) + \frac{1}{\omega_i^2} \sum_j \Delta \mathcal{F}_i^{(j)} h(t - t_i^{(j)}) \left(1 - \cos(\omega_i(t - t_i^{(j)}))\right) \quad (10)$$

where $\Delta \mathcal{F}_i^{(j)} = \mathcal{F}_i^{(j)} - \mathcal{F}_i^{(j-1)}$

2.3 Numerical Implementation

While the solution approach given in Equation 12 is already more computationally efficient than that of Equation 7, it is not yet very efficient. The problem is that each step depends on the entire force discretisation history and thus the computational cost of each step increases with every new force discretisation. To avoid this, a dynamic programming approach is implemented that makes use of the following useful identity:

$$\alpha \sin(\omega t + \beta) = A \sin(\omega t + \phi) + B \cos(\omega t + \theta)$$

where

$$\alpha = \sqrt{C^2 + D^2}$$

$$\beta = \arctan_2(C, \alpha) \in (-\pi, +\pi) \quad (11)$$

with

$$C = A \sin(\phi) + B \cos(\theta)$$

$$D = A \cos(\phi) - B \sin(\theta)$$

Applying this to the analytic solution $r_i(t)$ for the case of step-function loadings (12), we notice that we have the following relations at each force discretisation step:

$$\begin{aligned} r_i(t) &= A_i \sin(\omega_i t + \varphi_i) + \frac{1}{\omega_i^2} \sum_j \Delta \mathcal{F}_i^{(j)} h\left(t - t_i^{(j)}\right) \left(1 - \cos(\omega_i(t - t_i^{(n)}))\right) \\ &= \alpha_i^{(n)} \sin(\omega_i t + \beta_i^{(n)}) + \frac{\mathcal{F}_i^{(n)}}{\omega_i^2} \quad \text{with n s.t. } t^{(n)} \leq t < t^{(n+1)} \\ &= \alpha_i^{(n-1)} \sin(\omega_i t + \beta_i^{(n-1)}) - \frac{\Delta \mathcal{F}_i^{(n-1)}}{\omega_i^2} \cos(\omega_i(t - t_i^{(n-1)})) + \frac{\mathcal{F}_i^{(n)}}{\omega_i^2} \end{aligned} \quad (12)$$

Thus, clearly all values of α and β can be obtained recursively using the identity (11) by identifying the following parameters: $\omega = \omega_i$, $A = \alpha_i^{(n-1)}$, $\phi = \beta_i^{(n-1)}$, $B = -\frac{\Delta \mathcal{F}_i^{(n-1)}}{\omega_i^2}$ and $\theta = \omega_i t_i^{(n-1)}$. Once initial conditions are specified (which are given as $\alpha_i^{(0)} = A_i$ and $\beta_i^{(0)} = \varphi_i$) the problem can then be solved in this manner. This new approach no longer needs to sum over the entire force discretisation at each step, making it run in $O(d)$ as opposed to $O(d^2)$, which would be the case for the inefficient implementation.

In the WOBBLE package, pure modal analysis without consideration of rigid body motion is implemented in the `PureMA` class. With the efficient numerical implementation method identified as explained above, the algorithm for the `PureMA` solver can be summarised as shown below.

1. Compute or load the eigenmodes of the structure
2. Project the initial velocity and displacement into modal space
3. Project the forcing f_{ext} into modal space for each considered step loading
4. Compute the analytic solution for all time intervals between step loadings. Do this by finding the α , β , and offset terms using the efficient adaptive scheme
5. Evaluate the analytic solution at any points in time that are of interest

3 Theory of Rigid Body Motion

In case that the simulated object is not sufficiently fixed in space, it may move and rotate due to the applied loading. This additional movement is accounted for using the theory of rigid body motion (RBM). First, this section explains why RBM is necessary as a supplement to the modal analysis framework developed in Section 2. Next, the theory underlying RBM is presented and the interfacing between modal analysis and RBM is explained. Finally, comments are made on the numerical implementation of the joint modal analysis - RBM system and the implications this has on the performance of the WOBBLE code.

3.1 Need for Considering RBM Separately

RBM refers to the ways in which the studied body can rotate and translate in space without deforming. Thus, it may at first seem fully disjoint from the modal analysis theory discussed in Section 2, which is primarily concerned with how the non-rigid body deforms without moving as a whole. However, to a certain extent, RBM is in fact included in the modal analysis framework. For structures that are insufficiently constrained, finding the eigenmodes will lead to six eigenvectors with eigenvalue zero. An eigenvalue of zero means that the motion is not periodic and that the solution shown in Equation 7 does not apply. Instead, solving the differential equation of Equation 6 will lead to the solution shown in Equation 13.

$$\ddot{r}_i = \mathcal{F}_i \implies r_i(t) = r_{0i} + \dot{r}_{0i}t + \int_0^t \int_0^\tau \mathcal{F}_i(\tau') d\tau' dt \quad (13)$$

Orthogonalising the zero-eigenvectors and rewriting them using an easy-to-understand basis leads to Figure 1, which clearly shows that the space of vectors with eigenvalue zero corresponds to combinations of the six RBM degrees of freedom: translation along and rotation about each one of the three coordinate axes.

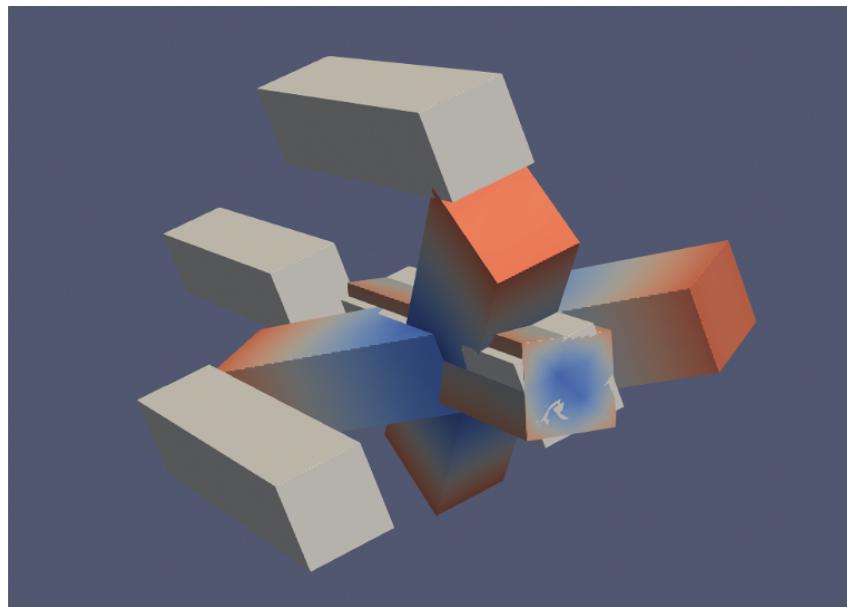


Figure 1: Modes with eigenvalue zero for a beam

Crucially, the RBM modes seen in Figure 1 exist in the tangent space used to derive the theory of modal analysis. Thus, they are conditioned on the assumption that all deformations are sufficiently small such that the system as a whole is governed by linear equations. This is not an issue for the translation modes, as translation is always linear and it is valid to simply superpose the translations along different axes, just like the different deformation modes are superposed in modal analysis. Furthermore, one can easily see that Equation 13 corresponds to the way in which Newton's second law is used in RBM to compute translations over time.

Rotation, on the other hand, is a highly non-linear phenomenon and thus large rotations, as are of interest for the WOBBLE package, cannot be simulated using the rotation eigenmodes of Figure 1 and the solution of Equation 13. Furthermore, for large angles it is no longer valid to simply superpose the rotations about the separate axes. Before examining in subsection 3.2 which non-linear theory applies instead, it is mathematically verified here that the rotation modes

indeed only apply to small rotations. This is done by showing that they can be derived using the small angle approximation.

Where y' and z' are the location of a mesh node relative to the body centre of mass, the (non-normalised) rotation mode about the x axis has the form shown in Equation 14, which is repeated for all nodes in the mesh and concatenated into one eigenvector.

$$v_{rot\ x} = \begin{bmatrix} 0 \\ -z' \\ y' \end{bmatrix} \quad (14)$$

Next, consider the diagram in Figure 2 (with the rotation exaggerated), which is used to describe the displacement due to an x-rotation as shown in Equation 15. Because this equation only simplifies to Equation 14 using the small angle approximation, this confirms that the rotation eigenmodes are not suitable for large rotations.

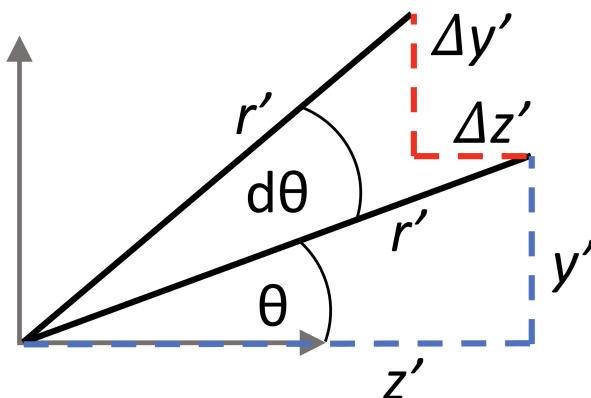


Figure 2: Rotation of a node by an angle $d\theta$, with the x-axis pointing out of the page

$$\begin{aligned} v_{rot\ x} &= \begin{bmatrix} 0 \\ -\Delta y' \\ \Delta z' \end{bmatrix} / \Delta\theta = \begin{bmatrix} 0 \\ r' \cos(\theta + d\theta) - y' \\ r' \sin(\theta + d\theta) - z' \end{bmatrix} / \Delta\theta = \begin{bmatrix} 0 \\ r' (\cos \theta \cos d\theta - \sin \theta \sin d\theta) - y' \\ r' (\sin \theta \cos d\theta + \cos \theta \sin d\theta) - z' \end{bmatrix} / \Delta\theta \\ &\approx \begin{bmatrix} 0 \\ r' (\cos \theta - d\theta \sin \theta) - y' \\ r' (\sin \theta + d\theta \cos \theta) - z' \end{bmatrix} / \Delta\theta = \begin{bmatrix} 0 \\ y' - z' d\theta - y' \\ z' + y' d\theta - z' \end{bmatrix} / \Delta\theta = \begin{bmatrix} 0 \\ -z' \\ y' \end{bmatrix} \end{aligned} \quad (15)$$

3.2 Theory Governing RBM

With the need identified to handle RBM separately, the question remains which theory can be used for the simulation of translation and rotation. To answer this question, it is first necessary to define the additional variables and the coordinate frames that are being considered. As shown in Figure 3, the two relevant coordinate frames are the global inertial frame G , which is fixed in space, and the body frame B , which is attached to the body centre of mass and translates and rotates with the body. The variable describing translation is thus the vector ρ_{cm} , so the position of the B frame in the G frame. Rotation is described by the vector θ , whose direction and magnitude

indicate the axis and angle of rotation respectively. It describes how the B frame is oriented relative to the axes of the G frame.

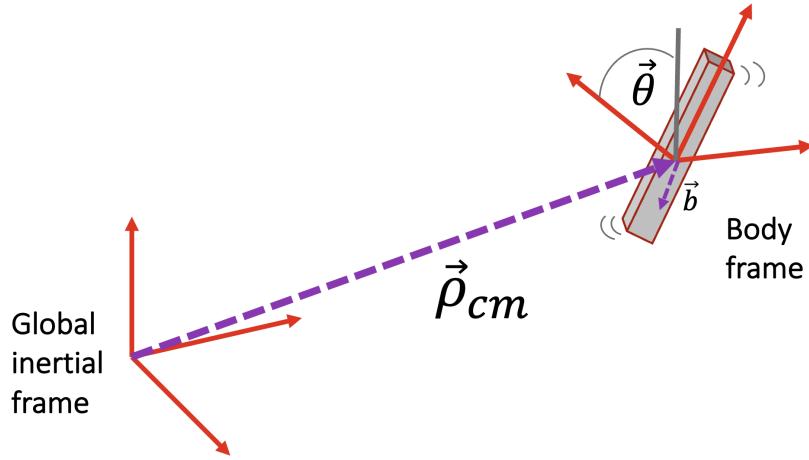


Figure 3: Relation between the G and B frames of reference

The equation for the time evolution of ρ_{cm} is simply the direct application of Newton's second law and is similar to Equation 13. It is shown in Equation 16, both for the case of a loading arbitrarily varying with time, and a loading that is constant over a time increment Δt . Importantly, f_{ext_G} is the same loading as used for the modal analysis in Section 2, with the difference that all loads in the same direction have been summed (collapsing the vector to length three) and the loading has been transformed into the G frame (which uses the later explained rotation matrix shown in Equation 23). It can be observed that, as expected due to linearity, handling translation using the RBM equation of Equation 16 has the same effect as using the zero-eigenvalue translation modes of the body and subsequently projecting the displacement into the G frame. This is the case because summing the forces along each axes and dividing by the mass is equivalent to projecting onto the mass-normalised translation eigenvectors.

$$\rho_{cm}(t) = \rho_{cm_0} + \dot{\rho}_{cm_0}t + \frac{1}{m_{tot}} \int_0^t \int_0^\tau f_{ext_G}(\tau') d\tau' d\tau \quad (16)$$

$$\rho_{cm}(t + \Delta t) = \rho_{cm}(t) + \dot{\rho}_{cm}(t)\Delta t + \frac{(\Delta t)^2 f_{ext_G}}{2m_{tot}}$$

For rotation the situation is more complex. The inertial term is now no longer the total mass of the body, but instead its moments of inertia, which are stored in the inertia tensor. This tensor is of the form as shown in Equation 17, with the terms computed using the expressions in Equation 18 (and the equivalent expressions for the other axes). Note that in Equation 18 the coordinates are in the B frame and thus relative to the center of mass

$$\hat{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \quad (17)$$

$$I_{xx} = \int_{m_{tot}} (y'^2 + z'^2) dm \quad (18)$$

$$I_{xy} = - \int_{m_{tot}} (x'y') dm$$

The obvious factor driving changes in the angular velocity and rotation angle of the body is the applied torque. The torque is computed, as shown in Equation 19, by summing the cross products of the nodal positions in the B frame and the applied forces. Similar to what was seen for the translation modes, this way of computing the torque is equivalent (up to a scale factor that is accounted for elsewhere) to projecting the force into modal space using the zero-eigenvalue rotation modes. This is because the shape of the rotation modes means that the dot product of mode and force results in the same operations as the cross product of nodal position and force.

$$T = \sum_{n=1}^N b_n \times f_{ext_n} \quad (19)$$

Additionally, gyrokinetic effects mean that the angular acceleration of the body is also influenced by its current angular velocity. This leads to the Euler rotation equation, which is shown in its re-arranged form in Equation 20 [4].

$$\alpha = \hat{I}^{-1} T - (\hat{I}^{-1} \times \Omega) \times \Omega \quad (20)$$

The highly non-linear manner in which the angular acceleration depends on the angular velocity means that, unless torque and angular velocity are about the same axis, it is not possible to compute an analytic solution for the rotation of the body over time. This calls for a numerical solution scheme, which is discussed in subsection 3.4.

Two final remarks on RBM theory should be made. Firstly, as no external potentials such as gravity are considered here, RBM alone will only lead to kinetic, not potential energy. This kinetic energy can be computed using Equation 21 and the found velocity and angular velocity vectors $\dot{\rho}_{cm}$ and Ω . Secondly, unlike translations, rotations cannot simply be added. While linear combinations of angular acceleration vectors or angular velocity vectors are possible, the combination of rotation angles requires a non-linear technique.

$$E_{K_{RBM}} = \frac{1}{2} m_{tot} (\dot{\rho}_{cm} \cdot \dot{\rho}_{cm}) + \frac{1}{2} \Omega^T \hat{I} \Omega \quad (21)$$

3.3 Interfacing between Modal Analysis and RBM

In order to combine both modal analysis and RBM into one simulation framework, several things need to be taken into account. As already indicated in Figure 3 and the surrounding discussion, the simulation is divided such that the deformation-vibrations of the body are handled using modal analysis in the B frame, whereas rotation and translation are computed using RBM theory in the G frame. Naturally, because RBM is handled separately, it is necessary to filter out the zero-eigenvalue modes before doing the modal analysis simulation.

Although the two theories are used to model different behaviour, they are not disjoint and it is important to realise that there is a coupling between the rotation and the deformation of the body. This coupling is bi-directional, meaning that the rotation impacts how the body deforms, but the deformation of the body also impacts its rotation. The effects of the coupling are summarised in Figure 4 and discussed below.

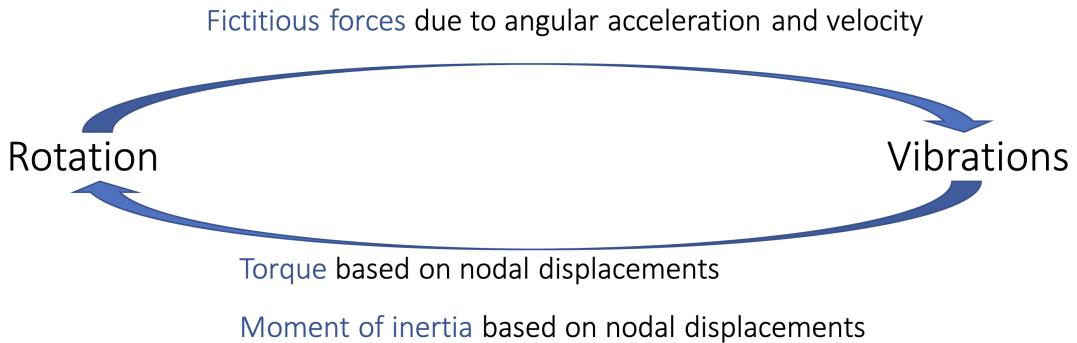


Figure 4: Bi-directional coupling between rotation and vibration

The rotation of the body influences its deformation through so-called fictitious forces, which are a consequence of Newton's laws only being valid in inertial frames of reference. When considering the motion of masses in a rotating frame such as the B frame, the inclusion of the fictitious force shown in Equation 22 allows handling dynamics as if the frame was inertial. Thus, Equation 22 can be used to determine an apparent force which, when projected into modal space, gives the excitement of the vibrational modes due to the rotation of the body. It should be noted that Equation 22 describes the force per mesh node, and that all nodes have to be considered.

$$f_{fict_i} = -m_i\alpha \times b - 2m_i\Omega \times \dot{b}_i - m_i\Omega \times (\Omega \times b_i) \quad (22)$$

In the other direction, the rotation is influenced by the deformation of the body in two ways. Firstly, the moment of inertia of the body is dependent on the spatial mass distribution and thus changes as mesh nodes are displaced in the body frame by vibrations. Secondly, the program is written such that forces are applied on individual nodes, so if a node is displaced, this can affect the magnitude and direction of the computed torque.

With the coupling of RBM and modal analysis identified, the question must be asked of whether these effects are significant enough to be modelled, or small enough to be discarded. To account for both options, the Wobble package contains options for either modelling RBM and body deformations separately, or in a coupled manner. The question of when which option is applicable is addressed in greater detail in subsection 4.1.

Independent of which approach is used, the final issue is how to superpose the displacements in the B frame with the rotation and displacement in the G frame. As the B frame is by definition centred at the body, it is more suitable to describe the joint displacement in the G frame. The transformation of the displacements from the B to the G frame can be done using the rotation matrix defined by Rodrigues' formula and shown in Equation 23 [5]. Note that here $|\theta|$ indicates the magnitude of the rotation and n_i indicate the components of the unit vector giving the axis of rotation. The superposition of the coordinates in the B frame with the coordinates of the B origin in the G frame are then done using Equation 24

$$R(\theta) = \begin{pmatrix} \cos |\theta| + n_1^2(1 - \cos |\theta|) & n_1n_2(1 - \cos |\theta|) - n_3 \sin |\theta| & n_1n_3(1 - \cos |\theta|) + n_2 \sin |\theta| \\ n_1n_2(1 - \cos |\theta|) + n_3 \sin |\theta| & \cos |\theta| + n_2^2(1 - \cos |\theta|) & n_2n_3(1 - \cos |\theta|) - n_1 \sin |\theta| \\ n_1n_3(1 - \cos |\theta|) - n_2 \sin |\theta| & n_2n_3(1 - \cos |\theta|) + n_1 \sin |\theta| & \cos |\theta| + n_3^2(1 - \cos |\theta|) \end{pmatrix} \quad (23)$$

$$\rho_{tot_i} = \rho_{cm} + R(\theta)b_i \quad (24)$$

3.4 Numerical Implementation

The final topic that has to be considered for the simulation of large translations and rotations is how they can be implemented numerically. Here, it is first discussed how the moment of inertia can be computed efficiently for a discretised mesh, before it is explained why numerical time integration is now necessary and how it is implemented. Finally, the algorithms for the coupled and uncoupled combinations of RBM and modal analysis are presented.

3.4.1 Computing the Inertia Tensor

As shown in Equation 18, finding the moments of inertia requires solving an integral over the entire mesh. While this can be done in Akantu, a computationally cheaper approach is to use the lumped mass matrix. The lumped mass matrix is found using the row-wise sum of the mass matrix and assigns each node a specific mass. Thus, using the lumped mass matrix means to model the body as a collection of point masses connected by springs. While less accurate than the full mass matrix, the lumped mass matrix makes it possible to rewrite the integrals of Equation 18 as sums, which are cheaper to compute. This leads to the moment of inertia expressions shown in Equation 25. It should be noted that the lumped mass matrix can similarly be used for computing the body centre of mass in case it is not known.

$$\begin{aligned} I_{xx} &= \sum^N (y_i'^2 + z_i'^2) m_i \\ I_{xy} &= - \sum^N (x_i' y_i') m_i \end{aligned} \tag{25}$$

3.4.2 Numerical Time Integration

One of the key advantages of the modal analysis approach developed in Section 2 for bodies not undergoing RBM is that it uses analytic time integration. This means that the simulation cost solely depends on the variation in the force (so on how many step functions are used), but not on the duration of the simulation or the time between force changes. Unfortunately, as explained in subsection 3.2, finding an analytic solution is not generally possible when accounting for rigid body rotation. Thus, the use of numerical time integration for the joint modal analysis RBM simulation framework is unavoidable. While this increases the cost of simulations, the WOBBLE package maintains the computational advantage that the cost can be significantly reduced by considering only a small number of relevant eigenmodes.

It has been chosen to implement numerical time integration using a simple explicit scheme comparable to the *Forward Euler* method. The accelerations, be it linear, angular, or in modal space, are all found for the current time step using the current parameters. Subsequently, the velocities are updated as shown in Equation 26. For the update of ρ_{cm} , the numerical integration is made more accurate by including not only the velocity, but also the known acceleration. This leads to the update rule shown in Equation 27, which is also used to compute the rotation increment $\Delta\theta$.

$$\begin{aligned} \dot{\rho}_{cm}^{(n+1)} &= \dot{\rho}_{cm}^{(n)} + \Delta t \frac{f_{extG}^{(n)}}{2m_{tot}} \\ \Omega^{(n+1)} &= \Omega^{(n)} + \Delta t \alpha^{(n)} \end{aligned} \tag{26}$$

$$\begin{aligned}\rho_{cm}^{(n+1)} &= \Delta t \dot{\rho}_{cm}^{(n)} + \frac{(\Delta t)^2 f_{extG}^{(n)}}{2m_{tot}} \\ \Delta\theta^{(n)} &= \Delta t \Omega^{(n)} + \frac{(\Delta t)^2}{2} \alpha^{(n)}\end{aligned}\tag{27}$$

Due to the non-linear nature of rotations, $\Delta\theta$ can, however, not simply be added to the current θ vector. Instead, as can be derived using Lie algebras and Rodrigues' formula, the combination is done using the multiplication of rotation matrices shown in Equation 28 [5, 6]. Because it is only the rotation matrix that is used in the other steps of the algorithm, it is most efficient to solely consider $R(\theta)$, not the θ vector itself, during the simulation. Nonetheless, storing $R(\theta)$ for all time steps means that, if desired, the θ vectors can be retrieved using additional computations and the inversion of Equation 23³.

$$R(\theta)^{(n+1)} = R(\Delta\theta^{(n)})R(\theta)^{(n)}\tag{28}$$

3.4.3 The SimpleRB Algorithm

The class `SimpleRB` in the `WOBBLE` package allows simulating both vibrations and RBM, while disregarding the coupling effects discussed in subsection 3.3. While less accurate, `SimpleRB` has the advantage of being computationally more efficient than its coupled counterpart. This is because modal analysis can still be done using the efficient analytic integration method, as it is not dependent on the rigid body motion. Furthermore, each step of the numerical time integration to find the translation and rotation of the body is comparably cheap. The algorithm for `SimpleRB` is summarised below.

1. Solve modal analysis for all time steps
2. Solve RBM for all time steps using the loop:
 - (a) Translate forces into G frame using current θ and Equation 23
 - (b) Update ρ_{cm} using Equation 16
 - (c) Compute α using Equation 20
 - (d) Use α to update Ω and θ
3. Superpose the two solutions using Equation 24

3.4.4 The CoupledRB Algorithm

As counterpart to `SimpleRB`, the `CoupledRB` class handles simulations where the full bi-directional coupling between the body deformation and rotation is accounted for. While this makes the simulation more accurate, it is also significantly more expensive. For one thing, the computational cost increases because the modal response can no longer be computed using analytic integration over long stretches of time. Instead, a new step loading has to now be considered at each time step and the cost increases accordingly. Furthermore, the cost is significantly increased because each simulation step requires two projections between physical and modal space - the fictitious force has to be projected into modal space and the nodal displacements have to be projected into physical space. The algorithm for one iteration of the `CoupledRB` solver loop is shown below.

1. Start with f_{ext} , $R(\theta)$, Ω , b , \dot{b} , ρ_{cm} , $\dot{\rho}_{cm}$ and \hat{I} ,

³This inversion is done using a *logmap* or an iterative scheme and is touched upon in subsection 7.2

2. Sum f_{ext} and translate it into G frame using current $R(\theta)$ and Equation 23
3. Update ρ_{cm} using Equation 16
4. Find the torque T using the current b and f_{ext} and Equation 19
5. Compute α using the T , \hat{I} and Equation 20
6. Use α to update Ω and $R(\theta)$ using Equation 26 to Equation 28
7. Compute f_{fict} using α , Ω , b , \dot{b} and Equation 22
8. Solve modal analysis for $f_{fict} + f_{ext}$ until the time $t^{n+1} = t^n + \Delta t$, which gives updates of b and \dot{b}
9. Update \hat{I} using the new b
10. Loop

4 Methods for Efficient Simulation

The presented theory can be computationally expensive to implement for large meshes since it scales quickly with the number of nodes. As a result, it is necessary to implement more efficient methods to reduce the simulation time. A number of methods are presented and discussed in this section which help reduce this increase in computational cost with increasing number of mesh nodes. First, it is explained under which circumstances coupling of RBM and vibrational motion can be neglected. Next, it is discussed how computational cost can be reduced by adopting a boundary approach. Finally it is explained how the smallest number of modes necessary for adequately describing the structure can be selected.

4.1 Neglecting Coupling

As discussed in Section 3, two options are available in the WOBBLE package for simulating the combination of vibrational displacements and RBM. While the SimpleRB option allows for cheap simulations by ignoring the coupling between rotation and vibrations, it comes at the cost of accuracy. The CoupledRB solver provides the additional level of accuracy but is significantly more expensive and thus slower. The question that users of the WOBBLE package then want answered is how to determine if using CoupledRB is necessary or if using SimpleRB is sufficient. To address this question it was decided to develop an approximate theory that will allow users to select a suitable method dependent on what angular accelerations and velocities they expect to encounter during the simulation.

Considering the bi-directional rotation-vibration coupling as summarised in Figure 4, it is found that out of the three coupling terms the fictitious forces can be expected to have the largest effect. This is because the other terms are affected by the vibrational displacements, which can be expected to be relatively small. Thus, the fictitious force will be used to judge which simulation class should be used. Considering the expression for the fictitious force (repeated in Equation 29) one sees that the Coriolis term can also be neglected as it is dependent on the vibrational velocities \dot{b} of the mesh nodes, which can be expected to be sufficiently small. Furthermore, the terms involving b are approximated using the initial nodal positions in the B frame, without considering the small vibrational displacements.⁴

$$f_{fict} = -m\alpha \times b - 2m\Omega \times \dot{b} - m\Omega \times (\Omega \times b) \quad (29)$$

⁴It is feasible to make small-value approximations in the derivation of this theory as it is intended as a conservative guideline, not a strict rule, of which simulation class to use. Furthermore, the use of this theory means that, in case SimpleRB is used, displacements and velocities truly are sufficiently small to be neglected.

The effect of the fictitious force is most easily judged by its magnitude. As shown in Equation 30, this magnitude depends on the angles θ_1 and θ_2 between the nodal position vectors b and the angular acceleration and angular velocity vectors. To be conservative and not suggest using SimpleRB when CoupledRB should be used, the maximum possible magnitude of the fictitious force is considered. Thus, α and Ω are presumed to be orthogonal to the position vector of the node maximizing $m|b|$.

$$\begin{aligned} |f_{fict}| &= m|\alpha||b| \sin \theta_1 + m|\Omega|^2|b| \sin \theta_2 \\ \Rightarrow \max_{nodes n} (|f_{fict}|) &\leq \max_{nodes n} (m_n|b_n|) (|\alpha| + |\Omega|^2) \end{aligned} \quad (30)$$

One option then to decide which simulation class to use is to set a value f_{lim} for the maximum fictitious force that one accepts to be neglected. For example, if the user knows that they will force vibrations using a 100 Newton loading, they may choose to ignore fictitious forces up to 1 Newton. Given such a limit value f_{lim} , the $\alpha - \Omega$ decision boundary is then given by Equation 31. This means that if the expected maximum α and Ω give a value lower than the one given in Equation 31, SimpleRB can be used. In case it is higher, CoupledRB should be used.

$$|\alpha| + |\Omega|^2 \leq \frac{f_{lim}}{\max_{nodes} (m|b|)} \quad (31)$$

In practice it may alternatively be desired to specify a maximum vibrational displacement d_{lim} that one is willing to neglect. In this case it is necessary to make the additional assumption that the magnitude of the vibrational displacement in modal space can be approximated using the ratio of projected force over eigenvalue. This assumption comes forth from the theory discussed in Section 2 but gives rise to an additional error that one should be aware of when using this displacement-limited method to decide which simulation class to use⁵. Equation 32 shows how the assumption allows translating d_{lim} into an $\alpha - \Omega$ decision boundary. Note that here $m|b|$ represents the concatenated vector of $m_n|b_n|$ across all nodes n , $\frac{1}{\omega^2}$ represents the vector with entries equal to one divided by the ordered eigenvalues, and \odot indicates element-wise multiplication.

$$\begin{aligned} d_{lim} &= \max_{nodes} (Pr) \\ &\approx \max_{nodes} \left(P \left[\frac{\mathcal{F}_1}{\omega_1^2}, \frac{\mathcal{F}_2}{\omega_2^2}, \dots, \frac{\mathcal{F}_M}{\omega_M^2} \right] \right) \\ &= \max_{nodes} \left(P \left(P^T \odot \frac{1}{\omega^2} \right) f_{fict} \right) \\ &\geq \max_{nodes} \left(P \left(P^T \odot \frac{1}{\omega^2} \right) m|b| \right) (|\alpha| + |\Omega|^2) \\ \Rightarrow |\alpha| + |\Omega|^2 &\leq \frac{d_{lim}}{\max_{nodes} \left(P \left(P^T \odot \frac{1}{\omega^2} \right) m|b| \right)} \end{aligned} \quad (32)$$

This alternative formulation for the $\alpha - \Omega$ decision boundary is also useful as it allows setting an upper limit beyond which the Wobble package should not be used at all. This is because, as discussed in Section 2, the vibrational modelling of the package is based on linear

⁵In practice this means to prefer CoupledRB over SimpleRB in case that the expected $\alpha - \Omega$ combination is close to the decision boundary.

elastic theory, so too large displacements will mean using the theory beyond its applicable range. As an arbitrary example, one might say that vibrational displacements above 5 centimetres exceeds what is realistically in the linear elastic regime for a certain structure. Equation 32 then allows finding the range of α and Ω values for which the studied body can be simulated without encountering additional non-linear errors.

Further insights into how the here-developed method can be used in practice can be found in subsection 5.4, where it is applied and tested. It should be noted that the computations necessary for finding the decision boundaries can be computationally expensive. However, as they only have to be performed once per mesh and can then be reused for subsequent simulations, the computations are classified as offline computations and their high cost is not an issue.

4.2 Boundary Approach

In many applications, external forces will only be present on the surface of the mesh, and only the displacements of the nodes on the boundary will be of interest. One such application is, for example, when the WOBBLE package is used to simulate the response of a structure due to an impact with another body. In this section, it is first discussed which computational savings can be made when employing a boundary approach and partly disregarding the inner part of the mesh. Subsequently, it is then discussed how energy calculations can be performed efficiently without considering non-boundary nodes. Already here it shall be stressed that the boundary approach does not come at any cost to the accuracy of the simulation - energy will remain fully conserved and results will be exactly as if the entire system was simulated.

4.2.1 Computational Savings

There are two points at which computational cost can be saved through the use of the boundary approach. The first is during the projection of the applied force f_{ext} into modal space. As discussed in Section 2, this projection is done using the method repeated in the left part of Equation 33. Considering only the nodes on the boundary means that rows of both the eigenvector matrix and the force vector can be discarded. This reduces the size of P^T from $M' \times N$ to $M' \times N_b$ and the size of f_{ext} from N to N_b . Consequently, the cost of the force projection reduces by a factor of order N_b/N .

The second point where computational savings can be made is during the projection of the modal analysis solution into physical space for simulations not done using CoupledRB⁶. The projection is done using the relation shown in the right-hand part of Equation 33. Here, using the boundary approach again means to reduce the number of rows of P from N to N_b . As the projection has to be done for every time step for which the physical displacements are of interest, the overall cost of the operation reduces from $N \times M' \times N_t$ to $N_b \times M' \times N_t$, which is again a reduction factor of N_b/N . Importantly, while no information is lost when using the boundary approach force projection, here the computational savings come at the cost of no longer knowing the displacements of the nodes within the body. Thus, this approach should only be used when this information is not of interest.

$$\mathcal{F} = P^T f_{ext} \quad \text{and} \quad u = Pr \quad (33)$$

While it may be tempting to also apply the boundary approach in other parts of the simulation, this is not possible without inducing errors. Listed below are short explanations of where the

⁶CoupledRB cannot be used with the boundary approach since the fictitious forces which enter the algorithm are *volumetric* and not boundary forces

boundary approach may not be used and why.

Finding the eigenmodes

Here all nodes must be considered as using the boundary approach would have the effect of modelling the object as an empty shell, rather than a three-dimensional volume. However, as finding the modes is an offline computation, it is not a major issue that costs cannot be saved here.

Projection of initial conditions

The conceptual reason why the boundary approach cannot be used here is that knowing the displacement on the boundary is insufficient for describing the displacement throughout the body. Thus, only considering the surface nodes during the projection means to withhold necessary information from the model. This conclusion is confirmed by using a sample initial displacement to test the projection with and without boundary approach. As expected, there is a significant difference in the projected values.

Projection of fictitious forces in CoupledRB

As discussed above, the boundary approach can be used during the projection of f_{ext} into modal space. However, it may not be used for the projection of f_{fict} , as unlike f_{ext} it is a body force and thus also acts on the internal mesh nodes.

Projection into physical space in CoupledRB

In the other solvers of the WOBBLE package, the vibrational displacement and velocity are only projected back into physical space for the purpose of having the solution in physical space. Thus, if only the solution on the surface is of interest, the boundary approach can be used without problems. In the CoupledRB solver, however, the physical vibrational displacements and velocities are necessary for the time evolution of the system. As they need to be known for all mesh nodes, including the internal ones, the boundary approach cannot be used here.

4.2.2 Energy Computation for Boundary Approach

When the boundary approach is used to project displacement and velocity into physical space in the PureMA and SimpleRB solvers, it initially appeared that there was a problem of no longer being able to compute the vibrational energy. This is because the lack of information about the displacement and velocity of the internal nodes means that only the energies associated with the boundary nodes can be found directly in physical space. To avoid this problem, a method has been developed to compute the energy directly in modal space. The derivations are shown in Equation 34 and Equation 35 for kinetic and potential energy respectively.

$$E_{K_{vib}} = \frac{1}{2} \dot{u}^T M \dot{u} = \frac{1}{2} (Pr)^T M (Pr) = \frac{1}{2} \dot{r}^T P^T M P \dot{r} = \frac{1}{2} \dot{r}^T \dot{r} \quad (34)$$

$$E_{P_{vib}} = \frac{1}{2} u^T K u = \frac{1}{2} (Pr)^T K (Pr) = \frac{1}{2} r^T P^T M P r = \frac{1}{2} r^T \Lambda r \quad (35)$$

As follows from Equation 34 and Equation 35, finding the vibrational energy is always cheaper in modal than in physical space, even when all information is available in physical space. Thus, the modal energy methods have been adopted for all solvers, independent of whether or not the boundary approach is used.

4.3 Truncation of Modes

The total number of modes for large meshes can be extremely large, so it is generally unfeasible to simulate using all modes. Furthermore, the response of structures is typically dominated by low-frequency modes, so in most cases high frequency modes can be neglected [3, 7]. As a result, it is possible to simply truncate the number of considered modes. In this work, a commonly used method called the mode displacement method is implemented [8]. This involves keeping only m modes in the modal analysis, and entirely disregarding the others. The method remains otherwise unchanged.

A key consideration when using the mode displacement method is to determine how many modes one should use. Firstly, it is clear that higher frequency modes are generally of less importance than lower frequency modes due to the presence of a $\frac{1}{\omega_i^2}$ term in Equation 12. As a result, it often makes sense to take the first m modes with the lowest frequencies. A more commonly used standard in industry is based off of the percentage of the effective modal mass retained by the truncated modes relative to that of the full system [9, 10]. A modified version of this approach is considered here by defining an effective mass as follows:

$$m_j^{eff} = \frac{|P^T M f_{ext}|}{\omega_j^2} \quad (36)$$

The term $|P^T M f_{ext}|$ computes the inner product of f_{ext} with each mode (recall that $P^T M P = I$), and the factor $\frac{1}{\omega_j^2}$ is introduced to remove the effect of differing eigenvalues affecting the solution (as per Equation 12). Furthermore, one ignores the rigid body modes of P since these are always considered when applicable. Thus, to compute the required number of modes to retain, one can simply check that the ratio of m_j^{eff} with the truncated modes to that with all modes is at least 90%⁷. It is worth noting that, because the ratio is computed relative to the system with all modes, determining the number of necessary modes to keep requires computing Equation 36 for all modes of the structure. Even though this makes the computation expensive, this is not an issue as the investigation of the effective mass only needs to be conducted once for each mesh and forcing and can thus be classified as an offline computation⁸. Once a suitable number of modes is found, this method can then simply be used for all future simulations using the same set-up.

An example of how the effective mass method is used to select a suitable number of modes and an investigation into the effects of disregarding high-frequency modes can be found in subsection 5.3.

5 Numerical Verification

In order to validate our theoretical approach, a large number of simulations and validations have been conducted. In this section, this body of work is presented - including validations of approximations and simplifications used, and comparison of WOBBLE simulation results with both analytic theory and numerical results from Akantu. The chosen mesh is shown in Figure 5 and the three key types of forcing that will be used in the verification tests are illustrated in Figure 6.

⁷90% was chosen since this is the same cutoff required for the effective modal masses as per the EN 1998-1 standard [11]. We adopt the same cut-off for our modified approach.

⁸Additionally, one may choose to simply retain the same number of modes when the forcing only changes slightly

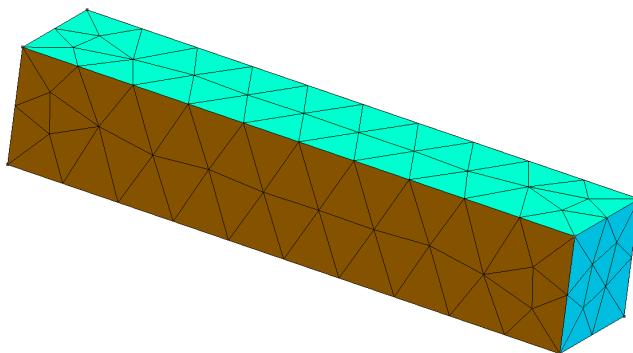
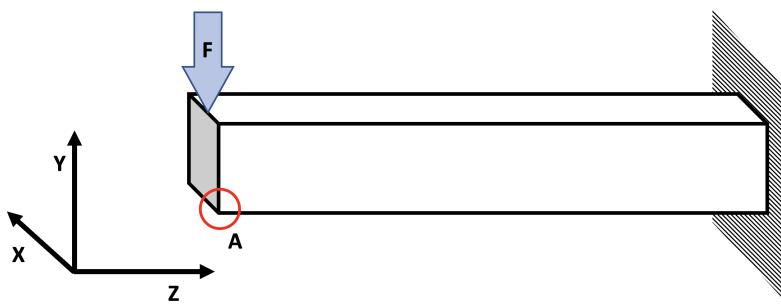
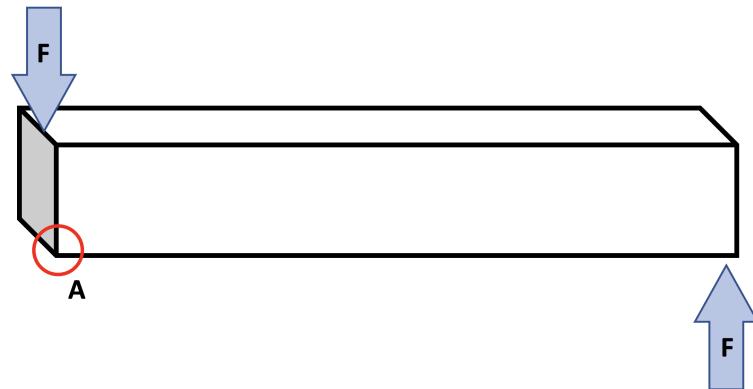


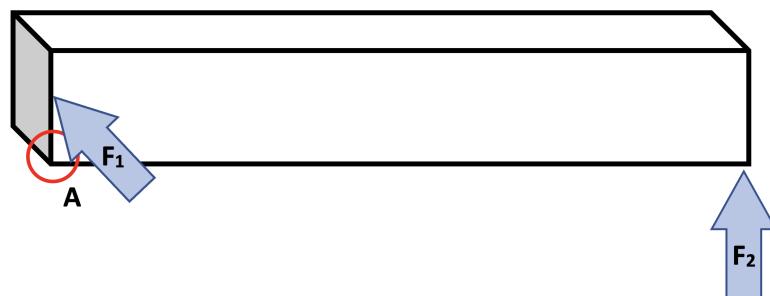
Figure 5: The beam mesh used for simulations in this work.



(a) Shear test with the beam fixed at one end. No rigid body motion is at play.



(b) Constant angular acceleration test. Rigid body motion is at play here.



(c) Complex forcing test. Rigid body motion is at play here. At first $F_1 = -100$, $F_2 = 100$, then $F_1 = 500$, $F_2 = 300$.

Figure 6: Types of forcing used for verification

In Figure 6 the origin of the mesh coordinate system is positioned at the point labelled A and oriented as shown in the shear test loading. The corner mesh node at A will repeatedly be used as a point of reference during the verification tests.

5.1 Mesh Statistics and Material Properties

In order to compare simulations, a single mesh was decided upon for this work, and it is shown in Figure 5. Considering the application of clearing space debris, a mesh of a beam was chosen since this is close to the shape of the claw-like arms that may be used. Meshing was done using Gmsh [12]. The parameters and statistics for the mesh are summarised in Table 1. Since the order of the mesh can change the modes and thus the behaviour of the system, it was deemed important to use a second order mesh for this work. This is because bending, which is a key component of many modes, is a second order phenomenon and can thus be better modelled using a second order mesh.

Table 1: Mesh statistics for the considered beam mesh

Nodes	Surface Nodes	Elements	Element Type	Dimensions	Order
579	426	254	Tetrahedra	1x1x5	2

The material properties were chosen to be linear elastic and similar to those of steel, with a density of $8000 \text{ [kg m}^{-3}\text{]}$ and a Young's modulus of $200 \times 10^9 \text{ [Pa]}$. The Poisson's ratio was set to zero.

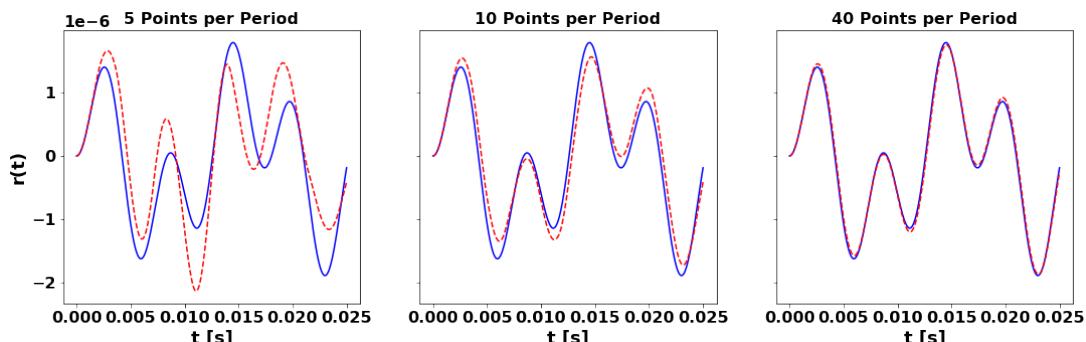
5.2 Convergence of Step Function Approximations

As discussed in Section 2, a key part of the theory underlying the WOBBLE modal analysis solver is that the applied loadings are discretised in time using step functions. In order to examine the ability of this step function approximation to accurately mimic the dynamics of the system, a comparison is made to a simplified analytic case. In particular, a single mode was forced with a projected forcing of the form $\mathcal{F} = \cos(\omega_F t)$. This system has the known analytic solution:

$$\ddot{r} + \omega^2 r = \cos(\omega_F t) \text{ with } r(0) = \dot{r}(0) = 0$$

$$\Rightarrow r(t) = \begin{cases} \frac{t}{2\omega} \sin(t\omega), & \text{if } \omega_F = \omega \\ \frac{1}{\omega^2 - \omega_F^2} (\cos(\omega_F t) - \cos(\omega t)), & \text{else} \end{cases} \quad (37)$$

The ability of step functions to accurately predict the analytic solution was investigated in both the case of non-resonance ($\omega_F \neq \omega$) and the case of resonance ($\omega_F = \omega$). Results are shown in Figure 7 and it is observed that relatively few step functions per period are required to obtain an accurate approximation.



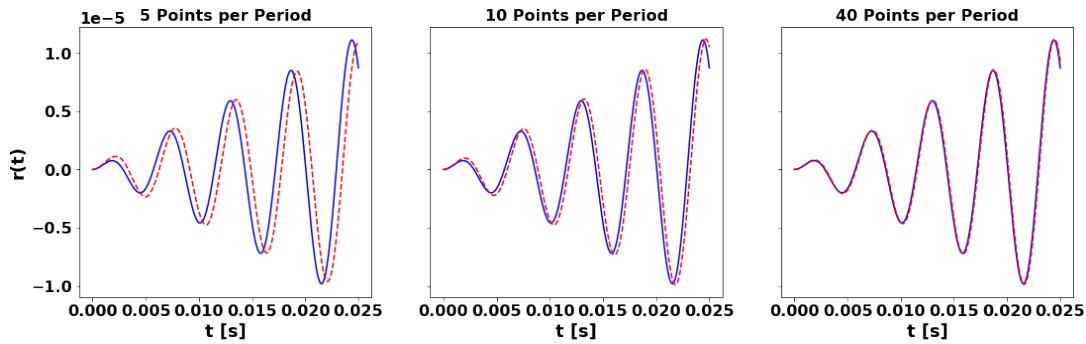


Figure 7: Numerical solution (red) and analytic solution (blue) for the non-resonant case (top) with $\omega_F = 400$ and $\omega = 1096$ and the resonant case (bottom) with $\omega_F = \omega = 1096$.

5.3 Validation of Mode Truncation Method

The mode truncation method discussed in subsection 4.3 was applied to the given mesh to determine the number of necessary modes to keep. Specifically, the method was applied for the *shear test* and *complex forcing* loading cases shown in Figure 6. The results of the effective mass computations for these two loading cases are shown in Table 2. Using the discussed 90% criterion, the table shows that only the first mode is necessary for the shear test loading and only modes 1, 2, and 4 are necessary for the complex forcing.

To visualise the effects of using different numbers of modes, simulations were conducted in the case of too few modes and more modes than necessary. Figure 8 shows the results of these simulations for the shear test loading and the displacement of the node at A. As expected, it is seen that along the main axes of displacement (y and z), using one mode suffices for modelling the behaviour. The results along the x-axis, however, notably differ between low and high numbers of modes. This is within the limits of what is predicted by the effective mass method, as the small scale of the x-displacement means that the modes governing this displacement do not notably contribute to the total effective mass. To be consistent across simulations and loadings, it was decided to use ten modes for all further verification tests. This slightly higher number of modes than suggested by the effective mass method was selected to ensure that the other validation tests would not be overly affected by (albeit small) mode truncation errors.

Table 2: Summary of the effective masses for a shear force (left) and a complex forcing (right)

Mode	ω	m^{eff}	
		Val	%
1	39224	1.62e-03	94.91%
2	39237	1.65e-05	0.97%
3	1084895	6.68e-06	0.39%
4	1198351	3.34e-05	1.96%
5	1200573	7.43e-08	0.00%
6	2467403	6.19e-12	0.00%
7	7095980	3.10e-06	0.18%
8	7102093	3.72e-08	0.00%
9	9815132	7.14e-07	0.04%
10	20273710	1.95e-07	0.01%
Total	-	1.71e-03	100%

Mode	ω	m^{eff}	
		Val	%
1	1337869	3.27e-06	27.87%
2	1339773	8.24e-08	0.70%
3	4325042	1.75e-06	14.85%
4	7646647	5.78e-06	49.24%
5	7652958	4.04e-07	3.44%
6	9869694	4.43e-10	0.00%
7	17392735	3.93e-07	3.34%
8	21995779	5.66e-08	0.48%
9	22077620	4.41e-09	0.04%
10	39486567	4.18e-09	0.04%
Total	-	1.17e-05	100%

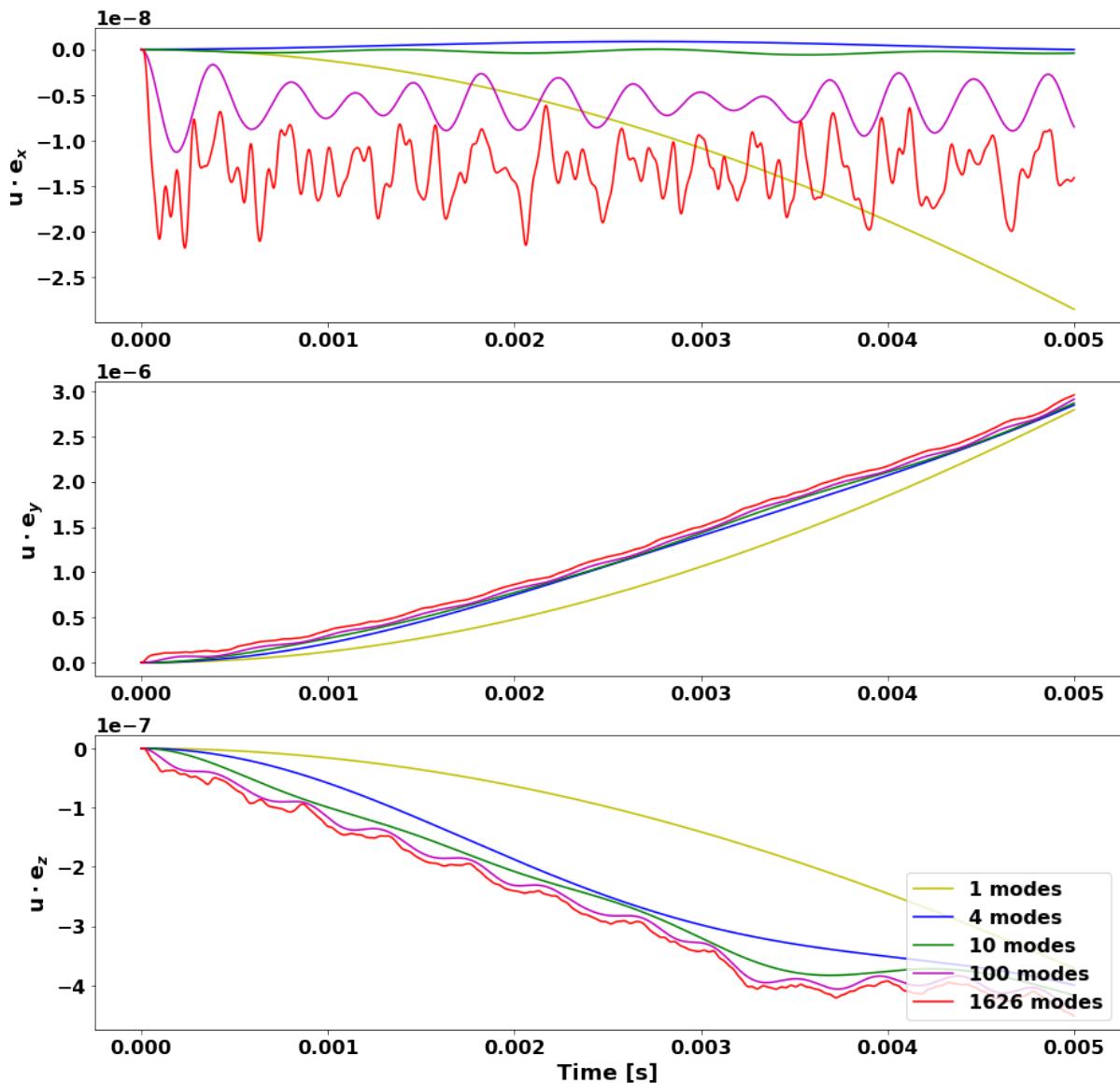


Figure 8: Convergence analysis of the displacements at the point A with respect to the number of modes.

5.4 Effects of Decoupling

In subsection 4.1 a theory was developed to allow users to choose between using SimpleRB and CoupledRB. Here, this theory is tested and it is illustrated how it may be used in practice. The displacement-limited option shall be used to determine the $\alpha - \Omega$ decision boundaries and, based on educated guesses of what may apply in practice, it has been chosen that displacements under $1\mu m$ are negligible and displacements above $5cm$ indicate deformation beyond the linear regime⁹. This means that the decision boundaries are at $|\alpha| + |\Omega|^2 = 1.19$ and $|\alpha| + |\Omega|^2 = 59570$ respectively.

To make this information easy to understand and apply by users of the WOBBLE code, it is visualised on a plot as shown in Figure 9. The green region indicates that the fictitious forces are expected to lead to displacements less than $1\mu m$, so that the SimpleRB class can be used.

⁹It should be noted that the precise choice of these values does not affect the validity of the verification conducted here.

The yellow region indicates that the effect of the fictitious forces is beyond what can be ignored and that thus CoupledRB should be used. Finally, the red region indicates that any such $\alpha - \Omega$ combinations will lead to displacements so large that they go beyond the linear elastic assumption underlying the WOBBLE code.

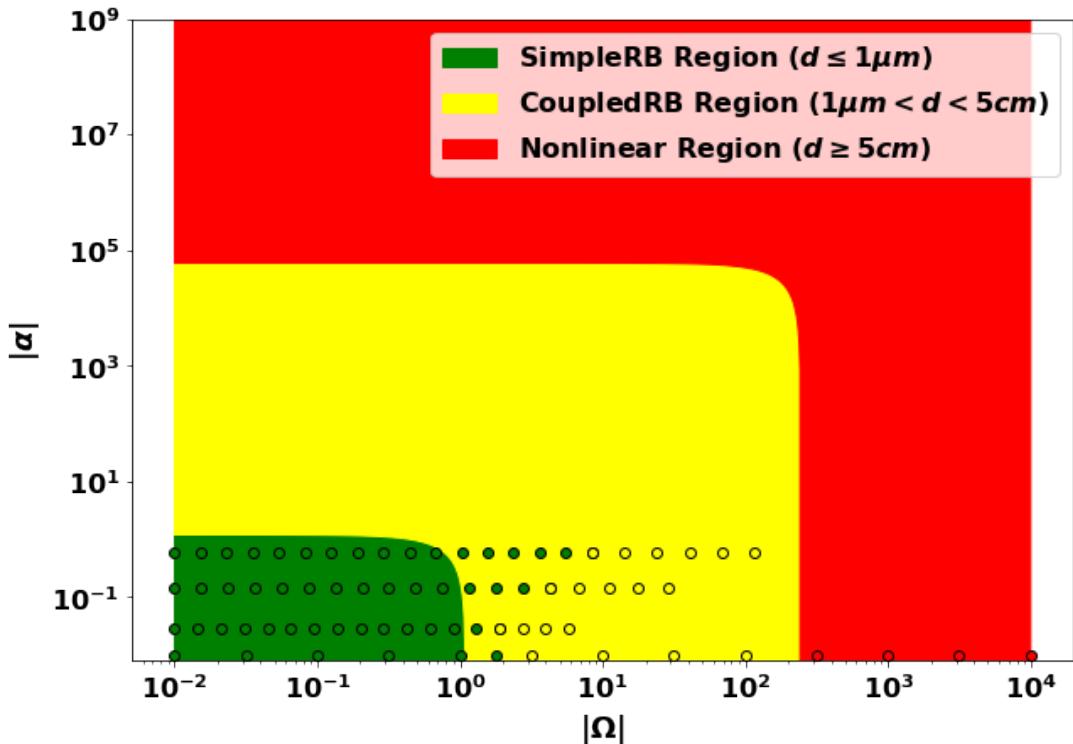


Figure 9: Visualisation of $\alpha - \Omega$ decision boundaries

Of course it is necessary to verify the validity of Figure 9 numerically. This is done using the two types of tests described below, which both make use of the RMS displacement values across all nodes.

- **Constant Angular Velocity:** The system is subjected to no external forces but is set with an initial angular velocity about the x axis. SimpleRB and CoupledRB results are then compared in order to determine the displacement difference due to the fictitious forces.
- **Constant Angular Acceleration:** The system is subjected to a forcing $+F$ on one side and a forcing $-F$ on the other (as depicted in Figure 6), leading to a pure rotation with a constant angular acceleration. At each timestep of the simulation, the results of SimpleRB and CoupledRB are compared as before.

The results of these tests are indicated in Figure 9 using the small coloured circles, where the position of the circles indicate the tested $\alpha - \Omega$ combinations (with $\alpha = 0$ simply shown at the bottom of the log plot). The circle colour indicates the magnitude of the displacement difference between the SimpleRB and CoupledRB simulations, with the colours corresponding to the same displacement limits as used for the regions of the plot.

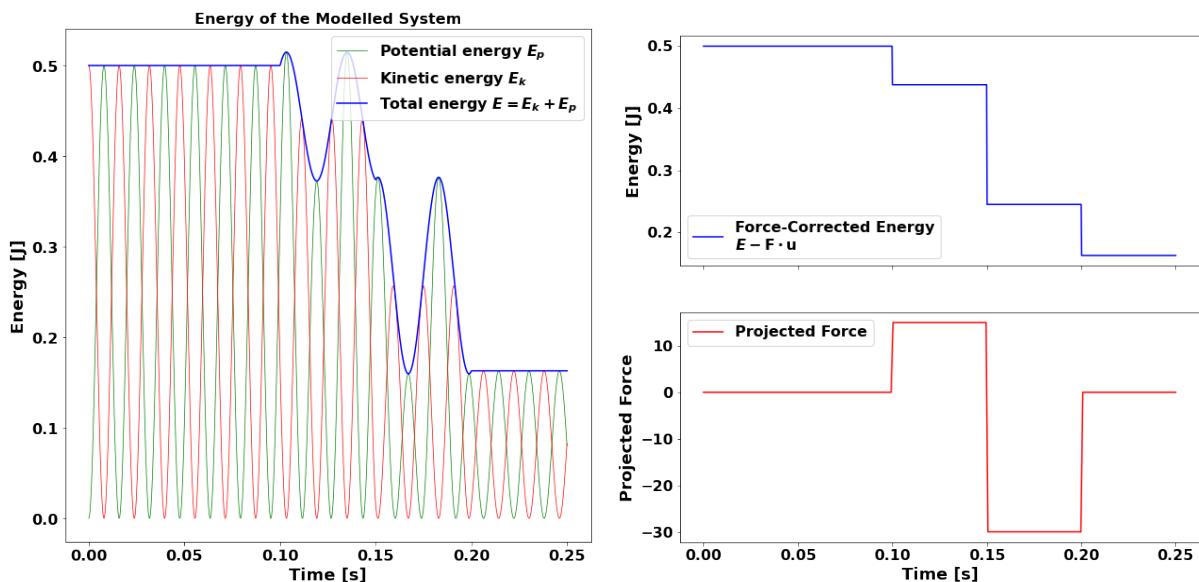
Clearly, Figure 9 shows that there is very good general agreement between the numerical test results and the theoretical decision boundaries. For some α the green-yellow decision boundary is notably more conservative than the numerical results show to be necessary. However, this is not a major issue as the decision plot errs on the side of using a more exact method, so users will not experience larger errors than expected.

5.5 Energy Considerations

In order to verify that the simulation methods work as expected, energy considerations were taken into account and examined. Here, two different situations in which energy conservation is expected are considered and analysed in terms of the computed energy values.

5.5.1 Step Function Loading on a Single Mode

Here, the case of the beam clamped on one side was considered in modal space. A single mode was forced with a projected force consisting of four step functions and starting and ending with zero force. Thus, constant energy was expected for the start and the end of the forcing. The initial projected velocity for the same mode was set to one. Furthermore, it was also expected that the force-corrected energy $E - f_{ext} \cdot u$ would be constant over each step loading. As shown in Figure 10, all of the expected behaviour was observed in the simulation results.



(a) Potential, kinetic and total energy of the system over the forcing. Constant total energy is observed when no force is exerted, as expected.
(b) Force profile and behaviour of the force-corrected energy. It is seen that the force corrected energy is constant for each step loading, as expected.

Figure 10: Behaviour of the energy due to a step-function loading of a single mode of frequency $\omega = 198$.

5.5.2 Constant Rotation

Constant energy is also expected in the case of a beam rotating with no external force applied. In SimpleRB there is no potential energy so the energy is purely kinetic and energy conservation is a direct sign of the beam maintaining a constant angular velocity. For CoupledRB the situation is slightly more complex, as the coupling means that there is a transfer between rigid body kinetic energy and internal kinetic and potential energy.

Numerical results for the case of no applied force and an initial angular velocity of one radian per second about the x-axis are shown in Figure 11. As shown in the figure, the energy for SimpleRB is indeed purely kinetic and is perfectly conserved. For CoupledRB, there is a transfer between kinetic and potential energy as expected. The plot of total energy, however, shows that there is a small periodic variation and that energy is thus not perfectly conserved.

However, it is decided that this variation is negligible as it is bounded and is smaller than the potential and total energies by respective factors of 10^{-6} and 10^{-12} ¹⁰. As expected, the energy in CoupledRB is also predominantly kinetic.

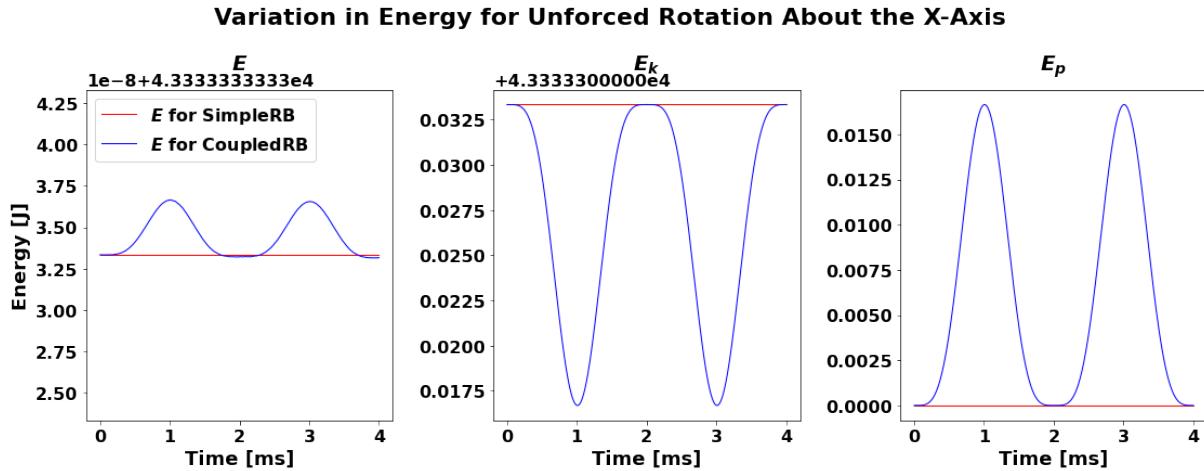


Figure 11: Comparison of the energy computed with CoupledRB and SimpleRB in the case of pure rotation with a timestep of $\Delta t = 10^{-5}$.

5.6 Timestep Considerations

Prior to interpreting simulation results, it is very important to validate that the simulation is converged with respect to the used timestep. Due to the non-linear nature of the phenomenon, it is expected that especially the accurate modelling of rotation will require small enough timesteps. To provide insights into the order of magnitude that can be expected for suitable timesteps and to ensure the validity of the here-discussed verification tests, a sample convergence analysis is conducted using SimpleRB for the cases of accelerating rotation and complex forcing (with the loadings as shown in Figure 6).

The case of constant angular acceleration was investigated for an angular acceleration with magnitude $|\alpha| = 0.144$, since in combination with the low expected angular velocities this lies in the decoupled regime (see subsection 5.4). The results are shown in Figure 12, and it is found that even a timestep as large as one second is sufficient to model changes in rotation.

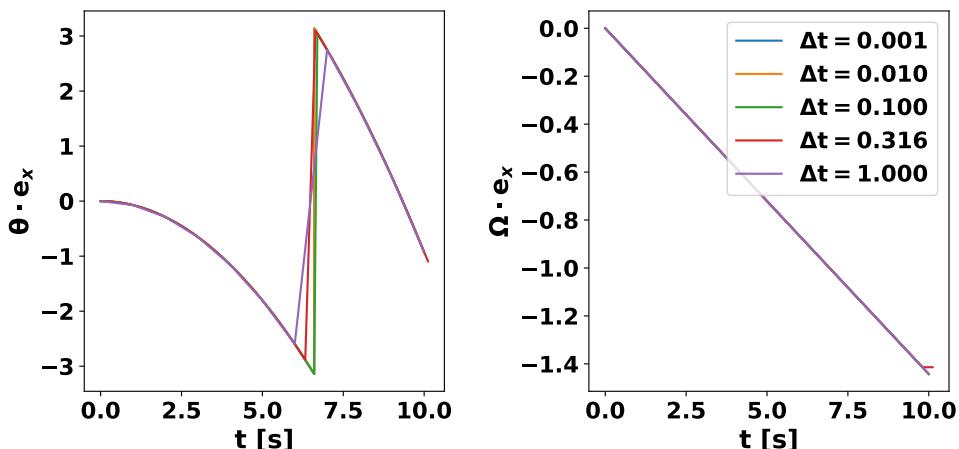


Figure 12: Convergence analysis w.r.t. the time step for pure rotation modelled with SimpleRB.

¹⁰Note the different scales on the three plots.

The results for the convergence analysis using the complex forcing are shown in Figure 13. The SimpleRB approach was again found to converge extremely rapidly, though expectedly a slightly smaller timestep is required to model this more intricate behaviour. The results are found to be almost converged for a timestep of $\Delta t = 0.1$ and fully converged for $\Delta t = 0.01$.

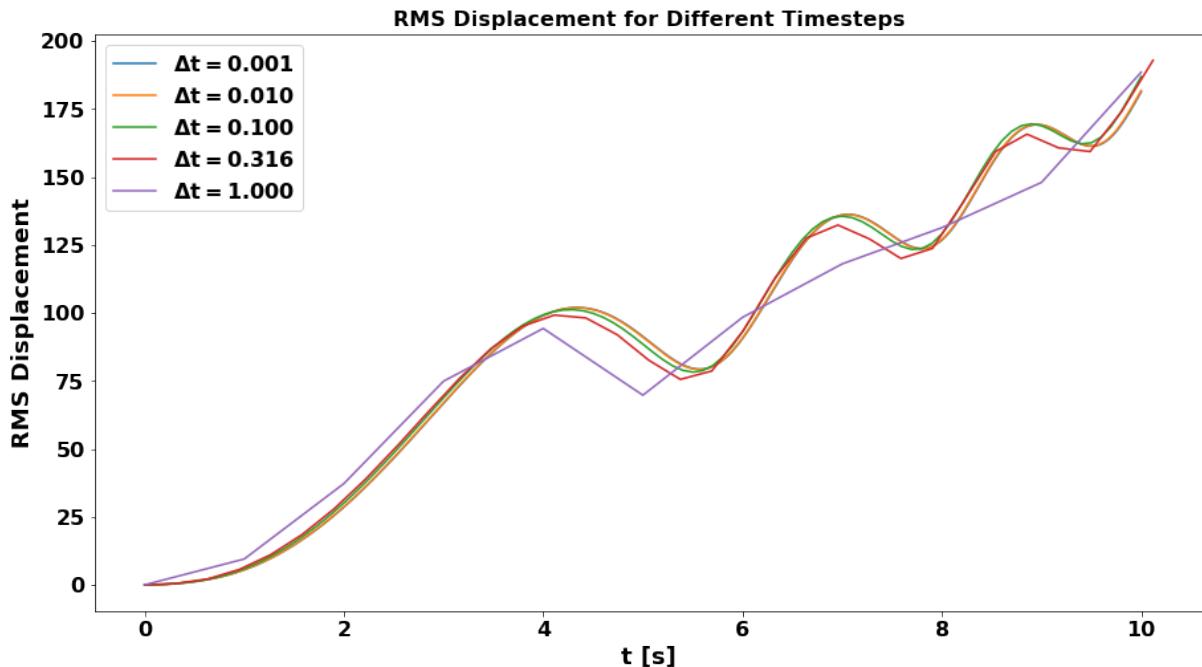


Figure 13: Convergence analysis w.r.t. the time step for complex forcing modelled with SimpleRB.

In general, the applicable timestep depends on the considered mesh and the applied loading, and thus users are recommended to conduct their own convergence studies whenever using the WOBBLE RBM packages. However, the convergence analysis conducted here already works to highlight one of the key advantages of the WOBBLE package in comparison to Akantu. While WOBBLE can safely be used in the considered scenarios for a time step of $\Delta t = 0.01$ the stable timestep of Akantu for the mesh is $\Delta t = 1.08 \times 10^{-5}$, which is significantly smaller, leading to more expensive computations.

5.7 Comparison with Akantu and Analytic Solutions

The most important verification of the WOBBLE package is the comparison against current state-of-the-art models such as Akantu, and comparison against known analytic solutions. To this end, a thorough comparison is detailed in this section for the three types of forcings depicted in Figure 6.

5.7.1 Shear Test for Clamped Beam

The shear test for a clamped beam is a classic scenario which has been previously studied in some detail. Various theories are possible, with three popular linear elastic theories being the Euler–Bernoulli, Rayleigh and Timoshenko theories. A thorough derivation of the behaviour predicted from the Euler-Bernoulli theory is presented in [13]. From this theory, it is expected that the maximum deflection of the steady state solution at the end of the beam is given as shown

below [13], where L is the beam length, \tilde{E} is the Young's modulus, and \tilde{I} is the area moment of inertia of the beam cross section:

$$u(L) = \frac{f_{ext}L^3}{3\tilde{E}\tilde{I}} \quad (38)$$

and with $\tilde{\rho}$ the material density and \tilde{A} the beam cross-sectional area, the first natural frequency of the vibrations is given as [13, 14, 15]:

$$\omega = 1.875^2 \sqrt{\frac{\tilde{E}\tilde{I}}{\tilde{\rho}\tilde{A}L^4}} \quad (39)$$

A numerical test was conducted to test this theory with $f_{ext} = 500$. The expected results from theory are $\omega = 203$ and $u(L) = 1.25 \times 10^{-6}$. Good numerical agreement was observed, with values of $\omega = 191$ and $u(L) = 1.26 \times 10^{-6}$. These findings are visualised in Figure 14. It is important to note that Equation 38 refers to the steady-state solution, thus meaning the average solution over a period of oscillation from the modal analysis prediction.

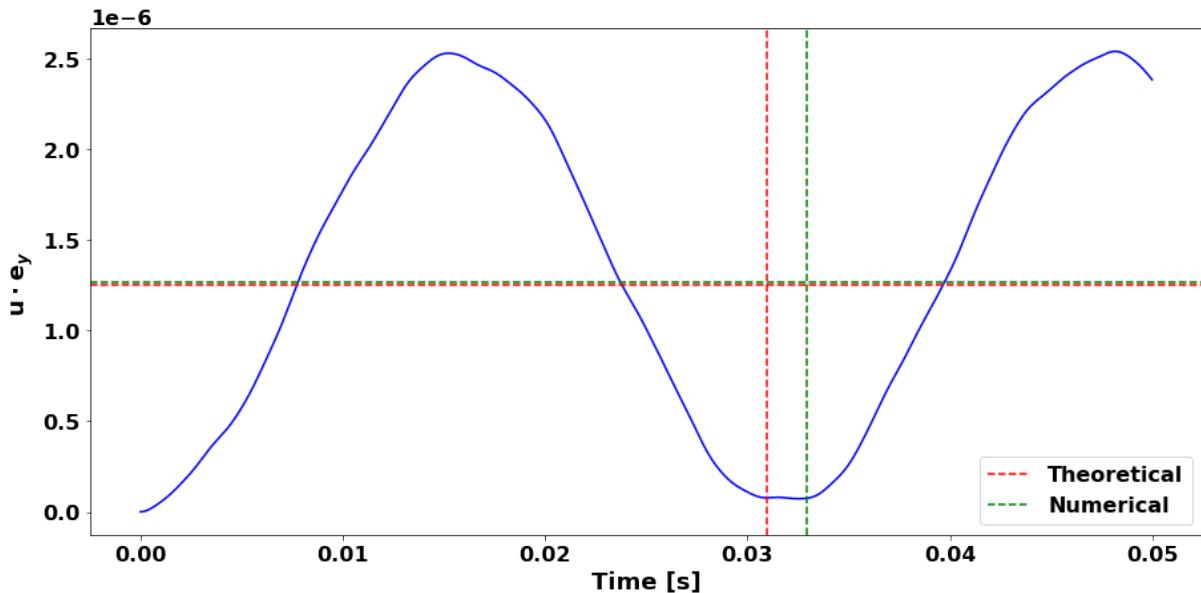


Figure 14: Predictions of WOBBLE (green) and Euler-Bernoulli beam theory (red).

Furthermore, exact agreement is not expected. This is because the Euler-Bernoulli theory makes assumptions in addition to linear elasticity - namely, it assumes that plane sections of a beam remain plane. This assumption holds in the case of elongated beams. However, the beam considered in this work may not be sufficiently elongated for this assumption to hold. Furthermore, there is a general consensus among the literature that of the three theories, the Timoshenko theory is best able to capture the real dynamics [16], although it is still imperfect [17].

5.7.2 Pure Rotation with Constant Angular Acceleration

In order to validate the ability of the WOBBLE package to model rigid body motion, the pure rotation situation depicted in Figure 6 was examined using a loading of ± 100 Newtons and the results are shown in Figure 15. For comparison, the node located at A was examined

Along the main directions of rotation, extremely good agreement is observed between Akantu and the WOBBLE simulations using SimpleRB and CoupledRB. Importantly, WOBBLE

was able to correctly predict that there should not be any displacement along the rotation axis (the x axis). Akantu, however, predicts small but growing deviations along this axis. After an investigation into the nature of the used mesh and observing that the same spurious displacement along the rotation axis occurs in the case of constant unforced rotation, it was concluded that the unexpected Akantu behaviour is most likely the result of small asymmetries in the mesh, which lead to spurious body deformations even when none are intended. In WOBBLE the mesh asymmetries are presumably hidden in the computation of the eigenmodes and inertia tensor and thus do not notably impact the simulations results.

Accelerating rotation about the x axis

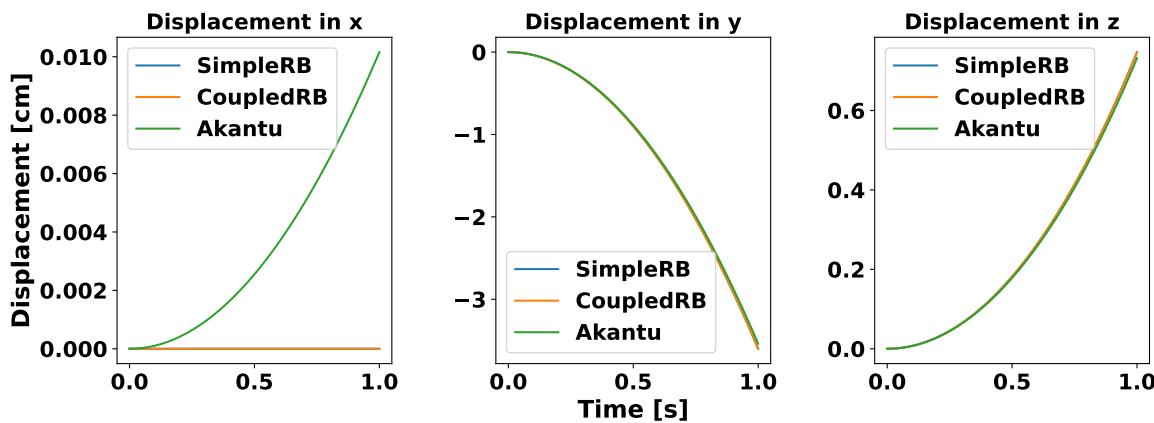


Figure 15: Akantu and WOBBLE results for accelerating rotation about the x-axis.

5.7.3 Combined Rotation, Translation and Modal Analysis

The final verification was undertaken using the complex forcing shown in Figure 6, which is expected to lead to rotation, translation and deformation vibrations. Again, for comparison the node located at A was examined and extremely good agreement was observed in all direction, as depicted in Figure 16.

Response to complex forcing

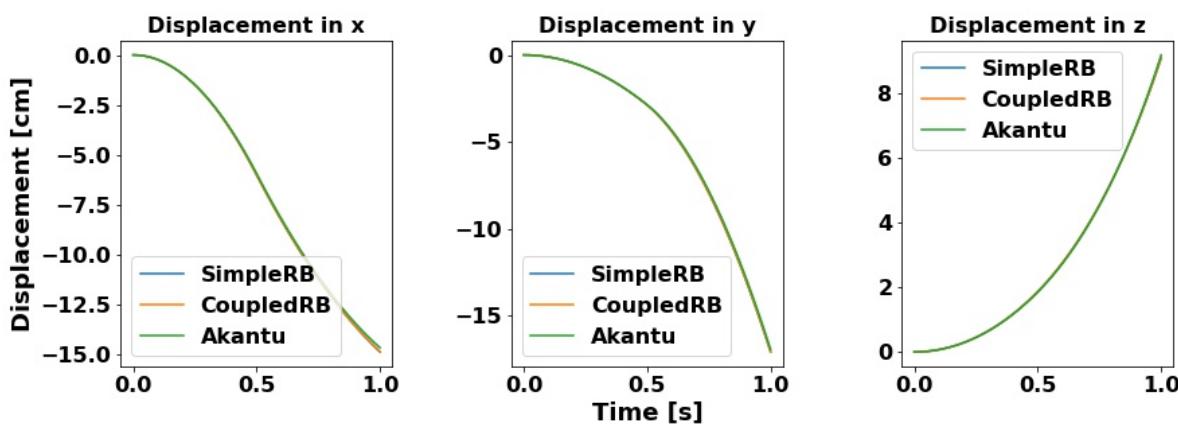


Figure 16: Akantu and WOBBLE results for complex forcing.

Furthermore, the RMS error between WOBBLE and Akantu was investigated. As shown in Figure 17, the error was found to be lower than 2% of the Akantu total RMS displacement. This is a relatively low error percentage and can potentially be explained by the mesh asymmetry effects observed for the accelerating rotation test. Thus, strong agreement between WOBBLE and the existing Akantu software is concluded.

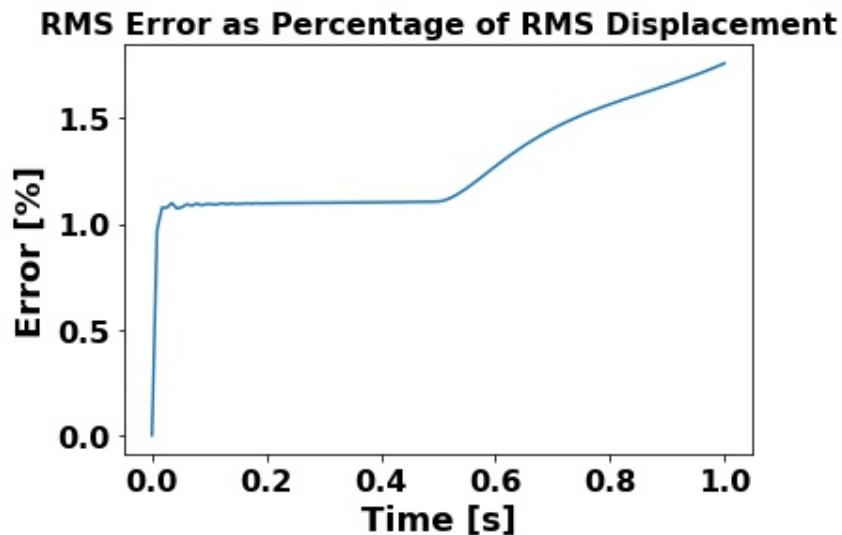


Figure 17: RMS error as percentage of Akantu RMS displacement for complex forcing

6 Case Study

With the WOBBLE code verified, it is desired to test its performance on a hypothetical test case. With applications in mind as might be encountered by *Clearspace*, it was decided to model the response of the previously discussed beam to a loading representing the collision with another object. Specifically, a loading is applied for 10 ms to the front face of the beam (so where the x-coordinate is zero) in the area where the y-coordinate is above 0.3m and the z-coordinate is greater than 4.1m . The total normal loading in the x-direction is 100 Newtons and the total shear loading in the z-direction is 10 Newtons. This loading is visualised in Figure 18 and was chosen as it is expected to induce a variety of different structural responses combining vibrations and RBM. First, the performance of the WOBBLE simulation is compared to Akantu. Then, the computational savings of using the boundary approach for WOBBLE are assessed.

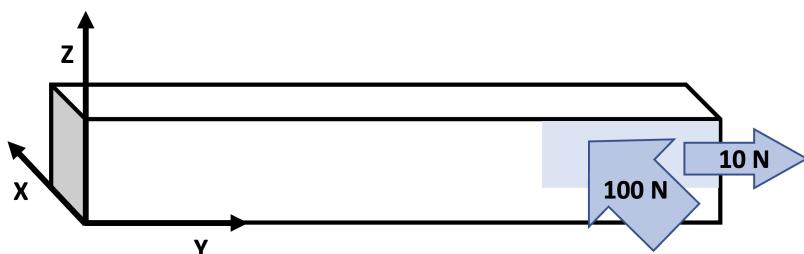


Figure 18: Case study forcing indicating a collision for 10 ms

6.1 Comparison with Akantu

The response of the structure is simulated over a total duration of one simulation second using Akantu and SimpleRB. Importantly, after the SimpleRB simulation it was confirmed that the angular acceleration and angular velocity values encountered throughout the simulation are indeed sufficiently small to justify neglecting coupling. The results are, for one thing, again compared using the displacement of the node A at the mesh origin. Furthermore, they are compared using the RMS error as percentage of the Akantu RMS displacement. This leads to the results shown in Figure 19. While there is a difference between the results of the two simulations it is rather small at around four percent. Importantly, the error percentage also does not grow over time and would thus presumably not increase in case that the simulation duration was extended.

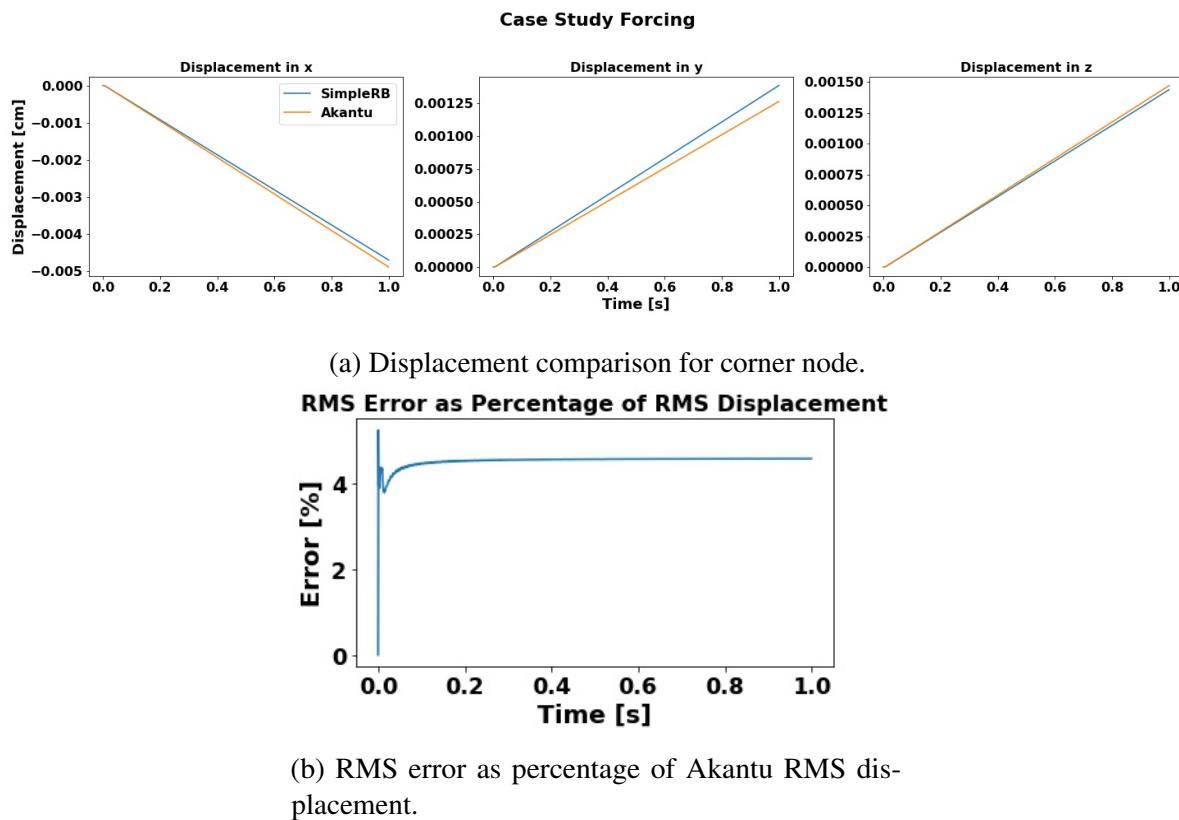


Figure 19: Case study simulation results

In addition to the results of Figure 19, another key result of the case study is that the computational time taken by Akantu and SimpleRB was measured. While the precise times vary between repeated simulation runs, on average it was found that Akantu took around 190 seconds whereas SimpleRB only took around 1.3 seconds. This is a time saving by a factor of almost 150 and highlights the significant computational savings that are possible when using the WOBBLE package for simulations such as the one studied here. Importantly, the SimpleRB simulation was run for a time step of 8.7×10^{-4} , which is 100 times greater than the stable time step required for the Akantu simulation. This relatively small time step was used because the very short duration of the applied force required a sufficiently fine timely resolution. However, as was demonstrated in subsection 5.6, this high timely resolution is not necessary for SimpleRB when the force no longer changes so rapidly, so then a larger time step could be used. Thus, the implementation of a varying time step in SimpleRB, as discussed in Section 7, is likely to lead

to further computational savings.

6.2 Computational Savings of Boundary Approach

For the SimpleRB simulation discussed above, computations were made even more efficient by using the boundary approach and thus only considering nodes on the surface of the structure. To quantify how significant the impact of the boundary approach is on the computational savings, a secondary investigation was done for the case study. Here the above loading was simulated with and without using the boundary approach and the computation time was measured and averaged over repeated runs. This was furthermore repeated for both a coarser and a finer mesh. The results are presented in Table 3¹¹.

Table 3: Computational cost of repeated case study runs

Mesh	Boundary approach	Average time [s]	Standard deviation [ms]
Coarse (237 nodes, 178 boundary nodes)	✓	0.61	25.9
	✗	0.63	36.4
Regular (579 nodes, 426 boundary nodes)	✓	1.34	54.8
	✗	1.50	57.5
Fine (1141 nodes, 798 boundary nodes)	✓	3.30	38
	✗	3.61	186

Clearly the computational savings are notable, as for the regular and fine meshes the boundary approach lowers the cost by around 10%. From the discussion in subsubsection 4.2.1 it follows that the computational savings of the boundary approach are related to the difference between the total number of nodes and the number of boundary nodes. Specifically, where C_1 denotes the cost independent of the boundary approach and C_2 indicates the cost related to the boundary approach, the total cost is as shown in Equation 40. Thus, the difference in cost should be a constant multiple of the difference between total and boundary number of nodes, as shown in Equation 41.

$$\begin{aligned} \text{Regular cost: } & C_1 + C_2 N \\ \text{Boundary approach: } & C_1 + C_2 N \frac{N_b}{N} \end{aligned} \tag{40}$$

$$\Delta \text{cost} = C_2(N - N_b) \tag{41}$$

To check whether the observed computational savings indeed obey this relation, the results of Table 3 were plotted in Figure 20. As the graph shows, the linear fit of the results (weighted by the inverse of the error bars and constrained to go through zero) indeed passes through all error bars and is sufficiently close to the experimental curve to suggest that computational savings of the boundary approach indeed scale linearly with $N - N_b$.

¹¹Note that these computation times were found using the EPFL NOTO server on the 3rd of June 2022. For each result the `%%timeit` functionality was used to run the simulations for 7 runs with 20 loops each.

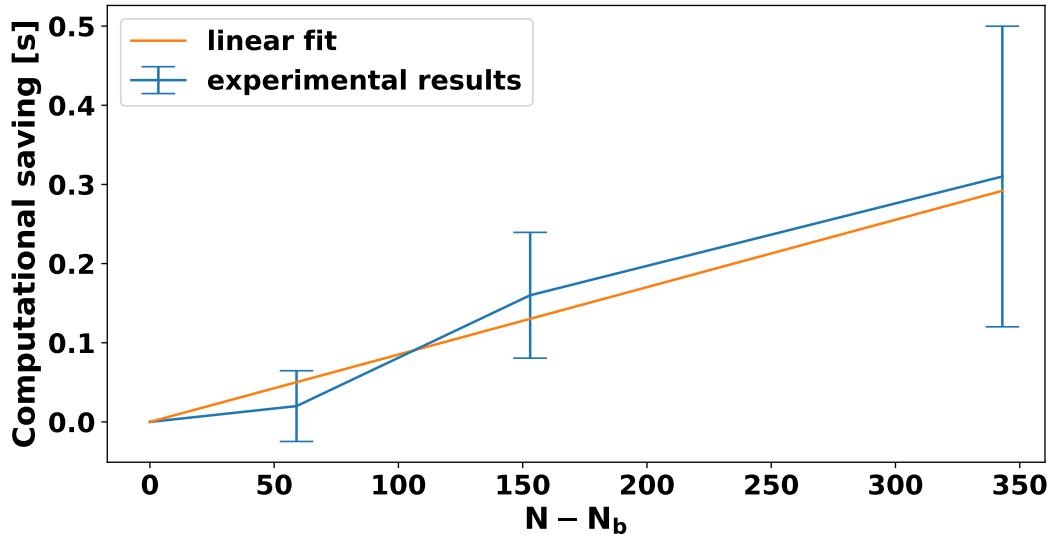


Figure 20: Computational savings of the boundary approach

7 Recommendations for Future Work

The presented work leaves scope for a number of possible extensions which are presented in this section. The tasks with higher priorities are presented in subsection 7.1, and other areas to examine are presented in subsection 7.2.

7.1 Strongly Recommended Topics

The presented body of work represents a good tool for simulating solid mechanics, as seen by Section 6. However, a number of improvements can be made. In particular, three key recommendations have been identified - including damping, adding the ability to model multiple bodies, and implementation in C++ within Akantu.

Damping

The case considered in this report was without damping (see Equation 1). However, in many applications, this assumption of no damping is not valid and an extra term needs to be added to the governing equation as follows:

$$M\ddot{u} + C\dot{u} + Ku = f_{ext} \quad (42)$$

However, the addition of this term leads to complications in the diagonalisation step, which no longer always yields a diagonal system. The case where the system is diagonalisable is known as Rayleigh or classic damping, and occurs when $CM^{-1}K = KM^{-1}C$ holds. However, whilst Rayleigh damping is a convenient mathematical description of damping, it is not always realistic [18], which has lead to many authors opting for non classically damped systems [19, 20]. Unfortunately, more realistic models suffer from the fact that the system is no longer diagonalisable. To deal with this, a frequently used assumption is to ignore off-diagonal terms, but this also can have issues [21].

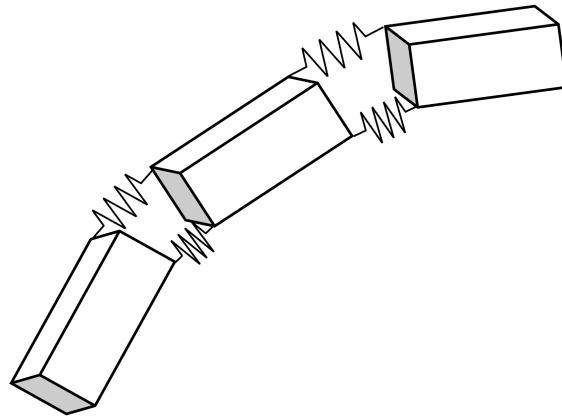
Modal analysis requires the ability to diagonalise the system. As a result, despite the issues mentioned above, it is proposed to implement Rayleigh damping. This is preferable since it is easily implementable - the only change to the presented theory is that the decoupled ODEs (see Equation 6) now contain an additional term depending on \dot{r} . More complex models can be investigated in the future if they are deemed necessary.

Integration with Connected Bodies

In many applications, a single body (or a single mesh) is not realistic. Objects of interest often consist of multiple interconnected parts with different materials. As mentioned in the introduction, an example of such an application is ClearSpace [22], which aims to use a satellite with multiple robotic arms as depicted in Figure 21.



(a) An example application showing the need for connected bodies. Image taken from [22].



(b) Example means to simulate arms using multiple beams connected by springs.

Figure 21: Illustration of method to simulate more complex bodies using spring connections.

To model structures consisting of multiple interconnected parts, it is possible to simply add an additional external force between the connected nodes to keep them together. This means the introduction of a von Neumann boundary condition for certain modes. An easy way to model this force is by using a spring-like force i.e. $f_{connection} = -Kx^2$, where x is the distance between the two nodes that should be connected. With this force included, it is then possible to conduct the simulations with WOBBLE as usual. An illustration of how this approach could simulate complex bodies such as robotic arms is depicted in Figure 21.

Implementation Within Akantu

In this work, many comparisons with Akantu were undertaken. It was found that the WOBBLE package gives good agreement with Akantu and may be able to give better predictions in some situations, such as the one depicted in Figure 15. This shows that WOBBLE can complement the existing Akantu framework, which makes it a good candidate for incorporation into the software.

Furthermore, WOBBLE is currently written in Python. A faster implementation can be achieved by rewriting the software in C++, which is the language that Akantu is written in. Thus, it would make sense to directly implement the routines of WOBBLE into Akantu, which would speed up the WOBBLE code even further and would also allow it to be part of a state-of-the-art software package for solid mechanics.

7.2 Further Areas to Investigate

Besides the three key recommendations discussed in subsection 7.1, there are also a number of other possible areas of investigation. These are presented in this section, ordered by importance according to the authors.

Investigation of Unidirectional Coupling

As discussed in subsection 5.4, the key component of coupling is the inclusion of fictitious forces. This coupling occurs from the rigid body solver to the modal analysis solver. However, in the `CoupledRB` algorithm, coupling is also undertaken in the opposite sense i.e. the torque and moment of inertia needed for rigid body mechanics are updated based on the modal analysis step (see subsubsection 3.4.4). However, this update is not as important and it may be possible to simply ignore the effects of coupling in this direction. Specifically, by assuming that the positions of the nodes do not change much i.e. $b \approx b_0$ and furthermore $\dot{b} \approx 0$, one can entirely neglect updating the moment of inertia and the torque. Furthermore, the small displacement assumption for the nodes means that the fictitious forces can be computed using only the initial nodal positions and without the Coriolis term. Together, this removes the need for the costly backprojection from modal to physical space, speeding up the code.

In short, this means that an interesting candidate method to investigate is one implementing unidirectional coupling in the discussed direction. It is worth noting that this method may have problems with energy conservation since the coupling is one-way, but numerical simulations will have to be undertaken to determine whether this is problematic. If it is not, this proposed method may provide a useful intermediate alternative between `SimpleRB` and `CoupledRB` in terms of both speed and accuracy.

Non-Uniform Time Array

The `WOBBLE` RBM classes are currently implemented so that they calculate the solution at uniformly spaced times. This is disadvantageous in that it is not possible to vary the timestep during a single simulation. As discussed in Section 6, having a varying timestep can be very beneficial in terms of computation speed. Although with the current implementation a varying timestep could be achieved by running new simulations with the initial conditions set to the previous results, it would be easier and more efficient to allow for non-uniform time arrays within `WOBBLE` directly.

Better Mode Truncation

As discussed in subsection 4.3, the `WOBBLE` package implements the mode displacement method for mode truncation. However, there exist other more accurate methods that are also possible, which better take into account the truncation errors arising from dropping modes. Two of such other methods are the mode acceleration method and the mode truncation augmentation method, as discussed by Neela [8]. Neela furthermore undertook a comparative analysis between the methods and found that the mode augmentation method was best able to mimic the true solution. Thus, adding the implementation of the mode augmentation method to `WOBBLE` may lead to further increases in accuracy.

Forces in the G Frame

Currently, only external forces acting in the body frame are specifiable in WOBBLE. However, there exist applications in which forces acting in the inertial frame (e.g. gravity) may be relevant. Currently, implementing this would require transforming the gravitational force from the inertial frame to the body frame at each timestep. This is inefficient and if in future work body forces are of interest, a new, more efficient method for dealing and reading body forces should be implemented.

Fourier Approximation of External Forcings

The current step-loading approach for approximating external forces is sufficient in most cases, as shown in subsection 5.2. However, for oscillatory excitations, it is natural to use a different approximation in terms of sinusoidal functions. As a result, another possibility to specify the external forcing using sinusoidal approximations may be beneficial since it may require fewer terms and thus speed up simulations.

Auto-detection of Modes

Determining which modes to retain for simulation requires an investigation into the effective modal masses of the various modes, as was summarised in Table 2. An implementation of a method to automatically determine which modes to keep would thus be advantageous.

More Accurate Numerical Approximations

As per subsubsection 3.4.2, numerical integration in WOBBLE is undertaken using *Forward Euler* approximations. As seen by Section 6, this integration scheme is sufficient to obtain good results. However, better results can obviously be obtained by using higher order methods. Naturally, an investigation into the trade off of accuracy and computation speed will be necessary.

Nonlinear Elasticity

Wobblē is based on linear elastic theory, but in reality materials are rarely linear elastic. Even common structural materials such as aluminium are better modelled with more general nonlinear elastic models [23]. As a result, future work should investigate the inclusion of more sophisticated nonlinear elastic material models, such as the Neo-Hookean or Mooney-Rivlin models or even higher order elastic models. An enormous quantity of different nonlinear elastic material models have been proposed [24, 25], so the exact choice of model to use is often problem-dependent. An important point to note is that nonlinear elasticity means that the stress is now dependent in a nonlinear fashion on the positions of the nodes, so that the stiffness matrix K is no longer constant. This means that finding the eigenmodes from the matrix $M^{-1}K$ is no longer so simple since this matrix is no longer a constant. Thus, the extent to which the use of non-linear elasticity models is feasible in Wobblē will have to be assessed.

Finding Rotation Angles from the Rotation Matrix

In the WOBBLE package, the rotation of the simulated body is found using the rotation matrices $R(\theta)$. If it is desired to then find the rotation vectors θ , a logmap is used. However, the logmap has the well-known issue that it has a singularity for $|\theta| = \pi$. To circumvent this, a numerical approximation method, such as the Newton Raphson method, should be implemented for the inversion of $R(\theta)$.

8 Conclusion

The aim of this project was to develop an accurate and efficient simulation software that allows users, such as the engineers at *Clearspace*, to model the dynamics of solid bodies in large rotations and translations. This should be done over long periods of time but at a low computational cost. In light of everything that has been discussed in this report, we can confidently claim that the developed WOBBLE package fulfils these requirements.

In this report, the theory underlying the WOBBLE package was presented and explained. Firstly, there is the theory of modal analysis, which allows decoupling the governing equations of solid mechanics and solving the uncoupled differential equations in modal space. While the offline computation of finding the eigenmodes can be expensive, the computational cost of the actual simulations is cheap and can be further reduced by only retaining a small number of important eigenmodes. The discretisation of the applied loading using step functions allows for analytic time integration, which means that the modal analysis solution does not have to be numerically integrated over time, but instead only evaluated at the times of interest. In WOBBLE, modal analysis for structures fixed in space is implemented in the `PureMA` class.

The second half of the WOBBLE theoretical foundation is the theory of rigid body motion. As the modal analysis approach is based on linear elastic theory, it is not suitable for modelling large rotations that go beyond the tangent space. Thus, rigid body mechanics is used for modelling the translation and rotation that bodies undergo when not sufficiently fixed in space. While the non-linearity of the equations governing rotation means that numerical time integration is now a necessity, the computational cost remains low as modal analysis continues to be used for modelling the body deformations and because large timesteps can be used. In WOBBLE, the combination of modal analysis and RBM is implemented in two different ways. `SimpleRB` models body deformations and RBM separately and thus disregards the coupling between the two. It is computationally cheap but has lower accuracy than its counterpart `CoupledRB`. `CoupledRB` accounts for the bi-directional coupling between body rotation and deformation, which comes at the cost of more expensive computations.

Next, methods were presented that increase the efficiency of the WOBBLE simulations. Firstly, this included a theory using user-prescribed limits on forces or displacements to determine whether `SimpleRB` can be used or if the higher level of accuracy of `CoupledRB` is required. Secondly, the boundary approach was developed. Here only the nodes on the surface of the mesh are considered and computational savings for projections in both directions between modal and physical space were found to scale linearly with the difference between the total number of nodes and the surface number of nodes. The third method for efficient simulation is the effective mass method, which allows the determination of which modes to retain for a specific forcing.

With the WOBBLE code implemented in accordance with the presented theory and principles, the subsequent step was to verify that it indeed produces reliable results. First, the validity of the step function approximation for applied forces was confirmed through the comparison of simulation results and analytic solutions. Next, the mode truncation method and the method for deciding between `SimpleRB` and `CoupledRB` was considered. For both methods, good agreement was found between theory and numerical results. In line with what is expected, the mode truncation method prioritises the coordinate axes along which displacements are large. Thus a larger number of modes is necessary if it is desired to accurately model displacements in secondary directions. The decision method for choosing between coupled and uncoupled RBM methods is reliable but slightly more conservative than necessary. However this is not seen as a big problem as the method errs on the side of rather using `CoupledRB`.

The next part of the verification was to consider the computed system energy. First, the

forcing of one specific mode was used to confirm that the modal analysis part of the solver perfectly obeys energy conservation. When the situation of constant unforced rotation was considered, it was found that SimpleRB too gives full energy conservation, but CoupledRB gives rise to small oscillations in total energy. However, these oscillations are of constant amplitude and of a much smaller scale than both potential and kinetic energy. Thus, they are not seen as a concern. A performed convergence analysis subsequently showed that WOBBLE can be run for much larger timesteps than Akantu, which is another major contributor to the low computational cost of the simulation package. The final part of the verification was to compare the WOBBLE results to the analytic results of Euler-Bernoulli beam theory as well as to numerical results computed using Akantu. Strong agreement was shown across all of these comparisons, with the relative RMS error between WOBBLE and Akantu not exceeding 2% for the case of a complex forcing involving deformation, translation, and rotation. The results found for pure accelerating rotation furthermore suggest that the WOBBLE simulation may be less susceptible than the Akantu results to spurious displacements caused by mesh asymmetries.

After verifying the WOBBLE code, the package was utilised to simulate a forcing that could possibly be encountered in space. For this, a case study scenario based on a $10ms$ impact was devised and simulated using WOBBLE and Akantu. For one thing, good agreement was again found between the simulation results of the two softwares. The displacement of the corner of the simulated beam was predicted almost identically by both codes, and the RMS error across all nodes was small at a constant 4%. Furthermore, the computational costs were measured and it was found that WOBBLE reduced the computation time by a factor of almost 150, from around 190 seconds to around 1.3 seconds. A secondary investigation subsequently used the case study loading to show that using the boundary approach gives computational savings as expected and allowed the WOBBLE case study simulation to be around 10% faster than if all nodes had been considered.

Given the high performance and drastic reduction in computation time offered by WOBBLE, it is highly recommended that the package is rewritten in C++ and integrated into the existing Akantu framework. In addition to making the code even faster, this will allow users of Akantu to benefit from a solid mechanics model specifically designed to offer cheap simulations for bodies in large translation and rotation. Other key recommendations which will significantly extend the applicability of the WOBBLE code are that Rayleigh damping should be incorporated and that a method for the interfacing between connected bodies should be developed.

References

- [1] EPFL Laboratoire de Simulation en Mécanique des Solides. *Akantu*. 2020. URL: <https://akantu.ch/> (visited on May 19, 2022).
- [2] F Ma and CH Ng. “On the orthogonality of natural modes of vibration”. In: *Mechanics Research Communications* 31.3 (2004), pp. 295–299.
- [3] O. A. López and M. Cruz. “Number Of Modes For The Seismic Design Of Buildings”. In: *Earthquake Engineering And Structural Dynamics* 25 (1996), pp. 837–855.
- [4] D. Garanin. “Rotational motion of rigid bodies”. In: (2008).
- [5] Howard Haber. *Three-Dimensional Rotation Matrices*. 2012. URL: <https://scipp.ucsc.edu/~haber/ph216/> (visited on May 19, 2022).
- [6] O.C. Zienkiewicz, R.L. Taylor, and D.D. Fox. *The Finite Element Method for Solid and Structural Mechanics*. 7th ed. Amsterdam: Elsevier, 2014.
- [7] Ansys. *Mode Participation Factor and Effective Mass*. 2020. URL: https://courses.ansys.com/wp-content/uploads/2020/10/Lesson4_ModeParticipationFactor.pdf (visited on May 18, 2022).
- [8] Sravan Kumar Neela. “Mode Truncation Analysis in Forced Response of Structural Dynamics”. MA thesis. The University of Texas at Arlington, Dec. 2011.
- [9] Autodesk. *How many modes should I run in a Modal (Natural Frequency) Analysis?* 2017. URL: <https://knowledge.autodesk.com/search-result/caas/sfdcarticles/sfdcarticles/How-many-modes-should-I-run-in-a-Modal-Natural-Frequency-Analysis.html> (visited on May 18, 2022).
- [10] Dlubal. *How many mode shapes do I need to consider in the dynamic analysis?* 2019. URL: <https://www.dlubal.com/en/support-and-learning/support/faq/000242> (visited on May 18, 2022).
- [11] Eurocode CEN. “Design of structures for earthquake resistance, part 1-1: general rules, seismic actions and rules for buildings, EN 1998-1”. In: (2004).
- [12] Christophe Geuzaine and Jean-François Remacle. “Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities”. In: *International journal for numerical methods in engineering* 79.11 (2009), pp. 1309–1331.
- [13] Scott Whitney. “Vibrations of cantilever beams: Deflection, frequency, and research uses”. In: *Website: Apr 23.10* (1999).
- [14] Amrita Vishwa Vidyapeetham Virtual Labs. *Free Vibration of a Cantilever Beam (Continuous System)*. 2022. URL: <https://vlab.amrita.edu/?sub=3&brch=175&sim=1080&cnt=1> (visited on May 19, 2022).
- [15] Tom Irvine. *Bernoulli-Euler Beams*. 2022. URL: <https://endaq.com/pages/bernoulli-euler-beams> (visited on May 19, 2022).
- [16] Anneke Labuschagne, NF Janse van Rensburg, and AJ2480028 van der Merwe. “Comparison of linear beam theories”. In: *Mathematical and Computer Modelling* 49.1-2 (2009), pp. 20–30.
- [17] JM Duva and JG Simmonds. “The usefulness of elementary theory for the linear vibrations of layered, orthotropic elastic beams and corrections due to two-dimensional end effects”. In: (1991).
- [18] Aldo Sestieri and SR Ibrahim. “Analysis of errors and approximations in the use of modal co-ordinates”. In: *Journal of sound and vibration* 177.2 (1994), pp. 145–157.
- [19] Fai Ma, Matthias Morzfeld, and Ali Imam. “The decoupling of damped linear systems in free or forced vibration”. In: *Journal of Sound and Vibration* 329.15 (2010), pp. 3182–3202.

- [20] Sondipon Adhikari. *Structural dynamic analysis with generalized damping models: analysis*. John Wiley & Sons, 2013.
- [21] Matthias Morzfeld, N Ajavakom, and F Ma. “Diagonal dominance of damping and the decoupling approximation in linear vibratory systems”. In: *Journal of Sound and Vibration* 320.1-2 (2009), pp. 406–420.
- [22] ClearSpace. *ClearSpace - A mission to make space sustainable*. 2022. URL: <https://clearspace.today> (visited on May 18, 2022).
- [23] Natalie Rauter and Rolf Lammering. “Numerical simulation of elastic wave propagation in isotropic media considering material and geometrical nonlinearities”. In: *Smart Materials and Structures* 24.4 (2015), p. 045027.
- [24] Michel Destrade, Giuseppe Saccomandi, and Ivonne Sgura. “Methodical fitting for mathematical models of rubber-like materials”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473.2198 (2017), p. 20160811.
- [25] L Angela Mihai and Alain Goriely. “How to characterize a nonlinear elastic material? A review on nonlinear constitutive parameters in isotropic finite elasticity”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473.2207 (2017), p. 20170607.
- [26] Christopher Hoen. “An Engineering Interpretation of the Complex Eigensolution of Linear Dynamic Systems”. In: *Presented at IMAC XXIII* (Jan. 2005).
- [27] M Aenlle, Martin Juul, and Rune Brincker. “Modal mass and length of mode shapes in structural dynamics”. In: *Shock and Vibration* 2020 (2020).
- [28] Dlubal. *Manual RF-DYNAM Pro*. Dlubal. 2020.
- [29] Comsol. *Eigenfrequency Studies*. 2019. URL: https://doc.comsol.com/5.5/doc/com.comsol.help.sme/sme_ug_theory.06.43.html (visited on May 19, 2022).

A Notes on Using the WOBBLE Package

The code for the WOBBLE package can be found online at github.com/lldriever/WOBBLE. In addition to the source code of the simulation package, the repository contains a number of sample IPython notebooks and data files. These sample files can be used to reproduce the plots shown in this report and are intended to function as a starting point for other investigations that users of WOBBLE may want to conduct. Importantly, to use the code in the repository it will also be necessary to install the Python version of Akantu.

Here, a set of fundamental instructions are provided on how to use the WOBBLE package. They are intended as a starting point which users can then build upon using the provided sample notebooks. First, geometry and mesh files are discussed before it is explained how forces are input into WOBBLE. Then, a flowchart is presented that outlines the decisions and steps that are necessary for running a WOBBLE simulation. Finally, a sample snippet of code for setting up and running a SimpleRB simulation is provided.

Geometry and Mesh Files

In WOBBLE it is possible to either provide a geometry file of the type `.geo` or to directly provide a suitable mesh file of the format `.msh`. Should a geometry file be provided, WOBBLE will create a suitable second order mesh using Gmsh [12]. In case that boundary conditions are to be applied, the provided geometry or mesh file should contain named groups that can be used to identify the applicable nodes.

Inputting Forces

Entering forces into WOBBLE is done using specifically formatted text files. The first row of the file provides the times (in seconds) at which the specified forces are to be applied. All rows below specify the force (in Newtons) applied to the separate degrees of freedom, i.e., the x, y, and z directions for each node. Importantly, the file only specifies the force for unblocked degrees of freedom - if a node is for example clamped along all three axes, the force file will be shorter by three rows. The horizontal delimiter used between values are single blank spaces.

Sample force files are provided in the git repository. If users desire to create their own force files, it is recommended to write a script that loads the applicable mesh and uses boolean masks to identify which degrees of freedom are to be forced how much and at what time.

Decisions when Running a WOBBLE Simulation

The flow chart shown in Figure 22 provides an overview of the main choices and steps that users have to make when running a simulation using WOBBLE. Note that for the box marked `*`, `save_modes=True` means that the computed eigenmodes will be stored after computation such that they can then be used for more efficient simulations in future. Computing the modes may be very expensive, but it is advised to set `num_modes='all'` and thus compute all modes. This is advised as it means that the modes will not have to be re-computed in case that more modes are desired in future.

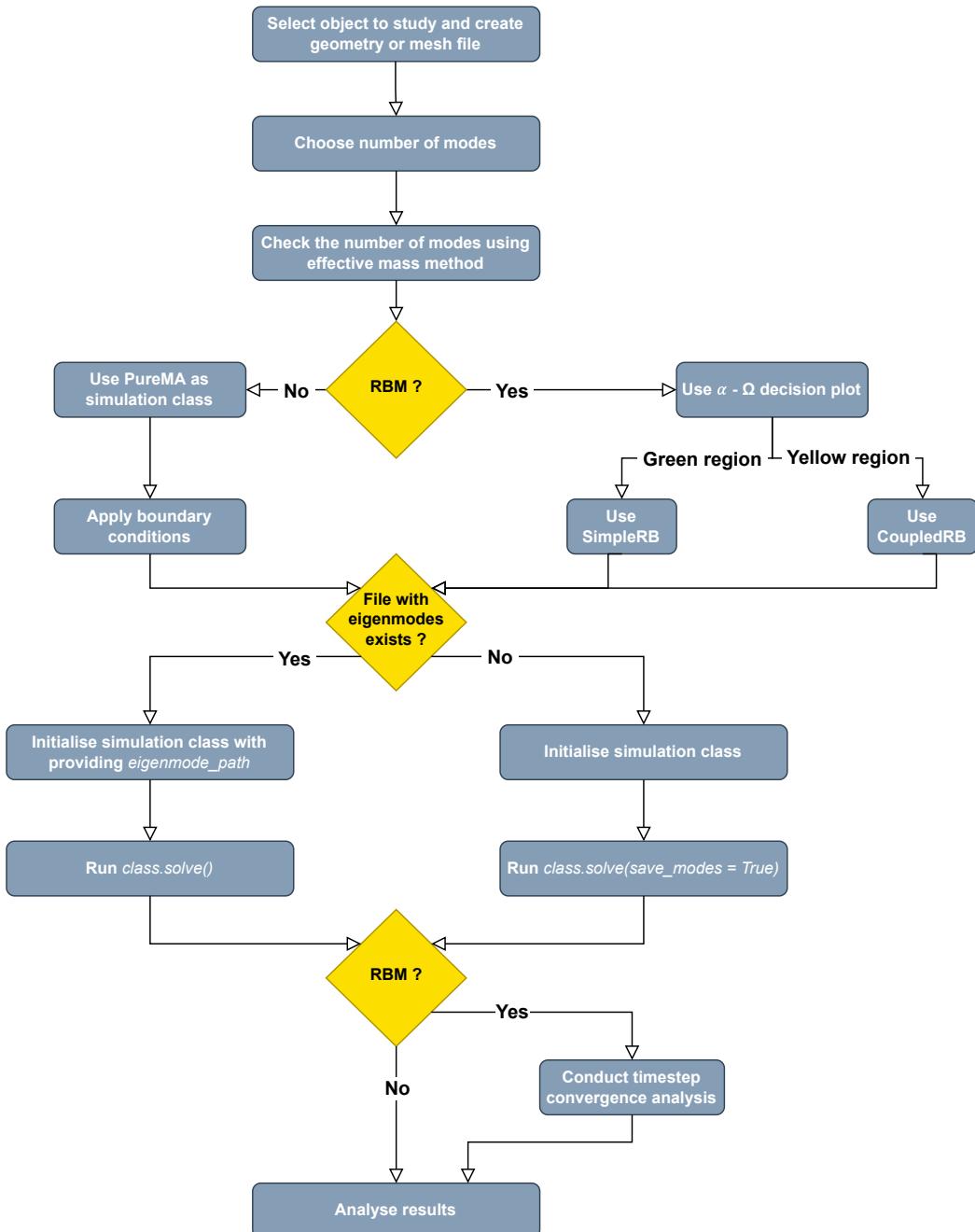


Figure 22: The key decisions and steps in running a WOBBLE simulation

Sample Code Snippet

Here, a sample piece of code is provided to illustrate how easy it can be to run a WOBBLE simulation. All of the simulation classes (PureMA, SimpleRB, and CoupledRB) are equipped with `.solve()` methods which act as wrapper functions for all of the methods usually required for setting up and running a simulation. Thus, running a simulation is as easy as shown in Listing 1 - one only needs to initialise the simulation and call `.solve()`. All of the computed information can then be accessed using the members of the class or by calling additional methods. For examples of how this is done, consult the example notebooks in the repository.

```
1 sim=SimpleRB('Test_Simulation',           # Name of the simulation object
2                 3,                         # Number of dimensions
3                 0.01,                      # Timestep
4                 1.5,                       # Total time
5                 mesh_file='beam.msh',
6                 material_file='material.dat',
7                 num_modes=10,
8                 force_path='force.txt',
9                 eigenmode_path='mode.csv')
10
11 sim.solve()
```

Listing 1: Example of running SimpleRB