An Integrated CAD Strategy for Nonlinear Dynamics of 3D Suspended Bridges

Giovanni Formica[1], Franco Milicchio[2], Mauro Murer[3]

[1]University of Roma Tre, formica@uniroma3.it
[2]University of Roma Tre, franco.milicchio@uniroma3.it
[3]University of Pavia, mauro.murer01@universitadipavia.it

Corresponding author: Franco Milicchio, franco.milicchio@uniroma3.it

Abstract. We propose a numerical strategy for predicting the nonlinear response of suspended bridges subject to aerodynamical loading conditions, directly implemented in the common CAD scripting language Python, without interfacing with any external tools, and tested on Autodesk Maya. This work addresses the problem of the simultaneous design and simulation, and in case, health monitoring, of complex structures undergoing highly nonlinear behaviour within times that are comparable with the 3D solid generation and rendering.

The implementation benefits from a theoretical framework which allows us to reduce the fully 3D mechanical problem of the suspended bridge to a beam-like problem accounting for the main nonlinear dynamical features. The same rules governing the theoretical reduction serve as instructions for rendering in 3D data structures the beam response under varying loads.

This code is useful for researchers in the field of nonlinear dynamics of suspended bridges and for practicing engineers seeking automated conceptual design tools. With further extensions, the code could solve sophisticated 3D conceptual design problems in structural engineering, mechanical engineering and architecture practice.

Keywords: suspended bridge, 3D-1D model reduction, nonlinear dynamics fully integrated in CAD
DOI: https://doi.org/10.14733/cadaps.2019.1046-1062

1 Introduction

CAD/E platforms gained nowadays an important role for various engineering applications, as the increasing number of research in the last two decades shows. As early works, in [1] the advantages of integrated CAD systems within a structural engineering context have been explored, identifying several aspects of integration to be targeted, such as 2-D drafting and 3-D modelling, graphical and nongraphical design information, CAD data structures and their user interface, and, more in general, drafting functions with other engineering applications.

More recently, researchers dealt with the link between CAD and CAE models, investigating the robustness and efficiency of both modeling representations and algorithms. In particular, we mention [7] where a high level topology based on mixed shape representation has been put forward to support B-Rep and polyhedral models simultaneously. In [6] the authors introduced a common data model able to implement CAD/CAE integration, and containing parameters required for both CAD modeling and CAE analysis. In [17] improved optimization algorithms are associated with the CAD/CAE integration technique, thus obtaining an open and integrated framework for performing structural design optimization within commercial software tools. In the same directions, [19] presented a hybrid software platform bringing various CAD and CAE tools into one design toolkit. Here, the interaction of the user with the 3D objects is performed through direct manipulation, icon-based graphical user interface, and command line, as well as scripting (*e.g.*, Rhinoscript in Rhinoceros, MEL in Maya or MAXScript in 3ds Max). Integrating CAD environments with analysis tools is still a very active field of research, where different models and relationships between representations need to be concurrently handled and updated with robust links [16]. For a more thorough state of the art one can refer to [18], where a unified representation architecture has been eventually proposed, including unified data management, unified display and post-processor and unified data model.

Especially for complex structures, as suspended bridges, there is an increasing interest to use powerful and friendly CAD/E softwares in order to have at least a preliminary design and parametric sensitivity analysis for safety assessments purposes and health monitoring [14, 10]. This point is emphasized by the several efforts made in the literature to bridge CAD/E with structural analysis softwares through external interface tools. Among these, recent developments can be found in [12] where a model for ship hydrodynamics simulations of hulls has been presented and then implemented in a C++ software; such a software uses the Open CASCADE Community Edition (OCE) open source library to import and query CAD models' mesh generation modules to automatically generate computational grids from the CAD geometry of the hull. In [20] a novel approach, both accurate and efficient, that interactively simulates the physical properties of a CAD model undergoing parametric modifications, and more in general topological changes, has been presented and implemented without expensive re-computations. Such an approach, implemented in MATLAB, is a direct simulation with accuracy control based on an offline computation of a generic solution parametrizing the design modifications; thorough the proper generalized decomposition model reduction technique, in combination with R-functions, linear elastic solution are then cheaply and quickly evaluated online in the CAD model. A Python code for general 3D topology optimization has been proposed in [21]. Within the context of compliance minimization with a fixed volume constraint, the code implements the Bi-directional Evolutionary Structural Optimization method, and then adopting the Abaqus Scripting Interface provides access to ABAQUS advanced tools for finite element analyses. Finally, in order to improve building information modeling (BIM), an integrated platform named SATBIM—targeting not only information modelling, but also structural analyses and visualisation—has been proposed in [15] for the design support of mechanised tunnelling processes. The novelty of such a platform is essentially in the multi-level numerical simulations that enable the modelling on different Levels of Detail (LoD) for each physical component, process information, and analysis type. As concerning the structural analysis, the numerical models for soil-structure interaction in tunnelling use an *ad hoc* Python code, specifically written for these purpose, that connects KRATOS Multiphysics, an open-source platform for FE simulation, with Revit and GiD by means of several Dynamo programs running under Revit.

In the present work, we propose an effective and fully integrated structural analysis within the CAD platforms themselves. The key idea is to *adopt a theoretically coherent reduced model* starting from a 3D complex model such as for suspended bridges.

Reduced models suitable for both accurate and efficient 3D structural analyses that run in different softwares have been investigated in the literature. We mention in particular [8], where an alternate "algebraic reduction" method is proposed to combine the generality of 3D finite element analysis, computationally expensive for thin structures, and the computational efficiency of a 1D beam analysis. This is achieved via a dual-representation framework: the geometry of the thin beams is captured via a 3D finite element mesh, while the physics is

captured via a 1D beam model. Additionally, one dimensional beams are employed in topology [4], where a reduced model allows integrating optimization procedures directly into a CAD environment. In the same context of nonlinear dynamics of slender beams, in [5] a numerical strategy has been developed in MATLAB and directly coupled with a versatile and general-purpose FEM softwares like COMSOL MULTIPHYSICS.

Contribution Our proposed strategy implements the reduced model presented in [3] within a unique Python environment working for both 3D geometric modeling, structural analysis, and graphical rendering of suspension bridges, allowing designers to comfortably generate models and, within times comparable to the model generation itself, getting back the nonlinear dynamics response under varying in time loading conditions. Such strategy has been fully integrated into Autodesk Maya, where not only engineers, but also the entertainment industry may exploit its results for the structural response, as our implementation is based only on Python and may be integrated immediately into other CADs such as 3ds Max, Abaqus, or Open CASCADE SDK.

Outline The manuscript is organized as follows. In Section 2 we introduce the nonlinear one-dimensional beam model, utilized then in Section 3 to solve the reduced problem via a Galerking approximation. The numerical results are finally highlighted in Section 4, where the suspension bridge is analyzed subject to wind velocity.

2 Nonlinear beam model

The One-Dimensional (1D) beam model is obtained as reduction of a Three-Dimensional (3D) model through a geometric procedure which obeys to a series of kinematics assumptions accounting for the nonlinear nature of the bridge strain fields. In particular, we follow the theory of suspension bridges deforming in 3D space described in [2, 3], where the strain is first defined with respect to both reference and deformed 3D configurations, and then reduced to a 1D beam-type space. Kinematic assumptions are finally corroborated by mechanical hypotheses of the linearization of the equations of motion with respect to the deformed beam configuration.

It's worth mentioning that we limit here the description of the loading conditions to the case of aerodynamic loads. Although it's straightforward to introduce in the nonlinear beam model both autonomous forces (*e.g.*, aeroelastic loads) and non-autonomous dynamic forces (*e.g.*, traffic-induced excitations), as widely argued in [9] the dynamic aeroelastic instabilities is one of the key point in analysis and design of suspended bridges. In line with the actual purposes of the present work which addresses to CAD numerical issues, we then give in this Section just a brief description of the nonlinear beam model, remitting to the references here reported.

2.1 Kinematic and geometric assumptions

The reference configuration $\mathcal{B}$ of the deck-girder is represented by the position vector of the base line (*i.e.*, the centerline) described by $\mathbf{r} = x\,\mathbf{e}_1$ as pictured in Figure 1, with $x \in [0, l]$, l being the longitudinal length of the bridge, and by the set of directors $(\mathbf{b}_2, \mathbf{b}_3) = (\mathbf{e}_2, \mathbf{e}_3)$ giving the orientation of the cross section (see Figure 2).
A material point of the deck-girder system in the configuration $\mathcal{B}$ is then represented by

$$\mathbf{x} = \mathbf{r} + \mathbf{s}\,, \tag{1}$$

where $\mathbf{s} = x_2\,\mathbf{b}_2 + x_3\,\mathbf{b}_3$ is the position vector of the material points of the cross section with respect to the base point.

The two suspension cables are supposed to be positioned in the elastic catenary equilibrium state under their own weight, so that each point of the cable corresponds to

$$\mathbf{r}^{\pm} = \mathbf{x}^{\pm} + h(x)\,\mathbf{e}_2\,, \tag{2}$$

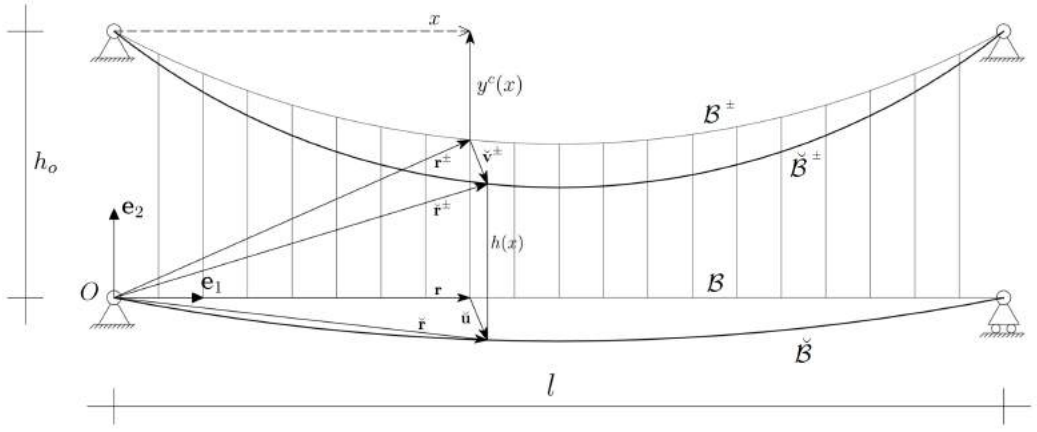

Figure 1: Inertial reference frame (O, $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$) and the reference configuration $\mathcal{B}$. The position vectors of the suspension cables (catenary configurations) and the hanger attachment points to the deck are described by $\mathbf{r}^{\pm}$ and $\mathbf{x}^{\pm}$, respectively; the position vector of the base line of the deckgirder is described by $\mathbf{r}$. All quantities with superscript $(\breve{\bullet})$ refer to the deformed configuration.

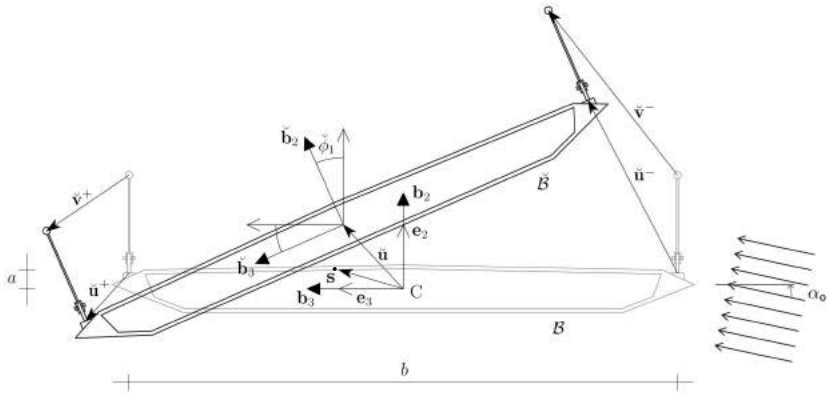

Figure 2: Planar model of the suspension bridge: reference configuration $\mathcal{B}$, equilibrium configuration $\mathcal{B}^{\circ}$, under dead loads, and actual configuration $\breve{\mathcal{B}}$ due to live loads. All quantities with superscript $(\breve{\bullet})$ refer to the deformed configuration. Arrows on the right depict the wind direction, generically oriented with an initial angle of attack α_o with respect to the axis $\mathbf{e}_3$.

where $\mathbf{x}^{\pm} = \mathbf{r} \pm b/2\,\mathbf{e}_3 + a\,\mathbf{e}_2$ is the position of the two systems of vertical hangers (distinguished as $\pm$) anchored to the deck, b and a being the distance between the hangers and the vertical distance between the center of mass of the deck-girder cross section and the upper (structural) plane of the deck, respectively. In (2), $h(x)$ is a continuous function that describes the undeformed length of the hangers according to $h(x) = h_o + y_c(x)$, with h_o being the height of the towers above the deck, and $y_c(x)$ the function satisfying the catenary equilibrium:

$$y_c(x) = \frac{H_c}{m_c\, g}\left(\cosh\left(\frac{m_c g l}{H_c}\left(\frac{1}{2} - \frac{x}{l}\right)\right) - \cosh\left(\frac{1}{2}\frac{m_c g l}{H_c}\right)\right), \tag{3}$$

H_c being the horizontal component of the tension acting in each cable, whose mass is m_c. The interested reader can find in [9] all the details to formally obtain the above equation.

2.2 Equations of motion

The suspension bridge problem is obtained by linearizing the set of reduced (1D) equations that govern the nonlinear (3D) motion of the structure. In particular, we assume the inextensibility of hangers, as they don't appreciably affect the bridge response due to their significant elasto-geometric stiffness with respect to those of the whole bridge (see [9, 3]). The introduced constraint leads to treat the hanger tensions as reactive forces in the equations of motion. We further assume the bridge to be unshearable, thus neglecting the shear stiffness terms referred to the deck-girder.

Then, the governing equations are cast in non dimensional form by using the deck span l as characteristic length and $1/\omega_b = \sqrt{\rho A\, l^4/EJ_3}$ as characteristic time, where ρA is the deck-girder mass per unit length, and EJ_3 is its bending stiffness about the local axis $\mathbf{b}_3$. The other non dimensional variables and parameters, concerning both inertial and stiffness terms, are listed below:

$$\iota = \sqrt{\rho J_1/(\rho A l^2)}\,, \qquad \epsilon_c = (\rho A)_c/\rho A\,, \qquad \delta_c = H_c\, l^2/EJ_3\,, \tag{4a}$$

$$\kappa_c = (EA)_c\, l^3/(EJ_3\, l_c)\,, \qquad \kappa_h = (EA)_h\, l^2/EJ_3\,, \qquad \kappa = EA\, l^2/EJ_3\,, \tag{4b}$$

$$\gamma = GJ_1/EJ_3\,, \qquad \beta = EJ_2/EJ_3\,, \tag{4c}$$

where $(EA)_c$, $(EA)_h$, and EA denote the axial stiffness of the suspension cables, hangers, and deck-girder, respectively; GJ_1 is the torsional stiffness about $\mathbf{b}_1$; finally, $(\rho A)_c = m_c/l_c$ is the cable mass per unit length, denoted by l_c. The non dimensional damping coefficients used are:

$$\bar{c}_1 = \omega_b c_1\, l^2/(EJ_3)\,, \quad \bar{c}_j = \omega_b c_j\, l^4/(EJ_3)\,, j = 2, 3\,, \tag{5}$$

c_1 being the damping coefficient of the torsional motion, c_2 and c_3 the damping coefficients of the flexural motion along $\mathbf{e}_2$ and $\mathbf{e}_3$ respectively, according to the following expression [3]: $c_i = 2\,\zeta_i\,\omega_i\,m_i$, for $i = 1, 2, 3$, with ζ_i being the damping ratio along the ith axis direction.

Consequently, from now on all the geometric quantities (namely x, y_c, b), and the displacement field u_j will be considered as divided by l, while the time variable t multiplied by ω_b, so that the non dimensional equations for span-wise uniform bridges turn out to be

$$\begin{aligned}
&\left(\iota^2 + b^2/2\,\epsilon_c(x)\right)\,\ddot{\phi}_1 + \bar{c}_1\dot{\phi}_1 - \gamma\phi_1'' - b^2\delta_c\phi_1'' - b^2\kappa_c y_c'' \int_0^1 y_c'\phi_1'\,\mathrm{d}x - \lambda_1 = 0\,,\\
&(1 + 2\,\epsilon_c(x))\,\ddot{u}_2 + \bar{c}_2\dot{u}_2 + u_2'''' - 2\delta_c u_2'' - 2\kappa_c y_c'' \int_0^1 y_c' u_2'\,\mathrm{d}x - \lambda_2 = 0\,,\\
&\ddot{u}_3 + \bar{c}_3\dot{u}_3 + \beta u_3'''' - \lambda_3 = 0\,.
\end{aligned} \tag{6}$$

The overdot $\dot{f}$ denotes differentiation with respect to non dimensional time, *i.e.*, $\dot{f} = {}^1\!/_{\omega_b}(\partial_t f)$, whereas the prime f' indicates differentiation with respect to the non dimensional space coordinate, *i.e.*, $f' = l(\partial_x f)$, for any function $f(x,t)$. The space and time-dependent external forces are represented by λ_1, λ_2 and λ_3, bridge weight included.

2.3 Aerodynamic loads

The aerodynamic loads, obtained from the nonlinear quasi-steady formulation, allow the system of governing equations to be integrated in the time domain. Following [3], the aerodynamic loads are represented by

the forces $F_2(x,t)$ and $F_3(x,t)$ and moment $M(x,t)$ per unit reference length expressed in the fixed frame $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ as

$$\begin{aligned} F_2(x,t) &= L(x,t)\cos(\alpha_{\mathsf{w}}) - D(x,t)\sin(\alpha_{\mathsf{w}})\,, \\ F_3(x,t) &= L(x,t)\sin(\alpha_{\mathsf{w}}) - D(x,t)\cos(\alpha_{\mathsf{w}})\,. \end{aligned} \tag{7}$$

Here $F_2(x,t)$ is the projection along the reference axis $\mathbf{e}_2$ of the aerodynamic non dimensional lift $L(x,t)$ and drag $D(x,t)$ while $F_3(x,t)$ is the projection along the reference axis $\mathbf{e}_3$. The aerodynamic forces depend on the effective angle of attack α_{e} expressed as:

$$\begin{aligned} \alpha_{\mathsf{e}}(x,t) &= \phi_1(x,t) + \alpha_{\mathsf{w}}(x,t) + \alpha_{\mathsf{o}}\,, \\ \alpha_{\mathsf{w}}(x,t) &= \frac{\dot{u}_2(x,t) + \frac{b}{4}\dot{\phi}_1(x,t)}{\bar{U} + \dot{u}_3(x,t)}\,, \end{aligned} \tag{8}$$

where $\bar{U}$ is the non dimensional wind speed defined as $\bar{U} = U_{\mathsf{w}}/(l\,\omega_b)$, neglecting $\dot{u}_3(x,t)$, while α_{o} is the initial wind angle of attack. The function $\alpha_{\mathsf{w}}(x,t)$ in (8) represents the instantaneous angle between the direction of the free-stream speed U_{w} and that of the relative velocity of the bridge with respect to the flow. Thus, the nonlinear expressions of the aerodynamic loads read:

$$\begin{aligned} D(x,t) &= P\; b\; C_D(\alpha_{\mathsf{e}})\,, \\ L(x,t) &= P\; b\; C_L(\alpha_{\mathsf{e}})\,, \\ M(x,t) &= P\; b^2\; C_M(\alpha_{\mathsf{e}})\,, \end{aligned} \tag{9}$$

where $P = \frac{1}{2}\frac{\rho_a\, U^2}{\rho A\, \omega_b^2}$ is the non dimensional dynamic pressure, $\rho_a = 1.225\ kg/m^3$ being the air density. The nonlinear form of the static aerodynamic coefficient curves was obtained as

$$\begin{aligned} C_D(\alpha_{\mathsf{e}}) &= 0.0695 + 0.1332\ \alpha_{\mathsf{e}} + 4.7033\ \alpha_{\mathsf{e}}^2 - 1.1667\ \alpha_{\mathsf{e}}^3 \\ &\quad - 24.1297\ \alpha_{\mathsf{e}}^4 \\ C_L(\alpha_{\mathsf{e}}) &= 0.0496 + 6.3567\ \alpha_{\mathsf{e}} - 2.5893\ \alpha_{\mathsf{e}}^2 - 109.3908\ \alpha_{\mathsf{e}}^3 \\ &\quad - 2.6756\ \alpha_{\mathsf{e}}^4 + 786.1796\ \alpha_{\mathsf{e}}^5 \\ C_L(\alpha_{\mathsf{e}}) &= 0.0330 + 1.1379\ \alpha_{\mathsf{e}} - 0.7673\ \alpha_{\mathsf{e}}^2 - 27.0033\ \alpha_{\mathsf{e}}^3 \\ &\quad + 4.5606\ \alpha_{\mathsf{e}}^4 + 249.0324\ \alpha_{\mathsf{e}}^5\,. \end{aligned} \tag{10}$$

Note that the aerodynamic forces are modelled by including the frequency-dependent unsteady aerodynamic characteristics—nonlinear functions of the effective angle of incidence—and are based on a nonlinear functional form accounting for viscous flow, thickness effect, and large flow separation at varying angles of incidence. To obtain the corresponding coefficients of such forces, CFD (Computational Fluid Dynamics) programs are needed: the present values are deduced from the wide numerical campaign conducted in [2], where the reader can find all the specific details. More specifically, the computations are performed by two of the most used CFD commercial softwares, *i.e.*, FLUENT by ANSYS, and DVMFLOW by COWI, which in [2] are validated and compared each with other.

The aerostatic coefficients associated with lift, drag, and aerodynamic moment exhibit a nonlinear behavior even at moderately large angles of attack. The best polynomial fitting for the aerostatic coefficients associated with drag, lift and aerodynamic moment includes terms up to fifth order. Therefore, the expression of the

space and time-dependent external forces are as follows:

$$\begin{aligned}\lambda_1 &= M(x,t)\,,\\ \lambda_2 &= F_2(x,t) + \frac{(\rho A + 2\,(\rho A)_c)\,g}{\rho A\;\omega_b^2 l}\,,\\ \lambda_3 &= F_3(x,t)\,.\end{aligned} \tag{11}$$

Finally note that one of the key input parameter is U_{w}, which has to be intended either as a fixed velocity or a prescribed function of time for simulation purposes.

2.4 Stress recovery

The following constitutive equations allows us to recover stress distributions in the beam, listed as axial stress components and bending moment's, respectively:

$$N = \kappa\,(\nu - 1)\,, \qquad N^{\pm} = N_c + \kappa_c\,(\nu^{\pm} - 1)\,, \tag{12a}$$

$$M_3 = \mu_3\,, \qquad M_2 = \beta\,\mu_2\,, \qquad M_1 = \gamma\,\mu_1\,, \tag{12b}$$

with ν and $\nu^{\pm}$ being the stretch strains of the deck and of the cables, respectively, while μ_i for $i = 1, 2, 3$ are the curvature strain components. In this context, we are working under the assumptions of an Euler-Bernoulli model, where shear strains are taken identically zero, *i.e.*, we assume the unshearability of the deck-girder. Thus, the shear stress components are neglected in (12), and are obtained from the corresponding equilibrium equations involving the x-derivatives of bending moments. Such an assumption is supported by the fact that the boxed-type cross section and the high flexural compliance of the structure allow to neglect shear deformations in the deck. This point has been widely argued and proved in [9], and it leads to an Euler-Bernoulli beam model, rather than a Timoshenko's one. The present computational framework, although explicitly implemented for the former model, could be easily extended to consider the latter. Computational costs increase moderately, in reason of the fact that the discrete equations of the numerical problem are derived through a reduced order model of a Galerkin approximation, as detailed in Sec. 3.

The strain measures are deduced within a linearization process imposed in a generic deformed configuration $\check{\mathcal{B}}$, $\check{\mathcal{B}}^{\pm}$ (see Figure 1). The linearized rotation matrix is:

$$\mathbf{R} = \begin{bmatrix} 1 & -\phi_3 & \phi_2 \\ \phi_3 & 1 & -\phi_1 \\ -\phi_2 & \phi_1 & 1 \end{bmatrix}\,, \tag{13}$$

while the linearized deck-girder stretch, shear strains, and flexural curvatures are respectively given by:

$$\nu = 1 + \partial_x u_1\,, \tag{14a}$$

$$\eta_2 = -\phi_3 + \partial_x u_2 = 0\,, \qquad \eta_3 = \phi_2 + \partial_x u_3 = 0\,, \tag{14b}$$

$$\mu_1 = \partial_x \phi_1\,, \qquad \mu_2 = \partial_x \phi_2\,, \qquad \mu_3 = \partial_x \phi_3\,. \tag{14c}$$

The linearized flexural rotation angles and flexural curvatures that take into account the unshearability constraint (*i.e.*, $\check{\eta}_2 = 0 = \check{\eta}_3$) read:

$$\phi_2 = -\partial_x u_3\,, \qquad \phi_3 = -\partial_x u_2\,, \tag{15a}$$

$$\mu_2 = -\partial_x^2 u_3\,, \qquad \mu_3 = -\partial_x^2 u_2\,. \tag{15b}$$

Cables' linearized stretches are given by

$$\nu^{\pm} = \nu^{c}\left[1 + \cos^2\theta^{c}\left(\partial_x v_1^{\pm} + \partial_x v_2^{\pm}\tan\theta^{c}\right)\right], \tag{16}$$

while, by linearizing also the kinematics of the hanger attachment points to the deck, the cables' displacements can be expressed as

$$\mathbf{u}^{(1)\pm} = \left(u_1 \pm \frac{b}{2}\phi_2 - a\phi_3\right)\mathbf{e}_1 + \left(u_2 \pm \frac{b}{2}\phi_1\right)\mathbf{e}_2 + (u_3 + a\phi_1)\,\mathbf{e}_3\,. \tag{17}$$

3 Implementation

3.1 Galerkin approximation

We recast the equations of motions (6) in a vectorial form as

$$\boldsymbol{M}(x)\,\ddot{\boldsymbol{w}}(x,\,t) + \boldsymbol{C}\,\dot{\boldsymbol{w}}(x,\,t) + \boldsymbol{s}\Big(\boldsymbol{w}(x,\,t)\Big) = \boldsymbol{\lambda}(x,\,t) \tag{18}$$

where $\boldsymbol{w}(x,\,t) = \Big[\phi_1(x,\,t),\,u_2(x,\,t),\,u_3(x,\,t)\Big]^{\top}$ is the vector collecting the three kinematic fields; $\boldsymbol{M}(x) = \mathrm{diag}\Big[\iota^2 + \frac{b^2}{2}\epsilon_{\mathrm{c}}(x),\,1 + 2\epsilon_{\mathrm{c}}(x),\,1\Big]$ and $\boldsymbol{C} = \mathrm{diag}\Big[\bar{c}_1,\,\bar{c}_2,\,\bar{c}_3\Big]$ are the diagonal matrices of mass and damping, respectively; finally, $\boldsymbol{s}\Big(\boldsymbol{w}(x,\,t)\Big)$ assembles all the remaining nonlinear terms in (6), depending on the kinematics descriptors $w_i(x,\,t)$ and their spatial derivatives.

Within a standard weak formulation, using a Galerkin approximation for a problem reduction in the eigenmode space of n effective eigenmodes, *i.e.* discretizing $\boldsymbol{w}$ as $\boldsymbol{w}(x,\,t) = \boldsymbol{\Xi}(x)\,\mathbf{q}(t)$, with

$$\boldsymbol{\Xi}(x) = \begin{bmatrix} \xi_1(x)\;\cdots\;\xi_n(x) & 0\;\cdots\;0 & 0\;\cdots\;0 \\ 0\;\cdots\;0 & \xi_1(x)\;\cdots\;\xi_n(x) & 0\;\cdots\;0 \\ 0\;\cdots\;0 & 0\;\cdots\;0 & \xi_1(x)\;\cdots\;\xi_n(x) \end{bmatrix},$$

and considering $\xi_k(x) = \sin(k\,\pi x)$ for $k = 1,\,2,\,\ldots,\,n$, the resulting discrete form of the equation of motion (18), turns out to be

$$\mathbf{M}\,\ddot{\mathbf{q}}(t) + \mathbf{C}\,\dot{\mathbf{q}}(t) + \mathbf{s}\Big(\mathbf{q}(t)\Big) = \mathbf{l}(t)\,, \tag{19}$$

which correspond to a standard modal separation.

The equation (19) is then solved by a Runge-Kutta scheme, suitably rearranging them as ODEs as follows:

$$\begin{aligned} \dot{\mathbf{z}}_1(t) &= \mathbf{z}_2(t) \\ \dot{\mathbf{z}}_2(t) &= \mathbf{M}^{-1}\,\Big(\mathbf{l}(t) - \mathbf{C}\,\mathbf{z}_2(t) - \mathbf{s}\Big(\mathbf{z}_1(t)\Big)\Big)\,. \end{aligned}$$

In addition to kinematics, we also consider modal analysis in our computations. Limiting to the initial conditions of the rest configuration and imposing $\mathbf{q}(t) = \mathbf{v}\,\exp(i\,\omega\,t)$, the eigenvalue problem derived from (19) turns out to be $\omega^2\,\mathbf{M}\,\mathbf{v} = \mathbf{K}\,\mathbf{v}$, with $\mathbf{K}$ being the initial stiffness matrix.

3.2 Up-scaling operators: from 1D to 3D

Since the physical model itself is, in essence, a one-dimensional beam, we need up-scale operators that map geometry and fields from $\mathbb{R}$ to $\mathbb{R}^3$. In the following we will refer then to points $x \in \mathbb{R}$ of the one-dimensional beam problem, discretized into n points x_i with $i = 1, \ldots, 2, \ldots, n$, along the beam axis.

The discretized beam axis then defines a *graph*, or a one dimensional *cell complex* [11, 13], upon which, embedded in $\mathbb{R}^3$, we construct the whole geometrical model of the bridge, as well as mapping physical fields from the domain to the three dimensional euclidean space.

Deck cross sections and bridge cables are generated by a map $c : x_i \mapsto \mathcal{S}_i \times \mathcal{C}_i$, where $\mathcal{S}_i$ is the polygonal cell complex modeling the deck cross section, and $\mathcal{C}_i$ represents the cable cross section—usually assumed as S^1, *i.e.*, a circle. By extruding $\mathcal{S}_i$, we reconstruct the whole bridge deck, and similarly, the extrusion of $\mathcal{C}_i$ produces the cable geometry. Additionally, hangers $\mathcal{H}_i$ are considered separately by extrusion of a hanger cross-section along the normal to the deck surface, and connected to both the cable and deck; we remind the reader that hangers are homogenized in the physical model, and rigidly attached to the deck (cf. Section 2.1). Hence, the union of cables, hangers, and deck sections yields the full geometric representation of the bridge (see Figure 3).

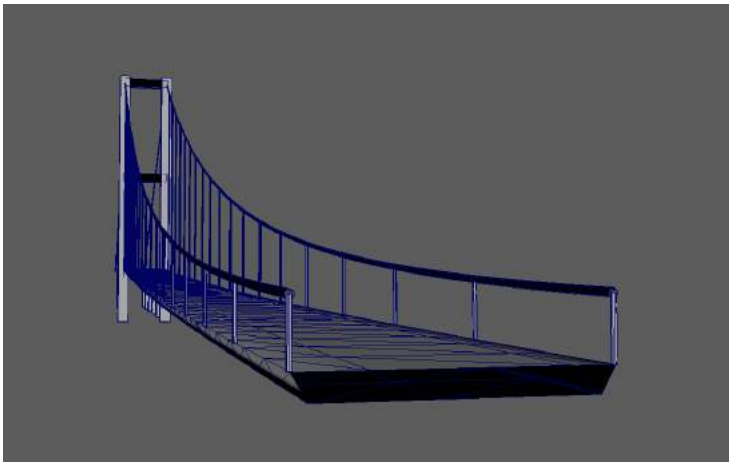

Figure 3: Vizualization of the cross-section during rendering procedure.

The *deformed* shape of the bridge, retrieved upon solving the mechanical problem for each time step, is therefore obtained by applying translation $\boldsymbol{u}$ and rotation $\boldsymbol{\phi}$ tensor field maps collected in the solution field vector $\boldsymbol{w}(x, t)$ to $\mathcal{S}_i \times \mathcal{C}_i$.

4 Numerical results

The entire suspension bridge is parametrically generated starting from few data, *i.e.* l and b, deck span and width, r_{c} and r_{h}, cable and hanger radius, respectively. For simplicity's sake, the generation of the bridge cross-section presently comes from a fixed shape where its thickness and height are obtained directly from b (see Figure 2), however, the generalization to custom polygonal shapes is trivial. Towers and side parts of the bridges are generated separately, since they are not actually involved in the mechanical simulation.

On the other hand, all the mechanical properties are derived from m_{c} and m_{d}, cable and deck masses, and from E_{c} and E_{d}, cable and deck Young modulus, respectively. Finally, the aerodynamic loading input conditions are the wind free-stream speed U_w, and the initial wind angle of attack α_{o}; self-weight loads are directly taken from the geometrical information.

In the following figures we present some meaningful output we can obtained from the dynamical analysis of the bridge, running in Autodesk Maya. As test case, we considered the *Runyang bridge*, the longest Chinese suspension bridge with a span of 1490 meters, a deck width of about 36 meters, depth of 3 meters,

tower's height of 210 meters, and distance between hangers of 16 meters.The mechanical parameters used are: $m_c = 3817\,\text{kg/m}$, $m_d = 18387\,\text{kg/m}$, $E_c = 2.0\,\text{GPa}$ and $E_d = 2.1\,\text{GPa}$.

Validation tests All the numerical results are in agreement with more accurate analyses using the beam model that the present work implemented. Our implementation has been validated by an extensive campaign, and we report here the key results. More details on the capability of the beam model to numerically capture the nonlinear dynamical response of suspended bridges can be found in [9, 3].

In order to show the accuracy of our numerical scheme we depict in Figure 4 the mid-point deflection varying in time for the bridge-beam subject to a wind velocity $U_w = 70\,\text{m/s}$, corresponding to the flutter velocity (see [3]), and with a zero initial angle of attack α_o. Our simulation is compared with the one obtained by the model proposed in [3], here taken as reference, showing a good agreement between the two.

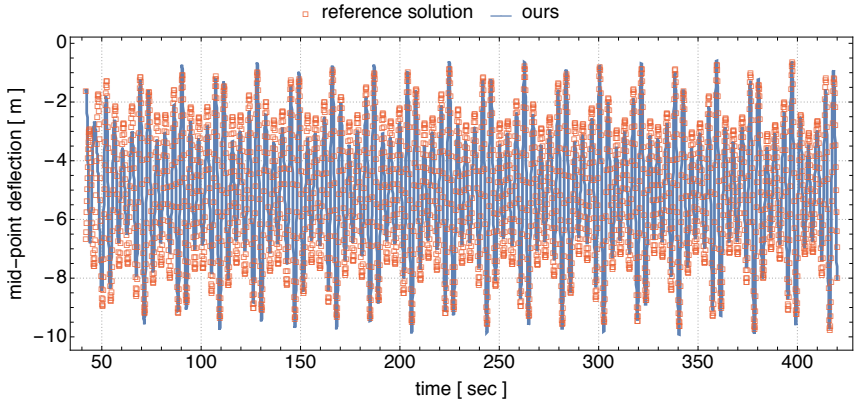

Figure 4: Mid-point deflection varying in time for wind velocity $U_w = 70\,\text{m/s}$. Comparison between reference simulations in [3] and ours.

By repeating several times such simulation for different values of wind velocity, we report in Figure 5 and in Figure 6 the maximum deflection and the maximum twist angle computed in absolute value during the time evolution, respectively. Again, a good agreement is proved with the reference solution in [3].

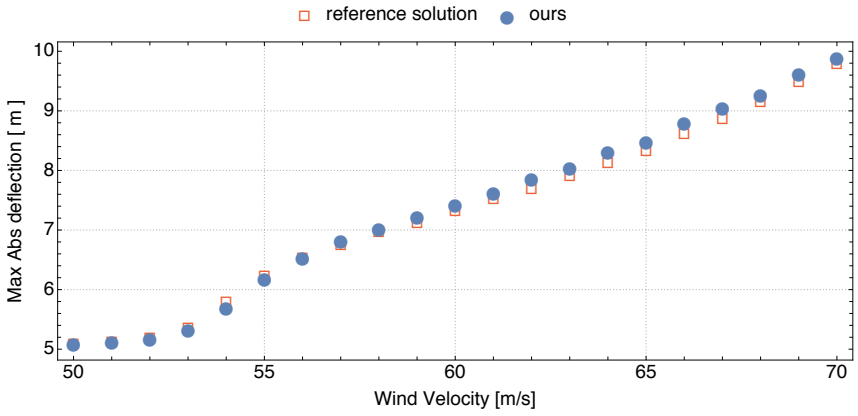

Figure 5: Maximum bridge deflection in absolute value for different values of wind velocity. Comparison between reference solution in [3] and our simulations.

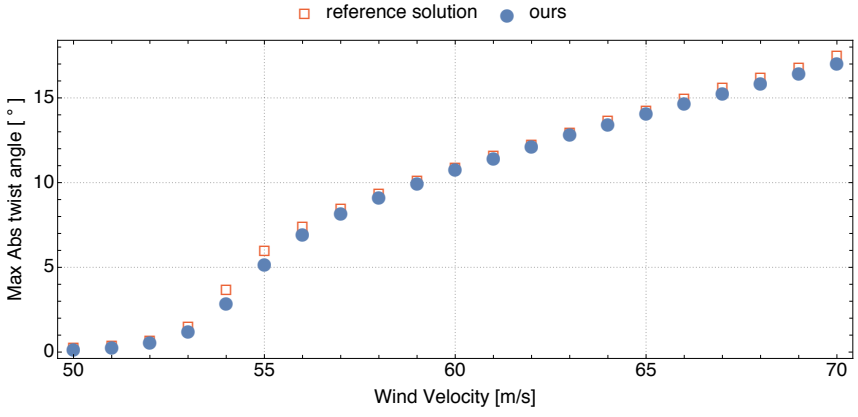

Figure 6: Maximum bridge twist-angle in absolute value for different values of wind velocity. Comparison between reference solution in [3] and our simulations.

Mechanical tests We report here two numerical tests relative to two different wind loading conditions. The first one is a Heaviside step function multiplied by the flutter velocity $U_w = 70\,\mathrm{m/s}$, with the objective of appreciating the damping capacities of the structure. The second one is an oscillating function with amplitude equal to the flutter velocity, so as to explore resonance conditions.

Figure 7 shows for the first test loading history, resulting deflection and aeroelastic force. The wind velocity is zeroed after $t = 200\ s$. Starting from this time, oscillations reduce in a damping phase elapsed in about $100\ s$. The results referred to the second test are summarized by Figure 8. After a smooth increasing, the wind velocity is set to a sinusoidal function with a frequency equal to the first vibration mode of the structure, *i.e.*, 0.092 Hz circa. Due to the damping capacity, the structure does not exhibit increasing vertical displacement amplitudes, although it works closely to the resonance. As well known, such a type of analyses are useful to investigate dynamic instability, by considering wider ranges of forcing amplitudes.

In Figure 9 we show few frames extracted from the bridge live animation while the flutter analysis runs In the first frame the structure is in the initial deformed configuration under self-weight loads. Then, a velocity

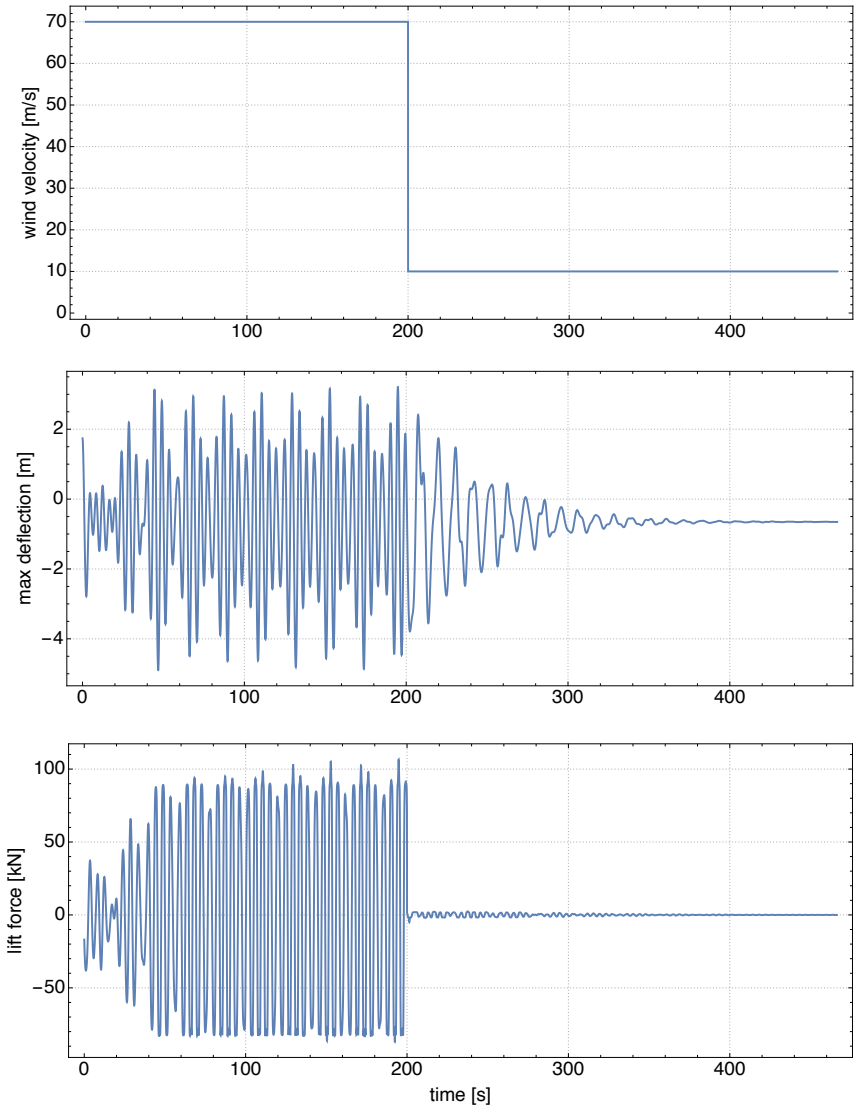

Figure 7: Dynamic behaviour of suspension bridge subject to constant force equal to flutter speed. From up to down, wind speed imposed history, maximum deflection, aeroelastic force.

field equal to *flutter conditions* is imposed. Hence, the whole bridge starts oscillating, and then, in the last frame we can clearly see vertical and torsional displacements of the cross section of the deck.

We finally mention the capabilities of the code to appreciate the beam stress component distributions field by color maps directly done on the rendered deforming structure, as reported in Figure 10.

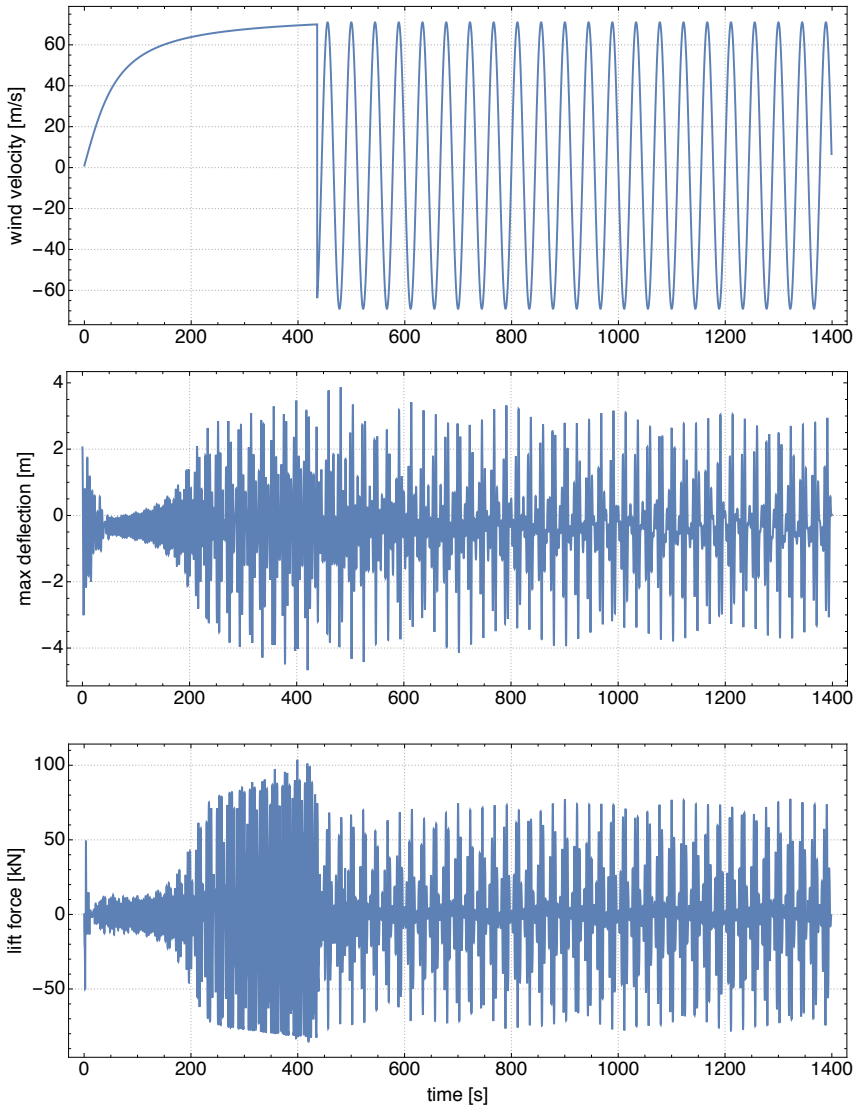

Figure 8: Dynamic behaviour of suspension bridge subject to oscillating force about the flutter speed. From up to down, wind speed imposed history, maximum deflection, aeroelastic force.

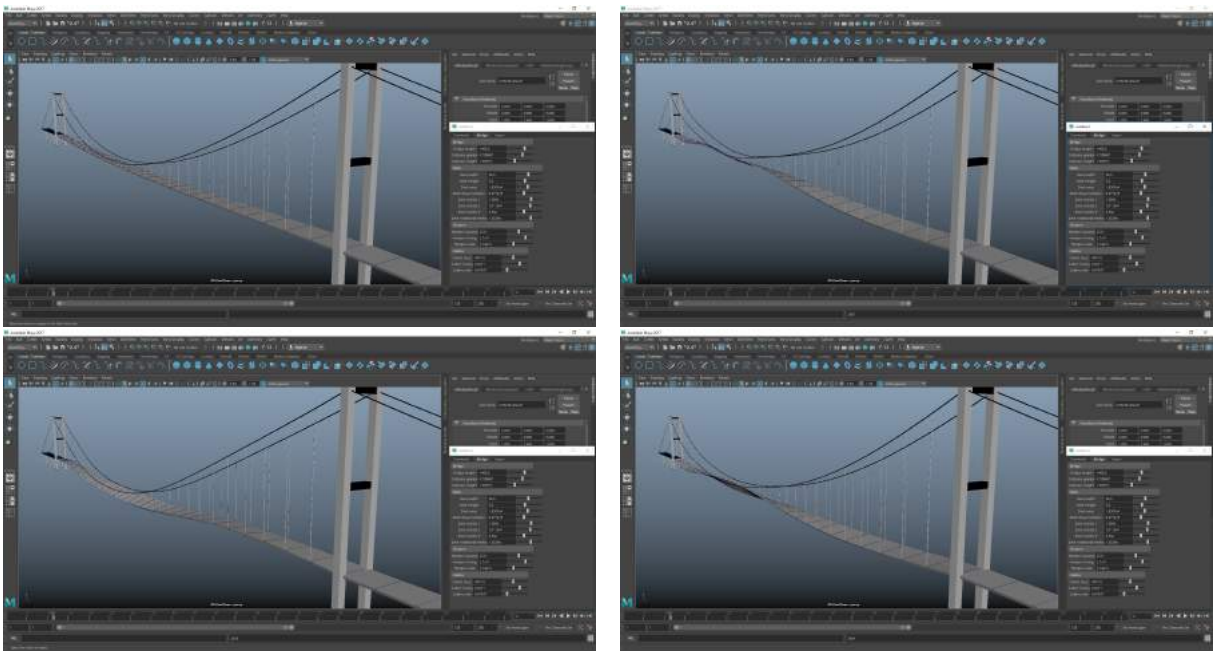

Figure 9: Frames extracted from the flutter analysis completely integrated in Maya.

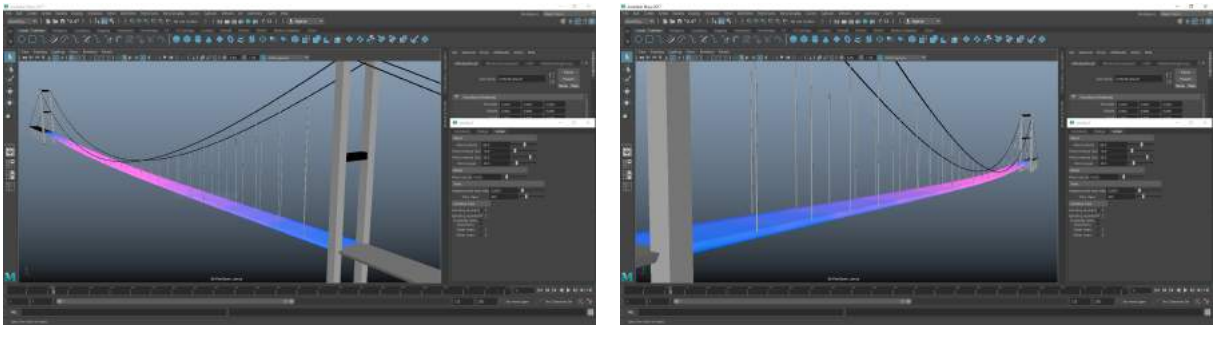

Figure 10: Color map of the bending moments plotted along the suspension bridge's deck.

Benchmarks In Figure 11 we report the comparison between the Autodesk Maya rendering time and our numerical solver's. As one can easily appreciate, per each time step the mechanical solution is obtained in roughly half the time employed in rendering. Moreover, as expected, both times decrease as the mesh size becomes coarser. We remind the reader that, in our context, the mesh size has to be intended as non-dimensional, equal to $1/n$, n being the number of 0-cells as introduced in Section 3.2.

A more interesting result is depicted in Figure 12, where the scalability of the numerical solver is investigated. Here, the average mechanical analysis time is plotted against the total number of simulated time steps, showing a linear scalability. Such a feature proves to be not dependent on the mesh sizes.

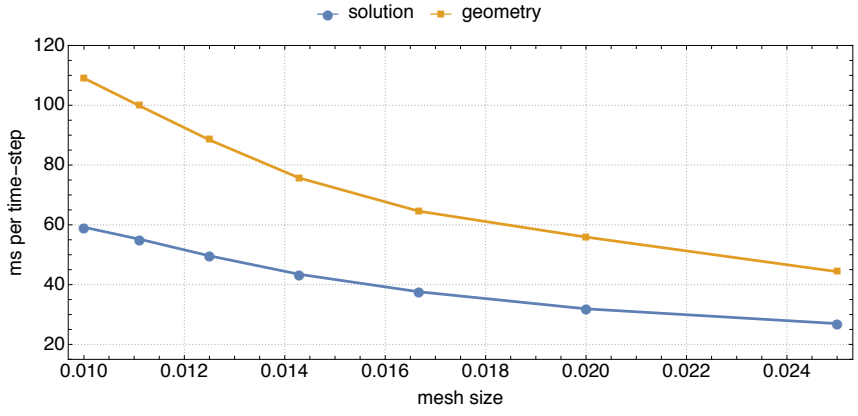

Figure 11: Average CPU times in milliseconds (ms) per each time step of the numerical structural response solver (in blue), compared with geometrical rendering times (in yellow), plotted against the non-dimensional mesh size.

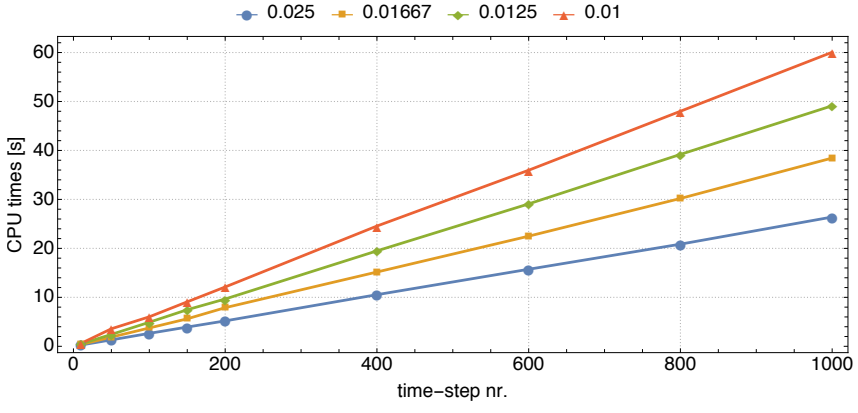

Figure 12: Scalability of the numerical structural response solver: the average time in seconds is plotted against the number of time steps, with respect to non-dimensional mesh sizes of $0.025, 0.01667, 0.0125$, and 0.01 (blue, yellow, green, and red, respectively).

5 Concluding remarks

A numerical dynamic solver fully embedded in Autodesk Maya has been proposed. Such a solver accurately predicts the nonlinear response of suspended bridges subject to aerodynamical loads. The suspended bridge is modelled as 1D nonlinear beam, mapped though up-scaling operators to a full 3D structural model, together with the solution kinematic fields of the mechanical problem. This has been obtained according to a reduced model framework, proposed in several papers (*e.g.*, [9]).

The implemented code shows linear scalability with respect to the number of time steps involved in the simulations. Such property is conserved independently of the mesh size that generates both the computational mesh as well as the rendering one. In addition, CPU times employed to solve each analysis time-step are roughly half times employed by Maya to generate the current deformed 3D bridge.

ORCID

Giovanni Formica, http://orcid.org/0000-0002-6410-8083
Franco Milicchio, http://orcid.org/0000-0002-4875-4894
Mauro Murer, http://orcid.org/0000-0001-9013-4966

REFERENCES

[1] Anumba, C.: Functional integration in cad systems. Advances in Engineering Software, 25(2), 103–109, 1996.

[2] Arena, A.: Aeroelasticity of suspension bridges using nonlinear aerodynamics and geometrically exact structural models. http://hdl.handle.net/10805/1731, Sapienza University of Rome, 2012.

[3] Casalotti, A.; Arena, A.; Lacarbonara, W.: Mitigation of post-flutter oscillations in suspension bridges by hysteretic tuned mass dampers. Engineering Structures, 69(Supplement C), 62 – 71, 2014.

[4] Cuillière, J.C.; François, V.; Nana, A.: Automatic construction of structural cad models from 3d topology optimization. Computer-Aided Design and Applications, 15(1), 107–121, 2017.

[5] Formica, G.; Arena, A.; Lacarbonara, W.; Dankowicz, H.: Coupling fem with parameter continuation for analysis of bifurcations of periodic responses in nonlinear structures. Journal of Computational and Nonlinear Dynamics, 8(2), 021013–8, 2012.

[6] Gujarathi, G.; Ma, Y.S.: Parametric cad/cae integration using a common data model. Journal of Manufacturing Systems, 30(3), 118–132, 2011.

[7] Hamri, O.; Léon, J.C.; Giannini, F.; Falcidieno, B.: Software environment for cad/cae integration. Advances in Engineering Software, 41(10), 1211–1222, 2010.

[8] Jorabchi, K.; Danczyk, J.; Suresh, K.: Algebraic reduction of beams for cad-integrated analysis. Computer-Aided Design, 42(9), 808–816, 2010.

[9] Lacarbonara, W.: Nonlinear Structural Mechanics: Theory, Dynamical Phenomena and Modeling. Springer US, 2013.

[10] Madsen, D.A.: Engineering Drawing & Design. Delmar, Clifton Park, NY, 2012. ISBN 1111309574.

[11] Milicchio, F.; Paoluzzi, A.; Bertoli, C.: A visual approach to geometric programming. Computer-Aided Design and Applications, 2(1–4), 2005.

[12] Mola, A.; Heltai, L.; DeSimone, A.: A fully nonlinearpotential model for ship hydrodynamics directly interfaced with cad data structures. In Proceedings of the International Offshore and Polar Engineering Conference, 815–822. International Society of Offshore and Polar Engineers, 2014.

[13] Munkres, J.R.: Elements of algebraic topology. CRC Press, 2018.

[14] Nahm, Y.E.; Ishikawa, H.: A new 3d-cad system for set-based parametric design. International Journal of Advanced Manufacturing Technology, 29(1-2), 137–150, 2006.

[15] Ninić, J.; Koch, C.; Stascheit, J.: An integrated platform for design and numerical analysis of shield tunnelling processes on different levels of detail. Advances in Engineering Software, 112(Supplement C), 165–179, 2017.

[16] Tierney, C.M.; Nolan, D.C.; Robinson, T.T.; Armstrong, C.G.: Managing equivalent representations of design and analysis models. Computer-Aided Design and Applications, 11(2), 193–205, 2013.

[17] Wang, D.; Hu, F.; Ma, Z.; Wu, Z.; Zhang, W.: A cad/cae integrated framework for structural design optimization using sequential approximation optimization. Advances in Engineering Software, 76(Supplement C), 56–68, 2014.

[18] Xia, Z.; Wang, Q.; Wang, Y.; Yu, C.: A cad/cae incorporate software framework using a unified representation architecture. Advances in Engineering Software, 87(Supplement C), 68–85, 2015.

[19] Zboinska, M.A.: Hybrid cad/e platform supporting exploratory architectural design. Computer-Aided Design, 59(Supplement C), 64–84, 2015.

[20] Zhu, L.; Li, M.; Martin, R.R.: Direct simulation for cad models undergoing parametric modifications. Computer-Aided Design, 78(Supplement C), 3–13, 2016.

[21] Zuo, Z.H.; Xie, Y.M.: A simple and compact python code for complex 3d topology optimization. Advances in Engineering Software, 85(Supplement C), 1 –11, 2015.