

A Computational Nonlinear Aeroelastic Framework with Multiple Capabilities

Carlos Eduardo de Souza

carlozesouza@gmail.com,

Roberto Gil Annes da Silva

gil@ita.br, and

Adolfo Gomes Marto

agmarto@gmail.com

Department of Aeronautical Engineering

Technological Institute of Aeronautics, São José dos Campos, SP, Brazil

ABSTRACT

This paper describes a computational aeroelastic framework intended to be capable of running different types of analyzes called from a single code. The environment of aeroelastic simulations requires the ability of dealing with multiple fields, including structural dynamics, unsteady aerodynamics, models coupling, controls, etc. To overcome the use of different softwares for each analysis phase, a single framework capable of performing linear and nonlinear aeroelastic analyses is proposed. The driver code is written in C++, taking advantage of object oriented programming, with extensive use of classes, but being able to use Fortran routines as well. It is designed for continuous growth, in a modular architecture, so that libraries can be compiled together bringing new capabilities. For that, it is necessary only to create interface functions internally. At the moment, an unsteady vortex-lattice method aerodynamics solver and spline coupling functions are implemented inside the code, and a corotational finite element solver is used as an external library. Examples of applicability are given, including structural dynamics, unsteady aerodynamics and nonlinear aeroelastic time response analyses. The present stage of the work is very promising for the expansion of the aeroelastic capabilities, as well as aerodynamics and structural analyses. The addition of further finite element models comprises the next planned goal in this development.

Keywords: Nonlinear aeroelasticity, numerical framework, UVLM

1. INTRODUCTION

Aeroelasticity is historically defined by the relationship among three fields: aerodynamics, aeroelasticity and mechanical vibrations. Its multidisciplinary characteristic requires the use of tools that are essentially different in nature, since they must be dedicated to solve each part of the aeroelastic problem. The present work is related to the development of a numerical tool dedicated to solve nonlinear aeroelastic problems, but able to be used in standalone related analyses.

The specific problem that motivated this study is the nonlinear aeroelastic response of thin wings subject to large displacements in a generic unsteady flight condition. This analyses is sought to be performed with a good computational cost. The problem of nonlinear aeroelasticity of highly flexible aircraft associated to long span wings permits the use of beam models for the structural part of the

problem. Presently, the interest in micro air vehicles (MAVs) turned the attention to the problem of the aeroelasticity of plate models.

Another aspect is the use of more efficient materials, often in the form of fiber-reinforced composite materials. The use of composite materials allows an overall increase in the structural efficiency of an airframe, with a better weight-to-thrust relation and large flexibility with positive safety factors. However, large displacements may compromise the aeroelastic behavior due to modifications in the aerodynamic loading of lifting surfaces, leading to strong non-linear effects such as geometric nonlinearities and flow separation. Furthermore, wings with small thickness bring the importance of using a shell finite element (FE) in the structural part of the problem to adequately model such type of structure, that are not adequately modeled by beam elements.

Different aerodynamic formulations were used in the large displacements problem investigation, including Peter's finite state methods [21] or Navier-Stokes [2] or Euler [18] formulations. The use of unsteady vortex-lattice methods (UVLM) has also been attempted, because of its implementation simplicity and moderate reduction in computational costs.

Large displacements and rotations are characteristics of the flapping wing problem, and the use of UVLM has been shown to be important because of the influence of the wake in the computation of propulsion forces [23] in flight dynamics analyses. Another work [12] applied the UVLM to the study of the aerodynamics of flat wings, rigid or with prescribed dynamic twist (pitching), but still without a flexible structural model. Banerjee [3] investigated the aeroelasticity of MAV with the use of a 3-D, free wake UVLM coupled with a membrane model with displacements calculated using Fourier series. Wings are simulated as membranes supported by a rigid frame in forward flight at a constant angle of attack. Stanford and Beran [20] couple the UVLM with a corotational FE to the investigation of flapping wings frame. The framework showed good results for structural response and aerodynamic forces prediction.

Application of UVLM to very flexible aircraft (VFA) wings is presented in [25] and [17]. The structural model is commonly represented by beam models, like a elastic beam coupled with rigid-body rotations [24] or geometrically-nonlinear composite beam [17]. For the structural modeling, the capability to model composite shells is part of the desired design goals. The use of finite difference approach for shell modeling [26] does not allow a direct modeling of local characteristics of a composite-made wing. The use of a corotational methodology [5, 6, 20] allows the simulation of composite plates wings, where aspect ratio considerations used for formulation of beam elements are not valid.

In [1] a flat plate model was tested in a subsonic wind-tunnel. The experimental setup aimed identifying aeroelastic characteristics of the model and, in particular, the flutter onset. It showed a good correlation between the experimentally identified and predicted flutter mode shape and onset speed. A large displacement LCO was observed in the post flutter behavior or under certain flow perturbations. Further aeroelastic investigation in the direction of large displacements modeling have been motivated by the observed LCO characteristics.

In the continuation of that research work, the "University of Michigan's Nonlinear Membrane Shell Solver" (UM/NLAMs, [6]) was used in the computational structural dynamics (CSD) analysis side of the aeroelastic problem. This solver has been extended to include laminated composite materials. An UVLM formulation has been implemented based on the formulation given in [13] and augmented with a few stall models.

This paper describes the efforts in creating an efficient nonlinear aeroelastic framework dedicated to the investigation of orthotropic material influence in the nonlinear aeroelastic response of very flexible composite wings. Taking in mind such an aeroelastic framework is not only capable of performing nonlinear analyses, but also able to deal with more common linear problems, it is supposed to be used as a research tool but also as a daily working tool for an aeroelasticist.

2. AEROELASTIC FORMULATIONS

The aeroelasticity is discussed here under its general aspects and the main mathematical tools applied to the study of the proposed problem are emphasized. The aeroelasticity relates the inertial, aerodynamic and elastic forces acting on a structure. The Collar's triangle is an efficient way for representing such relations [7], and better identify which are the disciplines that interact at each type of analysis. From the time when the triangle was proposed, many other interactions were identified as a result of the aeronautics development in last decades. Moreover, expansion of the knowledge is constant, being pushed by the need for more efficient aircraft, among other reasons. Later, control techniques were added to the problem, and presently it is part of the problem, in a field called aeroservoelasticity. Figure 1(b) depicts the Collar's triangle.

The aeroelastic equation of motion takes the classical form of

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

where \mathbf{M} , \mathbf{C} and \mathbf{K} are the structural mass, damping and stiffness matrices, \mathbf{q} is the displacement vector including translational and rotational degrees of freedom. The external forces applied to the structural model, given by \mathbf{f} , are obtained from the aerodynamic loading distribution and interpolated to the structural grid points.

The loading distribution is a function of the aerodynamic nodal displacements \mathbf{q}_a and velocities $\dot{\mathbf{q}}_a$, the resulting flow velocity \mathbf{v}_{res} at a panel and the interpolation matrix \mathbf{G}_{as} and the vorticity strength distribution Γ :

$$\mathbf{f} = \mathbf{G}_{as}(\mathbf{q}, \dot{\mathbf{q}}_a) \mathbf{f}_a(\Gamma), \quad (2)$$

and the aerodynamic force vector is an indirect function of Γ , that depends on time, aerodynamic geometry and resulting velocity distribution.

The explicit dependence on the vorticity distribution comes from the vortex lattice formulation, in which the vorticity distribution on the wake is used to calculate the vorticity distribution on the lifting surface, and subsequently, the pressure distribution. The aerodynamic force at each aerodynamic panel 3/4 chord center is obtained from the resultant pressure value and area S_i of each panel i , $\mathbf{f}_{ai} = p_i(\mathbf{q}_a, \Gamma_i, t) S_i$.

Although being composed by structural and aerodynamic operators, the aeroelastic equation of motion is usually solved in the structural space. The aerodynamic forces are brought to the structural space by means of interpolation matrix \mathbf{G}_{as} . On the other hand, the structural displacements are used to update the aerodynamic model. This model coupling dictates the definition of solution strategies, based on the type of operators formulation.

Further manipulations of Eq. 1 can lead to various solutions of the aeroelastic problem. Previous works described the problems and the formulation used for the solution of the time marching problem [10, 11]. The reader addressed to the theoretical manual of the software ZAEROTM [27] for a good overview of solutions to aeroelastic stability problems.

3. FRAMEWORK DESCRIPTION

The formulation is included into an aeroelastic simulation framework composed by a driver code, called ASAero, with the UVLM embedded in it, and coupled to a CSD solver, that is called as an external library. The capability of working as a black box is a requirement established for the code implementation. In this case, the source code of a specific solver does not need to be available, but only compiled as a library. Only a description of input data format is necessary, so that an interface function can be created.

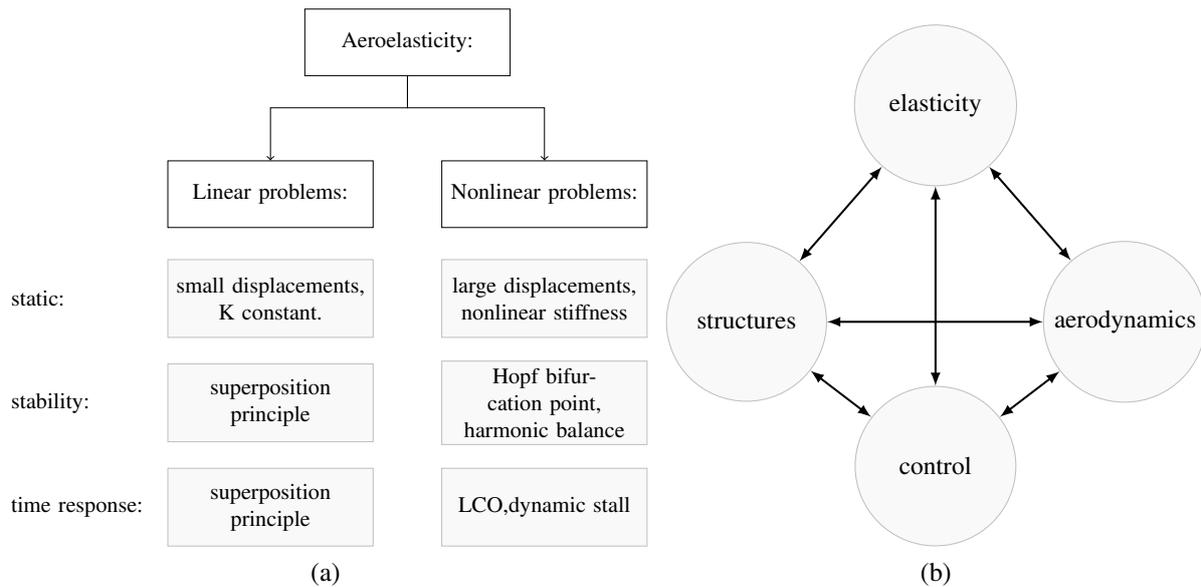


Figure 1: (a) Aeroelasticity and the possible solutions for linear and nonlinear problems. (b) Relationship among fields leading to aeroelasticity and servoaeroelasticity.

Another requirement consists in having a simplified input method, where one file is used for each analysis. To modify an analysis, it is necessary just to modify some control cards inside the analysis file. This simplifies a lot the creation of test matrices, and batch simulation, for example. The model is defined using a single text input file, similar to those used by commercial softwares NASTRANTM [15] or ZAEROTM [27]. In the present case, both the structural and the aerodynamic models are defined, as well as the coupling information and the simulation parameters.

The code is implemented in c++, what allows taking advantage of object orientated programming techniques. The code was written aiming to facilitate the management of information throughout the code, but still being a lean and fast code. Classes and data structures are used to handle input and output data, and, more important, simulation objects. Figure 2 presents an organization table of ASAero, identifying the different type of modules, functions and libraries that are used. Also, it is shown that the code is meant to grow, with space for further analysis functions, that may be use solver implemented internally or using external libraries.

Inside the framework, multiple analysis types can be established. The definition of an aeroelastic analysis procedure depends on the type of solvers for each side on the coupling method choice. In the present case, an explicit coupling method is used, what requires the use of generic coupling functions. As in the case of multiple analyses, the code is thought to have an increasing number of coupling methods implemented.

Figure 3 shows the flowchart of main file of ASAero. It is a very simple code, used for a management of analyses. The goal is to have a framework that increases with time, as new capabilities are inserted. These capabilities are then implemented inside new analysis processes, and can be self contained. If an external library is used to a certain analysis, it is still necessary to have an interface code that interprets the internal variables and send then to the external solver in an appropriate format.

Information handling is important for code efficiency, and in the present case special attention has been given to that. The model is constructed using a single text input file, with command fields that occupy one or more lines with 80 characters. Inside the code, an generic input class is used to read the input file and save inside special command data structures. Another dedicated function translates the command structures to the dedicated internal variables. The LAPACK software package, linked to the code as a library, brings efficiency to the solution of linear systems.

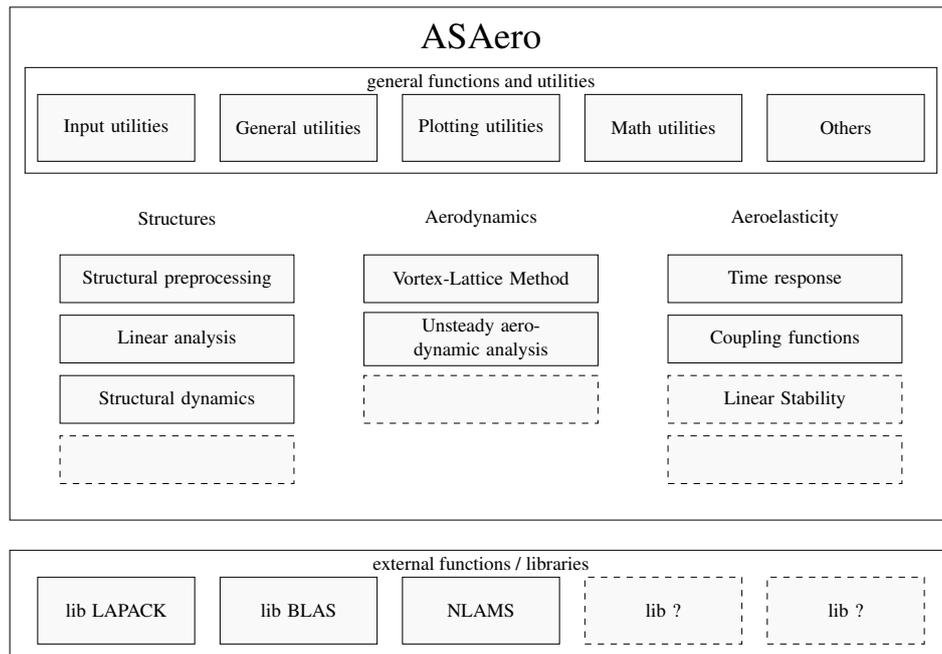


Figure 2: General view of ASAero components, with organization of internal functions and external libraries.

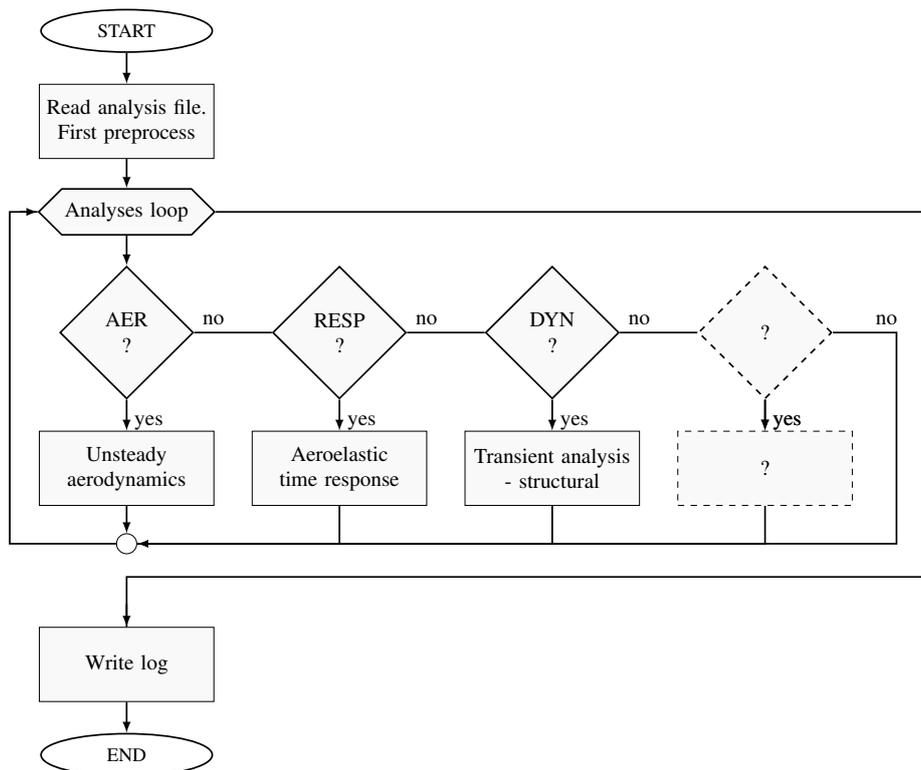


Figure 3: Flowchart of ASAero framework functionality with present capabilities and open boxes for future growth.

3.1. Preprocessing Utilities

An important feature of the present framework is the pre-processing capability. An utility library has been implemented for reading purposes, and converting functions transform the input cards into inter-

nal variables. Functions dedicated to prepare data for the structural analyses that were implemented for different types of elements, and are being improved since the work of [8]. It includes, so far, pre-processing of beam, flat triangular, and flat quadrilateral elements. Definition of element coordinate systems is also performed. Figure 4 shows the definition of nodes and coordinate systems used for triangular and quadrilateral elements. The quadrilateral elements are also used in the aerodynamic formulation.

The utility functions are also able of preprocessing laminated composites models where every layer is defined with different orientations and thickness. Multiple material types (equivalent to NASTRAN's MAT1 isotropic material and MAT8 orthotropic material) can be defined in the model. In orthotropic materials, the direction aligned with the largest elastic modulus is defined as being the material main direction. A laminate consists in combination of multiple layers stacked on top of each other, as seen in 5. These layers can be made of a single material or different ones. To take advantage of unidirectional composites properties, the best practice is to use layers with different orientations and thickness, what is known as the laminate stacking sequence or layup, where the material type can be the same or be a combination of various different ones

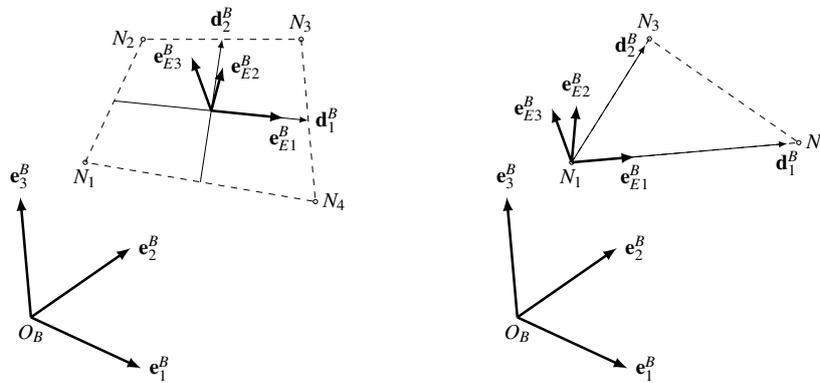


Figure 4: Example of coordinate systems calculated at different element types.

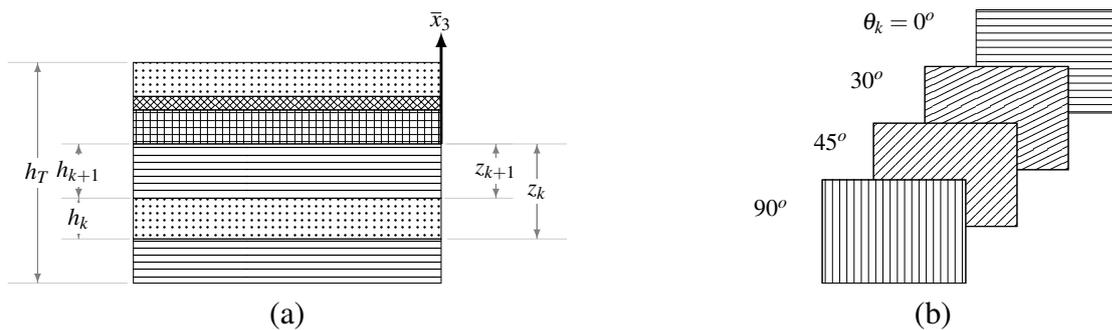


Figure 5: (a) Transversal view of a laminate with definition of laminate thickness parameters. (b) Example of a laminate with multiple orientations.

3.2. Structural transient analysis

One of the analysis types implemented consists in a structural transient analysis that is controlled by the ASAero framework, but using the UM/NLAMMS as CSD solver. The advantage is the possibility of a more generic analysis control, with wider possibilities for the body's prescribed motion and applied forces.

This module is a simplified version of the aeroelastic analyses, since there is no need for aerodynamic calculation or interpolation functions. The call to UM/NLAMMS is exactly the same as in the previous case, using the same functions and information. The main difference consist in the applied forces module, that needs to consider the transient information that is handled by a special dynamic load class.

For the present work, some features have been added to the CSD solver, what became version 2.0 of UM/NLAMMS. The most important is the capability of dealing with different number of materials, including laminated composites with an arbitrary number of laminae. The formulation for that was already included in the paper of [14] but had not been implemented computationally. Among others, the version 2.0 has the following modifications compared with v1.1: •reading of a single input file containing all the information necessary to perform an analysis. •unification of versions - the present version is a single code for static, dynamic and externally coupled analyses. The choice is made inside the main function, that routes the information to the right sub routine. This allows the program to be maintained as an unique core, not being necessary to create new functions to comply with different codes . •compilation as a library (libnlams.a) and creation of an interface function. It facilitates the evolution of the code as a single core since all the versions are unified, and there's no need to call each function separately, but only an interface function. Besides that, the code can be distributed as a "black-box", if necessary. •analysis of laminated composite plates without restriction in the number of layers; also: use of both isotropic and orthotropic material and various elements properties and materials in the same model.

Since the structural solver is called as an external library, all the input/output steps are performed trough a direct data access, using Fortran language features. The time advancing procedure is actually implemented in the driver code, because it is there that the coupling processes are called. The UM/NLAMMS does not receive any information about the aerodynamic model, but only forces already interpolated to the nodes.

3.3. Unsteady aerodynamics

To solve for unsteady aerodynamics, a panel method is used. The UVLM has been implemented based on the codes available in [13] but instead of Fortran, c++ is used. Classes and data structures are used though the code to handle input and output data, and, more important, simulation objects.

The aerodynamic model is defined using the concept of aerodynamic macro-elements (*AME*), which are continuous surfaces that are divided into a finite number of aerodynamic panels, according to parameters defined by the user. These macro-elements are associated with a set of structural FE, and an internal routine identifies which are the nodes contained in both sides. The interpolations are then applied separately to each set *AME/FE*.

A c++ class is created for the vortex-ring element, and two derived classes are used to deal with bound and wake elements. The bound elements need to handle more information, like the orientation at its center, for example. Also, the effective angle of attack need to be calculated only on bound elements, and not on wake elements. The domain discretization is performed in a similar way as in the Doublet-Lattice method, in which an AME is defined for each lifting surface. This AME is then divided into panels with an arbitrary number of divisions in the chordwise and spanwise directions. Once the wing is discretized, the next step is the discretization of the wake, following a similar procedure. In this scheme, the differentiation of "panels" and "vortex-ring elements" needs to be established:

- panels are the divisions of the lifting surfaces, and occupy the same position in space, since they are superimposed to the surfaces. In the UVLM, they are used only as interface between the aerodynamic and physical models, and the associate grid points are used to interpolate displacements from the structural model.

- vortex-ring elements are one of the methods that allow the solution of the Laplace equation in the potential methods, and is the core of the present UVLM implementation. It can be classified into two types: bound and wake elements. The bound elements are those superimposed to the LS, and it is necessary to establish a coordinate system at each time step, first to calculate the induced velocity and later to calculate the pressure distribution. For the wake elements, however, only the information of the corner points position is necessary.

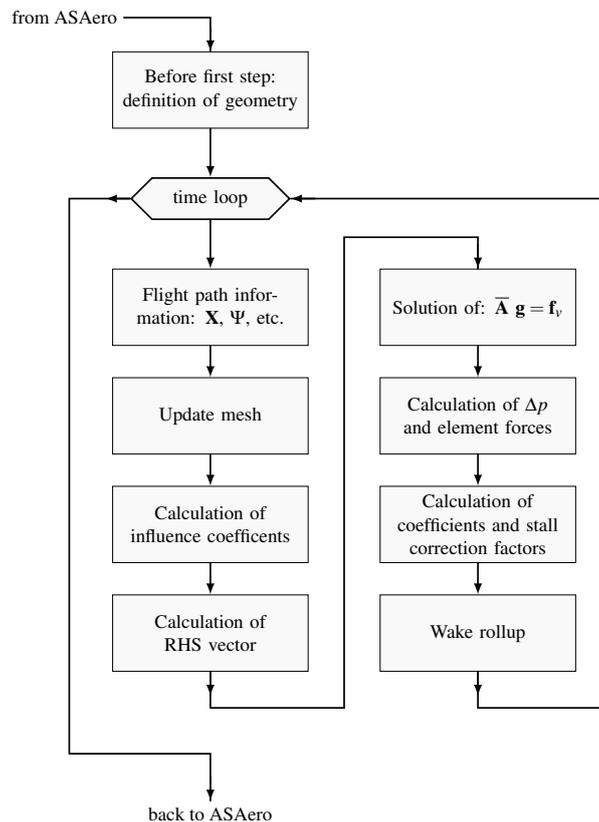


Figure 6: Flowchart of the aerodynamic solver, when running in standalone mode - without interaction with structural solver.

The above characteristics of vortex-rings elements are very suitable for a c++ programming style, as in [12]. In the present case, a generic Vortex Element class is defined, and the Bound and Wake classes are derived from that. All the operations related to geometry updating and information handling are contained inside the class, what brings big advantages in terms of code management and comprehension.

Inside the code, the geometry of panels and elements is defined based on grid points, in a similar way to what is done in the FE formulation. There are three sets of points, whose number remain fixed during the analysis:

- GRIDA: the corner points of panels. They are placed on the lifting surfaces, and depend on the user defined chordwise and spanwise discretization.
- GRIDB: the corner points of bound elements. They are initially created using the GRIDA position, but displaced backwards 25% of each element chord. The geometry of bound elements are then defined based on their position.
- GRIDW: the corner points of wake elements. They are all created also during the discretization step, but placed all at the trailing edge bound element points (last GRIDB point on each row). As the surface moves forward, these points starting moving. The first wake points remain connected

to the last bound points. If the wake is set free, then the other points move according to an inertial and induced velocities.

The above scheme is shown in Fig. 7, where the 3 main discretization steps are presented. In (a), the macro-element as defined by the user, by the position of root leading edge (RLE) and tip leading edge (TLE) points, and root (RC) and tip (TC) chords. In (b), the surface is divided in panels and the corner points are defined (GRIDA). In (c), the bound vortex-ring elements are defined from the panels; the bound element and its (GRIDB) use the same numbering sequence of the aerodynamic panels and GRIDA points.

Matrices with displacement information are assembled for each coordinate system: (a) displacement of aerodynamic panels grid points in the IF ; (b) displacement of aerodynamic panels grid points in the BF ; (c) displacement of bound elements grid points in the IF ; (d) displacement of wake elements grid points in the IF ; and (e) induced displacement of wake elements grid points in the IF . The mesh update is divided into a few steps, using these displacement matrices listed above, as shown in Fig. 7(b). The first updating phase occurs when GRIDA is updated using interpolation of the nodal displacement on the FE model, or using the flight dynamics information. Because of that, it is performed outside the aerodynamics module. The other steps are all performed inside the module UVLM for any kind of analysis that calls the aerodynamic solver. The function to update bound elements at each time step is more complex than that of a wake element. The former needs to have the coordinate system, area, corner and mid points updated, while the latter needs to have only the information on the corner nodes updated.

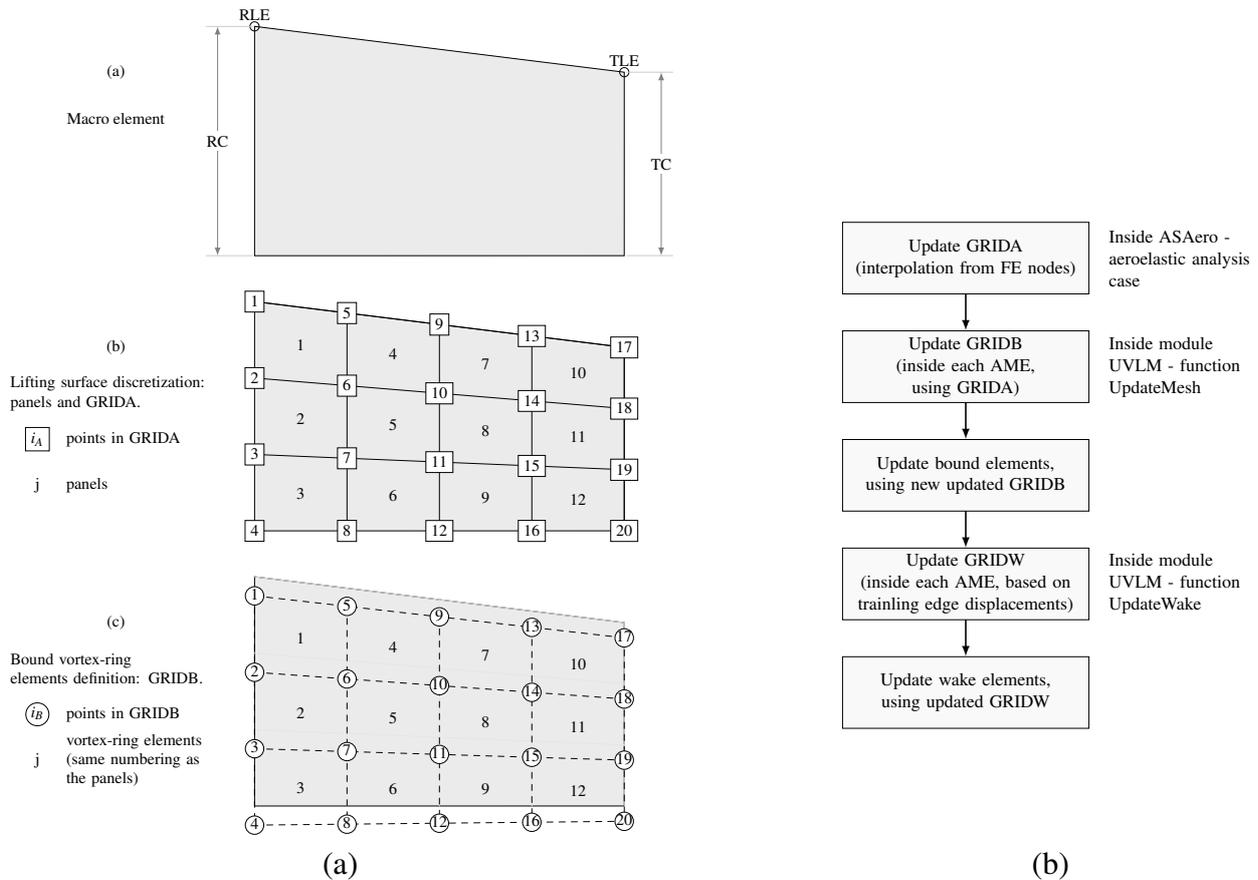


Figure 7: Discretization steps in the Vortex-Ring method, defined by coordinate of root leading edge (RLE), tip leading edge (TLE) points, root (RC) and tip (TC) chords. (b) Aerodynamic model update processes.

The stall correction procedure deals with various two different approaches and various possible static and dynamic stall models. To deal with that, a c++ class dedicated to manage the stall models is implemented. This class defines a airfoil property, and the functions are dedicated to handle the stall information and to define the correction applied to each bound element in an AME.

The stall models can be divided into two basic approaches: equivalent strip and bound element penalization. The equivalent strip approach brings the possibility of applying multiple 2D solutions, even static or dynamic stall models. The penalization functions are encapsulated into the vortex-ring element class that use the airfoil property to calculate the correction on the bound element pressure variation.

3.4. Aeroelastic time-response analysis

The aeroelastic analysis uses the internal aerodynamic solver and the external CSD solver. A flowchart with the aeroelastic time response simulation is seen on Fig. 8. The definition of each analysis phase is established by means of an input file, where the initial time of the perturbation and the time span for the sigmoid curve are informed to the program. The vehicle state is then calculated at each time step inside the "Prescribed Motion" block. The call to the UM/NLAMS is done through an interface function, that handles all the necessary information that need to be passed to the structural solver.

The final version of the implemented aeroelastic framework is divided into two main phases: the first is a pre-processing phase, when memory is allocated, the aerodynamic model is discretized, and the initial conditions are established. For this initial phase, the time step is set to $k = -1$, what indicates to some calculation modules that it is only a initialization step. The next step begins with the time step set to $k = 0$, the time is $t = k dt = 0$ and the aeroelastic phase begins.

Due to the characteristics of the aerodynamic solver, the time step tends to be larger than the required by the structural model, what creates difficulties for the use of implicit coupling methods. The choice comes then for an explicit method with subcycling in the structural part, contrary to what was used by [4], in which subcycling in the aerodynamic part was used. In this way, inside the global time-loop there is a structural-only time-loop, in which the structural solver is called using a smaller time-step. This has shown to be necessary, since the time step used for the aerodynamic model is fixed to comply with an uniform wake and wing discretization. In this case, the time is increased inside the structural loop with a time step Δt_s , until it reaches the next aeroelastic time step $t + \Delta t$. The value of Δt_s is defined independently of the global Δt and is based only in characteristics of the structural model.

The information of both aerodynamic and structural original undeformed meshes is saved and remains constant during the simulation. The transformation between inertial and body fixed system is always required for the definitions of RHS vector in the aerodynamic side and for the corotational formulation used in NLAMS. Displacement matrices are created for each mesh, and these are updated at each time step. For an initial condition in which the model is at rest, $\tilde{\mathbf{f}}_n$ is null, and at any other condition the vectors are updated with the previous time step values.

At each time step a prescribed motion module is called, defining the instantaneous system state variables, including the rotating frame transformation and skew matrices used by both structural and aerodynamic solvers. Once the internal structural time-loop is completed, the displacement and velocities are transported to the aerodynamic model, and the aerodynamic solver is called. The pressures are transformed in nodal forces, and the next time-step is ready to begin. Before that, all necessary information is recorded for post-processing. After completing the time loop it is only necessary to deallocate memory and write log files.

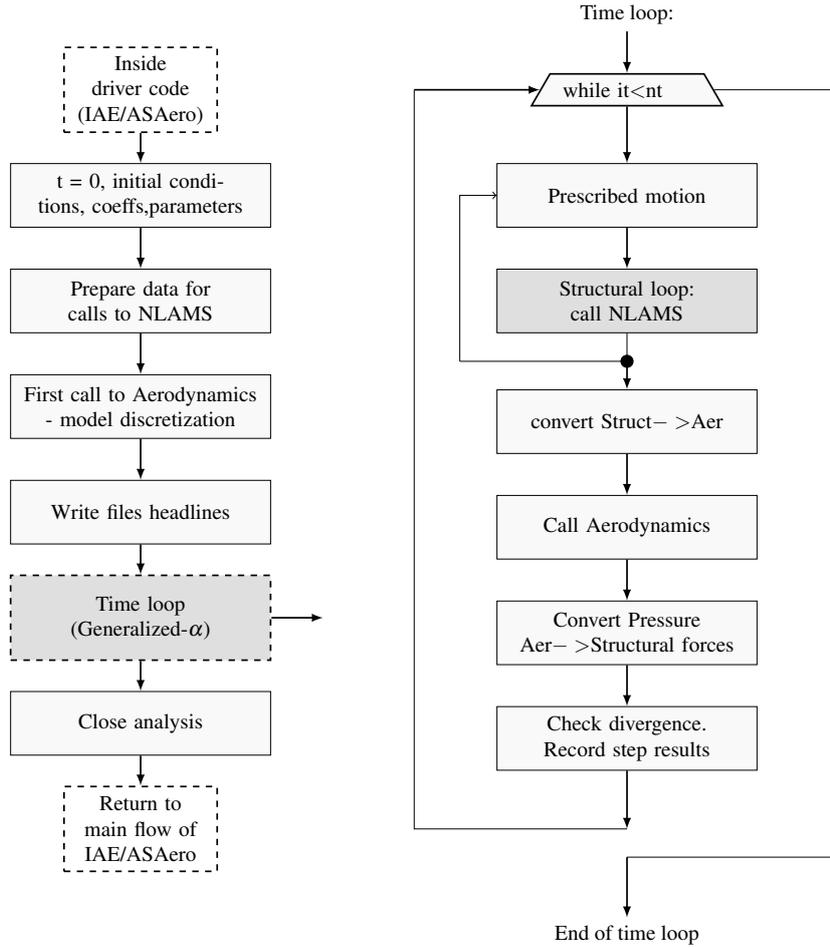


Figure 8: Aeroelastic simulation framework with UM/NLAMS and IAE/ASAero coupled.

3.5. Capabilities under developmet

Further investigations efforts are underway, besides the aeroelastic results shown in [10]. The idea is to demonstrate the code's ability to deal with multiple lifting surfaces and very large rotations. Here, a rotary model, equivalent to a double blade propeller, is shown. this kind of simulation allow the testing of various features can be tested: modeling of two separated aerodynamic macro-elements (AME); application of very large rotations; application of positive and negative incidence angles to the aerodynamic surfaces.

Figure 9(a) shows the dimensions of the rotary wing model. The two surfaces are rigid, what means that no structural model is coupled to the aerodynamics. It is composed by two surfaces, with a gap at the root, and placed symmetrically around the x_1 axis. A torsion is applied to the model, by means of two different incidence angles at the wing root and wing tip. These angles are applied antisymmetrically for each surface. The prescribed motion at the wing root is

$$\mathbf{u} = \{-v_\infty t \ 0 \ 0\}^T \text{ (m)} \quad \Psi = \{\psi \ 0 \ 0\}^T (\pi/180^\circ)(rad/s),$$

what means that the wing is moving in the $-x_1$ direction and rotating around the same axis at a speed of ψ degrees per second. This movement is shown in Fig. 9(b).

This kind of simulation still requires cautious validation, what was not done until the moment. However, Figure 10 shows four snapshots of simulation results for a model discretized into 15×5 bound elements and wake length of 20 chords, $v_\infty = 5m/s$ and $\psi = 18.85(rad/s) = 1080^\circ/s$. It is seen that the wake develops and follows the surfaces. Investigations are necessary to validate

forces and moments results, allowing the simulation of the aerodynamic loading on propellers or wind turbine blades, for example.

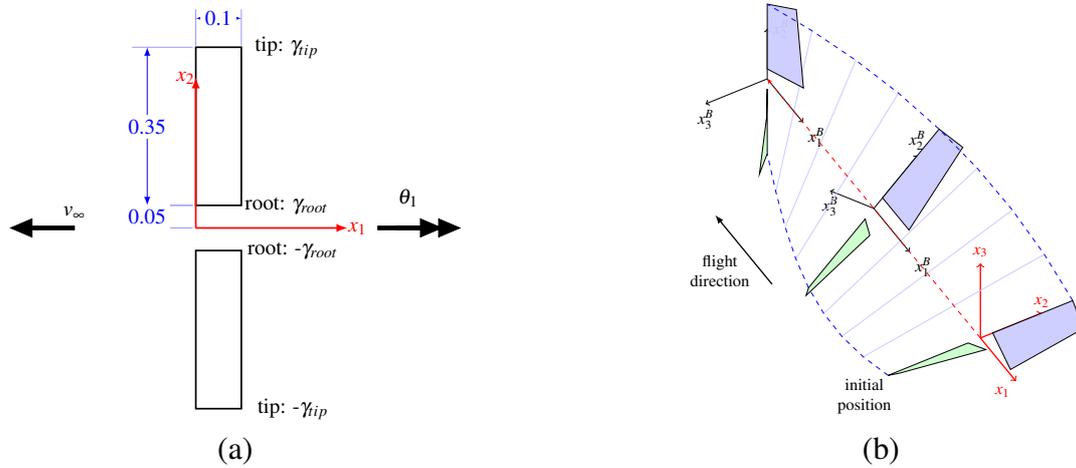


Figure 9: (a) Rotary wing model, defined with different incidence angles at root and tip. (b) Applied body dynamics: translation in the $-x_1$ direction and rotation around the same axis.

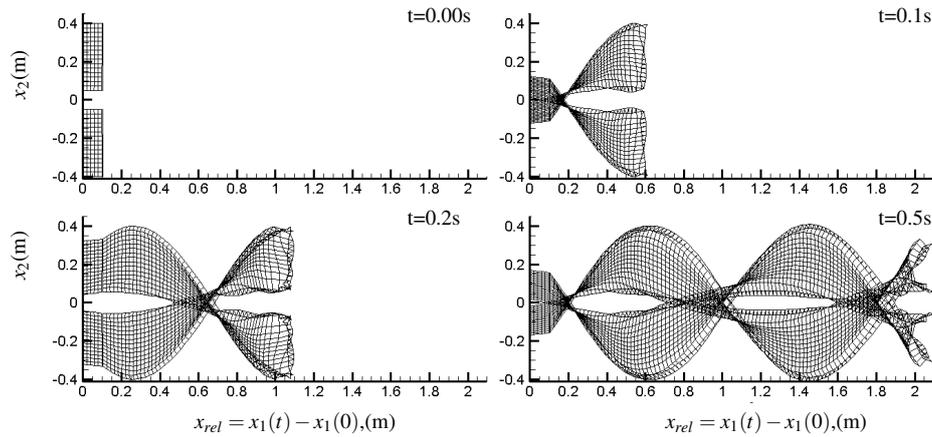


Figure 10: Snapshots of the rotary model for a free wake simulation.

4. NUMERICAL STUDIES

The framework was developed aiming the solution of problems related to large displacements and rotations. Numerical studies were presented in [10], [11], as well as experimental work performed to support the numerical implementation. The numerical framework is currently being used as a simulation tool for flat composite plates, representing very flexible wings. A typical aerodynamic structure is seen in Fig. 11, where a flat plate is subjected to an arbitrary movement in space, that includes linear displacements and large rotations referenced to an inertial frame. Also, multiple local reference frames are established, in the aerodynamic or in the structural part of the model.

Multiple simulations were performed, including structural only, aerodynamics only, and coupled nonlinear aeroelastic. In the remaining of this section, a few validation cases are briefly presented.

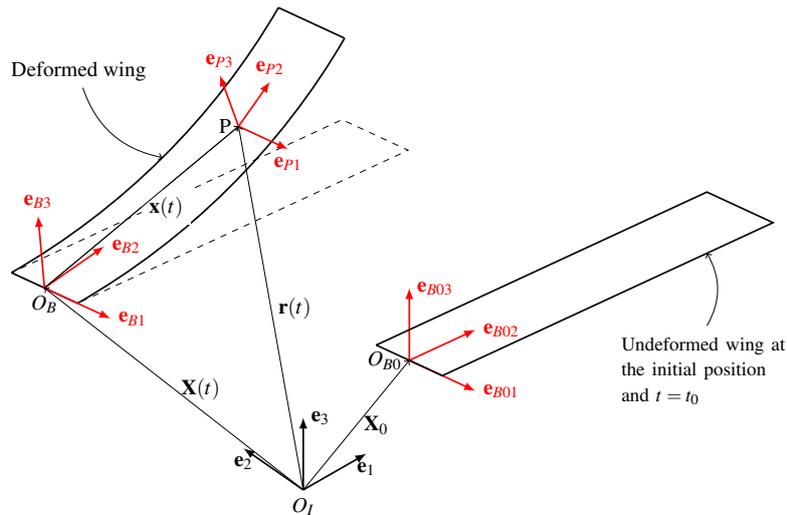


Figure 11: Typical problem under study: flat plate moving in an inertial frame subjected large displacements and rotations.

4.1. Aerodynamics

Results from previous works [10] depict the capability of the code do deal with unsteady aerodynamics through UVLM. For validation purposes, the example 1 of chapter 13 of [13] is used. This is an example of steady-state flight, with a solution for transient lift of a rigid wing. It means that no flight dynamics or aeroelasticity is taken into account, but the wing remains fixed and undeformed. The results are presented for a wing with the following parameters: rigid flat rectangular wing; aspect ratio from 4 to 200 ($= \infty$); angle of attack of 5° ; discretization of 13 panels along the span and 4 along the chord.

In the first time step the gradient $\partial\Phi/\partial t$ is large, and convergence is achieved soon after that. The suggested value for the time step in the reference is $U_\infty\Delta t/c = 1/16$. However, the numerical investigation showed a dependency of the time step with the the size of the wake panel exists. It is necessary to adjust the time step value so that the chord of each wake panel has to be the same of the wing panels [11]. Results then obtained compare very well with reference values, as seen in Figure 12.

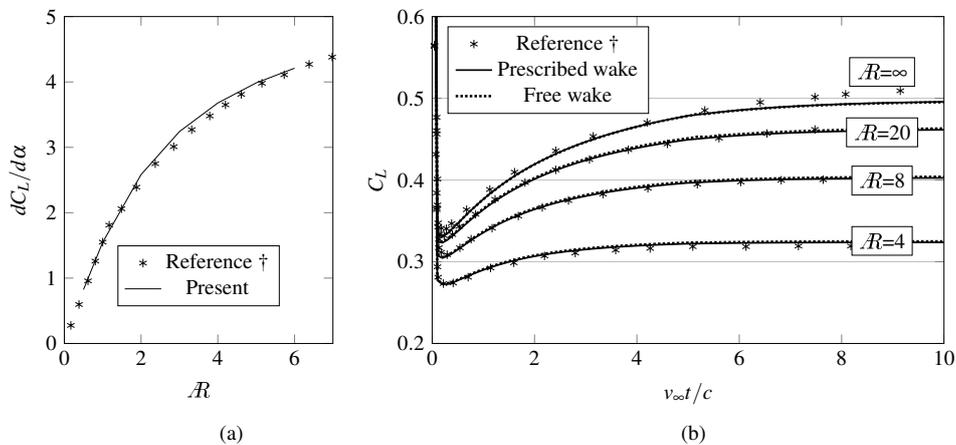


Figure 12: (a) Comparison of $dC_L/d\alpha$ for an increasing value of Aspect ratio. (b) Convergence of C_L with time step. Reference values from [13].

For periodic results, a validation similar to what has been presented by [16] is used. In this case, the Theodorsen's 2D transient lift is used to obtain a $C_L \times h$ curve. A plunging airfoil is used, for which the 2D lift function is reduced to:

$$l = \pi\rho_\infty [b^2\ddot{h} + 2v_\infty bC(k)\dot{h}] \quad (3)$$

where b is the semi-chord ($b = c/2$) and $C(k)$ is the Theodorsen's function [22]. In the above equation, the angle of attack components were removed from the original equation, and only the plunge degree of freedom is considered.

The applied harmonic plunging motion is $h = h_0 \sin(\omega_h t)$ and is evaluated for reduced frequencies k equal to 0.25, 0.50 and 0.75, where the reduced frequency is defined as $k = \omega_h c / 2v_\infty$. A wing with $\mathcal{R} = 100$ is modeled and analyzed with ASAero only. Plunge results are shown in Fig. 13 (a). It is seen that the overall behavior is very well captured. As the reduced frequency increases, the distance between the numerical and analytical curves for each reduced frequency tend to increase due to the influence of the unsteady wake and to the alternation of tip vortex direction as the wing plunges. To reduce the error, it is necessary to reduce the time step. This is achieved by adjusting the discretization: for lower time steps, more panels are necessary in the chord direction. An evaluation of the error dependence on the chordwise discretization is presented in Fig. 13 (b).

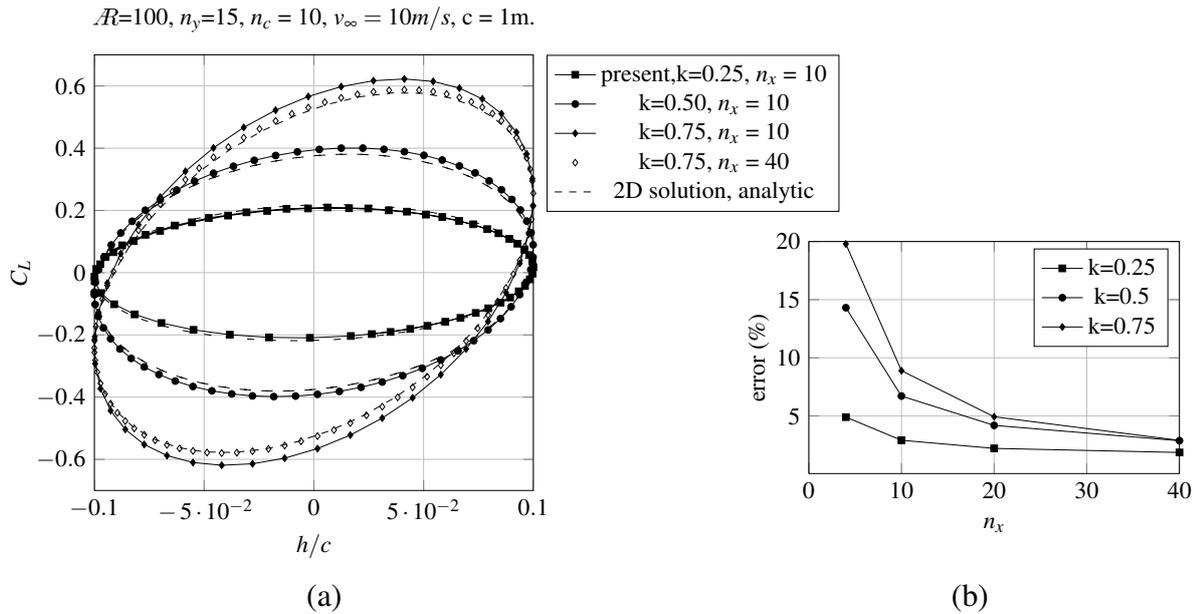


Figure 13: (a) Periodic response results compared to a 2D solution using Theodorsen's function for various reduced frequencies. (b) Evaluation of the error dependence on the chordwise discretization.

4.2. Study of a composite flat plate

A simple flat plate model is used to illustrate the ability of the present framework to capture LCO behavior associated to large displacements and composite material properties. This model represents the one used in the experimental studies presented in [9], being made of carbon roving and epoxy. Overall properties of the model are given in Tab. 1. For simplicity and reduction of total simulation parameters, only one material orientation value is used for the whole plate and the influence of material orientation angle θ_k in the time response results and nonlinear behavior consists in one investigation subject. Model dimensions are seen in Fig. 14(a) and the material orientation in Fig. 14(b).

The structural properties were adjusted according to the experimental results shown in [9]. For example, the average weight of the test models is 64 g, what takes the density to 1427 kg/m^3 . The elastic modules are initially estimated with generic values for E_1 , E_2 and G_{12} and a few linear modal analyses were performed to obtain frequencies approximated to the first three frequencies for each orientation.

The value of the factor S_r used in the generalized- α integration method affects the frequency spectrum response. For $S_r = 0.4$, only the first natural frequency is well captured, and the second one, that has higher energy in this case, appears with a very small amplitude. This happened because of the introduction of high dissipation effects associated to the numerical integration method [19]. For this reason the results show that the parameter S_r should be set as high as possible, to the limit of a Newmark solution. The higher value of S_r was necessary to allow the characterization of the second and third natural modes frequencies for the present model.

Table 1: Parameters for flat plate model: structural model, material properties and aerodynamic model parameters.

Model:		Material properties:		Aerodynamic model:	
dimensions (m):	0.3 x 0.1	E_1 (GPa):	110.0	ρ_∞ (kg/m^3):	1.225
number of elements:	150	E_2 (GPa):	5.25	chord (m):	0.1
number of nodes:	96	G_{12} (GPa):	7.5	\mathcal{R}	3.0
layup :	$[\theta_k]_3$	ν :	0.2		
lamina thickness (m):	5×10^{-4}	ρ_{equiv} (km/m^3):	1427.0		
total thickness (m):	1.5×10^{-3}				

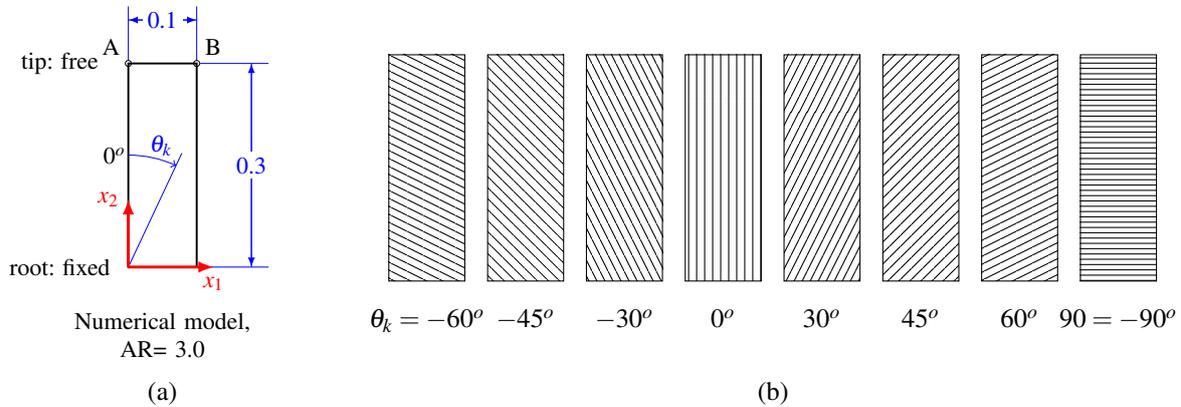


Figure 14: Flat composite plate model: (a) dimensions and position of analysis control points A and B; (b) example of different orientations in the same coordinate system.

A test performed with the structural transient analysis consists in a simulation of an experimental modal analysis (EMA) by means of an equivalent impact test. The plates used in the experiments are modeled using triangular finite elements. The analysis management and model update is all performed inside ASAero and the time step is performed by UM/NLAMMS, called from an interface function. The obtained results are then compared with experimental values. The material properties parameters E_1 , E_2 and G_{12} in the numerical model were adjusted aiming a better curve fitting for values of θ_k .

A transient force is applied to a node at the same position as in the EMA, following the equation below:

$$f(t) = A_{imp} [1 - \cos(\omega_{imp} t) / 2.] \quad (4)$$

where A_{imp} is the impact amplitude and ω_{imp} is the inverse of the impact time span : $\omega_{imp} = 1/\tau_{imp}$. A 2 s simulation is performed using the excitation given above. The displacement on a few points

on the wing are then recorded for a later post-processing. The same procedures used for the EMA described in [9] are used to obtain the FRFs that allow the modes identification. Figure 16 has time shots for the first few milliseconds after the impact is applied to the FE model.

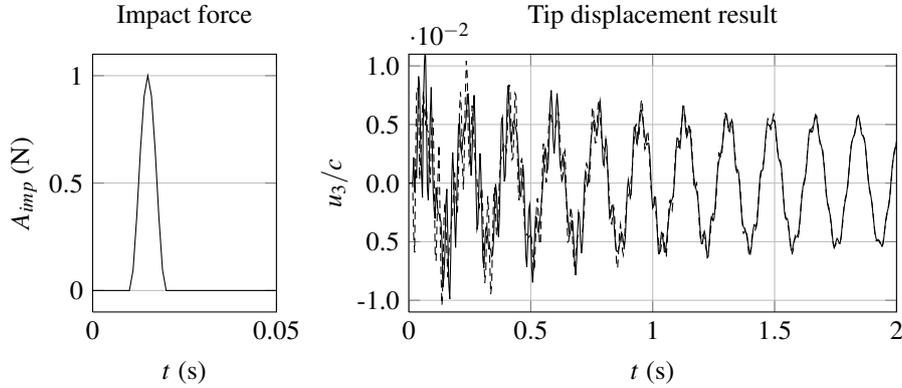


Figure 15: Applied force and tip displacement results for $\theta_k = 0^\circ$.

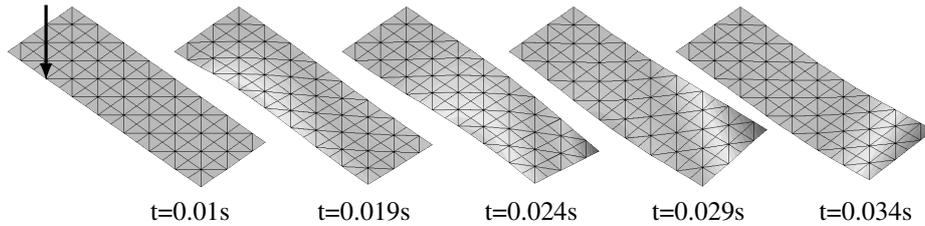


Figure 16: Time shots for the first milliseconds, showing the displacement in the FE model (displacements given in scale = 1:100) and the impact point.

Figure 17 presents the comparison between the natural modes shapes and FRFs obtained from the experimental modal analysis and the numerical impact test. It is observed a very good agreement between numerical and experimental modes. Regarding the FRFs curves, the case of $\theta_k = 30^\circ$ shows the largest difference. However, analyzing the experimental results, it is noticed that the FRF for this case does not show a smooth behavior for the second and third modes, above 45 Hz. For the other cases, all the three first modes are identified and the frequencies variation is captured.

4.3. Comparison with wind tunnel tests

The tests described in [9] were performed with the goal of observing the response to different material orientation θ_k and forward flight velocity v_∞ . The present numerical simulations were defined based on those results and consist in a forward flight at different velocities v_∞ between 5 and 20 m/s, but constant during each simulation case. The influence of root angle of attack α_{root} is another investigation parameter and is also set constant for each simulation. The wing tip is set free, and the position of points A and B are recorded during the simulation for characterization of nonlinear time response. The prescribed motion of the body frame is defined as

$$\mathbf{X} = \begin{Bmatrix} v_\infty t \\ 0 \\ 0 \end{Bmatrix} (m), \quad \Psi = \begin{Bmatrix} 0 \\ \alpha_{root}^o \\ 0 \end{Bmatrix} (\pi/180^\circ) \quad (5)$$

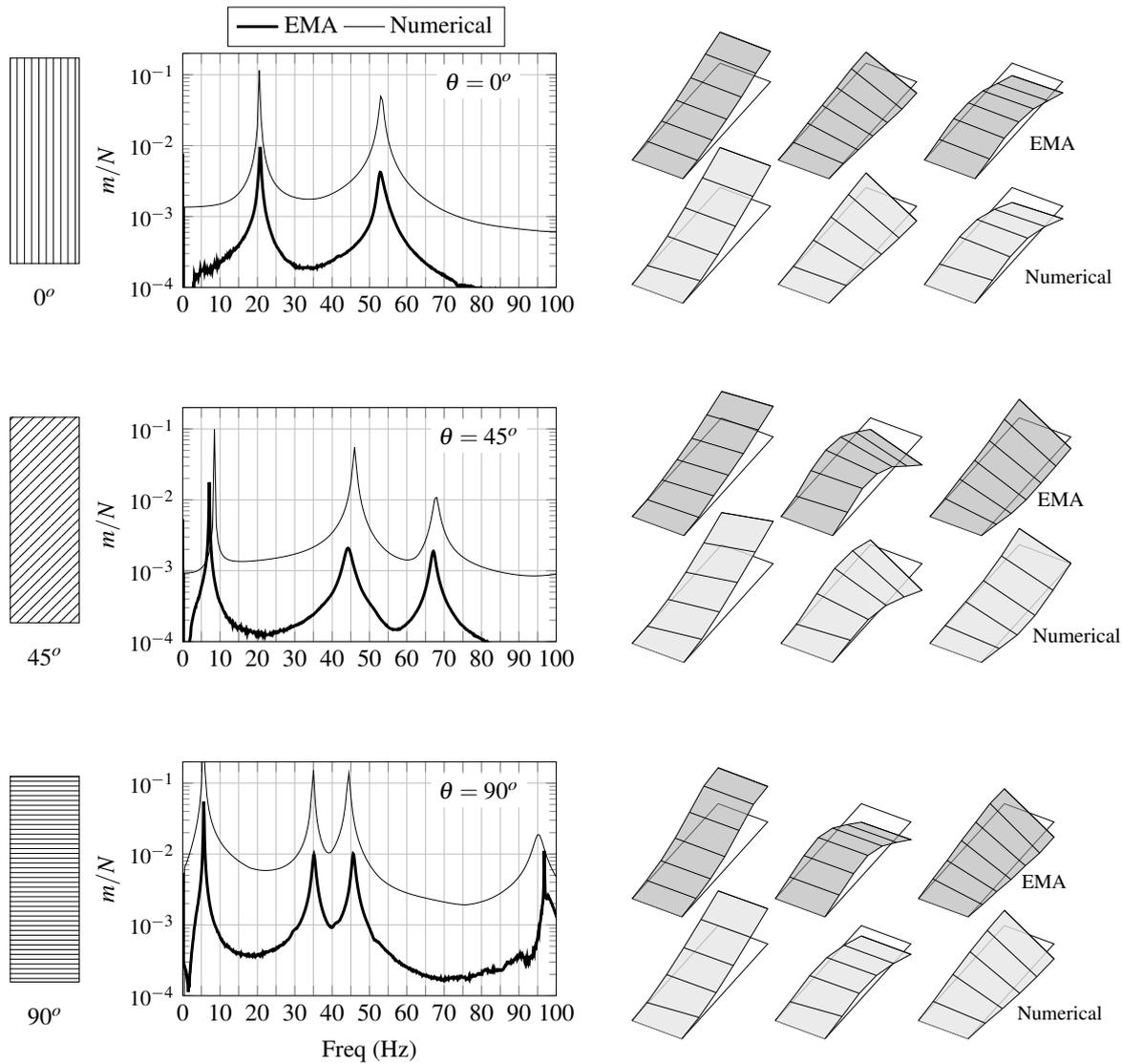


Figure 17: Comparison of natural modes obtained from the experimental modal analysis and the numerical impact test.

where v_∞ and α_{root} are simulation parameters to be modified for the investigations described below. The sections that follow present results and discussion for these tests.

Figure 18 shows a snapshot of a free wake simulation, where it is observed the rollup effects. The wing is divided into 15×5 panels and the wake is 10 chords long. This is the typical aerodynamic discretization for the remaining of this chapter, with exception of the free wake model. It was verified that the loading distribution is not significantly affected by the use of free wake model, and thus it was not considered. This fact results in less computational time for an aeroelastic simulation.

To compare numerical results with experimental ones, displacement results are processed to obtain singular value decomposition (SVD) curves for simulations with conditions similar to those explained in [9], obtaining the frequency spectrum for the different analyses. The displacement signals from points A and B (leading and trailing edge at tip - Fig. 14) are used, allowing the identification of bending and torsion modes, and showing the effects on frequency of fundamental modes distribution. The results are then compared by means of frequency spectrum at different velocities. The time response signal is not compared directly because the experimental one is a nondeterministic signal, presenting effects of wind tunnel turbulence.

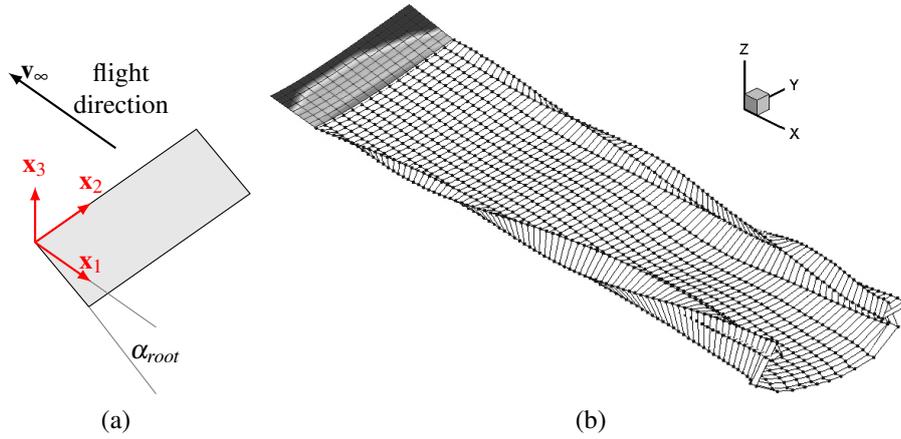


Figure 18: (a) The simulated forward flight condition. (b) The discretized aerodynamic model, with bound and wake elements.

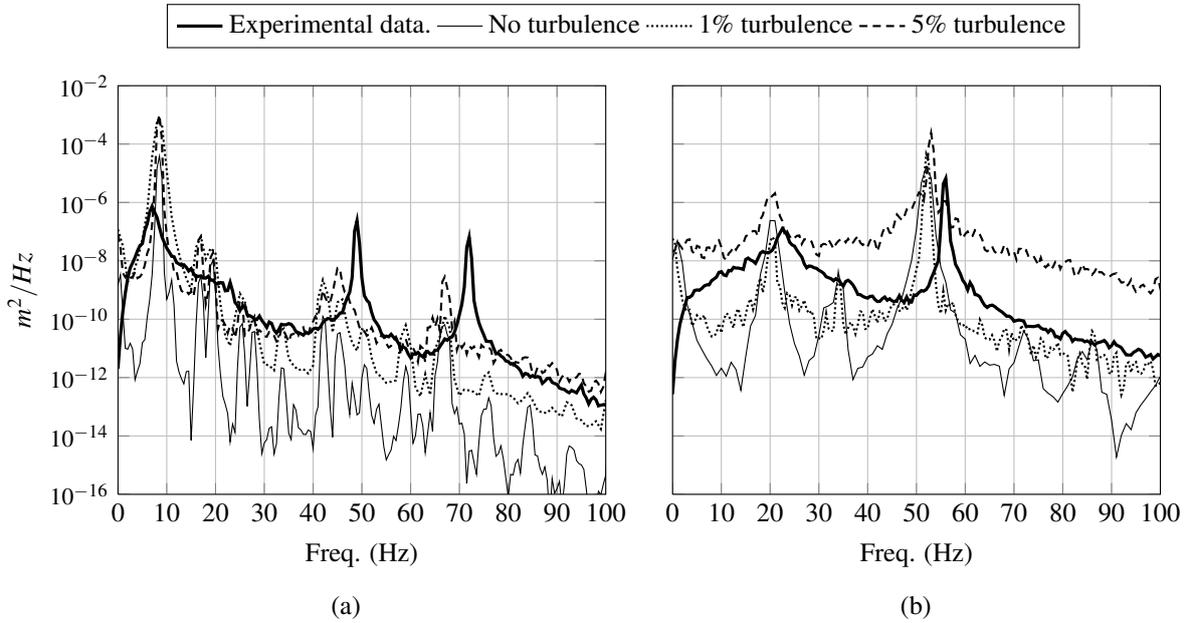


Figure 19: Frequency spectrum for (a) $v_\infty = 10$ m/s and $\theta_k = +45^\circ$, $S_r = 0.95$; (b) $v_\infty = 20$ m/s and $\theta_k = 0^\circ$, $S_r = 0.95$.

Figure 19 shows a comparison between experimental cases and three simulation ones, for $v_\infty = 20$ m/s and $\theta_k = 0^\circ$. Analyzing the experimental conditions, it is clear that the turbulence at the wind-tunnel test section has a strong influence in the displacement signal. A tentative wind perturbation was then added to the numerical simulations an external wind velocity perpendicular to the main flow direction. The c++ function `rand` is used, generating random numbers between -50 and +50, what is applied as a transversal wind, in the following way:

$$\mathbf{v}_{wind} = \{0 \quad 0 \quad 1\}^T \frac{\sigma_r}{100} A_t, \quad (6)$$

where the velocity \mathbf{v}_{wind} is included in the UVLM formulation as an external flow velocity component, A_t is the wind perturbation amplitude and σ_r is the random number generated by the `rand` function. The amplitude A_t is applied as a percentage of the free flow velocity v_∞ , and is related to the test

section characteristics. This random wind condition is applied throughout the simulation time and in composition with the body dynamics expressed in Eq. (5). The dotted lines in Fig. 19 show the spectrum response for this simulation. The increase in the curve level reflects the insertion of energy in the system, by means of the simplified turbulence model. There is no exact match on frequencies values, but the difference is small, and the overall behavior is very well captured by the numerical framework.

Results for a further simulation case, with $v_\infty = 10$ m/s and $\theta_k = +45^\circ$, are shown in Fig. 19. This condition did not present LCO behavior on tests. However, for the simulation without turbulence and $S_r = 0.95$ and a turbulence level of 1%, a LCO response appears, being identified by the appearance of harmonics in the frequency spectrum. A larger turbulence value brought the simulated curve closer to the experimental one, and also showed effects of the second and third modes. For this case, it is seem that a further refining on parameters is necessary for better reproducing the experimental curve.

5. CONCLUDING REMARKS

The paper described a numerical framework for use in multiple types of simulations related to aeroelastic problems. At the moment, it is able to solve for structural transient analyses, unsteady aerodynamics and nonlinear aeroelastic response in time.

The framework was designed to receive additions in form of new functions. This condition makes it attractive for research not only in aeroelastic problems, but for each independent field. An important characteristic is that it uses an input format similar to common engineering softwares, what is important for learning and for productivity. This framework is intended to be an work tool that can be used by an aeroelasticist in a daily basis.

For future, more capabilities are planned, aiming a larger applicability. A direct goal is to complete the validations listed in section 3.5., but also new implementations are directly applicable. Among the desired enhancements, two main goals are related to implementation of control techniques and structural optimization.

ACKNOWLEDGMENTS

The first and second authors acknowledge the partial support from the Instituto Nacional de Ciência e Tecnologia - Estruturas Inteligentes em Engenharia, INCT-EIE. The first author was also partially supported by the University of Michigan's Active Aeroelasticity and Structures Research Laboratory.

REFERENCES

- [1] A. B. Aguiar, C. E. de Souza, and A. G. Marto. Modelagem de uma asa aerolástica experimental em compósito (in portuguese). In *Congresso Nacional de Eng. Mecânica, 6., Campina Grande*, August 2010.
- [2] H. Aono, S. K. Chimakurthi, P. Wu, E. Sällström, B. K. Stanford, C. E. S. Cesnik, P. Ifju, L. Ukeiley, and W. Shyy. A computational and experimental study of flexible flapping wing aerodynamics. In *AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 48., Orlando*. AIAA, 2010.
- [3] S. P. Banerjee. *Aeroelastic Analysis of Membrane Wings*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, August 2007.
- [4] S. K. Chimakurthi. *A Computational Aeroelasticity Framework for Analyzing Flapping Wings*. PhD thesis, The University of Michigan, Ann Arbor, 2009.

- [5] S. K. Chimakurthi, C. E. S. Cesnik, and B. K. Stanford. Flapping-wing structural dynamics formulation based on a corotational shell finite element. *AIAA Journal*, 49(1):128–142, January 2011.
- [6] S. K. Chimakurthi, B. K. Stanford, C. E. S. Cesnik, and W. Shyy. Flapping wing CFD/CSD aeroelastic formulation based on a co-rotational shell finite element. In *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 17th AIAA/ASME/AHS Adaptive Structures Conference 11th AIAA*, 2009.
- [7] A.R. Collar. The expanding domain of aeroelasticity. *Journal of the Royal Aeronautic Society*, 50:613–636, 1946.
- [8] C. E. de Souza and J. S. O. Fonseca. Compliance optimization of laminated shell structures. In *COBEM 2003*, Abril 2003.
- [9] C. E. de Souza, A. G. Marto, R. G. A. da Silva, L. A. Inojosa, and E. L. Oliveira. Characterization of flexible composite wings through experimental and operational modal analyses. In M. A. Savi, editor, *Proceedings of the XV International Symposium on Dynamic Problems of Mechanis*, Búzios, Brasil, February 2013.
- [10] C. E. de Souza, R. G. A. Silva, and C. E. S. Cesnik. Nonlinear aeroelastic framework based on vortex-lattice method and corotational shell finite element. In *53rd Structures, Structural Dynamics, and Materials Conference (SDM)*. AIAA, April 2012.
- [11] C. E. de Souza, R. G. A. Silva, and Carlos E. S. Cesnik. An object-oriented unsteady vortex lattice method for aeroelastic analyses of highly flexible wings. In *10th World Congress on Computational Mechanics*, July 2012.
- [12] T. E. Fritz and L. N. Long. Object-oriented unsteady vortex lattice method for flapping flight. *Journal of Aircraft*, 41(6):1275–1290, 2004.
- [13] J. Katz and A. Plotkin. *Low Speed Aerodynamics*. Cambridge University Press, Cambridge, UK, 2nd. edition, 2001.
- [14] P. Khosravi, R. Ganesan, and R. Sedaghati. Co-rotational nonlinear analysis of thin plates and shells using a new shell element. *International Journal of Numerical Methods in Engineering*, 69:859–885, 2007.
- [15] R. H. MacNeal. The nastran theoretical manual. Technical report, MSC, 1972.
- [16] J. Murua. *Flexible Aircraft Dynamics with a Geometrically-Nonlinear Description of the Unsteady Aerodynamics*. PhD thesis, Imperial College London, London, May 2012.
- [17] R. Palacios, J. Murua, and R. Cook. Structural and aerodynamic models in nonlinear flight dynamics of very flexible aircraft. *AIAA Journal*, 48:2648–2659, 2010.
- [18] G. Seber and O. O. Bendiksen. Nonlinear flutter calculations using finite elements in a direct Eulerian-Lagrangian formulation. *AIAA Journal*, 46(6):1331–1341, June 2008.
- [19] C.M. Shearer and C.E.S. Cesnik. Modified generalized- α method for integrating governing equations of very flexible aircraft. In *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 47.*, Newport, 2006.
- [20] B. Stanford and P. Beran. An updated lagrangian shell and vortex lattice aeroelastic framework for flapping wings. In *IFASD 2009*, 2009.
- [21] W. Su and C. E. S. Cesnik. Nonlinear aeroelastic simulations of a flapping wing micro air vehicle using two unsteady aerodynamic formulations. In *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, number AIAA-2010-2887, pages 1–22, April 2010.
- [22] T. Theodorsen. General theory of aerodynamic instability and the mechanism of flutter. Technical Report NACA-TR-496, NACA, 1935.
- [23] M. S. Vest and J. Katz. Unsteady aerodynamic model of flapping wings. *AIAA Journal*, 34(7):1435–1440, July 1996.
- [24] Z. Wang. *Time-Domain Simulations of Aerodynamic Forces on Three-Dimensional Configurations, Unstable Aeroelastic Responses, and Control by Neural Network Systems*. PhD thesis, Virginia Polytechnic Institute and State University, 2004.

- [25] Zhicun Wang, P. C. Chen, D. D. Liu, and D. T. Mook. Nonlinear- aerodynamics/nonlinear-structure interaction methodology for a high-altitude long-endurance wing. *Journal of Aircraft*, 47(2):556–566, 2010.
- [26] Qiang Zhu. Numerical simulation of a flapping foil with chordwise or spanwise flexibility. *AIAA Journal*, 45(10):2448–2457, 2007.
- [27] ZONA Technology, editor. *ZAERO Theoretical Manual*. Version 8.0. ZONA Tech., 2007.