A continuation method for determining the speed dependent modal properties of large MDOF aeroelastic systems

Julien Heremans[a,b], Grigorios Dimitriadis[c], Vincent Denoël[a]

[a] *Structural & Stochastics Dynamics, University of Liège, Belgium*
[b] *F.R.S-F.N.R.S, Belgian Fund for Scientific Research*
[c] *Aerospace Engineering Department, University of Liège, Belgium*

Abstract

In this article, an arc-length continuation process is presented as an alternative to the classical p-k method to determine the pre-flutter modal properties of an aeroelastic system. The algorithm is based on a reformulation of the generalized eigenvalue problem into a set of nonhomogeneous algebraic equations and on the addition of a continuation equation. The reformulated system is then solved with several nonlinear solvers, and the performance of the resulting algorithms is compared to that of the p-k method on three examples. The analysis is performed mode-by-mode, initiated from wind-off conditions and gradually progressing until aeroelastic instability. The research findings highlight the efficiency of continuation methods, thanks to their ability to refine the wind speed mesh where the system experiences local variations related to rapid aeroelastic changes. The various versions of the proposed algorithm show faster convergence than the direct approach, but also excellent stability performance even in critical regimes. Finally, the mode-by-mode solution allows the use of a custom wind speed mesh for each mode separately and prevents mode swapping.

Keywords: flutter, modal analysis, nonlinear eigenvalue problem, pre-flutter analysis, aeroelastic design, critical velocity

Email addresses: julien.heremans@uliege.be (Julien Heremans), gdimitriadis@uliege.be (Grigorios Dimitriadis), v.denoel@uliege.be (Vincent Denoël)

1. Introduction

Modern design engineering faces numerous challenges when it comes to optimizing the material, cost, and geometries of the design products. The aim of modern design engineering is to achieve efficient designs that are not only lightweight and cost-effective but also highly reliable and safe. Among all the design operations that engineers carry out when designing a wing or bridge deck, flutter analysis is recognized as one of the most crucial tasks when assessing dynamic behavior.

Flutter is an aeroelastic instability, characterized by self-excited vibration that occurs in various types of structures such as aircraft and bridges. It can result in dangerous oscillations, or even total loss of the structure, which makes it a serious concern for design engineers. In practice, most flutter analysis methodologies result in aeroelastic equations of motion expressed in a wind-off modal basis, and formulated as a second order differential equation with frequency-dependent coefficients to model the aerodynamic loads. The solution of the homogeneous differential equation is obtained by solving a nonlinear eigenvalue problem, defining a critical state at which the transfer function is singular. This particular point may not be unique, in which case the system admits several critical states. On the contrary, it may not exist if the system is unconditionally stable, *i.e.* flutter-free.

The determination of the critical velocities is only one part of the design engineer's job, that must also focus on the behavior of the structure for all wind speeds prior to flutter, ensure comfort requirements, safety against fatigue, flutter condition assessment, etc. Beside the classical flutter safety recommendation which ensures that the critical velocity is far enough from the design wind speed [1] or airspeed, some other serviceability or comfort criteria may be based on the maximum allowable structural response and can be specified for instance for an absolute displacement, acceleration or rotation [2]. For wind speeds lower than the critical speed, a forcing (buffeting) term can be added on the right hand side of the governing equation [3, 4], so that the response amplitude can be determined to allow verification of these serviceability conditions. One efficient framework to determine the response of such large-scale structures is provided by the buffeting analysis in modal basis, which, besides the knowledge of all the structural and aeroelastic matrices required for a nodal analysis, necessitates also the establishment of a modal basis, possibly changing with wind speed. The determination of a modal basis is an additional task but is the price to pay for taking benefit of the large computational cost reduction associated with modal truncation. As a result, designers carrying out modal analysis are not only interested in the solution of the critical state, but also in all states prior to it. These states, referred to as *pre-flutter states* in this paper, are generally obtained by solving the nonlinear eigenvalue problem for progressively increasing airspeeds. By extension a *pre-flutter analysis* refers to the modal analysis of the aeroelastic system, consisting in computing the eigenvalues and eigenmodes of the fluid-elastic system.

For some simple models such as the pitch/plunge model, the determination of these states is quite straightforward, but this matter is rapidly complicated when considering more sophisticated models, as required by some modern applications in which several dozens of modes are sometimes considered [5, 6, 7, 8]. Today, an apparent difficulty originates from the mode tracking when solving the eigenvalue problem with frequency dependent matrices. Indeed, when multiple modes are considered, this frequency dependency may lead to mode swapping, one important inconvenience for existing solution techniques. A brief review of solution methods is presented in the next section.

This paper considers an alternative approach by investigating an explicit solution for the nonlinear generalized complex eigenvalue problem. This solution is carried out

mode-by-mode and hinders therefore any mode swapping to occur. A globally convergent algorithm is then constructed using arc-length continuation methods offering hence a systematic and quickly converging tool for determining the pre-flutter states in the manner of those used in structural nonlinear push-over analyses [9, 10].

The specificity of arc-length methods relies in a continuation equation that is added to a set of algebraic equations. Since we focus on a homogeneous critical problem, the nonlinear eigenvalue problem is first transformed into a set of algebraic equations by adding an eigenvector normalization equation. After formalizing the problem and the proposed solution, this article presents in detail the continuation process, and constructs accordingly the system of equations to be solved in the case of an aeroelastic problem. The method is then illustrated on three selected examples. The performance of the method in terms of convergence and computational cost are then compared to reference methods.

The fundamental theory of flutter is quite general, and can be specialized to aeronautical applications as well as civil engineering applications, even if of course, some characteristic quantities are intrinsically linked to each specific field. The developments presented in this article are perfectly general, and may be applied to any flutter application, be it a wing or a bridge deck. For this reason, a general terminology is used in the following so that the free stream airspeed will be sometimes referred to as *wind speed*, while *bridge deck width* may also be translated to *airfoil chord*.

2. Problem statement

The most fundamental equation governing the dynamic behavior of an aeroelastic system is

$$\mathbf{M}_s\ddot{\mathbf{x}}(t) + \mathbf{C}_s\dot{\mathbf{x}}(t) + \mathbf{K}_s\mathbf{x}(t) = \mathbf{p}(\mathbf{x}(t), \dot{\mathbf{x}}(t), \ddot{\mathbf{x}}(t)), \tag{1}$$

where $\mathbf{M}_s$, $\mathbf{C}_s$, $\mathbf{K}_s$ are the nodal structural matrices of mass, damping and stiffness, respectively expressed in kg, $\mathrm{N\,s\,m^{-1}}$ and $\mathrm{N\,m^{-1}}$, $\mathbf{x}(t)$ is the nodal displacement vector, $\dot{\mathbf{x}}(t)$ its time derivative and $\mathbf{p}$ is the aerodynamic nodal force vector in N. The form of $\mathbf{p}$ may significantly differ depending on the application, but can generally be modeled by an operator of the nodal displacement $\mathbf{x}(t)$, velocity $\dot{\mathbf{x}}(t)$ and acceleration $\ddot{\mathbf{x}}(t)$. Assuming a Ritz–Galerkin approach [11], the nodal displacement $\mathbf{x}(t)$ is expressed as a linear combination of m mode shapes Ψ_i $(i = 1, \ldots, m)$ and associated modal amplitudes $y_i(t)$. Hence $\mathbf{x}(t) = \mathbf{\Psi}\mathbf{y}(t)$, where $\mathbf{\Psi}$ gathers the column ordered mode shapes, and $\mathbf{y}(t)$ is a column vector collecting the modal coordinates $y_i(t)$. The choice of the modal basis is not discussed in this article as the proposed methodology is not specific to any modal basis. For instance, it can be the normal modes of vibration obtained with a detailed finite element model [12] of an aircraft, or the modes of a bridge deck under a reference aeroelastic loading, or $\mathbf{\Psi}$ may even be chosen equal to the identity matrix, in which case the modal basis is the same as the nodal basis. The following methodology applies regardless of the basis in which the problem is established.

Projection of (1) in the modal basis leads to

$$\mathbf{M}_s^*\ddot{\mathbf{y}}(t) + \mathbf{C}_s^*\dot{\mathbf{y}}(t) + \mathbf{K}_s^*\mathbf{y}(t) = \mathbf{p}^*(\mathbf{y}(t), \dot{\mathbf{y}}(t), \ddot{\mathbf{y}}(t)) \tag{2}$$

where the modal matrices $\mathbf{M}_s^* = \mathbf{\Psi}^T\mathbf{M}_s\mathbf{\Psi}$, $\mathbf{C}_s^* = \mathbf{\Psi}^T\mathbf{C}_s\mathbf{\Psi}$ and $\mathbf{K}_s^* = \mathbf{\Psi}^T\mathbf{K}_s\mathbf{\Psi}$ and the generalized modal force $\mathbf{\Psi}^T\mathbf{p}(\mathbf{y}(t), \dot{\mathbf{y}}(t))$ are introduced. For the sake of conciseness, it is proposed to drop the superscript $*$ to denote to modal quantities in the following.

In aeronautic applications the load vector usually admits a simple canonical form [13, 14] such that aeroelastic equations are linear and read

$$\mathbf{M}_s\ddot{\mathbf{y}}(t) + \mathbf{C}_s\dot{\mathbf{y}}(t) + \mathbf{K}_s\mathbf{y}(t) - \frac{1}{2}\rho U^2\mathbf{Q}(k)\mathbf{y}(t) = \mathbf{0} \tag{3}$$

where $\mathbf{Q}(k)$ is the complex aerodynamic force matrix, ρ is the free stream air density, U the free stream airspeed, $k = \omega b/U$ is the reduced frequency, ω is the frequency in rad/s, b is a characteristic length. Eq. (3) is a time–frequency domain equation. The aerodynamic load term involves the product of time and frequency dependent variables $\mathbf{Q}(k)$ and $\mathbf{y}(t)$, which is formally wrong from a mathematical standpoint since the problem should be formulated in the time or frequency domain, but not both at the same time. It needs to be interpreted as a convolution in the time domain, and not as a product of two functions of k and t respectively. To be explicit, in the frequency domain, Eq. (3) should be understood as

$$\left(-\mathbf{M}_s\omega^2 + \mathrm{i}\omega\mathbf{C}_s + \mathbf{K}_s - \frac{1}{2}\rho U^2 \mathbf{Q}\left(\frac{\omega b}{U}\right)\right)\mathbf{Y}(\omega) = \mathbf{0} \tag{4}$$

where $\mathbf{Y}(\omega)$ is the Fourier transform of $\mathbf{y}(t)$. Therefore this set of equations exactly corresponds to a second order differential system with airspeed and frequency dependent properties. Despite its lack of rigor, the time–frequency notation (3) is intensively used in aeroelasticity [4, 13, 15], where the aerodynamic loads are most often established in the frequency domain. For aircraft, the unsteady aerodynamic model is obtained from solutions of the potential flow equations, such as the Doublet Lattice Method [16] or the Source and Doublet Panel Method [15]. For bridges, the aerodynamic lift and drag forces and moment acting on a bridge deck segment are expressed by means of Scanlan derivatives A_i^*, P_i^* and H_i^*, see for instance [17, 18] and [4, 19]

$$\begin{aligned}
L &= \frac{1}{2}\rho U^2 B\left(KH_1^*\frac{\dot{h}}{U} + KH_2^*\frac{B\dot{\alpha}}{U} + K^2H_3^*\alpha + K^2H_4^*\frac{h}{B}\right)\\
D &= \frac{1}{2}\rho U^2 B\left(KP_1^*\frac{\dot{p}}{U} + KP_2^*\frac{B\dot{\alpha}}{U} + K^2P_3^*\alpha + K^2P_4^*\frac{p}{B}\right)\\
M &= \frac{1}{2}\rho U^2 B^2\left(KA_1^*\frac{\dot{h}}{U} + KA_2^*\frac{B\dot{\alpha}}{U} + K^2A_3^*\alpha + K^2A_4^*\frac{h}{B}\right),
\end{aligned} \tag{5}$$

where $K = 2k$ is a reduced frequency, $\alpha(t)$ is the displacement in torsion, and $p(t)$ and $h(t)$ are respectively the along and cross-wind displacements. Here again, the aerodynamic loads exhibit the time–frequency format, as discussed previously.

The formulation (5) is fully compatible with the very general formulation usually encountered in aeronautics $\frac{1}{2}\rho U^2\mathbf{Q}(k)\mathbf{y}(t)$ as used in (3) but without aerodynamic mass terms. More specifically, the vector $\mathbf{Q}(k)\mathbf{y}(t)$ is composed of triplets gathering the lift, drag and moment forces at each node. Each of them is expressed as

$$\frac{1}{2}\rho U^2 4k^2\begin{pmatrix} H_4^* + \mathrm{i}H_1^* & 0 & B(H_3^* + \mathrm{i}H_2^*)\\ 0 & P_4^* + \mathrm{i}P_1^* & B(P_3^* + \mathrm{i}P_2^*)\\ B(A_4^* + \mathrm{i}A_1^*) & 0 & B^2(A_3^* + \mathrm{i}A_2^*)\end{pmatrix}\begin{pmatrix} h\\ p\\ \alpha\end{pmatrix}, \tag{6}$$

where the Scanlan derivatives are all functions of k and the displacements h, p, and rotation α are written in the frequency domain.

Eq. (3) is solved for the system's eigenvalues at different airspeeds, altitudes and, for compressible flows, Mach number values in order to investigate the stability of the system at all flight conditions of interest. Several methods have already been proposed and are reviewed in the next section.

3. Existing Solutions

The eigensolution of (3) is usually obtained by defining a Laplace variable

$$p = \frac{\mathrm{d}}{\mathrm{d}t} \tag{7}$$

and substituting into (3) to obtain

$$\left(\mathbf{M}_s p^2 + \mathbf{C}_s p + \mathbf{K}_s - \frac{1}{2}\rho U^2 \mathbf{Q}(k)\right)\mathbf{y} = \mathbf{0}. \tag{8}$$

This definition of p is entirely equivalent to guessing a solution of the form

$$\mathbf{y} = \boldsymbol{\phi} \mathrm{e}^{pt} \tag{9}$$

where $\boldsymbol{\phi}$ is an eigenvector of the aeroelastic system and p the associated eigenvalue, and then substituting into (3). Consequently, p can be seen as both a Laplace variable or a system eigenvalue with the form $p = r + \mathrm{i}\omega$, such that its imaginary part is the frequency of oscillation

$$\omega = \frac{kU}{b}. \tag{10}$$

The airspeed U is treated here as an input parameter. There are two typical approaches for calculating p. The first one is called *determinant iteration* and determines r and ω such that

$$|\mathbf{M}_s p^2 + \mathbf{C}_s p + \mathbf{K}_s - \frac{1}{2}\rho U^2 \mathbf{Q}(k)| = 0. \tag{11}$$

As the determinant is complex, Eq. (11) constitutes a set of two equations with two unknowns that can be solved using a nonlinear solver at all airspeeds of interest. The solution is started at $U = 0$ where r and ω are obtained from the eigensolutions of the wind-off structural system. A set of determinant iterations is carried out for each system state until the critical airspeed.

The second method is called *frequency matching* (or frequency lining-up) where the eigenvalues of an augmented state space matrix $\mathbf{A}$ are evaluated

$$\mathbf{A} = \begin{pmatrix} -\mathbf{M}_s^{-1}\mathbf{C}_s & -\mathbf{M}_s^{-1}\left(\mathbf{K}_s - \frac{1}{2}\rho U^2 \mathbf{Q}(k)\right) \\ \mathbf{I} & \mathbf{0} \end{pmatrix} \tag{12}$$

such that $\Im(p) = kU/b$. Several algorithms have already been formulated to solve this problem when $\mathbf{A}$ is a constant matrix, such as the QZ-algorithm [20] or Cholesky decomposition [21] based solution. The solution is started near $U = 0$ (but not at $U = 0$ because k is infinite) with initial guesses for p and k from the solution of the wind-off system. After each eigenvalue calculation, k is updated from $k = \Im(p)b/U$ until the old and new values of k are nearly identical. Again, one set of frequency matching iterations is carried out for each mode.

The matrix $\mathbf{Q}(k)$ is obtained numerically at a set of predefined discrete reduced frequency values k_i; it is calculated at intermediate values of k by interpolation. The problem that arises is the fact that $\mathbf{Q}(k)$ contains aerodynamic mass stiffness and damping information but, due to their numerical nature, it is impossible to separate these contributions. Consequently, several flutter analysis methods have been developed, such as the k method [22], the p method [23], the p-k method [24], modified versions of the p-k

method [25, 26] and the g method [27]. For instance, in the modified p-k method, the matrix $\mathbf{Q}(k)$ is separated into its real and imaginary parts. Then, Eq. (8) becomes

$$\left(\mathbf{M}_s p^2 + \left(\mathbf{C}_s - \frac{1}{2k}\rho b U \Im(\mathbf{Q}(k))\right) p + \mathbf{K}_s - \frac{1}{2}\rho U^2 \Re(\mathbf{Q}(k))\right) \mathbf{Y}(\omega) = \mathbf{0} \tag{13}$$

such that $\Im(\mathbf{Q}(k))$ represents aerodynamic damping and $\Re(\mathbf{Q}(k))$ aerodynamic stiffness. Then, the p-k solution is obtained by calculating the eigenvalues of matrix

$$\mathbf{A} = \begin{pmatrix} -\mathbf{M}_s^{-1}\left(\mathbf{C}_s - \frac{1}{2k}\rho b U \Im(\mathbf{Q}(k))\right) & -\mathbf{M}_s^{-1}\left(\mathbf{K}_s - \frac{1}{2}\rho U^2 \Re(\mathbf{Q}(k))\right) \\ \mathbf{I} & \mathbf{0} \end{pmatrix}. \tag{14}$$

In civil engineering, the Scanlan derivatives for a particular bridge section are usually estimated from wind tunnel experiments [28, 29, 30]. Accounting for (5), the equation of motion (3) may be expressed as

$$\left[\mathbf{M}_s - \mathbf{M}_{\text{ae}}(k,U)\right] \ddot{\mathbf{y}} + \left[\mathbf{C}_s - \mathbf{C}_{\text{ae}}(k,U)\right] \dot{\mathbf{y}} + \left[\mathbf{K}_s - \mathbf{K}_{\text{ae}}(k,U)\right] \mathbf{y} = \mathbf{0} \tag{15}$$

where $\mathbf{M}_{\text{ae}}(k,U)$ is the aerodynamic mass matrix, $\mathbf{C}_{\text{ae}}(k,U)$ the aerodynamic damping matrix and $\mathbf{K}_{\text{ae}}(k,U)$ the aerodynamic stiffness matrix. As the aerodynamic load contributions are already separate, the issue of separating them out of a combined aerodynamic load matrix does not exist and there is no need to apply the modified p-k or g methods. Determinant iteration is carried out by solving

$$\left|\left[\mathbf{M}_s - \mathbf{M}_{\text{ae}}(k,U)\right] p^2 + \left(\mathbf{C}_s - \mathbf{C}_{\text{ae}}(k,U)\right) p + \left[\mathbf{K}_s - \mathbf{K}_{\text{ae}}(k,U)\right]\right| = 0 \tag{16}$$

while frequency matching is achieved by calculating the eigenvalues of matrix

$$\mathbf{A} = \begin{pmatrix} -\mathbf{M}^{-1}\left[\mathbf{C}_s - \mathbf{C}_{\text{ae}}(k,U)\right] & -\mathbf{M}^{-1}\left[\mathbf{K}_s - \mathbf{K}_{\text{ae}}(k,U)\right] \\ \mathbf{I} & \mathbf{0} \end{pmatrix} \tag{17}$$

where $\mathbf{M} = (\mathbf{M}_s - \mathbf{M}_{\text{ae}}(k,U))$.

A common problem with both determinant iteration and frequency matching is the fact that, as the airspeed varies, mode swapping can occur. For example, if the imaginary part of the eigenvalue of mode 1 is lower than that of mode 2 at a higher airspeed, the solution procedure will follow the lowest imaginary part and produce an eigenvalue branch that is a mix of the eigenvalues of modes 1 and 2. This situation can render the resulting stability plots difficult to interpret, especially when several modes are getting involved. Some methods based on scalar products of mode shapes or the Modal Assurance Criterion (MAC) [31, 32] have been proposed to alleviate the mode swapping issue and to reorder the output of the p-k method. As shown next, these methods are not unfailing and sometimes offer disappointing performance. Manual sorting turns out to be the last resort, but becomes rapidly cumbersome for larger systems.

Furthermore, the problem statement as formulated by the frequency matching method results in a waste of resources as, at every single iteration, all eigenvalues of (12) are calculated while only one of them is of interest. The unnecessary computational burden increases with the size of the model, and becomes significant when complex models are employed. Alternative algorithms may be more resource efficient by specifying *a priori* the mode number being tracked, but must anyway at least evaluate several eigenvalues at a time if these are close to another. The determinant iteration approach does not suffer from this inconvenience.

The article has thus far reviewed two categories of methods. However, the approach adopted in this article does not belong to either of the above methods. It relies on an explicit solution of the generalized complex eigenvalue problem to solve the flutter equations. It may therefore be described as an hybridization between the two approaches. By assuming a solution of the form of (9), the proposed method solves for both p and $\boldsymbol{\phi}$, such that there is no possibility for mode swapping to occur, since the eigenvectors $\boldsymbol{\phi}$ are orthogonal to each other at the same airspeed U. The process is initiated in wind-off conditions, as usual, and progresses towards the critical speed by implementing an arc-length continuation process rather than following a user predefined mesh.

4. Description of the method

4.1. Transformation of the eigenvalue problem into a nonhomogeneous problem

Since (15) is a linear problem, its solution is

$$\mathbf{y}(t) = \sum_{i=1}^{m} \boldsymbol{\phi}_i \mathrm{e}^{\lambda_i t}, \tag{18}$$

where m is the number of aeroelastic modes considered in the analysis, ϕ_i is the ith mode of the aeroelastic system and λ_i is used instead of p_i to emphasize that this is an eigensolution. This eigenvalue is such that $\lambda_i = -\xi_i\omega_{0,i} + \mathrm{i}\omega_i$ where $\omega_{0,i} = |\lambda_i|$ and ξ_i take the meaning of an undamped circular frequency and a damping ratio. Substituting (18) into (15) leads to

$$\lambda_i^2\mathbf{M}(\omega_i, U)\boldsymbol{\phi}_i + \lambda_i\mathbf{C}(\omega_i, U)\boldsymbol{\phi}_i + \mathbf{K}(\omega_i, U)\boldsymbol{\phi}_i = \mathbf{0} \tag{19}$$

where $\mathbf{M}(\omega, U) = \mathbf{M}_s - \mathbf{M}_{\mathrm{ae}}(\omega, U)$, $\mathbf{C}(\omega, U) = \mathbf{C}_s - \mathbf{C}_{\mathrm{ae}}(\omega, U)$ and $\mathbf{K}(\omega, U) = \mathbf{K}_s - \mathbf{K}_{\mathrm{ae}}(\omega, U)$. This procedure is fully compatible with aircraft flutter; for instance, if the modified p-k approach is used,

$$\mathbf{M}_{\mathrm{ae}}(\omega, U) = \mathbf{0},\ \mathbf{C}_{\mathrm{ae}}(\omega, U) = \frac{1}{2k}\rho b U \Im(\mathbf{Q}(k)),\ \mathbf{K}_{\mathrm{ae}}(\omega, U) = \frac{1}{2}\rho U^2 \Re(\mathbf{Q}(k)). \tag{20}$$

If n refers to the initial number of coordinates to formulate the problem —nodal or wind-off modal—, Eq. (19) defines a set of $2n$ nonlinear real equations, and stages $2(n+1)$ real unknowns for a given airspeed U; two for the real and imaginary parts of λ_i, and $2n$ for the real and imaginary parts of the complex eigenvectors $\boldsymbol{\phi}_i$. This is typical of eigenvalue problems where the underdetermination is associated with the freedom to normalize the eigenvectors in any desired way. Consequently, a nonlinear algebraic problem can be obtained by supplementing two additional equations to normalize the eigenvectors. The normalization conditions chosen here are

$$\Re\{\boldsymbol{\phi}_i\} \cdot \Re\{\boldsymbol{\phi}_i\} = 1 \quad \text{and} \quad \Im\{\boldsymbol{\phi}_i\} \cdot \Im\{\boldsymbol{\phi}_i\} = 1. \tag{21}$$

For a chosen airspeed, Eqs. (19) and (21) form a closed set of $2(n+1)$ nonlinear algebraic equations with $2(n+1)$ unknowns. Introducing the objective function $\mathbf{f}(\mathbf{x}, U)$, the problem may be expressed in the general form

$$\mathbf{f}(\mathbf{x}_i, U) = \mathbf{0}, \quad \text{with} \quad \mathbf{x}_i = \left[\Re(\lambda_i),\ \Im(\lambda_i),\ \Re(\boldsymbol{\phi}_i)^T,\ \Im(\boldsymbol{\phi}_i)^T\right]^T, \tag{22}$$

and solved numerically for each mode and all U of interest using classical nonlinear solvers, such as the Newton–Raphson [33], Krylov [34] or [35] methods. The convergence of such

algorithms may be quite sensitive to the initial guess and divergence may occur for some chosen U without apparent reason, especially when the aeroelastic system experiences rapid changes.

Note also that since (22) corresponds to a mode-by-mode approach, its solution for each mode can be carried out separately, possibly in parallel.

4.2. The continuation process

The continuation process is based on introducing a continuation equation to the above system (22). In parallel, the airspeed is now considered as a variable rather than a parameter. Introducing $\mathcal{D}$, the $2n+3$ dimensional space defined by the real unknowns of the problem —namely air speed U, eigenvalues λ_i and eigenvectors $\boldsymbol{\phi}_i$— and starting from a known point $\mathbf{p}_0 \in \mathcal{D}$ that is a solution of $\mathbf{f}(\mathbf{x}, U) = \mathbf{0}$, the chosen arc-length method consists in finding the intersection of the objective function $\mathbf{f}(\mathbf{x}, U)$ with the hypersphere of radius r, defined in $\mathcal{D}$ and centered on $\mathbf{p}_0 = (\mathbf{x}_0, U_0)$. The situation is illustrated in two dimensions in Figure 1, which plots two eigensolution branches —denoted by Mode 1 and Mode 2. Mode 1 is being tracked. Point $(\mathbf{x}_0, U_0)$ lies on Mode 1; the two intersections of this branch with the hypersphere centered at $(\mathbf{x}_0, U_0)$ are determined and the process is repeated by placing the new center at the rightmost root. The equation of the hypersphere is

$$\begin{aligned} r^2 \quad = \quad & \left(\frac{U - U_0}{U_{\text{ref}}}\right)^2 + \left(\frac{\Re(\lambda) - \Re(\lambda_0)}{\Re(\lambda_{\text{ref}})}\right)^2 + \left(\frac{\Im(\lambda) - \Im(\lambda_0)}{\Im(\lambda_{\text{ref}})}\right)^2 \\ & + \sum_{k=1}^{n} \left(\frac{\Re(\phi_k) - \Re(\phi_{0,k})}{\Re(\phi_{\text{ref},k})}\right)^2 + \left(\frac{\Im(\phi_k) - \Im(\phi_{0,k})}{\Im(\phi_{\text{ref},k})}\right)^2 \end{aligned} \tag{23}$$

where the quantities with subscript $_{\text{ref}}$ are scaling parameters chosen to make sure that $((U - U_0)/U_{\text{ref}})^2$ and each term of the dot product in (23) have the same order of magnitude, so that they all bring a similar contribution to the sphere radius. In total for each mode, there are $2n + 3$ unknowns and as many equations. The physical interpretation of the augmented system is important: in the current iteration, the solution of (22) is searched in space $\mathcal{D}$ so that the solution lies precisely at a distance r from the last converged point p_0. The introduction of rescaling parameters relative to any direction of $\mathcal{D}$ and therefore the choice of an hyperellipsoid for (23) makes this equation dimensionally consistent; each term in the sum is dimensionless, as is the radius of the hypersphere. As well as being very general, this form allows the user to deform the hypersphere in any direction, potentially adjusting the weight relative to any direction. The freedom in choosing the rescaling parameter is a feature discussed and illustrated later in Section 5.2. Introducing the rescaling matrix, $\mathbf{S}^{-1} = \text{diag}\big[\Re(\lambda_{\text{ref}}), \quad \Im(\lambda_{\text{ref}}), \quad \Re(\phi_{\text{ref},1}), \ldots, \quad \Im(\phi_{\text{ref},N})\big]$, the hypersphere equation can be written as

$$r^2 = \left(\frac{U - U_0}{U_{\text{ref}}}\right)^2 + (\mathbf{x} - \mathbf{x}_0)^T \mathbf{S}^T \mathbf{S} (\mathbf{x} - \mathbf{x}_0). \tag{24}$$

The system defined by (22) and (24) may then be solved to determine a new point on the branch. The identification of one single point, or one pre-flutter state, is referred to as one *step* within the algorithm and requires several *iterations* to be reached.

In practice, the process is initiated in wind off conditions and progresses in the increasing airspeed direction. Starting from $U_0 = 0$, where the initial state $\mathbf{x}_0$ is easily obtained from the wind-off solution, a nearby point on the branch is found by solving

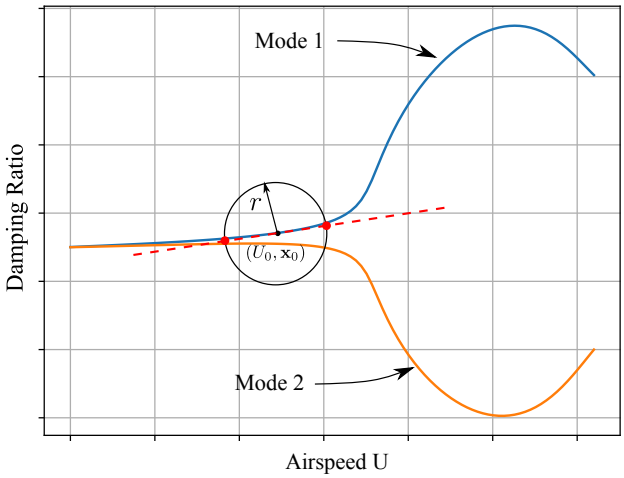


Figure 1: Illustration of the arc-length procedure in 2D space. The first iteration of the current step is investigated.

(22) and (24). For small sphere radius, there are two solutions, one at a higher one at a lower airspeed. The intersection at the highest of the two airspeeds is retained. This new point on the branch is then considered as the next starting point $(\mathbf{x}_0, U_0)$ and the process is repeated until flutter conditions are reached. The step length between two consecutive points in $\mathcal{D}$ is equal to r. However in a subspace $\mathcal{E} \subset \mathcal{D}$, the step length depends on the gradient of $\mathbf{f}$ in $\mathcal{D}$. For example, if the steps of length r separating three consecutive points of a branch are $(\Delta U_j, \Delta \mathbf{x}_j)$ and $(\Delta U_k, \Delta \mathbf{x}_k)$, the steps in the U-direction ΔU_j and ΔU_k may not be equal and, likewise, the step $||\Delta \mathbf{x}_j||$ may not be equal to $||\Delta \mathbf{x}_k||$. This is one of the advantages of arc-length methods with respect to the systematic solution of (22) for a constant airspeed step ΔU, as the latter is smaller only where required.

The arc-length algorithm presented here is adapted from [9], whose approach consists in solving (22) and (24) using numerical nonlinear solvers; it will be referred to in the present work as Riks method. Furthermore, it is possible to take advantage of the fact that the problem is quadratic in U to improve this method. Linearizing $\mathbf{f}(\mathbf{x}, U)$ in (22) around the i-th point on the branch, $(\mathbf{x}_i, U_i)$, and solving for $\mathbf{x}$ gives

$$\mathbf{x} = \mathbf{x}_i - \mathbf{J}_{\mathbf{x}}^{-1}(\mathbf{x}_i, U_i)\left[\mathbf{f}(\mathbf{x}_i, U_i) + \mathbf{J}_U(\mathbf{x}_i, U_i)(U - U_i)\right] \tag{25}$$

with $\mathbf{J}_{\mathbf{x}}(\mathbf{x}, U) = \dfrac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}, U)$ and $\mathbf{J}_U(\mathbf{x}, U) = \dfrac{\partial \mathbf{f}}{\partial U}(\mathbf{x}, U)$. Substituting (25) in (24) results in a quadratic equation in the sole unknown U

$$(1 + \mathbf{b}^T\mathbf{S}^T\mathbf{S}\mathbf{b})U^2 - 2(U_0 - \mathbf{a}^T\mathbf{S}^T\mathbf{S}\mathbf{b})U + U_0^2 - (r\ U_{\text{ref}})^2 + \mathbf{a}^T\mathbf{S}^T\mathbf{S}\mathbf{a} = 0 \tag{26}$$

with

$$\begin{aligned} \mathbf{a} &= \mathbf{x}_i - \mathbf{x}_0 - \mathbf{J}_x^{-1}(\mathbf{x}_i, U_i)\left[\mathbf{f}(\mathbf{x}_i, U_i) - \mathbf{J}_U(\mathbf{x}_i, U_i)U_i\right] \\ \mathbf{b} &= -\mathbf{J}_{\mathbf{x}}^{-1}(\mathbf{x}_i, U_i)\mathbf{J}_u(\mathbf{x}_i, U_i). \end{aligned}$$

This quadratic equation in U is easily solved analytically to obtain the two intersections with the hypersphere of radius r. As mentioned earlier, the rightmost solution is retained. Rejecting systematically the lowest root prevents any unwanted changes in the continuation direction. The new flight speed U is now used to evaluate a new $\mathbf{x}$ by solving (22)

using a Newton–Raphson or Broyden method. The idea of exploiting the quadratic nature of the continuation equation (24) was first raised by [36]. In the present work, this approach will be referred to as Crisfield's method.

It must be highlighted that both Crisfield's and Riks' methods are tracking strategies. All continuation methods augment the nonlinear system by adding one additional tracking equation. This larger system must be solved with a nonlinear solver. Therefore, a continuation algorithm combines a tracking procedure and a nonlinear solver.

Note that, in regions where the curvature of the eigensolution branch is very high, the linearization used to obtain (25) may result in an airspeed prediction that lies too far from the branch for the Newton–Raphson calculation to converge. In such situations, the value of r may need to be reduced, by means of an arc-length step adaptation procedure. Note also that, as long as the solver converges, using a coarse radius r only affects the resolution of the branch, not the accuracy, as the latter is ensured by the convergence criterion.

As a summary of this development section, a flowchart describing the workflow of the continuation methods is proposed in Figure B.20 of Section Appendix B. This workflow is illustrated for a full multi-modal *pre-flutter analysis*, including the three loops on the modes, the wind speed U and the iterative loop of the nonlinear solver.

4.3. Hypersphere radius adaptive refinement strategies

The choice of the radius r depends on the mesh density required by the user. It must be chosen based on the scaling parameters U_{ref}, λ_{ref}, ... that are calibrated on the expected orders of magnitude desired for each component $j = 1, \ldots, N$ of the step $x_j - x_{0,j}$ and $U - U_0$. The procedure for the selection of these scaling parameters is discussed in Section 5.2. The initial radius is typically chosen of order 1, but is highly correlated to the scaling parameters chosen by the user. After one or several iterations, the radius r may be updated to improve the behavior of the solution: decreasing r produces a better guess for the subsequent iterations, while increasing it will decrease the computational cost. Different strategies for adapting the radius to the gradient of the objective function have been proposed. One such strategy is to define an upper and a lower threshold for the number of iterations to convergence of the previous step above or below which the current radius is multiplied or divided by a chosen factor $m \in \mathbb{R}_0$ —for instance $m = 2$. One problem with this approach is that, depending on m, either the radius becomes rapidly very small (or very large), or it may take a long time for the radius to stabilize to the appropriate value when m is close to unity.

Another strategy depends on the number of iterations required for the Newton–Raphson system to converge at two consecutive points on the branch. If point $i - 1$ is obtained after m_{i-1} iterations and point i after m_i, a convergence ratio can be defined as

$$\beta_i = \frac{m_{i-1}}{m_i}. \tag{27}$$

Then, the value of the radius used to predict point $i + 1$ is given by

$$r_{i+1} = \beta_i r_i \tag{28}$$

such that β_i becomes an amplification or reduction factor. In practice, it is recommended to limit the number of iterations to around 20. If convergence has not been achieved after this number of iterations, it is possible to return to point i, or even $i - 1$, and reduce r by a bigger factor, *e.g.* $r_{i+1} = r_i/2$.

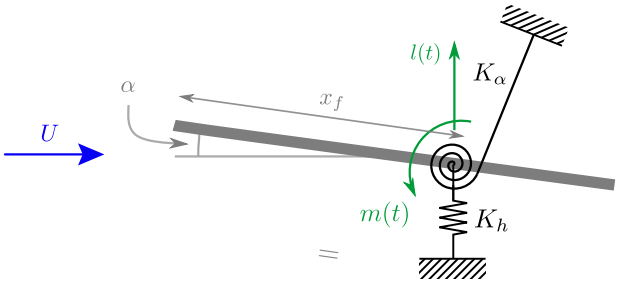


Figure 2: Schematic representation of the flat plate used in illustration 1 and 2.

While the first strategy allows the radius to take theoretically any value in a discrete set $\{r_0 \cdot m^p\}$ with $p \in \mathbb{Z}$, the second strategy is more flexible and allows r to take any value in $\mathbb{R}_0$. It is also more reactive as it accommodates brutal changes in gradients, and adapts in consequence the radius rapidly, while the first method would likely require many more updates of the radius before stabilizing. The second method is used in the following examples.

5. Illustrations

In this section, the behavior of the algorithm will be illustrated on three selected examples. The first two illustrations are both 2-mode applications; a bridge deck, and a theoretical airfoil. The last application consists in a generic 7-mode aircraft model. The robustness of the continuation methods introduced previously will be discussed, and their performance in terms of computational burden and accuracy will be compared investigating two criteria: the number of iterations and the number of function evaluations.

It is important to note that the number of iterations may not be appropriated to compare the computational efficiency of two different methods, as the load associated with an iteration is not necessarily the same for each method. The three continuation methods have similar workloads in a given iteration, but these workloads are different from that of the p-k method. Therefore computational efficiency can be discussed based on the number of iterations to compare the continuation methods between themselves, but not to compare the continuation and p-k methods. The same applies to the second criterion, the number of function calls.

It must also be kept in mind that a fair comparison of the cumulated number of iterations is not possible when two computed curves have different numbers of points; more points require inevitably more iterations. To allow for a fair comparison, the continuation methods are set to use the same refinement strategy and the same sphere initial parameters —see Table 2. Concerning the p-k method, the number of points was chosen to match approximately that of the other three methods.

5.1. Profiled bridge deck modeled with Scanlan's derivatives

The first test case is a pitch/plunge model of an idealized flat plate, namely the simplest model encapsulating the essential aerodynamic phenomena from which the unsteady aerodynamics of coupled aeroelastic systems may be investigated. The simplicity of this model makes it an ideal benchmark for validating computational tools and simulations.

Table 1: Parameters for the first two proposed case studies. All the parameters are measured in wind-off conditions ($U = 0$ m/s). Variables $\xi_{s,\alpha}$ and $\xi_{s,h}$ respectively refer to the structural damping in pitch and plunge motion.

Parameters	**App. 1**	**App. 2**	**Units**
Mass moment of Inertia I_α	2.46×10^6	0.0703	[kg.m^2/m]
Mass moment of inertia I_α	2.46×10^6	0.0703	[kg m^2/m]
Foil/Deck mass m_s	22 470	13.5	[kg/m]
Natural frequency $f_{s,h}$	0.1	5	[Hz]
Natural frequency $f_{s,\alpha}$	0.278	6.5	[Hz]
Damping ratios $\xi_{s,\theta}$ and $\xi_{s,h}$	0.3 / 0.3	0 / 0	[%]
Foil/Deck width c or B	31	0.25	[m]
Pitch axis x_f	$B/2$	$0.46 \cdot c$	[m]
Air density ρ	1.22	1.22	[kg/m^3]
Aeroelastic model	Theodorsen	Theodorsen+Jones	–

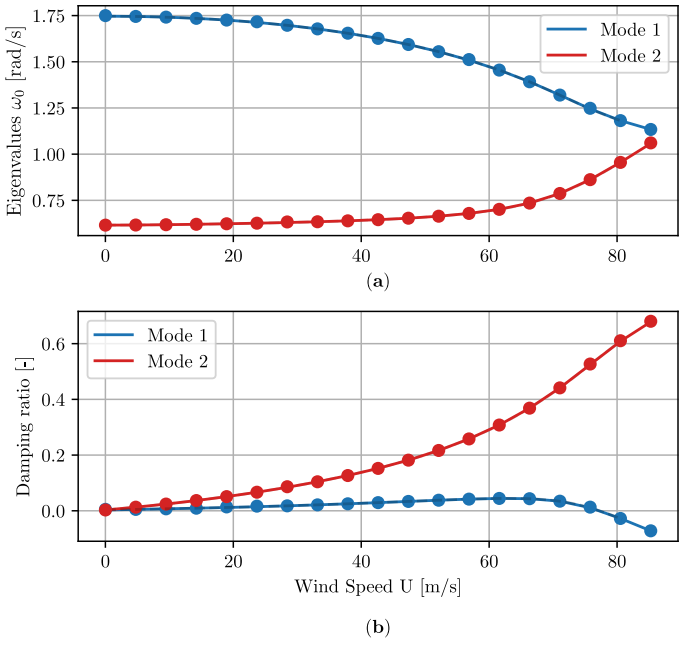


Figure 3: Application 1.Variation of natural frequencies and damping ratios with airspeed, calculated using the p-k method. The circle markers ● denote the pre-selected airspeeds at which the analysis was carried out.

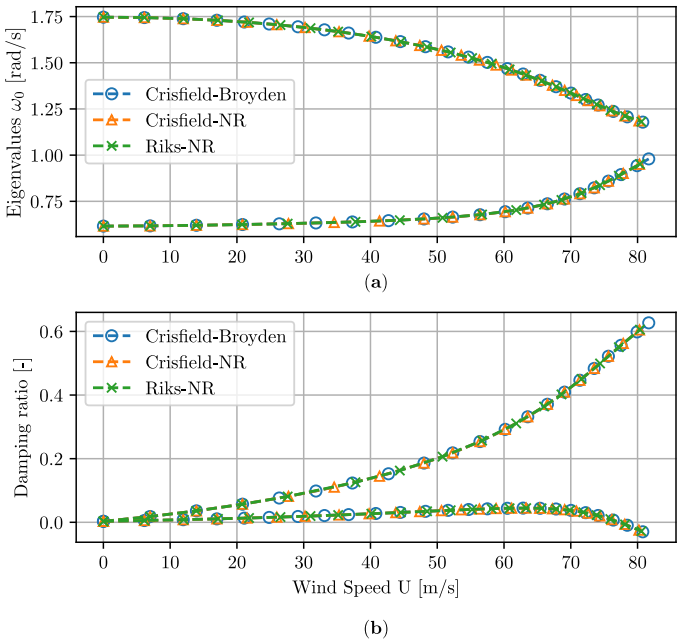


Figure 4: Application 1. Variation of natural frequencies and damping ratios with airspeed, calculated using the three continuation methods.

In this case study, a bridge deck is modeled as a pitch/plunge model of a flat plate, as proposed in the benchmark published by [37]. A drawing of the deck section subjected to aeroelastic loading is shown in Figure 2, while table Table 1 summarizes all the parameter values used in this example. Adopting Scanlan's nomenclature for the aeroelastic loads, the frequency domain representations of the lift $l(t)$ and moment $m(t)$ are described by the aeroelastic matrices $\mathbf{M}_{\text{ae}} = \mathbf{0}$, $\mathbf{C}_{\text{ae}}$ and $\mathbf{K}_{\text{ae}}$, obtained from equation (5),

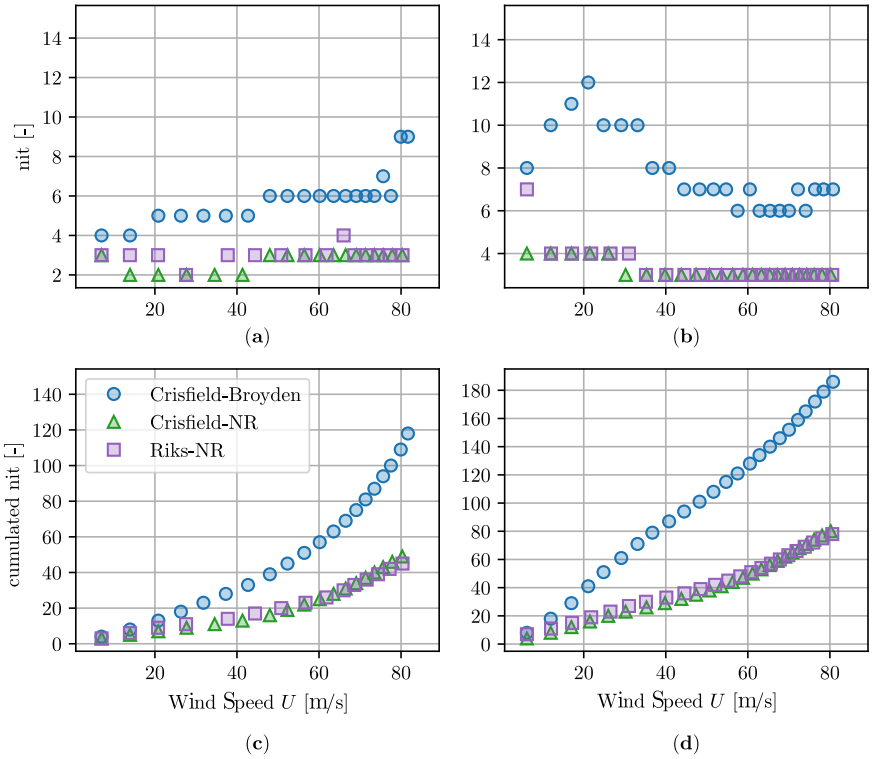
$$\begin{aligned} \mathbf{C}_{\text{ae}}(\omega, U) &= \frac{1}{2}\rho U B K \begin{bmatrix} H_1^*(K) & BH_2^*(K) \\ BA_1^*(K) & B^2A_2^*(K) \end{bmatrix} \\ \mathbf{K}_{\text{ae}}(\omega, U) &= \frac{1}{2}\rho U^2 B K^2 \begin{bmatrix} H_4^*(K)/B & H_3^*(K) \\ A_4^*(K) & BA_3^*(K) \end{bmatrix} \end{aligned} \tag{29}$$

where the flutter derivatives $H_1^*(K)$, $H_2^*(K)$, $H_3^*(K)$, $H_4^*(K)$ and $A_1^*(K)$, $A_2^*(K)$, $A_3^*(K)$, $A_4^*(K)$ are functions of the reduced frequency $K = \omega B/U = 2k$ and the wind speed U. These functions are analytically derived from [38], neglecting the added mass effects that are usually low in wind engineering applications. This leads to the following equations as established by [4] for the case where the rotation axis is located at the center of the plate ($x_f = c/2$)

$$\begin{aligned} H_1^* &= -V^*F, & A_1^* &= -\tfrac{V^*}{4}F \\ H_2^* &= \tfrac{V^*}{4}\left(1 + F + V^*G\right) & A_2^* &= -\tfrac{V^*}{16}\left(1 - F - \tfrac{2}{\pi}V^*G\right) \\ H_3^* &= \tfrac{V^*}{2\pi}\left(FV^* - \tfrac{\pi}{2}G\right) & A_3^* &= \tfrac{V^*}{8\pi}\left(FV^* - \tfrac{\pi}{2}G\right) \\ H_4^* &= \tfrac{\pi}{2}\left(1 + \tfrac{2}{\pi}V^*G\right) & A_4^* &= \tfrac{V^*}{4}G, \end{aligned} \tag{30}$$

where $V^* = 2\pi/K$, F and G are respectively the real and complex parts of Theodorsen's

Figure 5: Application 1. Number of iterations (nit) and cumulated number of iterations for Mode 1 in (a,c) and Mode 2 in (b,d). The conclusion regarding the computational loads of the continuation methods should be made with care, as the workload per iteration for each method is not exactly the same.



function $C(k) = F + \mathrm{i}G$, given by

$$C(k) = \frac{-J_1(k) + \mathrm{i}Y_1(k)}{-(J_1(k) + Y_0(k)) + \mathrm{i}(Y_1(k) - J_0(k))} \tag{31}$$

$k = \omega b/U$, $J_0(k_i)$, $J_1(k_i)$ are Bessel functions of the first kind and $Y_0(k_i)$, $Y_1(k_i)$ are Bessel functions of the second kind.

Once the aeroelastic loads are prescribed, the pre-flutter analysis can be carried out using the four methods presented in the previous section. The first method is the p-k method, and the others are three continuation methods: the Riks-Newton–Raphson (Riks-NR) couples the Riks method with a Newton–Raphson solver, while the Crisfield–Newton–Raphson (Crisfield–NR) and the Crisfield–Broyden techniques, respectively implement the Crisfield method together with a Newton–Raphson and Broyden solver, see for instance [33].

The results obtained with the p-k method are shown in Figure 3. In this figure, the circle markers • denote the pre-selected airspeeds at which the system was solved. It can be seen that the natural frequencies of the two modes vary with airspeed, approaching each other as the airspeed increases. Both damping ratios initially increase with airspeed but, for $U > 63$ m/s, the damping of mode 1 starts to decrease while the other increases faster. This phenomenon is the typical binary flutter mechanism, whereby one of the modes transfers all of its energy to the other, such that the latter becomes undamped at $U = 75.8$ m/s and negatively damped at higher airspeeds.

The flutter solutions obtained using the three continuation methods are shown in Figure 4. Every single point of every curve is converged, as it satisfies the condition of maximum required tolerance on the residual defined as the euclidean norm of the objective function $\mathbf{f}(\mathbf{x}, U)$,

$$\mathrm{Res} = \sqrt{\sum_{i=1}^{N} f_i^2(\mathbf{x}, U)} \leq \texttt{abstol} = 10^{-4}, \tag{32}$$

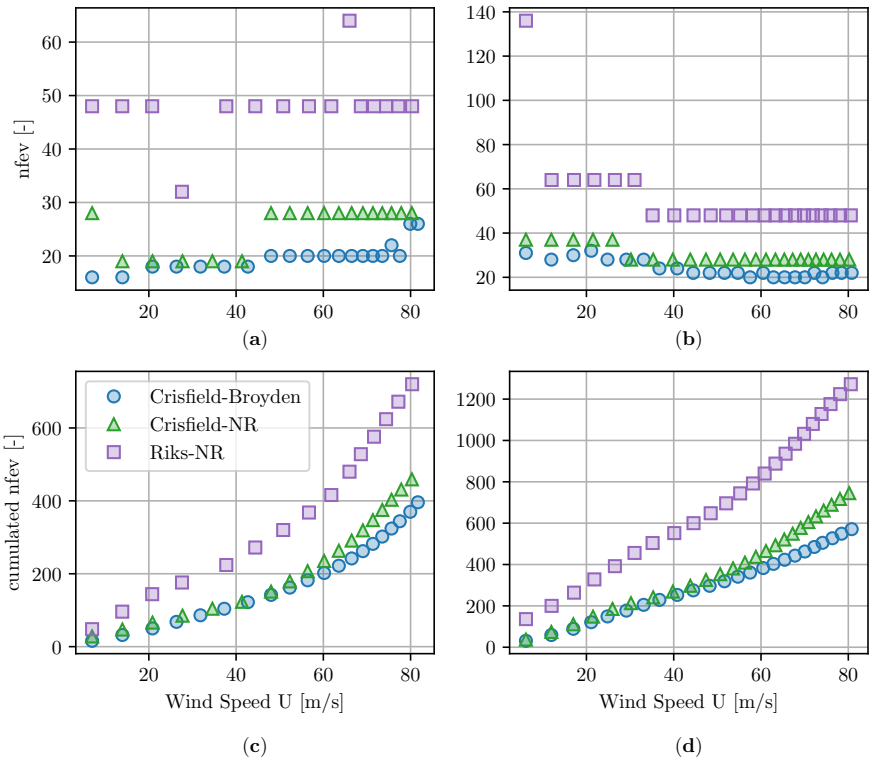
so that all results are the same and all curves are overlaid within the user defined tolerance.

Contrary to the p-k method, for which a uniform mesh in wind velocity U was imposed, each of the other three methods uses different custom nonuniform meshes. This is because of the adaptive refinement strategy of the sphere radius r, that increases or decreases the step between two consecutive points based on the number of iterations required by the method to converge, but also because of the gradient in $\mathcal{D}$. Each analysis is however initiated with the same algorithm parameters (see Table 2), and these are automatically adapted depending on the method's progress. The second refinement strategy presented in Section 4.3 was used for all examples presented in this article.

The convergence speed of the methods may be discussed by comparing the number of iterations required for each point to converge, depending on the chosen method. Figure 5 shows in (a) and (b) the number of iterations required by the four methods to obtain every point of the pre-flutter curve. It is seen that the Crisfield–Broyden methods is always more iteration demanding. This is because this method has an inexact or even no knowledge of the Jacobian matrix. As a consequence it progresses in a sub-optimal manner to find the solution of the system. The other two methods have equivalent performance concerning the number of iterations. Figures (c) and (d) show the cumulated number of iterations and depict the clear separation between the two groups of methods.

The number n_{fev} of evaluations of the target function $\mathbf{f}(\mathbf{x}, U)$ for the continuation methods is presented in Figure 6(a) and (b), and its cumulate is shown in (c) and (d). Clearly, Riks method requires significantly more function calls than the others with

Figure 6: Application 1. Number of calls (nfev) and cumulated number of calls to target function $\mathbf{f}(\mathbf{x}, U)$ for mode 1 in (a,c) and mode 2 in (b,d).



roughly 50 calls per point on average, as opposed to the Crisfield methods that require approximately half of that. The Broyden technique is slightly less expensive than the Newton–Raphson method, since it estimates the Jacobian based on the current and previous evaluations of $f(\mathbf{x}, U)$. For this flutter analysis, the Crisfield–Broyden method saves approximately 30% of the total number of calls with respect to the Crisfield–Newton–Raphson approach, and more than 130% with respect to the Riks method.

5.2. Pitch-Plunge model of an idealized flat airfoil

The second application is also a flat plate, but the lift and moment are expressed as [13]

$$m(t) = \left[\frac{\rho\pi b^4}{8}\omega^2\alpha + \pi\rho U e c^2 C^\dagger(k)\left(U\alpha + \mathrm{i}\omega h + \left(\frac{3}{4}c - x_f\right)\mathrm{i}\omega\alpha\right)\right.$$
$$\left. - \left(\frac{3}{4}c - x_f\right)\rho\pi b^2 U\mathrm{i}\omega\alpha + \left(x_f - \frac{c}{2}\right)\rho\pi b^2\left(-\omega^2 h + \left(x_f - \frac{c}{2}\right)\omega^2\alpha\right)\right] e^{\mathrm{i}\omega t} \quad (33)$$

and

$$l(t) = \left[\rho\pi b^2 U\mathrm{i}\omega\alpha + \pi\rho U c C^\dagger(k)\left(U\alpha + \mathrm{i}\omega h + \left(\frac{3}{4}c - x_f\right)\mathrm{i}\omega\alpha\right)\right.$$
$$\left. + \rho\pi b^2\left(-\omega^2 h + \left(x_f - \frac{c}{2}\right)\omega^2\alpha_0\right)\right] e^{\mathrm{i}\omega t} \quad (34)$$

with c referring to the chord, x_f the distance from trailing edge to the pitch axis, while $b = c/2$ is the half-chord and $e = x_f/c - 1/4$. The variables h and α_0 represent the modal amplitude for the pitch and plunge motions, respectively, and finally, the function $C^\dagger(k)$ stands for Jones' approximation of the Theodorsen function [39],

$$C^\dagger(k_i) = 1 - \frac{0.165}{1 - \frac{0.0455}{k_i}\mathrm{i}} - \frac{0.335}{1 - \frac{0.3}{k_i}\mathrm{i}}. \tag{35}$$

Using the definition of equation (3),

$$\begin{aligned}
&\frac{1}{2}\rho U^2 \mathbf{Q}(k) \\
&= \begin{pmatrix}
-\omega^2 m + \pi\rho U c C^\dagger(k)\mathrm{i}\omega & \rho\pi b^2 U\mathrm{i}\omega + \rho\pi b^2 \left(x_f - \frac{c}{2}\right)\omega^2 \\
-\omega^2 \rho\pi b^2 & +\pi\rho U c C^\dagger(k)\left(U + \left(\frac{3}{4}c - x_f\right)\mathrm{i}w\right) \\
-\pi\rho U e c^2 C^\dagger(k)\mathrm{i}\omega & (\frac{3}{4}c - x_f)\rho\pi b^2 U\mathrm{i}\omega \\
+\left(x_f - \frac{c}{2}\right)\rho\pi b^2\omega^2 & -\pi\rho U e c^2 C^\dagger(k)\left(U + (\frac{3}{4}c - x_f)\mathrm{i}\omega\right) \\
& -\left(x_f - \frac{c}{2}\right)\rho\pi b^2\omega^2 - \frac{\rho\pi b^4}{8}\omega^2
\end{pmatrix}.
\end{aligned} \tag{36}$$

This aeroelastic model is similar to that used in Section 5.1, excepted that Jones' approximation [39] is used, and that the equations presented here accommodate the possibility that the rotation axis is not located at the foil center $x_f \neq c/2$. The structural matrices are built according to Figure 2. Introducing the plate mass m_s per unit length,

$$\mathbf{K}_s = \begin{bmatrix} K_h & 0 \\ 0 & K_\alpha \end{bmatrix}, \quad \mathbf{C}_s = \mathbf{0}, \quad \mathbf{M} = \begin{bmatrix} m_s & S \\ S & I_\alpha \end{bmatrix} \tag{37}$$

where $I_a = \frac{m_s}{3}\left[c^2 - 3(x - c_f)x_f\right]$ is the mass moment of inertia of the plate per unit length, with respect to the rotation axis and $S = m\left(\frac{c}{2} - x_f\right)$ the static imbalance. The stiffnesses K_h and K_α are evaluated from the wind-off natural frequencies f_α and f_h.

The pre-flutter modal analysis is first carried out using the p-k method to give the results shown in Figure 7. There is one mode swapping around 30 m/s. This is in this case the lesser evil knowing that those points are by far supercritical —the critical velocity is slightly lower than 25 m/s— but illustrates well the possible shortcomings of the method. An attempt at automating data post-processing is presented in Section Appendix A to discard this problem of mode swapping by restoring mode shapes continuity by means of the scalar product of computed mode shapes and mode shapes obtained at the previous step. A detailed code snippet of the developed algorithm is proposed in Figure A.18. The processed output corresponding to the data shown is detailed in Figure 15 is represented in Figure A.19, illustrating a failure of the algorithm to prevent mode swapping, despite the sorting correction.

Concerning mode swapping, the continuation methods behave much better: there is no mode swapping because the analysis is conducted separately for each mode. In the post-critical regime, where the p-k method requires more iterations for convergence, the continuation methods converge without any trouble. However, the adaptive meshing feature of the continuation methods reduces automatically the radius r in this regime, as seen by the larger mesh density around 25 m/s in Figure 8. When the algorithm converges less rapidly, smaller radii are chosen as discussed in Section 4.3, and, on the contrary, a larger radius is selected when the solver converges faster. To illustrate the behavior of the algorithm, the variation of the radius with respect to U is shown in Figure 9. For small wind speeds, the radius is constant; it decreases significantly at around 25 m/s, only to increase again at higher airspeeds. Methods that converge faster are more likely to use a

Figure 7: Application 2. Variation of natural frequencies and damping ratios with airspeed, calculated using the p-k method. Mode swapping occurs around $U = 30$ m/s.

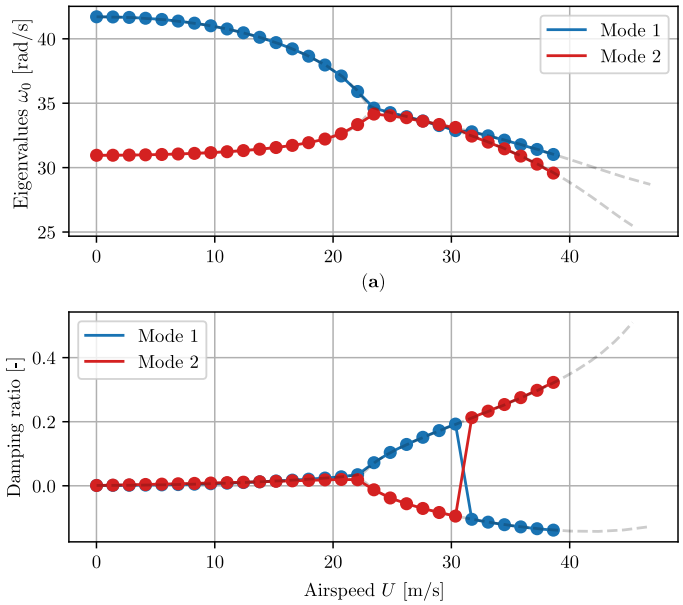
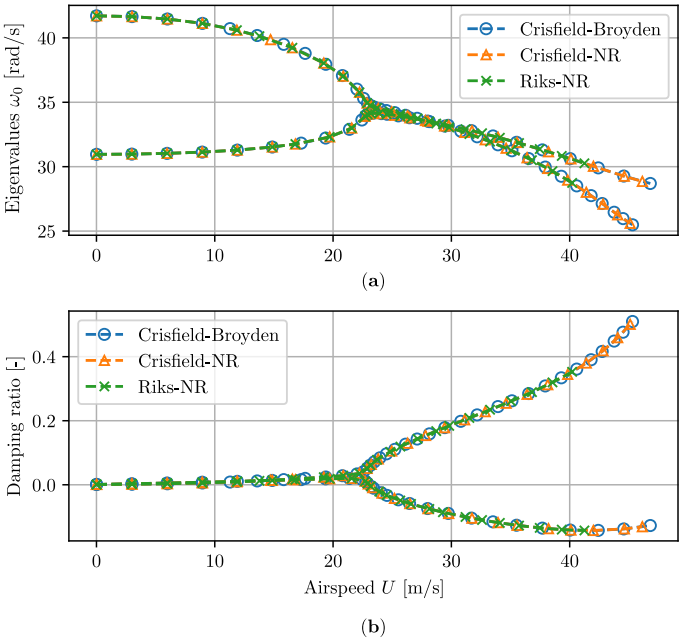


Figure 8: Illustration 2. Results of the pre-flutter analysis (a) Undamped eigenvalues and (b) Damping ratios for the first and second modes derived with the three continuation methods.



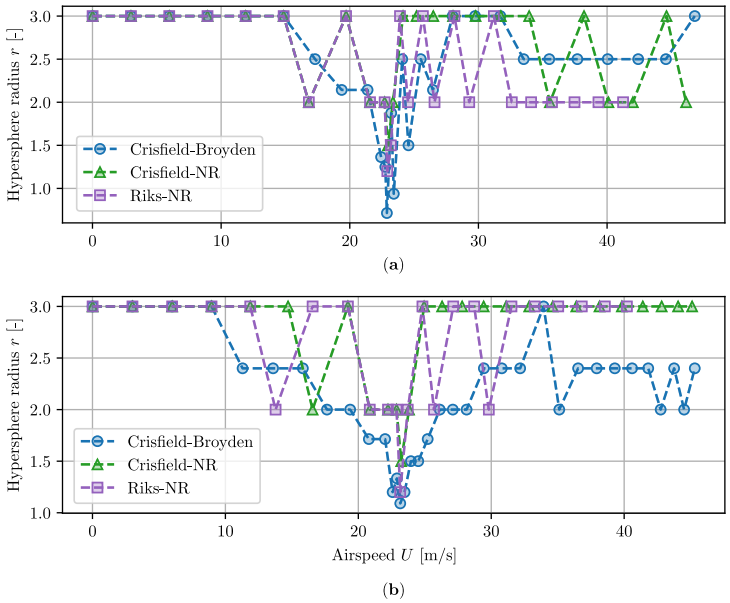


Figure 9: Application 2. Evolution of the sphere radius with respect to the wind speed U for the 3 continuation methods.

less dense mesh. For instance, the Crisfield–NR method seems to use globally larger radii than the other methods. The second reason for this local refinement is a consequence of the variation in gradients $\mathbf{J}_\mathbf{x}$ and $\mathbf{J}_U$. Indeed, the radius r^2 given in equation (24) is expressed as a weighted sum of $\Delta U^2 = (U - U_0)^2$ and the squared norm of the step vector $\mathbf{dx} = \mathbf{x} - \mathbf{x}_0$ derived from equation (25)

$$\mathbf{x} - \mathbf{x}_0 = -\mathbf{J}_\mathbf{x}^{-1}(\mathbf{x}_0, U_0)\left[\mathbf{f}(\mathbf{x}_0, U_0) + \mathbf{J}_U(\mathbf{x}_0, U_0)(U - U_0)\right]. \tag{38}$$

Therefore, if the component related to the norm of $\mathbf{dx}$ is negligible with respect to r, the distance between two consecutive points will mostly be driven by $U - U_0$.

The behavior of the adaptive refinement is also sensitive to the scaling parameters provided by the user. The parameter U_ref is probably the easiest to fix. For a radius $r \approx 1$, it can be chosen so that, in a given step where all variations $dx_j = \frac{x_j - x_{0,j}}{x_{\text{ref},j}}$ are zero for all j, the distance between two consecutive points $U_i - U_{0,i}$ is equal to rU_ref. The same principle is used to select $x_\text{ref,j}$, which must be chosen such that the largest expected step in the j direction $dx_j = \frac{x_j - x_{0,j}}{x_{\text{ref},j}}$ is at most equal to r. Hence, if all variations dU and dx_i are negligible, with $i \neq j$, the largest allowed step $x_j - x_{0,j}$ is $rx_{\text{ref},j}$.

The choice of the scaling parameter is also a tool for the improvement of the convergence of the method. For instance, if the user is able to identify directions in $\mathcal{D}$ along which no or almost no variation of the modal properties is observed, these can be discarded to drive the algorithm towards preferential directions where large gradients are observed, reducing hence the number of variables effectively contributing to the hypersphere radius. Such an approach may be referred to as a *cylindrical arc-length method* (see [41]), by reference to the shape of the continuity equation, initially spherical, that degenerates into a cylinder due to large x_ref. The use of a cylindrical continuation scheme is useful in large systems, where local contributions to the hypersphere radius of some directions in $\mathcal{D}$ are annihilated by a much larger fraction of other directions along which

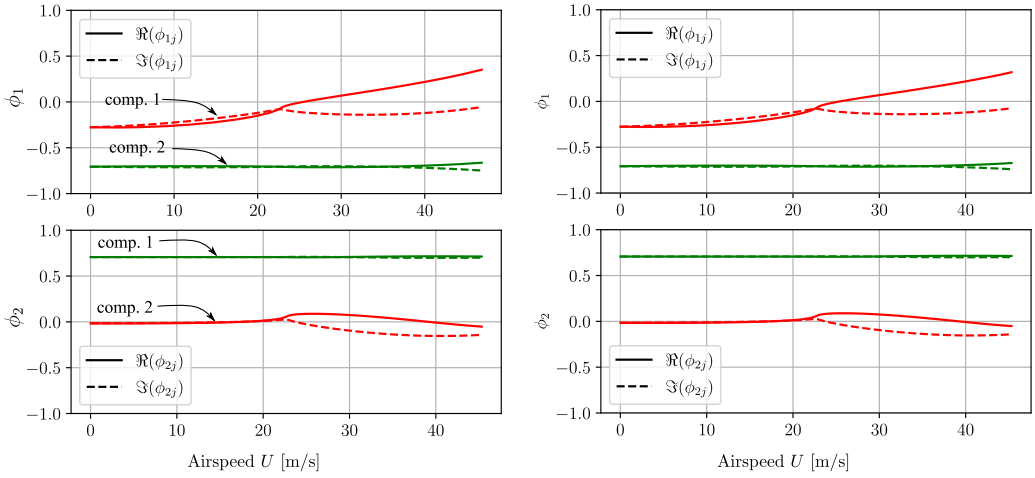


Figure 10: Application 2. Effect of the scaling parameters related to the mode shapes $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$. (a) Cylindrical arc-length: $x_{\text{ref},j} = 100$, and (b) spherical arc-length $x_{\text{ref},j} = 0.01$.

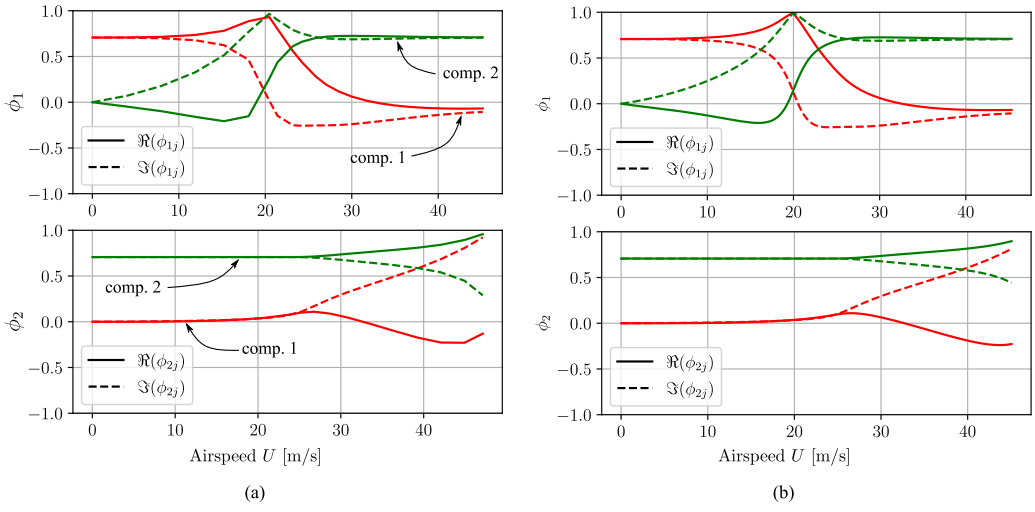


Figure 11: Application 2, with $x_f/c = 0.5$ instead of 0.46. Effect of the scaling parameters related to the mode shapes $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$. (a) Cylindrical arc-length: $x_{\text{ref},j} = 100$, and (b) spherical arc-length $x_{\text{ref},j} = 0.01$. All other parameters are identical to those given in Table 2.

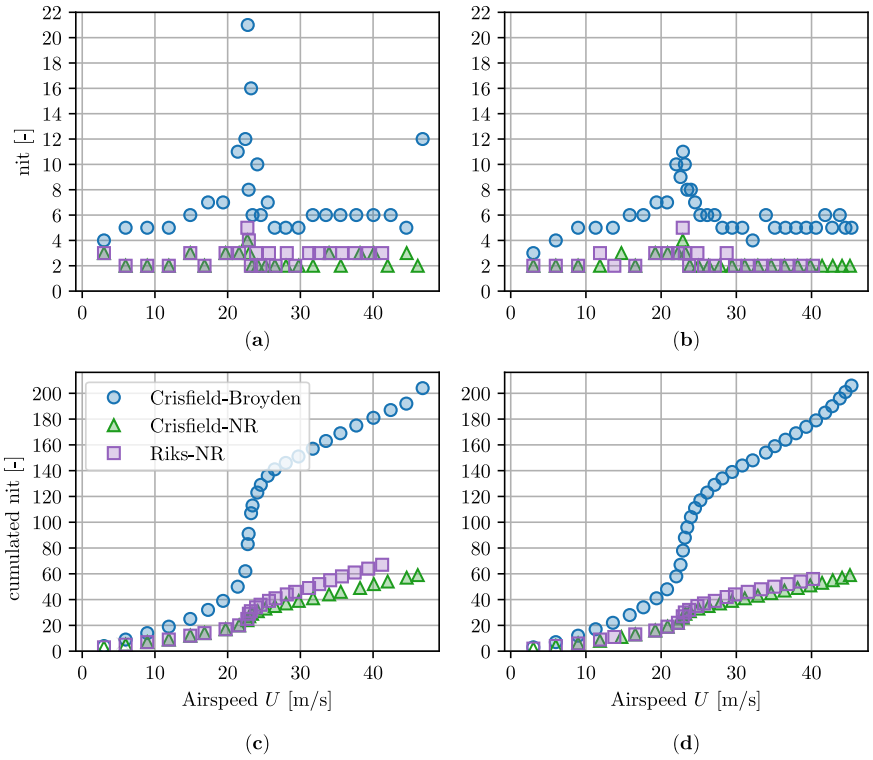


Figure 12: Application 2. Number of iterations (nit) and cumulated number of iterations for mode 1 in (a,c) and mode 2 in (b,d). The conclusion regarding the computational loads of the continuation methods should be made with care, as the workload per iteration for each method is not exactly the same.

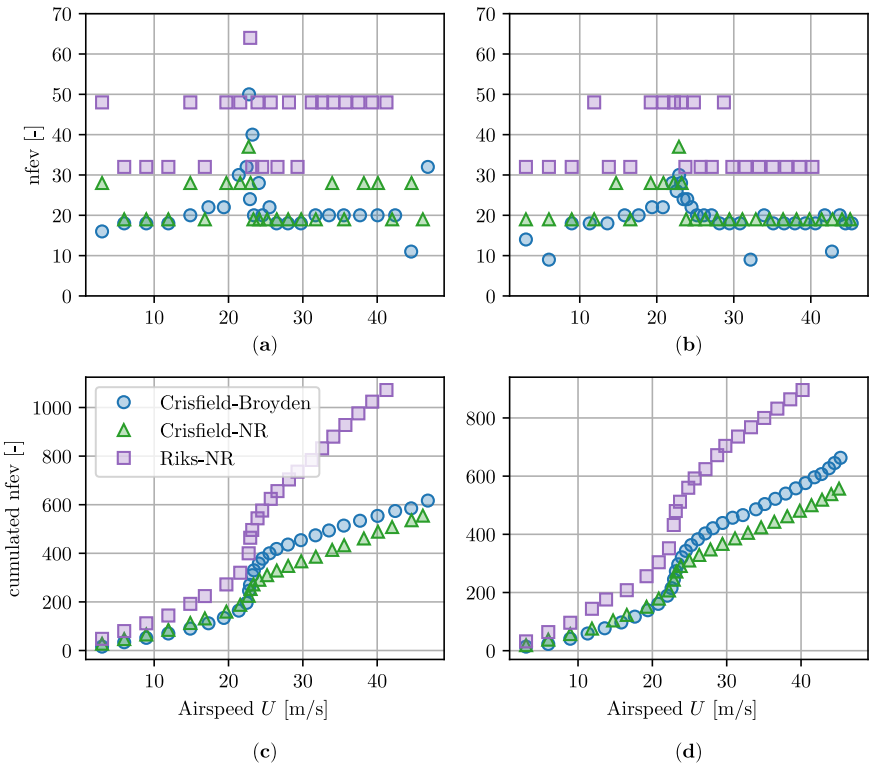


Figure 13: Application 2. Number of calls (nfev) and cumulated number of calls to target function $\mathbf{f}(\mathbf{x}, U)$ for mode 1 in (a,c) and mode 2 in (b,d).

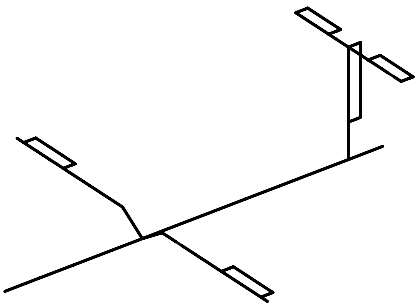


Figure 14: Schematic of the structural model of the GTA. Adapted from [40].

Table 2: Algorithms parameters used for the 3 proposed case studies.

Parameters	**App. 1**	**App. 2**	**App. 3**	**Units**
Initial radius r_0	0.7	3	4	[–]
Min. Radius $r_{\min}$	0.05	0.5	0.5	[–]
Min. Radius $r_{\max}$	2.	7	5	[–]
Ref. Speed U_{ref}	10	1	10	[m/s]
Ref. $\Re\{\lambda_i\}$	1000	0.2 / 0.3	1	[rad/s]
Ref. $\Im\{\lambda_i\}$	0.3	1	5	[rad/s]
Ref. coordinate $\phi_{ij,\text{ref}}$	1	100	1	[–]

no gradient is observed.

In the present case, the mode shapes exhibit a linear variation with airspeed U, and this tendency is already well captured with a few points, as seen in Figure 10(a). Adding more points by decreasing $\Re[\phi_{\text{ref}}]$ and $\Im[\phi_{\text{ref}}]$ will not improve the representation of $\boldsymbol{\phi}_i$, as illustrated in Figure 10, where a comparison between the mode shapes obtained with $\Re[\boldsymbol{\phi}_{\text{ref}}] = \Im[\boldsymbol{\phi}_{\text{ref}}] = 100$ in (a) and $\Re[\boldsymbol{\phi}_{\text{ref}}] = \Im[\boldsymbol{\phi}_{\text{ref}}] = 0.01$ in (b) is illustrated, but where no apparent difference is observed due to the low variations of the gradients $\frac{\partial \boldsymbol{\phi}}{\partial U}$. To illustrate the differences between cylindrical and spherical arc-length, the same comparison as that made in Figure 10 is carried out in Figure 11, where the pitch axis x_f of the considered structure is set to $0.5c$ instead of $0.46c$ to observe larger gradient variations in the modes shapes. This figure shows that the mode shapes obtained in (b) are significantly smoother than those in (a), but a reasonably good description was already achieved in (a). It is important to recall that each point in the curve is calculated with the same absolute tolerance and that the density of the mesh does not influence this accuracy. However, interpolation errors may sometimes be induced for instance if the modes shapes in Figure 11 (a) are used instead of those in (b). Therefore, the resolution of the resulting data must be verified to limit interpolation errors when interpolating between two consecutive points, as is required in order to pinpointed exactly the flutter airspeed [42, 43].

To analyze the performance of the methods, the number of iterations is compared in Figure 12 and contrasts well the behavior of the Riks and Crisfield methods. The latter reduce by a factor 2 or even 3 the number of iterations with respect to Riks method. It must be noted that around 23 m/s, the Crisfield–Broyden method failed to converge after 15 iterations — i.e. the maximum allowed number of iterations. The reason for this is that this point is close to the critical velocity, and important variations in the gradients of $\mathbf{f}(\mathbf{x}, U)$ are expected there, affecting the performance of Broyden's method, which estimates the Jacobian from previous and current values of $\mathbf{f}$. This particular point requires 21 iterations in total instead of the 15 allowed. Except for this detail, the Crisfield–Broyden and Crisfield–NR techniques perform equally well, as confirmed by Figure 13, reducing the number of function calls per iteration by a factor of 2 or 3, and dividing by 2 the total cumulated number of calls required for a full flutter analysis. The Crisfield–NR method slightly outperforms the Crisfield–Broyden approach as shown in Figure 13(c) and (d) in terms of function calls, but Newton–Raphson requires the evaluation of the Jacobian matrix which is estimated by Broyden's method. Hence, because the evaluation of $\mathbf{f}$ potentially represents two different computational loads, a time-based benchmarking seems to be the only way to arbitrate which method is the most time-efficient between Broyden and Newton–Raphson, if computational time is the governing concern.

5.3. Benchmark of the Generic Transport Aircraft (GTA)

The last illustration is a multi-mode model of a generic transport aircraft. The investigated prototype is taken from the ZAERO manual [44], and was used in the past by several authors [40, 45]. The model was generated using MCS-NASTRAN software and incorporates structural information from a simplified finite element model of the aircraft, whose mesh is schematically illustrated in Figure 14, and aerodynamic loads obtained using the doublet lattice method.

Using Roger's approximation [46], the aerodynamic forces $\mathbf{Q}(k)$ are written in the form

$$\mathbf{Q}(k) = \mathbf{A}_0 + \mathbf{A}_1 k + \mathbf{A}_2 k^2 + \sum_{j=1}^{n_l} \mathbf{A}_{2+j} \frac{k}{k+\gamma_j} \tag{39}$$

where $\mathbf{A}_j$ are $n_m \times n_m$ real matrices, n_m is the number of modes considered in the analysis and γ_j are aerodynamic lag coefficients given by

$$\gamma_j = -1.7 k_{\max} \frac{j}{(n_l+1)^2} \tag{40}$$

with $k_{\max}$ referring to the maximum reduced frequency at which the aerodynamic matrix $\mathbf{Q}(k)$ is evaluated by NASTRAN, and with $n_l = 4$. The number of retained modes is chosen equal to 7.

The variation of the natural frequencies and damping ratios with airspeed is displayed in Figure 15, revealing that the flutter mechanism is binary, while the other five modes are only slightly affected by aerodynamic effects, except their damping ratios which vary linearly with U. Figure 16 presents a less overloaded view of the eigenvalues, determined using the Crisfield–Broyden method in (a) and p-k method in (b). Once again, mode swapping can be observed at and airspeed of around 180 m/s in the predictions of the p-k method.

As this flutter analysis is conducted on a mode-by-mode basis, the inclusion of 7 modes in the system is not more difficult than the previous case studies involving only 2 modes. Naturally, this expansion entails an anticipated increase in computational workload: the single-mode analysis is now repeated 7 times instead of twice.

In addition, Figure 15 emphasizes one of the primary advantages of the arc-length method, which is its ability to fine-tune the mesh at specific wind speeds where refinement is necessary. In this example, modes 1 and 3 require a local refinement around 220 m/s, but a coarse mesh may be used elsewhere. This local refinement is well captured by continuation methods, in Figure 15(b) where the mesh density for mode 1 and 3 at 220 m/s is much higher than for speeds lower than 100 m/s. Furthermore, the mode-by-mode analysis allows a coarser mesh to be used for all other modes whose properties remain constant throughout the flight envelope. In this example, among the 7 modes in total, 5 of them remain mostly constant so that the local refinement required by mode 1 and 3 must not be transposed to other modes. Here for 7 modes in total, the absolute work load saving is not substantial. But when many modes are involved, mesh refinement allows for an appreciable potential reduction in the global computation load with respect to methods that use a uniform and likely fine mesh identical for all modes of the system.

The number of function evaluations is presented in Figure 17 for modes 1 and 3. It depicts once more the same tendency as observed in the first two illustrations: the two Crisfield methods converge more rapidly. For a full flutter analysis, the number of function evaluations is reduced by a factor of 3 approximately for modes 1 and 3.

Modes 2, 4, 5, 6 and 7 are also good candidates to illustrate the sensitivity of the algorithm to badly chosen scaling parameters U_{ref} and $\mathbf{x}_{\text{ref}}$. Taking a closer look, for

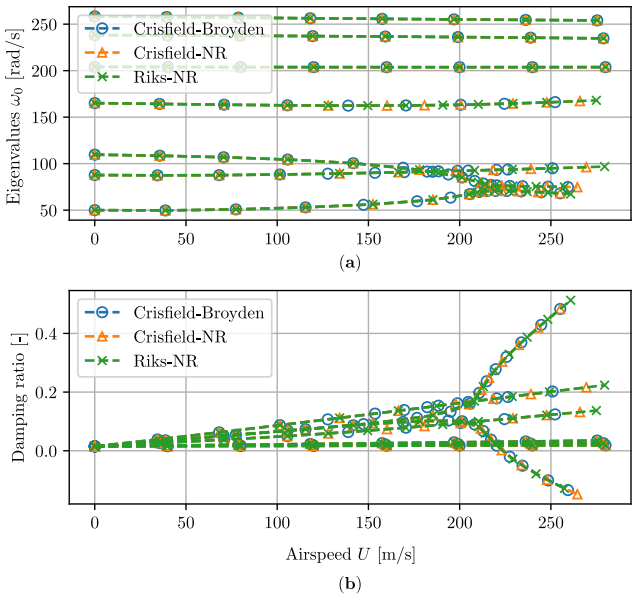


Figure 15: Illustration 3 (GTA Benchmark). Results of the *pre-flutter analysis* (a) Undamped eigenvalues and (b) Damping ratios derived with the 3 considered methods.

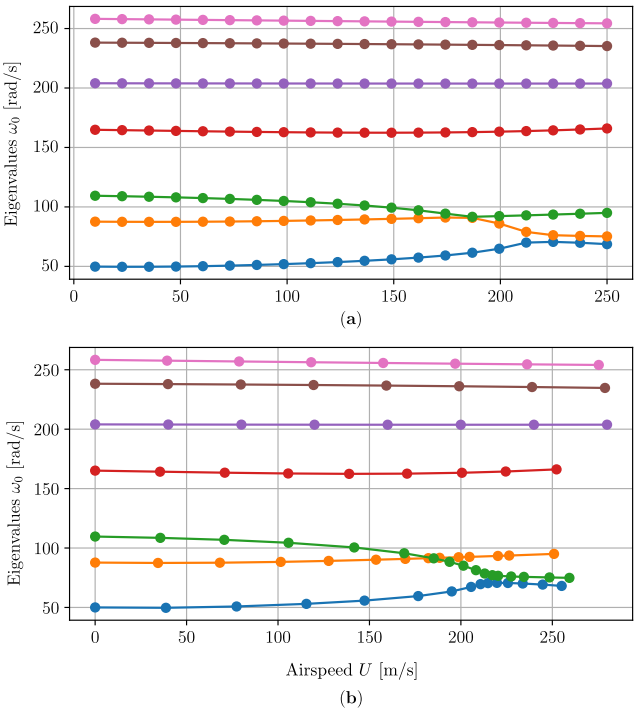


Figure 16: Illustration 3. Results of the *pre-flutter analysis* with (a) p-k method and with (b) Crisfield–Broyden. A mode swapping may be seen in (a) around 180 m/s.

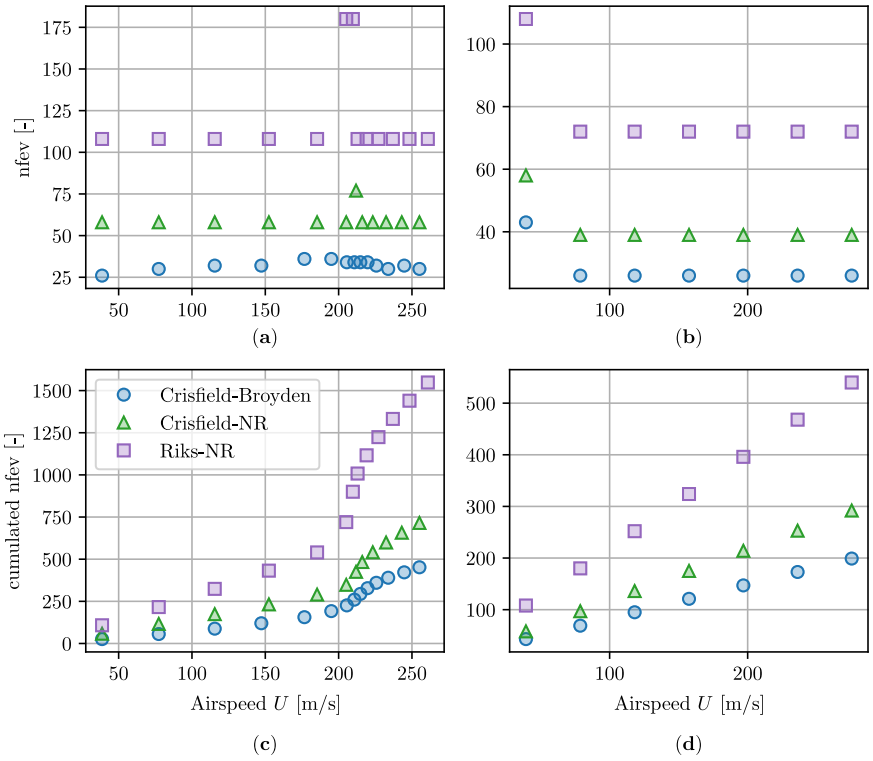


Figure 17: Application 3. Number of calls (nfev) and cumulated number of calls to target function $\mathbf{f}(\mathbf{x}, U)$ for mode 1 in (a,c) and mode 3 in (b,d).

instance at the 5th mode, it is evident that it does not affected significantly by the aerodynamic loads, at least within the specified airspeed range. This mode exhibits constant properties, including constant eigenvalue, almost constant damping ratio and constant mode shape; it is insensitive to U. As a consequence, the only variation expected in the space $\mathcal{D}$ will be exclusively in a direction parallel to U. In such a case, performing cylindrical steps by discarding the U direction from the line search direction —*i.e.* choosing $U_{\text{ref}} \to \infty$— results in a divergence of the algorithm. To illustrate this issue, if the analysis is conducted for the 5th mode with $U_{\text{ref}} = 100$ m/s instead of $U_{\text{ref}} = 10$ m/s as presented before, the algorithm diverges. Then, if the scaling parameters have a major importance on the quality of the local refinement performed by the algorithm, they could also potentially govern the stability of the solution when badly chosen. Therefore, their selection should sometimes be approached with careful consideration, especially in cases where gradients are unidirectional or almost unidirectional— *i.e.* nearly all columns of $\mathbf{J}_{\mathbf{x}}$ are empty. In these particular cases, the use of cylindrical arc-length methods is the most appropriate choice to ensure stability. One of the key principles to ensure the robustness of the method can be summarized as including in the continuation equation at least one direction of $\mathcal{D}$ that affects $\mathbf{f}(\mathbf{x}, U)$, and being aware that reducing the dimensions of the hypersphere does not necessarily improve the speed of convergence.

6. Conclusion

In this study, an arc-length continuation process was proposed as an alternative to the classical p-k method to perform flutter analysis of aeroelastic systems. The motivation was to construct a systematic and globally convergent tool to determine the variation of the system's modal properties with airspeed, providing crucial information for engineers engaged in modal vibration analysis, designing slender structures exposed to air flows. This process is applicable to any aeroelastic model, from bridge decks to aircraft wings.

The principle hinges on solving the nonlinear generalized eigenvalue problem for a chosen wind speed U to determine one *pre-flutter state*. The critical problem is first reformulated and transformed into a nonhomogeneous system of equations by incorporating the normalization condition. One continuity equation is then added to offer a smart tracking process. The process is initiated at wind-off conditions, and progresses step-by-step to the aeroelastic instability. The size of the steps are fixed by the hypersphere radius. In the proposed algorithm, the solution is carried out mode-by-mode, which means that each mode is considered separately. In that way, any undesired mode swapping phenomenon is prevented. Whenever modes behave independently from others, the algorithm automatically adapts and they are resolved with their own velocity mesh.

Two classes of continuation method have been described: Riks' approach and Crisfield's technique. The first consists in solving systematically the system of equations formed by the transformed critical problem and the continuity equation. The second method takes advantage of the knowledge of the analytical form of the continuity equation to solve it analytically. The second approach, was seen to be significantly faster than the first.

The continuation processes proposed here have proven to be appropriate to determine the flutter behaviour of a system, thanks to their ability to focus on regions where high local variations in modal properties are observed. By selecting suitable scaling parameter and sphere radius values, the user can guarantee that the algorithm will detect local variations that a uniformly distributed mesh —as those used with the traditional p-k method— could inadvertently miss. The fact that the analysis is conducted mode-by-mode brings another significant advantage regarding the use of computational resources. A custom mesh is used for each mode, which means that in large multi-mode systems, a coarser mesh will be chosen for modes that have mostly constant properties, resulting in a better optimization of the total computational load.

The performance of the methods was tested on three selected examples. In total, four techniques were assessed: the p-k method, Riks-Newton–Raphson, Crisfield–Newton–Raphson and Crisfield–Broyden. Among the three continuation methods considered, the Riks-Newton–Raphson was seen to be the most computationally expensive, requiring in total approximately 3 times more function evaluations than the Crisfield-based methods. The Crisfield–Broyden approach was the most efficient in terms of required number of calls to the objective function. No comparison of computational efficiency could be inferred between the p-k and continuation methods. Each continuation method demonstrated excellent stability performance, completing the flutter solution without mode swapping or convergence problems, even near critical points and in the post-critical regime, where the classical p-k method was shown to experience mode swapping in two of the test cases.

Declaration of conflict of interest

Julien Heremans reports financial support was provided by National Research Fund. If there are other authors, they declare that they have no known competing financial

```
def process_pre_flutter( U, w0, xi, phi ):

    # U: wind speeds vector of size nU
    # w0: matrix (2 x nU) containing eigenvalues
    # xi: matrix (2 x nU) containing damping ratios
    # phi: matrix (2 x 2 x nU) containing the eigen vectors
    #      consistently normalized

    w0_ = np.zeros( (2,len(U)), dtype='float64')
    xi_ = np.zeros( (2,len(U)), dtype='float64')

    w0_[:,0] = w0[:,0]
    xi_[:,0] = xi[:,0]

    dprod = np.zeros( 2, dtype='complex128' )
    for iU in range( 1, len(U) ):
        dprod[0] = phi[0,:,iU-1].T @ phi[0,:,iU] \
             + phi[1,:,iU-1].T @ phi[0,:,iU]
        dprod[1] = phi[0,:,iU-1].T @ phi[1,:,iU] \
             + phi[1,:,iU-1].T @ phi[1,:,iU]
        idmax = np.argmax( np.abs( dprod ) )
        idcmp = int(idmax == 0) # complementary index

        w0_[0,iU] = w0[idmax,iU]
        w0_[1,iU] = w0[idcmp,iU]

        xi_[0,iU] = xi[idmax,iU]
        xi_[1,iU] = xi[idcmp,iU]

    return w0_, xi_
```

Figure A.18: Function used to post-process the data obtained with the p-k method to prevent the mode swapping. The eigen modes given as inputs must be consistently normalized (see for example (21)).

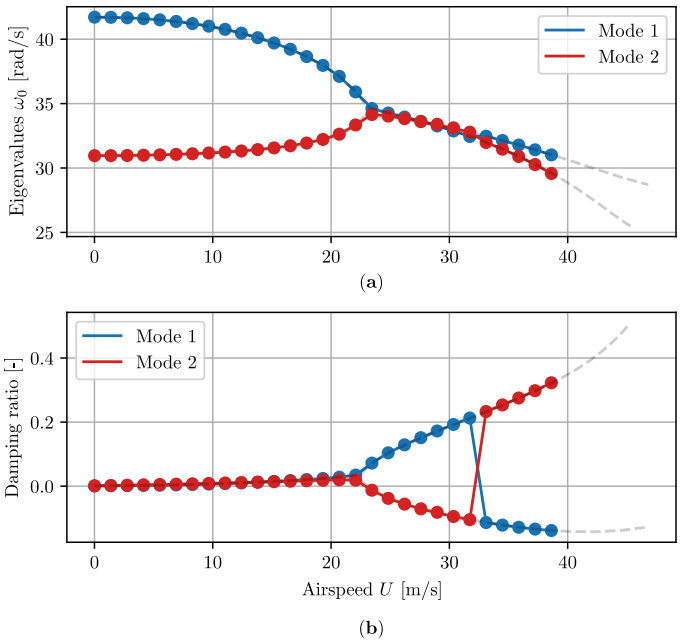


Figure A.19: Application 2. Failure of the algorithm that aims at preventing mode swapping. There is one mode swapping around 32 m/s.

interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This paper was partly supported by a doctoral grant of the FNRS Belgium `http://dx.doi.org/10.13039/501100002661` (Belgian Fund for Scientific Research) attributed to Julien Heremans. The involvement of Julien Heremans in this project was also made possible thanks to the research project FINELG2020 supported by the Service Public de Wallonie of Belgium `http://dx.doi.org/10.13039/501100009595`.

Appendix A. Prevention of mode swapping for the p-k method using eigen mode orthogonality

The p-k method is sometimes used together with a post-treatment to discard the issue of mode swapping. Figure A.18 shows the code that is used in order to sort modes at a given step, after they have been computed using the `eig` function. The chosen mode is the one that maximizes the dot product between the modes considered in the current step, and that kept in the previous step.

The corresponding results for application 2 are shown in Figure A.19, where the algorithm is seen to fail around 33 m/s.

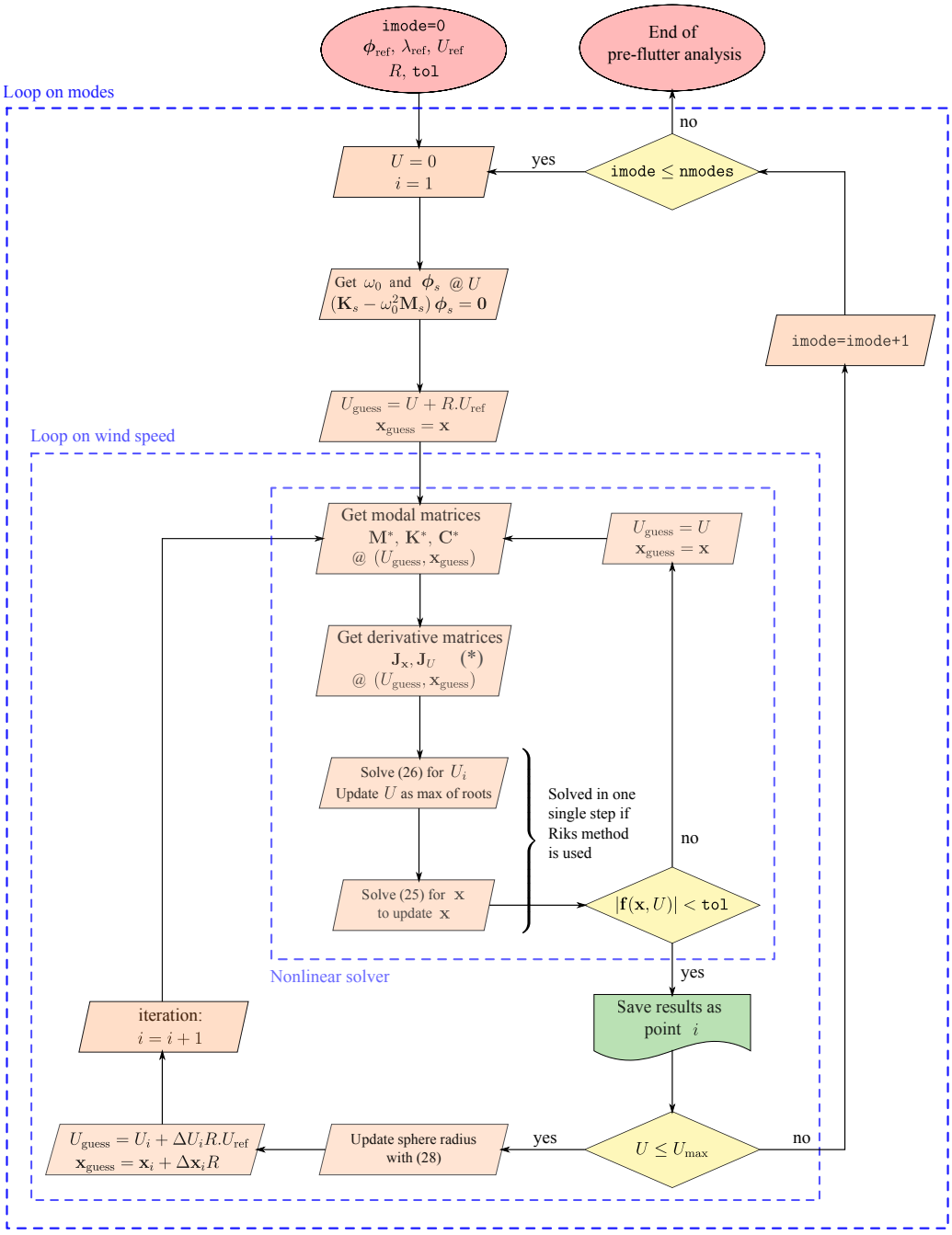


Figure B.20: Flowchart of a complete pre-flutter analysis. (*) If a Broyden method is used, the derivative matrices $\mathbf{J}_{\mathbf{x}}$ and $\mathbf{J}_U$ are estimated using the current and previous value of $\mathbf{f}(\mathbf{x}, U)$.

Appendix B. Workflow of a pre-flutter analysis

See Figure B.20.

References

[1] *EN 1992-1-4 Eurocode 1: Actions on structures - Part 1-4: General actions - Wind actions*, Bruxelles, 2005. CEN.

[2] *EN 1990:2002 Eurocode 0: Basis of structural design*, Bruxelles, 2002. CEN.

[3] Anurag Jain, Nicholas P. Jones, and Robert H. Scanlan. Coupled aeroelastic and aerodynamic response analysis of long-span bridges. *Journal of Wind Engineering and Industrial Aerodynamics*, 60:69–80, 1996. The Wind Engineering Society's 2nd UK Conference.

[4] Robert H. Scanlan. Problematics in formulation of wind-force models for bridge decks. *Journal of Engineering Mechanics*, 119(7):1353–1375, 1993.

[5] Xinzhong Chen, Ahsan Kareem, and Masaru Matsumoto. Multimode coupled flutter and buffeting analysis of long span bridges. *Journal of Wind Engineering and Industrial Aerodynamics*, 89(7):649–664, 2001. 10th International Conference on Wind Engineering.

[6] Xinzhong Chen and Ahsan Kareem. Advanced analysis of coupled buffeting response of bridges: a complex modal decomposition approach. *Probabilistic Engineering Mechanics*, 17(2):201–213, 2002.

[7] Hiroshi Katsuchi, Nicholas P. Jones, and Robert H. Scanlan. Multimode coupled flutter and buffeting analysis of the akashi-kaikyo bridge. *Journal of Structural Engineering*, 125(1):60–70, 1999.

[8] Quanshun Ding, Airong Chen, and Haifan Xiang. Coupled flutter analysis of long-span bridges by multimode and full-order approaches. *Journal of Wind Engineering and Industrial Aerodynamics*, 90(12):1981–1993, 2002. Fifth Asia-Pacific Conference on Wind Engineering.

[9] E. Riks. An incremental approach to the solution of snapping and buckling problems. *International Journal of Solids and Structures*, 15(7):529–551, 1979.

[10] E. Riks. Some computational aspects of the stability analysis of nonlinear structures. *Computer Methods in Applied Mechanics and Engineering*, 47(3):219–259, 1984.

[11] GB Warburton. Dynamics of structures, by ray w. clough and joseph penzien, mcgraw-hill, new york, 1993. no. of pages: 738. isbn 0-07-011394-7, 1995.

[12] Olek C Zienkiewicz, Robert L Taylor, and Jian Z Zhu. *The finite element method: its basis and fundamentals*. Elsevier, 2005.

[13] Grigorios Dimitriadis. *Introduction to Nonlinear Aeroelasticity*. Wiley, 2017.

[14] E. Jonsson, C. Riso, C. A. Lupp, C. E. S. Cesnik, J. R. R. A. Martins, and B. I. Epureanu. Flutter and post-flutter constraints in aircraft design optimization. *Progress in Aerospace Sciences*, 109:100537, 2019.

[15] G. Dimitriadis. *Unsteady Aerodynamics - potential and vortex methods*. Wiley - In print, 2024.

[16] E. Albano and W. P. Rodden. A doublet-lattice method for calculating lift distributions on oscillating surfaces in subsonic flows. *AIAA Journal*, 7(2):279–285, 1969.

[17] Robert H Scanlan and J Tomo. Air foil and bridge deck flutter derivatives. *Journal of Soil Mechanics & Foundations Div*, 1971.

[18] Robert H Scanlan. On flutter and buffeting mechanisms in long-span bridges. *Probabilistic engineering mechanics*, 3(1):22–27, 1988.

[19] Robert H. Scanlan, Jean-Guy Béliveau, and Kathleen S. Budlong. Indicial aerodynamic functions for bridge decks. *Journal of the Engineering Mechanics Division*, 100(4):657–672, 1974.

[20] C. B. Moler and G. W. Stewart. An algorithm for generalized matrix eigenvalue problems. *SIAM Journal on Numerical Analysis*, 10(2):241–256, 1973.

[21] George Lindfield and John Penny. Chapter 2 - linear equations and eigensystems. In George Lindfield and John Penny, editors, *Numerical Methods (Fourth Edition)*, pages 73–156. Academic Press, fourth edition edition, 2019.

[22] R. L. Bisplinghoff, H. Ashley, and R. L. Halfman. *Aeroelasticity*. Dover Publications, New York:, 1996.

[23] H. Hassig. An approximate true damping solution of the flutter equations by determinant iteration. *Journal of Aircraft*, 8(11):885–889, 1971.

[24] W. P. Rodden and E. Dean Bellinger. Aerodynamic lag functions, divergence, and the British flutter method. *Journal of Aircraft*, 19(7):596–598, 1982.

[25] W. P. Rodden. *Handbook for Aeroelastic Analysis*, volume 1. MSC/NASTRAN Ver. 65, 1987.

[26] Yingsong Gu and Zhichun Yang. Modified p-k method for flutter solution with damping iteration. *AIAA Journal*, 50(2):507–509, 2012.

[27] P. C. Chen. Damping perturbation method for flutter solution: The g-method. *AIAA Journal*, 38(9):1519–1524, 2000.

[28] Xinzhong Chen and Ahsan Kareem. Identification of critical structural modes and flutter derivatives for predicting coupled bridge flutter. *Journal of Wind Engineering and Industrial Aerodynamics*, 96(10-11):1856–1870, 2008.

[29] Gianni Bartoli, Stefano Contri, Claudio Mannini, and Michele Righi. Toward an improvement in the identification of bridge deck flutter derivatives. *Journal of Engineering Mechanics*, 135(8):771–785, 2009.

[30] Q. C. Li. Measuring flutter derivatives for bridge sectional models in water channel. *Journal of Engineering Mechanics*, 121(1):90–101, 1995.

[31] Randall Allemang. The modal assurance criterion - twenty years of use and abuse. *Sound & vibration*, 37:14–23, 08 2003.

[32] Miroslav Pastor, Michal Binda, and Tomáš Harčarik. Modal assurance criterion. *Procedia Engineering*, 48:543–548, 2012. Modelling of Mechanical and Mechatronics Systems.

[33] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, USA, second edition, 1992.

[34] Dana Knoll and David Keyes. Jacobian-free newton-krylov methods: A survey of approaches and applications. *Journal of Computational Physics*, 193:357–397, 01 2004.

[35] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19:577–593, 1965.

[36] M.A. Crisfield. A fast incremental/iterative solution procedure that handles snap-through. *Computers & Structures*, 13(1):55–62, 1981.

[37] Giorgio Diana, Stoyan Stoyanoff, Ketil Aas-Jakobsen, Andrew Allsop, Michael Andersen, Tommaso Argentini, Miguel Cid Montoya, Santiago Hernandez, Jose Angel Jurado, Hiroshi Katsuchi, Igor Kavrakov, Ho-Kyung Kim, Guy Larose, Allan, Guido Morgenthal, Ole Øiseth, Simone Omarini, Daniele Rocchi, Martin Svendsen, and Teng Wu. Iabse task group 3.1 benchmark results. part 1: Numerical analysis of a two-degree-of-freedom bridge deck section based on analytical aerodynamics. *Structural Engineering International*, 0(0):1–10, 2019.

[38] Theodore Theodorsen. General theory of aerodynamic instability and the mecanism of flutter. *NACA Report*, 496, 1935.

[39] Robert T Jones. Operational treatment of the nonuniform-lift theory in airplane dynamics. Technical report, NASA Langley Research Center Hampton (United States), 1938.

[40] G. Dimitriadis. Bifurcation analysis of aircraft with structural nonlinearity and freeplay using numerical continuation. *Journal of Aircraft*, 45(3):893–905, 2008.

[41] Manuel Ritto-Corrêa and Dinar Camotim. On the arc-length and other quadratic control methods: Established, less known and new implementation procedures. *Computers & Structures*, 86(11):1353–1368, 2008.

[42] Julien Heremans, Anass Mayou, and Vincent Denoël. Background/resonant decomposition of the stochastic torsional flutter response of an aeroelastic oscillator under buffeting loads. *Journal of Wind Engineering and Industrial Aerodynamics*, 208:104423, 11 2020.

[43] Julien Heremans, Margaux Geuzaine, and Vincent Denoël. A background/resonant decomposition based method to predict the behavior of 2-dof aeroelastic oscillators. *Journal of Wind Engineering and Industrial Aerodynamics*, 233:105290, 2023.

[44] ZONA Technology. *ZAERO Ver. 7.2: Theoretical Manual*. AZ, Scottsdale, 2004.

[45] M. Karpel, B. Moulin, and P. C. Chen. Dynamic response of aeroservoelastic systems to gust excitation. *Journal of Aircraft*, 42(5):1264–1272, 2005.

[46] Kenneth L. Roger. Airplane math modeling methods for active control design. 1977.