XLIII CILAMCE
21-25, November, 2022
FOZ DO IGUAÇU - PR - BRAZIL

Python Language Applied to Flutter Analysis in a VANT‘S 3D Wing.

Nicolle C. Parra[1], Hélio de A. Pegado[2]

[1]*Dept. of Civil Engineering, Federal University of Sergipe*
Rosa Elze, 49100-000, Sergipe/São Cristóvão, Brazil
nicollegcmparra@gmail.com
[2]*Dept. of Mechanical Engineering, Federal University of Minas Gerais*
Av. Pres. Antônio Carlos, 31270-901, Minas Gerais/Belo Horizonte, Brazil
heliopegado.eng.ufmg@gmail.com

Abstract. Flutter is one of the most well-known dynamic aeroelastic phenomena, known for its destructiveness and frequently associated with thin structures and less rigid materials, and has been widely used in the evolution of the aeronautical industry. The SAE Aerodesign competition simulates the design of UAVs - Unmanned Aerial Vehicles - and includes the study of the probability of aeroelastic phenomena in the project to qualify students and enthusiasts in the field. Aiming to enlarge access to Flutter analysis occurrence by students and produce an efficient study of the Flutter Critical Speed, the Python language was applied to create a program able to interact with the Aerodynamics through the Panel Method and the structural behavior by the Grid Method and return the Critical Speed calculated according to Method K, generating the V-G Diagram of the aeroelastic behavior of the plane. The wing of the UAV designed by Araras Aerodesign team in the system with two degrees of freedom was used as an example. The code was efficient in generating the expected result, checked according to the literature, and compared with commercial software.

Keywords: Flutter, Aeroelasticity, Python, Grid Method, VLM.

1 Introduction

Aeroelasticity is a field of study focused on the interconnection between aerodynamic forces and structural analysis, constituting one of the essential analyzes for dimensioning an aircraft. One of the most famous occurrence of Aeroelasticity in airplanes is Flutter, defined as coupling between the two main modes of vibration of a wing: bending and torsion. Therefore, the excitation produced by one mode increases force in the other, increasing amplitude of the movements and deformation. As Aerodynamic loads are applied in the wing, the energy provided by steam through the structure excites even more vibration until reach the structural limit of the wing. Beyond this limit, the structure can not handle its deformation and collapses. This article proposes a Flutter analysis in a UAV designed to SAE Aerodesign Competition through a Python program that calculates Aerodynamic forces, Structural deformation and provides Critical Flutter Speed as the main output. Three models were build to interact in the aeroelastic calculation: Aerodynamic Model, which calculates Aerodynamic Forces and Moment through Vortex Lattice Method - VLM, Structural Model and Aeroelastic Model. The main code consists of the Aeroelastic Model, responsible to generate Critical Flutter Speed by K-Method and Frequency, Damp and Velocity Diagram or V-g-f Diagram, while structural behavior is modelled by Grid Method. Every analysis provided by the code is verified by comparing results with certified references or with the results obtained by software, mainly Femap NASTRAN. The Flutter analysis were performed with Araras Aerodesign’s airplane data as an example. The ease of Python’s programs allows aeroelastic analysis within the reach of students with less access opportunities, expanding the frontiers of knowledge.

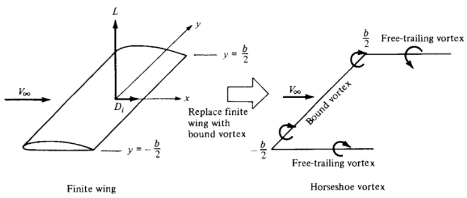
2 Methodology

Critical Flutter Speed is the main value to measure if an airplane is safe during flight. To calculate this parameter, previous Aerodynamic and Structural analysis must be done. The Python program interacts two side codes to obtain aerodynamic forces and structural behavior and receives these variables automatically. PyTornado library applies the Vortex Lattice Method described by **?**] and a function called "*structural_calculation*" studies the structure with Grid Method. Every step of the program is described on this section and preliminar project data of Araras Aerodesign's UAV was written as input, obtaining Critical Flutter Speed and V-g-f Diagram as outputs.

2.1 Vortex Lattice Method - Aerodynamic Model

Vortex Lattice Method is an application of Prandtl's Lift Line Theory which suits to represent the steam through a 3D wing. Prandtl modelled a wing with two free trailing vortices in each side and a bound vortex following the span line, as Figure 1, creating a shape called horseshoe. However, just one horseshoe is not necessary to cover all the changes in the steam while crossing the wing and multiple horseshoes must be created spanwise and chordwise, creating the main methodology behind VLM.

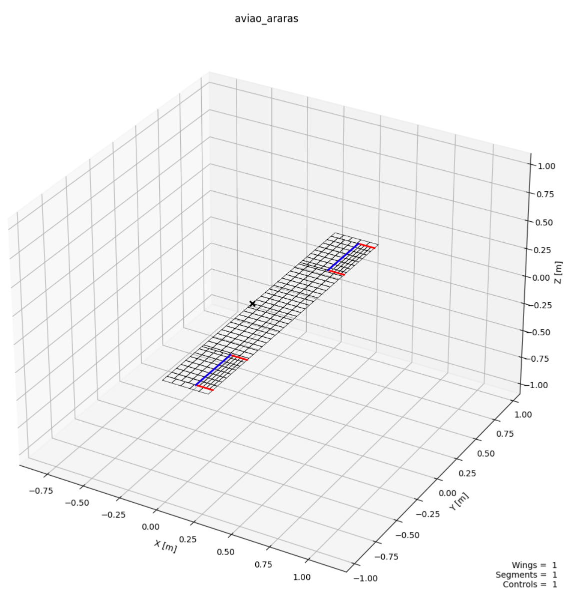
Figure 1. Horseshoe vortex throught a finite wing.



Source: **?**]

PyTornado library applies VLM on an input wing registered by the user, reading three JavaScript Object Notation - JSON files: *settings*, *state* and *aircraft*. JSON files are widely used due to easy data transference and intuitive interaction with the user. *Settings* file organizes all inputs and holds saving and running informations. *State* is where the user fills steam data and flight conditions out. The variables necessary are airplane true velocity, angle of attack and sideslip angle, altitude, air density, Mach number and body axis rotation. *Aircraft* holds airplane geometry. wing area, span, chrod, mass center and rotation center coordinates, as well as control surfaces position are listed on this file. Airfoils can be included in *airfoil* file, inside *aircraft*. The library exhibits plots - such asFigure 2 - and saves outputs into *results* file. The main code can access all the results automatically and include what is necessary in the next analysis, Structural Model.

Figure 2. 3D panned Wing ploted by PyTornado library.

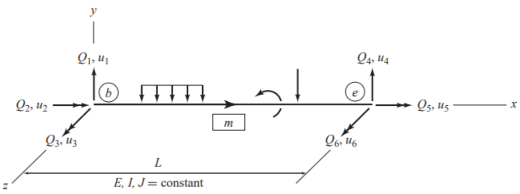


Source: Author.

2.2 Grid Method - Structural Model

One of the methods to study structural behavior is Grid Method **(author?)** [Kassimali], capable of representing tridimensional loads and non conventional geometries through the division of the structure in members, usually beams. Figure 3 represents a two node example, mathematically described with force and displacements relation (1).

Figure 3. Forces and displacements on a single grid element - Local coordinates.



Source: **(author?)** [Kassimali]

$$Q = ku + Q_f \tag{1}$$

Q_f is the force vector to a six degree of freedom member, analyzed with (2). Members which both of sides are fixed create forces on both ends b and e, different from a free end member which has only one node force. FS_b e FS_e are shear forces and FM_b e FM_e are bending moments. Torsion moment is also calculated and the sections were considered hollow circular.

$$Q_f = \begin{bmatrix} FS_b \\ FT_b \\ FM_b \\ FS_e \\ FT_e \\ FM_e \end{bmatrix} \tag{2}$$

A transformation matrix (3) must be used to represent local results in a global scale T, as well as member length L (4), global displacements v (5) and forces Q (6), according to [**?**].

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & cos\theta & sen\theta & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & cos\theta & sen\theta \\ 0 & 0 & 0 & 0 & -sen\theta & cos\theta \end{bmatrix} \tag{3}$$

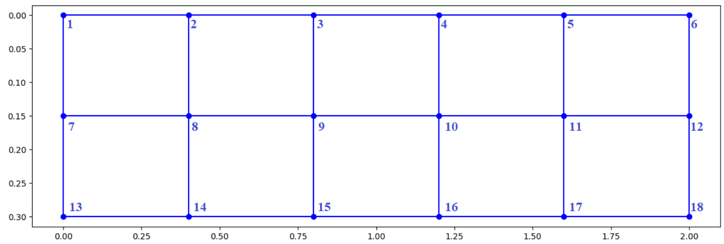
$$L = \sqrt{(X_e + X_b)^2 + (Z_e - Z_b)^2} \tag{4}$$

$$v = T^T u \tag{5}$$

$$F = T^T Q \tag{6}$$

Grid Method was compared to Femap NASTRAN to validate its use and displacement results, and its outputs are incorporated into Aeroelastic Model without any user command. The Grid created by Python's input function is on Figure 4 and Aerodynamic lift is distributed according to Stender's Methodology - more details available on **?**], as the Aerodynamic Model calculates the total lift and drag on the wing.

Figure 4. Grid drawed by Python's program.



Source: Author.

Functions included in the code register geometry and material parameters. Polar Moment of Inercia, Moment of Inercia are chosen according to input of a number corresponding to each section previously written on the code. The material registered leads the code to complete with the right Elasticity and Shear modulus. Circular, Rectangular and Coffin type sections are already available to the user, and the materials are wood, aeronautic aluminium - same as Nastran FEMAP in order to compare both analysis - and Carbon Fiber. The user can enter new sections and materials.

2.3 Aeroelastic Model - K Method and V-g-f Diagram and Main Code.

Flutter calculation is the most important output of the Python code. Due to its dynamic nature, the movement equation (7) defines wing oscillation pattern. Force vector R is related to matrix M and stiffness K. Although there are many kinds of Flutter, the code calculates a binary system with bending (8) and torsion (9). Both movements are a relation between bending inercia moment I_γ and torsion I_θ, bending stiffness k_γ and torsional k_θ.

$$R = Mr'' + Kr \tag{7}$$

$$I_\gamma \ddot{\gamma} + k_\gamma \gamma = 0 \tag{8}$$

$$I_\theta \ddot{\theta} + k_\theta \theta = 0 \tag{9}$$

Analyzing a steam with air velocity V, forces assume aerodynamic values and the system is described as complex due to aerodynamic matrix nature. In order to write the aeroelastic system equation, natural and reduced frequencies are incorporated to transform variables to modal coordinations. The system can be represented with (12).

$$M_q q'' + K_q q = \frac{\rho V}{2}\frac{b}{k}\Phi^T AIC_I \Phi q' + \frac{\rho V^2}{2}\Phi^T AIC_R \Phi q \tag{10}$$

$$Aq'' + \rho V B q' + (\rho V^2 C + E)q = 0 \tag{11}$$

$$Aq'' + Eq = \frac{\rho V^2}{2} Q q \tag{12}$$

Q matrix has all reduced frequency related terms and the velocity is the only unknown variable. The wing is a 2 degree of freedom or binary system, whose stiffness are represented by torsion and bending strings. This procedure follows the steps in Hancock et al. [2].

After the transformation, identity matrix could be used in the trivial dynamic system equation with multiples degrees of freedom (13). It is possible to reorganize (12) into matrix form in (14) and (15). D stands for structural damp, considered zero in the studied model, while A, E, C and B are Inertia, Structural Stifness and Aerodynamic damping matrix, divided into real C and imaginary parts B.

$$Iq' - Iq' = 0 \tag{13}$$

(12) na forma matricial em (14) e (15)

$$\begin{bmatrix} I & 0 \\ 0 & A \end{bmatrix} \begin{Bmatrix} q' \\ q'' \end{Bmatrix} - \begin{bmatrix} 0 & I \\ -(\rho V^2 C + E) & -(\rho V B + D) \end{bmatrix} \begin{Bmatrix} q \\ q' \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{14}$$

$$\begin{Bmatrix} q' \\ q'' \end{Bmatrix} - \begin{bmatrix} 0 & I \\ -A_{-1}(\rho V^2 C + E) & -A_{-1}(\rho V B + D) \end{bmatrix} \begin{Bmatrix} q \\ q' \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{15}$$

(15) can be solved assuming the result of x as $x_0 e^{\lambda t}$, leading to eigenvalues λ calculations. They are known as a complex conjugate and real or imaginary nature defines the system stability. According to Hancock et al. [2], if a complex eigenvalue has a positive real part, it is statically unstable.

Non stationary aerodynamic is incorporated on the calculation through reduced frequency and each mode frequency variate according to air speed value. The code draws a diagram relating frequency and damp results to a range of velocity from 0 to $180 m/s$, but the user can change these limits. Whether K Method is used to calculate Critical Flutter Speed, the only physically representative value is the one obtained when damping is zero. The lack of structural damping D affects the others values and their physical representation is inaccurate.

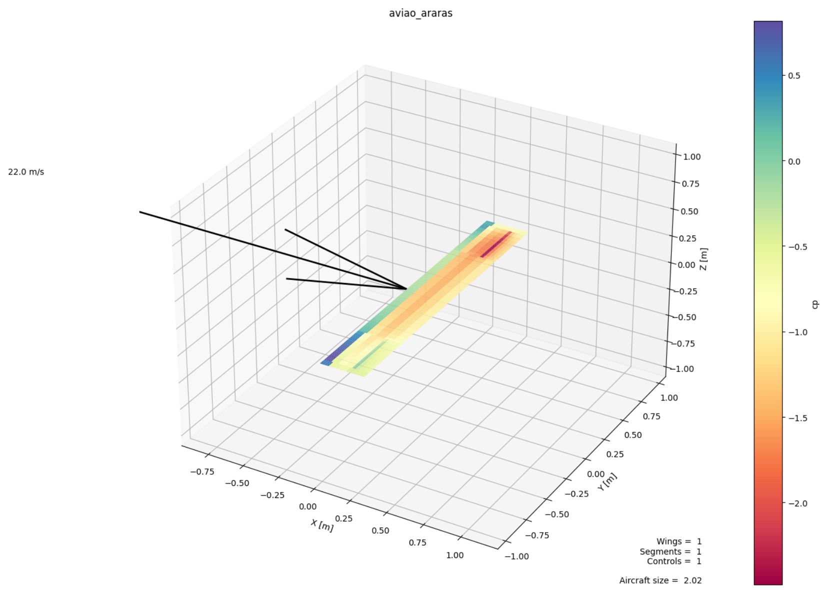
It is expected the user to complete three JSON files - *state*, *aircraft*, *settings* - and a single Python code called input_data as input to the code. Aerodynamic model is the only one initialized through command prompt typing "*pytornado -v –run settings/filename.json*" inside PyTornado directory. "Filename" must be substituted by file's name in user's computer. The main code reads all of PyTornado's results and the user needs to run the main code without any further process. V-g-f Diagram is plotted and Critical Flutter Speed is attributed to the value which damping turns zero. This can happen on Torsional or Bending mode.

3 Results and Discussions

Outputs obtained in each model, Aerodynamic, Structural and Aeroelastic, were validated according to literatur or comparison with Femap NASTRAN. After validation, Araras Aerodesign UAV's wing were calculated by the Python's program as an example.

PyTornado library plot a pressure pattern as one of the results, visible on Figure 5. According to **?**], this is the pattern expected when analyzing a wing in a cruise flight condition.

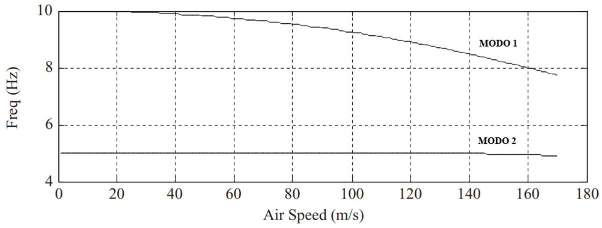
Figure 5. Pressure pattern ploted by PyTornado library.



Source: Author.

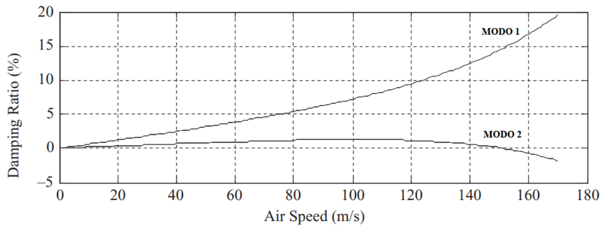
(author?) [Wright and Cooper]'s example of Flutter calculation, described as a [2] application, were tested and results compared with the ones on the book. As reported by literature results Figure 6, Figure 7 and author's results Figure 8 and Figure 9, V-g-f Diagrams are exactly the same and the Critical Flutter Speed is $151 m/s$. This process certifies correctness of Aeroelastic calculation within the code.

Figure 6. Frequency x Velocity Diagram to a 4 panel and e 2 structural elements according to Literature.



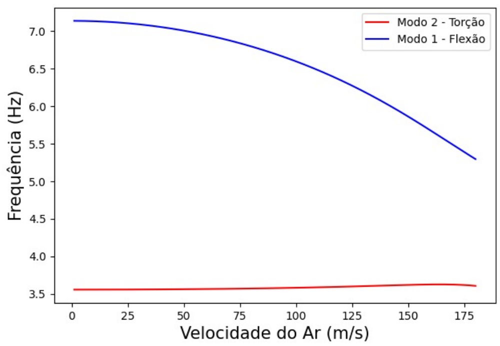
Source: [Wright and Cooper]

Figure 7. Damping x Velocity Diagram to a 4 panel and e 2 structural elements according to Literature.



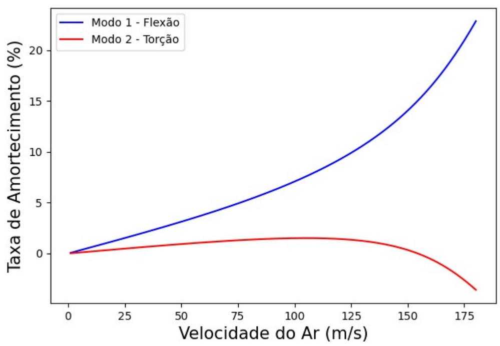
Source: [Wright and Cooper]

Figure 8. Frequency x Velocity Diagram to a 4 panel and e 2 structural elements according to Python's program.



Source: Author.

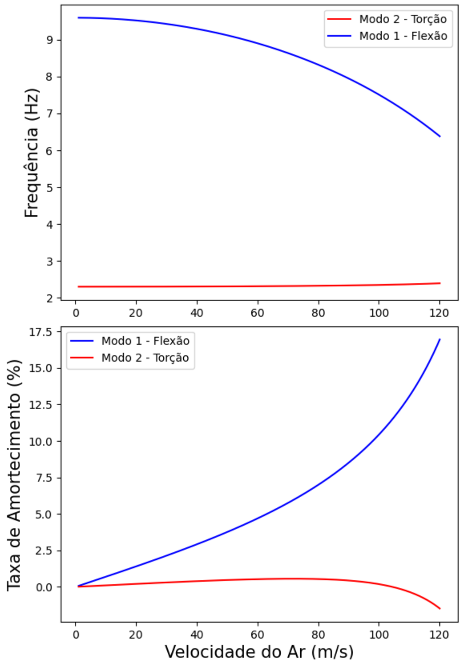
Figure 9. Damping x Velocity Diagram to a 4 panel and e 2 structural elements according to Python's program.



Source: Author.

The results concerned to Araras Aerodesign UAV are on **??** and Critical Flutter Speed is close to $100m/s$. Araras airplane's flight envelope, as well as other UAVs registered in SAE Aerodesign Competition, has a range of velocity between 0 and $30m/s$, implying that Flutter is beyond flight operation. The result also converges with Flutter Speed calculated by the member of Araras Aerodesign team.

Figure 10. V-G Diagram to Araras Aerodesign's wing.



Source: Author.

It is also possible to vary geometric parameters and the material of the wing and investigate its effects on Flutter occurrence. The decreasing on bending frequency but increase of binary Flutter velocity with the increase of span length as observed while the change in the material demonstrated a huge effect of material's stiffness on Flutter: the more rigid a material is, less possibility of the phenomenon is observed.

4 Conclusions

The main code and Python language are capable of performing an initial aeroelastic analysis with its own Panel Method study and applying the Grid Method to represent the structural behavior of an UAV wing built for the SAE Aerodesign competition. Automatically, the input data were read and interacted to plot the V-G Diagram as output data, which allows the user to identify the Critical Flutter Velocity, objective that guided the creation of this program. Preliminary data from the aircraft designed by the Araras Aerodesign team is enough to obtain a first value of the Critical Speed, calculated using the V-G Diagram at $110m/s$. The result is outside of the aircraft's Flight Envelope and indicates that the destructive phenomenon does not occur in the flight. It is still possible to analyze the effects of a wide variety of geometric parameters, on the Flutter Velocity, in addition to changing the material and geometric section. It was observed that the increase in wingspan on a very rigid wing does not cause significant changes in the occurrence of Binary Flutter but decreases bending frequency significantly. The wing material is crucial to the Critical Flutter Speed, with the most rigid material being preferable from a purely aeroelastic point of view. The program performs its objectives without high computational cost, free of charge and wide student's accessibility. It is easily editable and produces a study which can be replicated by students who are interested in the area. No paid software is needed to run the code, in addition to studying and expanding knowledge in the area of Aeroelasticity.

Acknowledgements. This work is dedicated to Federal University of Sergipe, SAE Aerodesign Competion and students around Brazil who are interested in the field. Thanks to professor Hélio Pegado.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[Kassimali] A. Kassimali. *Matrix analysis of structures*.
[2] G. J. Hancock, J. R. Wright, and A. Simpson. On the teaching of the principles of wing flexure-torsion flutter. *The Aeronautical Journal (1968)*, vol. 89, pp. 285 – 305, 1985.
[Wright and Cooper] J. R. Wright and J. E. Cooper. *Introduction to aircraft aeroelasticity and loads*.