

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/260983266>

A beam to 3D model switch for transient dynamic analysis

Conference Paper · January 2012

CITATIONS

6

READS

126

4 authors, including:



Mikhael Tannous

12 PUBLICATIONS 85 CITATIONS

[SEE PROFILE](#)



Patrice Cartraud

Ecole Centrale de Nantes

110 PUBLICATIONS 1,837 CITATIONS

[SEE PROFILE](#)



David Dureisseix

Institut National des Sciences Appliquées de Lyon

193 PUBLICATIONS 1,278 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



wood science for conservation of cultural heritage [View project](#)



Ship Ultimate Strength under Whipping [View project](#)

A BEAM TO 3D MODEL SWITCH IN TRANSIENT DYNAMIC ANALYSIS

Mikhael Tannous¹, Patrice Cartraud¹, David Dureisseix², and Mohamed Torkhani³

¹GéM, Ecole Centrale de Nantes
address

e-mail: {mikhael.tannous,patrice.cartraud}@ec-nantes.fr

² Lamcos, INSA de Lyon
address

e-mail: david.dureisseix@insa-lyon.fr

³ EDF, R&D, Clamart
address

e-mail: mohamed.torkhani@edf.fr

Keywords: dynamics, finite element, switch

Abstract. *Many industrial problems involving slender structures in transient dynamics require a 3D model for a better understanding of local or non linear effects that occur along a small period of time, whereas a simplified beam model can be sufficient for simulating the linear phenomena occurring for a long period of time.*

This paper proposes a method which enables to switch from a beam to a 3D model during a transient dynamic analysis solved with a time integration technique. Thus, this method allows to reduce the computational cost while preserving a good accuracy.

Starting with the beam model, at switch moment t_s , a 3D solution is constructed as the sum of a cross-section rigid body displacement corresponding to the classical Timoshenko kinematical assumption and a 3D correction which accounts for cross-section deformation. This correction is obtained from the solution of a static problem and may be computed on three consecutive time steps (the switch instant, the previous and the following steps) thus allowing to make a velocity and acceleration corrections based on the three static computations. The dynamic simulation is initialized at t_s by the displacements, velocities and accelerations being computed.

The method is validated through comparisons with a reference solution corresponding to the 3D model solution used during all the simulation.

1 INTRODUCTION

Many industrial problems in transient dynamics, such as the simulation of turbine accidents, require a $3D$ model for a better understanding of non linear local effects such as rubbing, that occur along a small period of time. However, a $3D$ model for the entire rotor and the entire stator used during the whole simulation will result in an unaffordable computational time. We therefore, present a method that can dramatically reduce the computational time, for problems where the $3D$ non linearities are restricted in space and time.

For non linear phenomena that are restricted in time, one can use a time integration scheme switching technique. An implicit/explicit or explicit/implicit switching algorithm may reduce the computational costs such as described in [14, 15, 13, 16]. However, since in turbine accidents simulations, the non linearities are also restricted in space, time integration scheme switching will not lead to optimal computational cost.

Other approaches in the literature deal with reducing the computational cost of $2D$ or $3D$ models by using different mesh densities or models. Adaptive meshing techniques can take the form of a local refinement of a $3D$ model, such as in [10, 11, 17] or a beam to $3D$ refinement as done in [9] for the simulation of a bridge deformation. The beam and the local $3D$ refinement are connected by mathematical constraints. Domain decomposition techniques are also well known in this field to reduce computational cost while being accurate and examples can be found in [8], [7], [3, 12] and others. For this same purpose FETI domain decomposition methods were developed and used in [5, 2, 1, 6, 4].

These CPU time reducing techniques are very interesting. However, since non linearities are restricted in space and time, the optimal computational cost can not be achieved unless no $3D$ model at all is present during the linear stage, which means the use of a beam model. The turbine rotor with rubbing was modelled by a beam in [19] and [20]. However, the beam model is not adapted to treat accurately local rotor stator interactions and a $3D$ model is needed for a better understanding of local non linear effects. Therefore, a numerical strategy which enables to use a $3D$ or beam model at different stages of the transient analysis allows to reduce the computational cost while preserving a good accuracy is proposed here, as illustrated in Fig. 1. In fact, the simulation starts at $t = t_0$ with a beam model for a linear simulation, and switches at $t = t_{s1}$ to a $3D$ model when a non linear phenomenon is to take place. The simulation is to be switched back at $t = t_{s2}$ to the beam model for of the rest of the simulation that ends at t_f , if no more non linear phenomena are present.

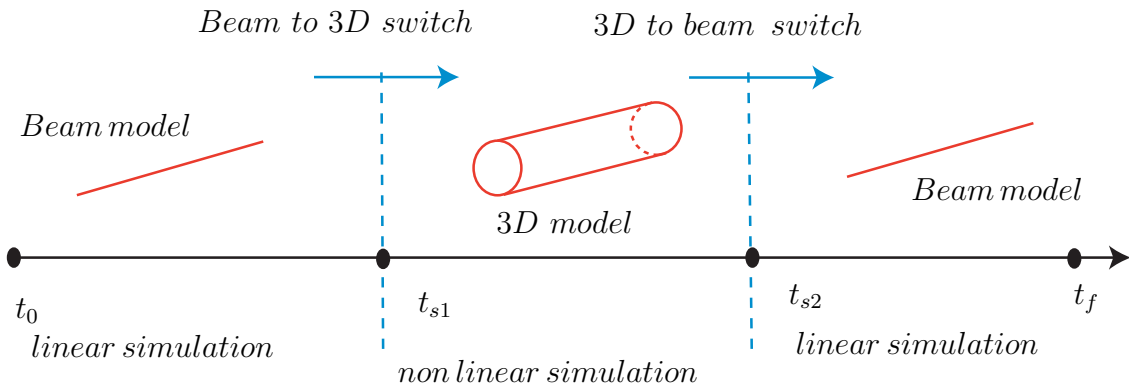


Figure 1: Beam to $3D$ switch

This raises the problem of the switch from one model to another and this paper is about a

beam to 3D model switch, the 3D to beam model switch is not addressed here.

2 MATHEMATICAL BASICS OF THE SWITCH

A beam model simulation that started at $t = 0$ is to be switched for a 3D model simulation at $t = t_s$. Starting with the 3D model at $t = t_s$ requires the collection of the beam model solution at t_s and transforming this solution to have a suitable 3D model initialization at the same moment.

The fundamental dynamic equation of a beam at $t = t_s$ can be written:

$$\mathbf{M}_b \ddot{\mathbf{U}}_b + \mathbf{C}_b \dot{\mathbf{U}}_b + \mathbf{K}_b \mathbf{U}_b = \mathbf{f}_b \quad (1)$$

where, $\mathbf{M}_b$, $\mathbf{C}_b$, and $\mathbf{K}_b$ are respectively the mass, damping and stiffness matrices of the beam model.

The 3D model at $t = t_s$ can be described by:

$$\mathbf{M}_{3D} \ddot{\mathbf{U}}_{3D} + \mathbf{C}_{3D} \dot{\mathbf{U}}_{3D} + \mathbf{K}_{3D} \mathbf{U}_{3D} = \mathbf{f}_{3D} \quad (2)$$

where, $\mathbf{M}_{3D}$, $\mathbf{C}_{3D}$, and $\mathbf{K}_{3D}$ are respectively the mass, damping and stiffness matrices of the 3D model.

Suppose that we start with the beam model, at switch moment ($t = t_s$), we have to construct the 3D solution $\mathbf{U}_{3D}$ from the beam solution. This is performed first by decomposing the 3D displacement into a cross-section rigid body displacement corresponding to the classical Timoshenko kinematical assumption $\mathbf{P}\mathbf{U}_b$, and a 3D correction $\mathbf{U}_{3Dc}$ which accounts for cross-section deformation:

$$\mathbf{U}_{3D} = \mathbf{U}_{3Dc} + \mathbf{P}\mathbf{U}_b \quad (3)$$

We therefore need to generate $\mathbf{P}\mathbf{U}_b$ and to compute $\mathbf{U}_{3Dc}$ in order to construct the 3D model displacement at t_s .

2.1 Generating $\mathbf{P}\mathbf{U}_b$

$\mathbf{P}\mathbf{U}_b$ is obtained through a projector matrix $\mathbf{P}$ which transforms the beam displacement vector into a 3D rigid body displacement. However, it is not easy to build $\mathbf{P}$ because it depends on the relationship between the beam mesh and the 3D mesh, which may change from one cross-section to another. Instead, we will generate $\mathbf{P}\mathbf{U}_b$ as a whole.

Let N_{ij} a node that belongs to the i^{th} cross-section of the 3D model, $\mathbf{P}\mathbf{U}_b^{ij}$ is the displacement of N_{ij} computed for a cross-section rigid body displacement. The cross-section to which belongs N_{ij} has G_i for its gravity center. The i^{th} beam node has a displacement $\mathbf{U}_b^i$ and a rotational displacement θ_b^i . With the finite element software used, such as Code_Aster or Abaqus, we write a python loop that manages to compute $\mathbf{P}\mathbf{U}_b^{ij} = [\mathbf{P}\mathbf{U}_{bx}^{ij}, \mathbf{P}\mathbf{U}_{by}^{ij}, \mathbf{P}\mathbf{U}_{bz}^{ij}]$ for each node N_{ij} as follows:

$$\mathbf{P}\mathbf{U}_b^{ij} = \mathbf{U}_b^i + \mathbf{N}_i \mathbf{G}_i \wedge \theta_b^i \quad (4)$$

2.2 Computing $\mathbf{U}_{3Dc}$

Due to the decomposition of the 3D displacement according to Eq. (3), the 3D model initialisation will be performed through the 3D correction $\mathbf{U}_{3Dc}$. Thus, inserting Eq. (3) in Eq. (2) at

$(t = t_s)$ gives:

$$\mathbf{M}_{3D}(\ddot{\mathbf{U}}_{3Dc} + \mathbf{P}\ddot{\mathbf{U}}_b) + \mathbf{C}_{3D}(\dot{\mathbf{U}}_{3Dc} + \mathbf{P}\dot{\mathbf{U}}_b) + \mathbf{K}_{3D}(\mathbf{U}_{3Dc} + \mathbf{P}\mathbf{U}_b) = \mathbf{f}_{3D} \quad (5)$$

Since we have one equation with three unknowns, then making the following assumptions:

$$\begin{aligned} \dot{\mathbf{U}}_{3Dc} &= \mathbf{0} \\ \ddot{\mathbf{U}}_{3Dc} &= \mathbf{0} \end{aligned} \quad (6)$$

will result in a displacement correction $\mathbf{U}_{3Dc}$ that corresponds to a static computation for the 3D model, at $t = t_s$, and that is the solution of the following equation:

$$\mathbf{K}_{3D}\mathbf{U}_{3Dc} = \mathbf{f}_{3D} - \mathbf{M}_{3D}\mathbf{P}\ddot{\mathbf{U}}_b - \mathbf{C}_{3D}\mathbf{P}\dot{\mathbf{U}}_b - \mathbf{K}_{3D}\mathbf{P}\mathbf{U}_b \quad (7)$$

The computations of $\mathbf{P}\dot{\mathbf{U}}_b$ and $\mathbf{P}\ddot{\mathbf{U}}_b$ can be done in the same way as $\mathbf{P}\mathbf{U}_b$ by substituting in Eq. (4) $\mathbf{U}_b$ by $\dot{\mathbf{U}}_b$ or $\ddot{\mathbf{U}}_b$.

Then one can switch to the 3D model which is initialized at $t = t_s$ by:

$$\begin{aligned} \mathbf{U}_{3D} &= \mathbf{U}_{3Dc} + \mathbf{P}\mathbf{U}_b \\ \dot{\mathbf{U}}_{3D} &= \mathbf{P}\dot{\mathbf{U}}_b \\ \ddot{\mathbf{U}}_{3D} &= \mathbf{P}\ddot{\mathbf{U}}_b \end{aligned} \quad (8)$$

With this choice, only the displacements are different from the cross-section rigid-body displacement assumption at the switch instant. However, the initialization of the accelerations and velocities are a critical issue in many integration schemes such as several integration schemes of the Newmark integration scheme family, or the central difference scheme. Since no correction has been developed for the velocities or accelerations, it means that they will be different with the 3D reference ones. In most cases of study shown later in this paper, this difference is around 5% on velocities and accelerations, which seems to be quite small. However, this induces a transient phenomenon in which high frequency oscillations occur, which are an artefact of this initialization. These high frequency oscillations may lead the solution to diverge. In order to vanish these oscillations, one can insert a numerical damping or make velocities and accelerations correction as detailed in Section 3.

3 INITIALIZING THE 3D SOLUTION

3.1 Numerical damping (HHT integration technique)

A numerical damping in the integration scheme can filter these high frequency oscillations without any other influence on the solution. The HHT integration scheme has been used to filter the numerical oscillations. This numerical damping needs to be maintained on the several time steps following the switch in order the high frequency oscillations vanish, as shown in the following results. However, a more attractive method does exist and can reduce considerably the high frequency oscillations that appear after switching and is detailed in Section 3.2.

3.2 A triple static switch procedure

As Eq. (8) shows, the high frequency oscillations are generated by a poor initialization of the velocities and accelerations, since these later are generated from the beam solution and are

not completely adapted to the 3D model. The hypothesis taken in Eq. (6) is too strong and generates high frequency oscillations. However, assuming that the displacement initialization is adapted to the 3D model, then a strategy enabling a better initialization of the velocities and accelerations based on the displacement correction can be built with the integration scheme and thus eliminates the high frequency transient phenomenon that occurs after switching.

We therefore first check the displacement correction on a static problem to prove its efficiency and then, according to the integration scheme being used, construct velocity and acceleration initializations.

3.2.1 The switch for static problems

The switch for a static problem may be seen as a particular case of the dynamic one. It is investigated here to test if the 3D displacements after switching are close to a reference 3D static solution. The beam fundamental equation for a static problem is:

$$\mathbf{K}\mathbf{U}_b = \mathbf{f}_b \quad (9)$$

The 3D fundamental equation is:

$$\mathbf{K}_{3D}\mathbf{U}_{3D} = \mathbf{f}_{3D} \quad (10)$$

The 3D displacement can be divided as explained earlier in Eq. (3), and that will lead to define $\mathbf{U}_{3Dc}$ as the solution of:

$$\mathbf{K}_{3D}\mathbf{U}_{3Dc} = \mathbf{f}_{3D} - \mathbf{K}_{3D}\mathbf{P}\mathbf{U}_p \quad (11)$$

This static correction $\mathbf{U}_{3Dc}$ summed with $\mathbf{P}\mathbf{U}_b$ is compared to a reference solution for the same 3D model mesh, computed by solving Eq. (10). This has been performed on several mesh types, cross-sections shapes and boundary conditions, and the difference between the computed displacements and the reference solution has been found to be negligible, indicating that the displacements are well corrected by this switch method.

3.2.2 Basics of the triple static switch procedure

Since a static switch provides an accurate correction of the 3D displacements, a correction of the velocities and accelerations can be built using three static switch procedures at three consecutive time steps. The velocities and accelerations at $t = t_s$ will be computed from the displacement values at this instant and at one step before this instant and one step after. From these values, velocities and accelerations will be obtained through the same expressions as those used in the time integration technique chosen for solving the problem. This procedure is based on the knowledge of the time integration technique used to solve for the dynamic equation of motion.

The integration technique mostly used in Code_Aster is implicit, and especially the Newmark integration scheme. In [18], we find a detailed description of the Newmark integration technique. The velocity is computed by:

$$\dot{\mathbf{U}}_{n+1} = \dot{\mathbf{U}}_n + (1 - \gamma)\Delta T\ddot{\mathbf{U}}_n + \gamma\Delta T\ddot{\mathbf{U}}_{n+1} \quad (12)$$

where $\mathbf{U}$, $\dot{\mathbf{U}}$, and $\ddot{\mathbf{U}}$ are respectively the displacements, velocities and accelerations. ΔT denotes the step time. n is the current step and $n + 1$ is the following one. One has also β and γ being the Newmark parameters. The displacements are computed by:

$$\mathbf{U}_{n+1} = \mathbf{U}_n + \Delta T \dot{\mathbf{U}}_n + \Delta T^2 \left(\frac{1}{2} - \beta \right) \ddot{\mathbf{U}}_n + \Delta T^2 \beta \ddot{\mathbf{U}}_{n+1} \quad (13)$$

The Newmark integration scheme is consistent because it can be proved that one has:

$$\lim_{\Delta T \rightarrow 0} \frac{\dot{\mathbf{U}}_{n+1} - \dot{\mathbf{U}}_n}{\Delta T} = \ddot{\mathbf{U}}_n \quad (14)$$

and

$$\lim_{\Delta T \rightarrow 0} \frac{\mathbf{U}_{n+1} - \mathbf{U}_n}{\Delta T} = \dot{\mathbf{U}}_n \quad (15)$$

Thus we define the acceleration at $t = t_s$ by:

$$\begin{aligned} \ddot{\mathbf{U}} &= \frac{\dot{\mathbf{U}}_{n+1} - \dot{\mathbf{U}}_n}{\Delta T} = \frac{\frac{\mathbf{U}_{n+1} - \mathbf{U}_n}{\Delta T} - \frac{\mathbf{U}_n - \mathbf{U}_{n-1}}{\Delta T}}{\Delta T} \\ &= \frac{\mathbf{U}_{n+1} - 2\mathbf{U}_n + \mathbf{U}_{n-1}}{\Delta T^2} \end{aligned} \quad (16)$$

and initialize the 3D model by the following velocities and accelerations:

$$\dot{\mathbf{U}}_{3D} = \frac{[\mathbf{P}\mathbf{U}_b + \mathbf{U}_{3Dc}]_{t_{s+1}} - [\mathbf{P}\mathbf{U}_b + \mathbf{U}_{3Dc}]_{t_{s-1}}}{2\Delta T} \quad (17)$$

$$\ddot{\mathbf{U}}_{3D} = \frac{[\mathbf{P}\mathbf{U}_b + \mathbf{U}_{3Dc}]_{t_{s+1}} - 2[\mathbf{P}\mathbf{U}_b + \mathbf{U}_{3Dc}]_{t_s} + [\mathbf{P}\mathbf{U}_b + \mathbf{U}_{3Dc}]_{t_{s-1}}}{\Delta T^2}. \quad (18)$$

This method is as accurate as the time step is small, since it is demonstrated that for sufficiently small time steps, it is consistent with the Newmark integration scheme in use. Nevertheless, the time step in use with the Newmark family is always finite, and thus the transient phenomenon will still be observed on the first 4 to 5 time steps that follows the switch. However, the high frequency oscillations present a very low amplitude, and vanish in the scope of the following 4 to 5 time steps that follows the switch. We therefore can propose this initialization of the 3D model as a proper beam to 3D model switching technique.

4 APPLICATION EXAMPLES

In this section, we present a simple numerical example that illustrates the efficiency of the beam to 3D model switch for dynamic cases. In fact, the method has been validated on more complex cases, for different type of cross-section shapes, loadings and boundary conditions. In the case considered here, the beam model is a Timoshenko beam model, with a rectangular cross-section, having the following dimensions: width 0.012 m, height 0.01 m and a clamped a length of 0.1m. The beam is made with a steel material with density $\rho = 7800 \text{ kg/m}^3$, Young modulus $E = 2.1 \times 10^{11} \text{ N/m}^2$ and Poisson coefficient 0.3. One side of the beam is fixed, the other one is subjected to a concentrated dynamic load equal to $\mathbf{f}(t) = 100 \times t \times e^{-1.1t}$ at its surface center. Fig. 2 illustrates the 3D model. The 3D model is quadratically meshed with approximately one thousands nodes. The switch instant is $t_s = 1.5 \text{ s}$, at which the beam simulation is switched to the 3D model, with the same boundary conditions and loading. The 3D solution after switching is compared to a reference solution, which is a 3D solution obtained on

the same 3D model for a simulation that starts at $t = 0$ and lasts three seconds.

This work is done twice. The switch from the beam model to the 3D model is performed

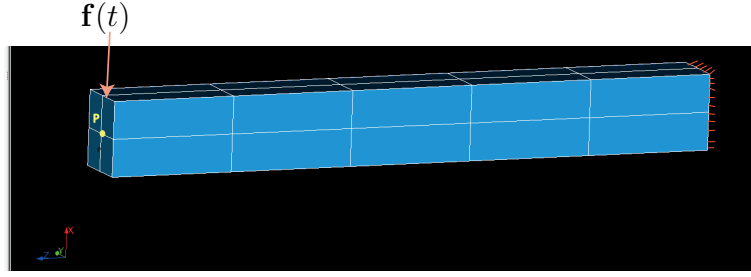


Figure 2: The 3D model under study

first using the approach described in Section 3.1 (static correction only) and second with the initialization built from the 3D displacements computed at three different time steps (see Section 3.2.2).

We compare the displacements, velocities and accelerations of node P where the load is exerted and that belongs to the 3D model as shown on Fig. 2, and the corresponding point that belongs to the beam model.

Fig. 3 shows the displacement results. First we can see a certain difference between the beam solution and the 3D reference solution. This difference is very small, but still remarkable if we make a zoom. Immediately after switching, the 3D solution turns out to be very accurate and is very close to the reference one. Both methods exhibit practically the same precision regarding the displacements.

However, as shown on Fig. 4, which represents a velocity comparison, or Fig. 5 which rep-

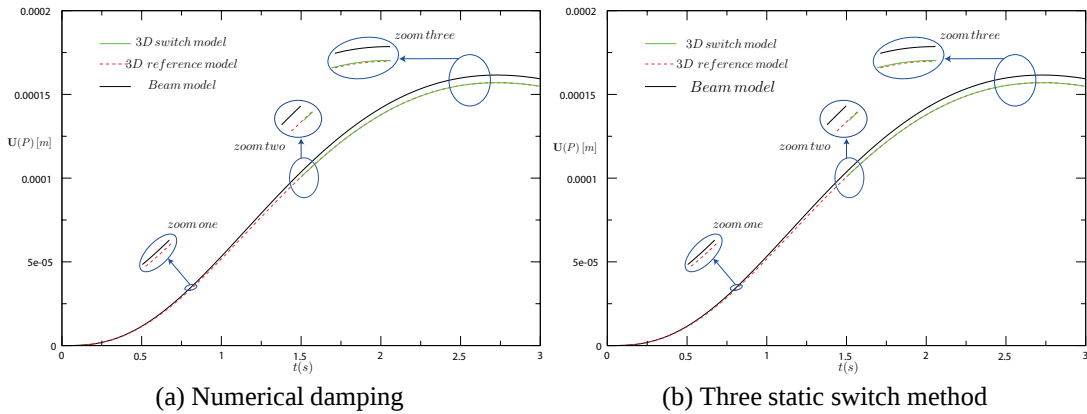


Figure 3: Beam to 3D model switch: displacement analysis

resents an acceleration comparison, immediately after switching, high frequency oscillations with large amplitude occur in the case where there is only the static correction. For a HHT integration scheme with $\alpha = 0.25$, in our case 35 time steps (0.02625 s) were sufficient for the 3D solution to converge to the reference one.

If a triple static switch procedure is performed, the velocities do not present any oscillations; however, very small oscillations occur on the accelerations and vanish shortly after switching. The results show that both methods work, but the triple static switch procedure appears to be more accurate.

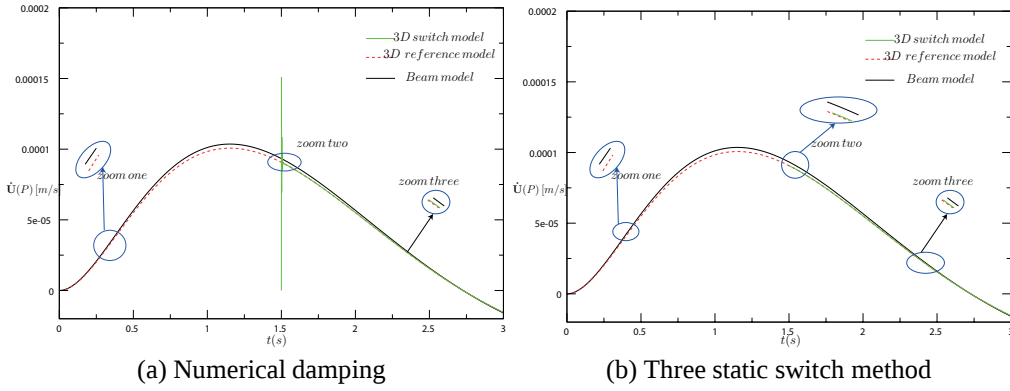


Figure 4: Beam to 3D model switch: velocity analysis

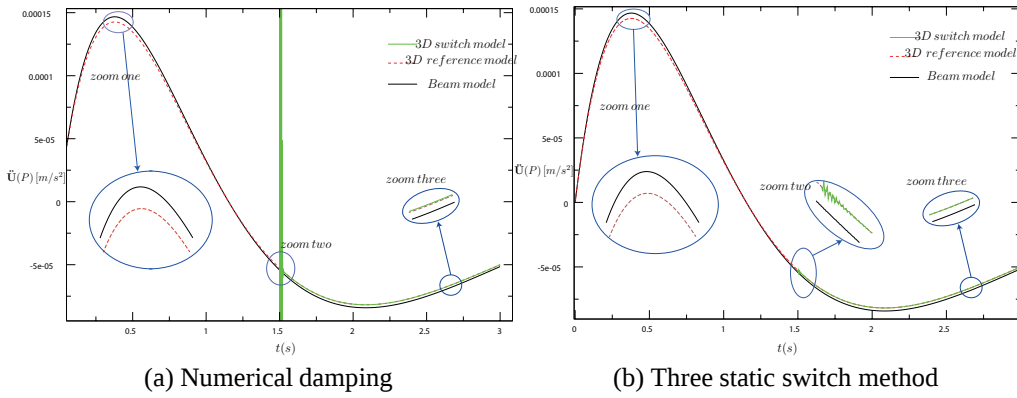


Figure 5: Beam to 3D model switch: acceleration analysis

5 CONCLUSIONS

We have proposed a numerical method that enables to switch from a beam to a 3D model when a 3D description is required only on a small part of space and time domains. This technique enables to save computational time, with a good accuracy. The authors are currently developing this method in the case of rotating machine for application to the slowing down of unbalanced turbine rotors with local interactions and frictions.

ACKNOWLEDGMENTS

The authors thank the French National Research Agency (ANR) in the frame of its Technological Research COSINUS program. (IRINA, project ANR 09 COSI 008 01 IRINA)

REFERENCES

- [1] P. Avery and C. Farhat. The feti family of domain decomposition methods for inequality-constrained quadratic programming: Application to contact problems with conforming and nonconforming interfaces. *Computer Methods in Applied Mechanics and Engineering*, 198:1673–1683, 2009.
- [2] P. Avery, G. Rebel, M. Lesoinne, and C. Farhat. A numerically scalable dual-primal substructuring method for the solution of contact problems part i: the frictionless case.

- Computer Methods in Applied Mechanics and Engineering*, 193:2403–2426, 2004.
- [3] L. Champaney. A domain decomposition method for studying the effects of missing fasteners on the behavior of structural assemblies with contact and friction. *Computer Methods in Applied Mechanical Engineering*, 208:121–129, 2012.
 - [4] C. Farhat and J. Li. An iterative domain decomposition method for the solution of a class of indefinite problems in computational structural dynamics. *Applied Numerical Mathematics*, 54:150–166, 2005.
 - [5] C. Farhat, J. Mandel, and F.-X. Roux. Optimal convergence properties of the feti domain decomposition method. *Computer method in Applied Mechanics and Engineering*, 115:365–385, 1994.
 - [6] C. Farhat, K. Pierson, and M. Lesoinne. The second generation feti methods and their application to the parallel solution of large-scale linear and geometrically non-linear structural analysis problems. *Computer Methods in Applied Mechanics and Engineering*, 184:333–374, 2000.
 - [7] V. Faucher and A. Combescure. A time and space mortar method for coupling linear modal subdomains and non-linear subdomains in explicit structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 192:509–533, 2003.
 - [8] P. Gosselet and C. Rey. Non-overlapping domain decomposition methods in structural mechanics. *Archives of Computational Methods in Engineering*, 11:1–50, 2005.
 - [9] P. Kettil and N.-E. Wiberg. Application of 3d solid modeling and simulation programs to a bridge structure. *Engineering with Computers*, 18:160–169, 2002.
 - [10] M. Kimura, H. Momura, M. Mimura, H. Miyoshi, T. Takaishi, and D. Ueyama. Adaptive mesh finite element method for pattern dynamics in reaction-diffusion systems. In *Proceedings of the Czech-Japanese Seminar in Applied Mathematics, Kuju Training Center, Oita, Japan*, 2005.
 - [11] X. Li and N.-E. Wiberg. Implementation and adaptivity of a space-time finite element method for structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 156:211–229, 1998.
 - [12] O. Lloberas-Valls, D.J. Rixen, A. Simone, and L.J. Sluys. Domain decomposition techniques for the efficient modeling of brittle heterogeneous materials. *Computer Methods in Applied Mechanical Engineering*, 200:1577–1590, 2011.
 - [13] N. Mahjoubi, A. Gravouil, and A. Combescure. Coupling subdomains with heterogeneous and incompatible time steps. *Computational Mechanics*, 44:825–843, 2009.
 - [14] L. Noels, L. Stainier, and J.-P. Ponthot. Combined implicit/explicit algorithms for crashworthiness analysis. *International Journal of Impact Engineering*, 30:1161–1177, 2004.
 - [15] L. Noels, L. Stainier, and J.-P. Ponthot. Combined implicit/explicit time-integration algorithms for the numerical simulation of sheet metal forming. *Journal of Computational and Applied Mathematics*, 168:331–339, 2004.

- [16] K. C. Park, S. J. Lim, and H. Huh. A method for computation of discontinuous wave propagation in heterogeneous solids: basic algorithm description and application to one-dimensional problems. *international Journal for Numerical Methods in Engineering*, 4285, 2012.
- [17] A. Plaza, M. Padron, and G. Carey. A 3d refinement/derefinement algorithm for solving evolution problems. *Applied Numerical Mathematics*, 32:401–418, 2000.
- [18] D. Rixen. Multi-body dynamics: time integration, 2002.
- [19] S. Roques. *Modlisation du comportement dynamique couple rotor-stator d’une turbine en situation accidentelle*. PhD thesis, Ecole Centrale de Nantes, 2007.
- [20] S. Roques, M. Legrand, P. Cartraud, C. Stoisser, and C. Pierre. Modeling of a rotor speed transient response with a radial rubbing. *Journal of Sound and Vibration*, 329:527–546, 2009.