PyAeroElast V1.0

|  |
| --- |
|  |
| User manual |
| **Stanislav Karpuk** |

Table of Contents

[Nomenclature 2](#_Toc4847166)

[1. General Code Description 3](#_Toc4847167)

[2. Code Structure 3](#_Toc4847168)

[3. Input description 4](#_Toc4847169)

[4. Running the Code 8](#_Toc4847170)

[5. Post-processing 8](#_Toc4847171)

[6. Function description 9](#_Toc4847172)

[1.1. Rigid Body Aerodynamics 9](#_Toc4847173)

[1.2. Divergence Speed 9](#_Toc4847174)

[1.3. Aeroelastic lift distribution to a symmetric maneuver 10](#_Toc4847175)

[1.4. Control Power and Control Reversal Speed 10](#_Toc4847176)

[1.5. Flutter Speed 11](#_Toc4847177)

[Appendix A. Validation of the Diederich method applied to the modified lift distribution analysis 12](#_Toc4847178)

[References 16](#_Toc4847179)

[Appendix B. Derivation of the angle-of-attack due to a symmetric maneuver 16](#_Toc4847180)

# Nomenclature

= aerodynamic influence coefficient

, , = structural influence coefficients

= wing lift-curve slope

= moment coefficient about the aerodynamic center

= load factor

= wing area

= airplane weight

= weighted coefficient for area calculation

= wing chord

= aileron chord

= airfoil lift coefficient

= distance from the elastic axis to the center of gravity

= distance from the elastic axis to the aerodynamic center

= incidence angle

= wing half span

= roll rate

= dynamic pressure

= aileron reversal dynamic pressure

= wing spanwise Y-coordinate

= rate of change of zero-lift angle of attack with aileron deflection

= rate of change of moment coefficient due to aileron deflection

= angle-of-attack

= zero-lift angle-of-attack

= aileron deflection

= non-dimensional Y-coordinate along the wing span

= column matrix

= row or a multidimensional matrix

= rigid-body coefficient

= elastic coefficient

# General Code Description

PyAeroElast is a Python set of scripts to estimate static and aeroelastic behavior of an airplane conceiver-type wing of medium to high aspect ratios. The code uses aeroelastic strip theory and semi-empirical relations to estimate aerodynamic behavior and aeroelastic responses of the wing.

The primary use of PyAeroElast is to perform rapid estimations of aeroelastic behavior of the aircraft and exploration of the V-n flight envelope to identify potential failure cases. Static aeroelastic analysis includes rigid body aerodynamic coefficients and lift distribution, divergence speed, lift distribution for symmetric maneuvers due to elastic effects, aileron power coefficients and aileron reversal speed, and estimation of flutter speed.

# Code Structure

PyAeroElast consists of the main script where all required input is initialized and a set of aeroelastic tools. The input Python scripts include all required information for the analysis desired and directories of the input and output. The Block diagram of the PyAeroElast is shown in Figure 1.

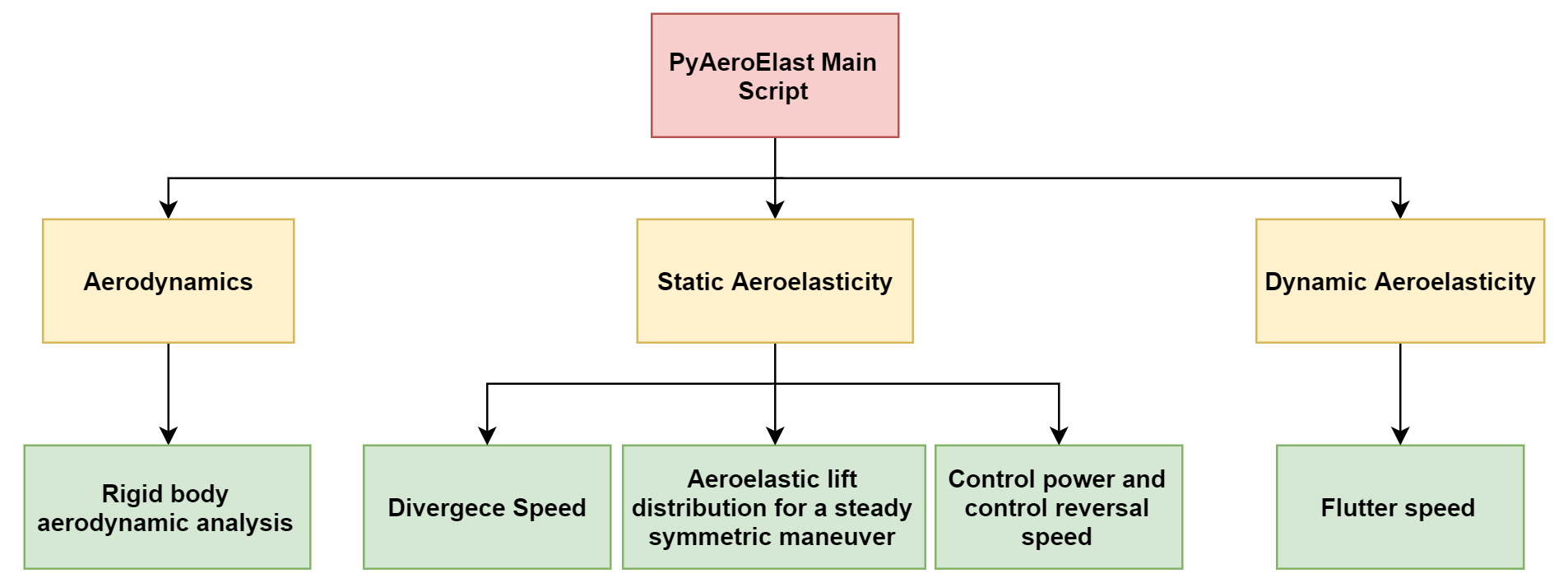


Figure 1. PyAeroElast code block diagram.

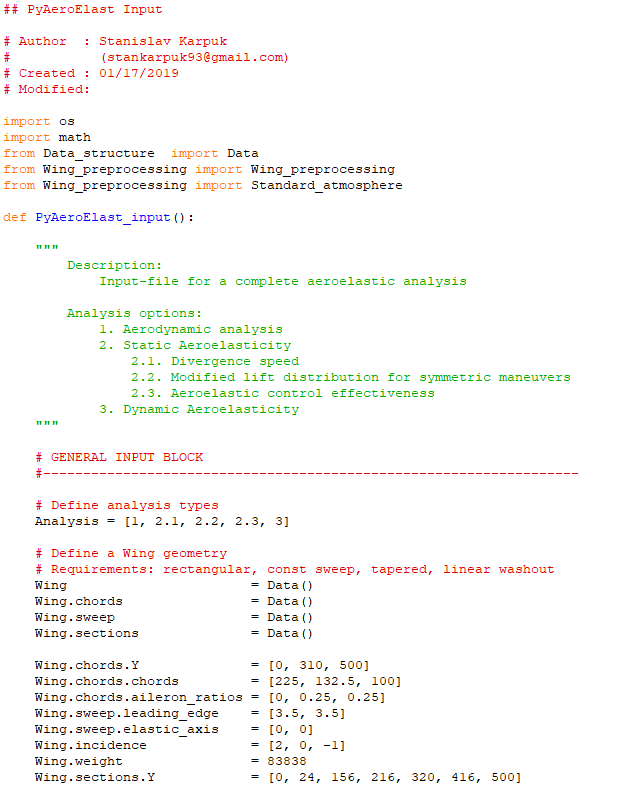
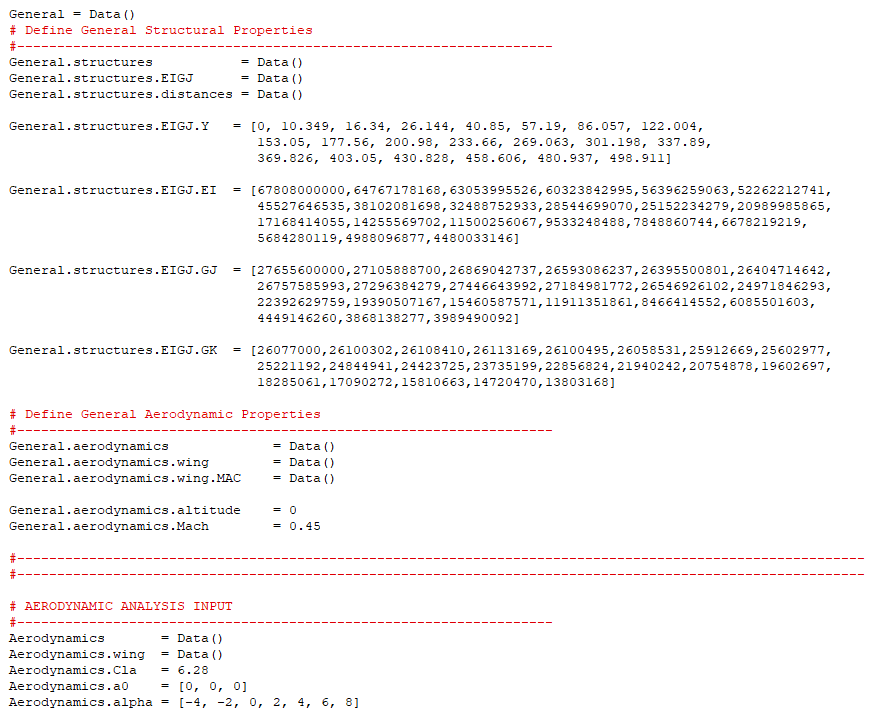
Possible analysis performed by PyAeroElast can perform can be decomposed into three major blocks: Aerodynamic analysis, Static Aeroelastic analysis, and Dynamic aeroelastic analysis. Each block has several types of analysis that can be performed. At this moment, only Static Aeroelasticity block has multiple types of analysis. Expansion of the code functionality will be available in the future versions. Functionality of each block and analysis constraints are further described in the Section 6.

# Input description

The PyAeroElast input file is created as a python function which is divided into sections depending on types of analysis one wants to perform. A typical input python file structure is shown in Figure 2. Each input parameter and its application are described below

**Defines a set of analyses to perform**

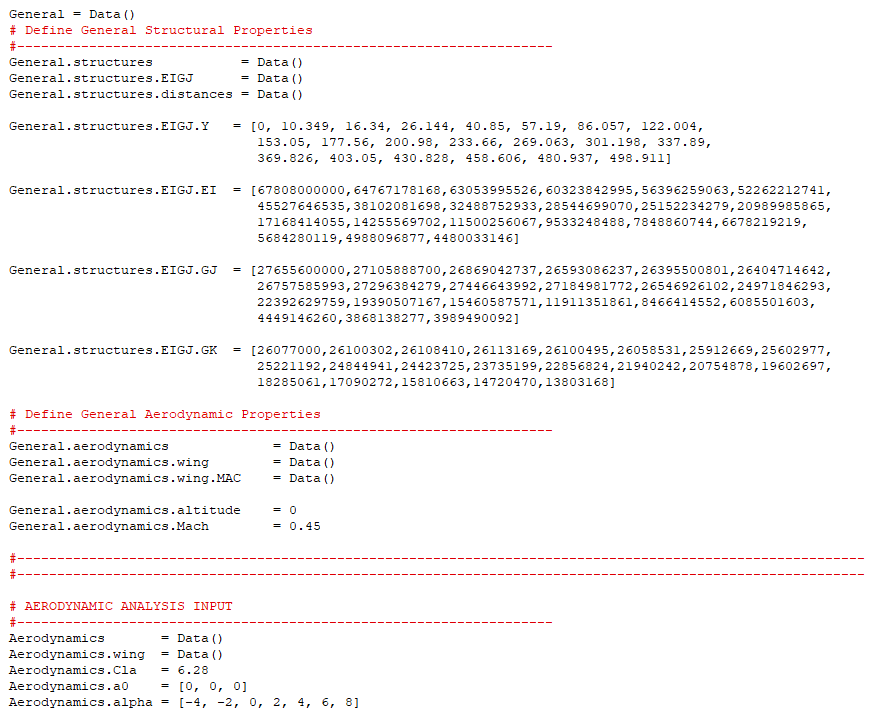
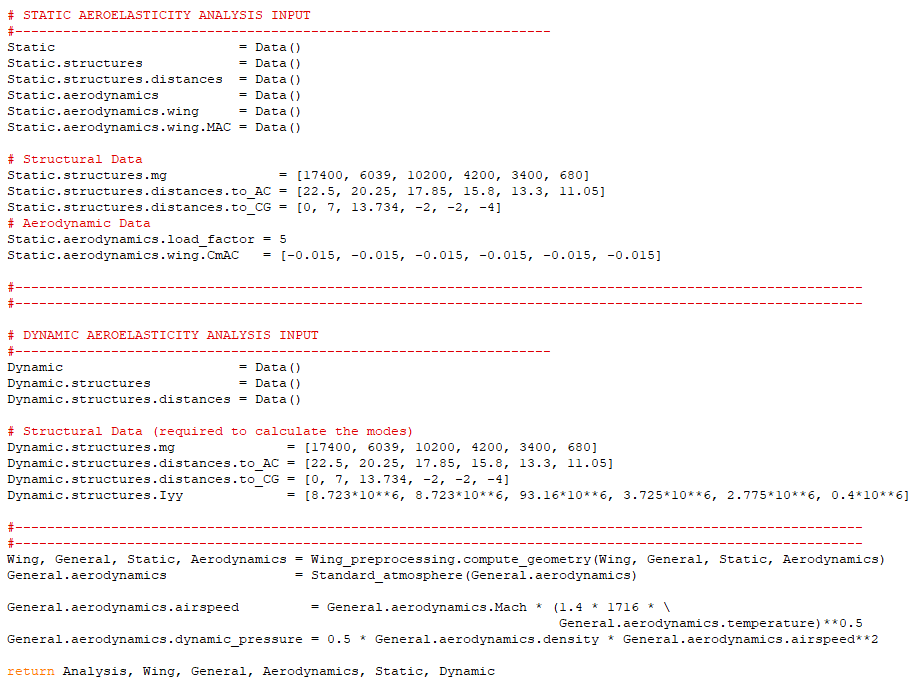
**Defines wing planform parameters**



**Defines properties specifically for dynamic aeroelastic analysis**

**Defines general parameters for any analysis**

**Defines properties specifically for Aerodynamic analysis**



**Defines properties specifically for static aeroelastic analysis**

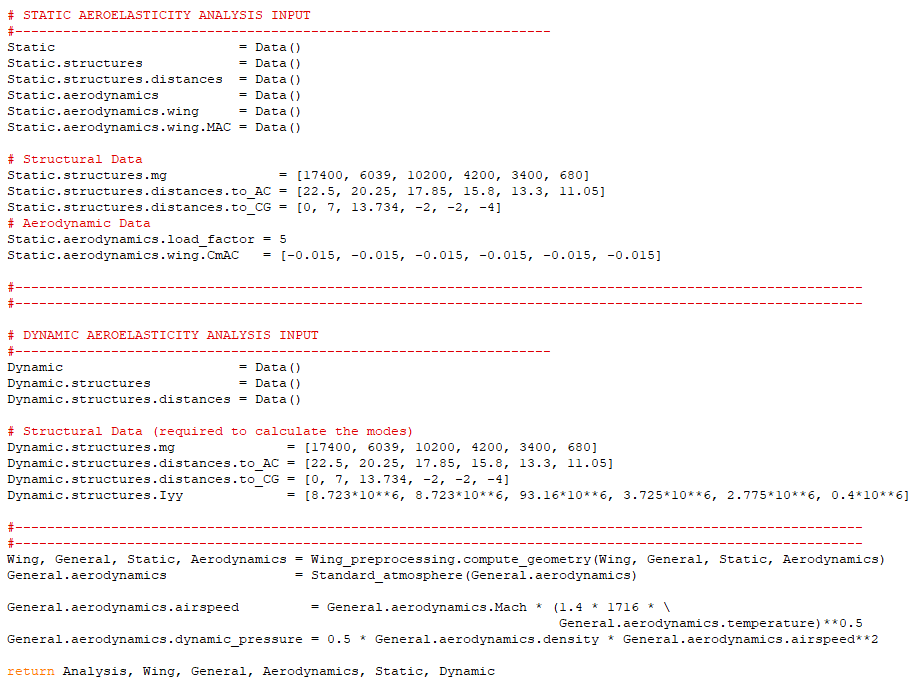


Figure 2. A sample input python file for a complete aeroelastic analysis.

The input file consists of several sections:

1. **Analysis types**. It determines what types of analysis the user wants to perform. Numbers of each types of analysis are provided in the commend above the start of the input
2. **Wing Planform Definition**. It sets the wing planform up. PyAeroElast lets the user specify a planform of any shape if important assumptions are satisfied.

Wing.chords.Y – specifies the wing stations along the wing in inches. The first and last numbers define the root and the tip respectively. Any number of sections can be specified.

Wing.chords.chords – defines chord lengths at each Y-station in inches. The number of elements must be same as in the Wing.chords.Y.

Wing.chords.aileron\_ratios – defines an aileron to chord ratio at each Y-station. The number of elements must be same as in the Wing.chords.Y. This input is required only if control effectiveness analysis is performed.

Wing.sweep.leading\_edge – specifies the wing leading edge sweep between the stations in degrees. The number of elements must be one less than in the Wing.chords.Y.

Wing.sweep.elastic\_axis - specifies the elastic axis sweep between the stations in degrees. The number of elements must be one less than in the Wing.chords.Y.

Wing.incidence – defines incidence angles for each section on the wing in degrees. The number of elements must be same as in the Wing.chords.Y.

Wing.Weight – specifies the wing total weight in pounds. This input is only required during the static symmetric maneuver analysis.

Wing.sections.Y – defines particular wing sections that will be used in all analyses. Such input philosophy is motivated by the fact that aeroelastic analysis uses a lumped mass method which requires centroid positions between the sections. The number of elements can be arbitrary but must be an odd number. For rigid body aerodynamic and divergence speed analyses (where mass properties are not required), more elements can be specified using np.linspace(Wing.chords.Y[0], Wing.chords.Y[len(Wing.chords.Y)-1], num=25), where num defines the number of sections.

1. **General analysis parameters**. This section includes structural parameters and aerodynamic parameters required for most studies. Structural parameters are defined as a set of lists that specify the IEGJ curves along the wingspan.

General.structures.EIGJ.Y – defines Y-coordinates along the wingspan in inches.

General.structures.EIGJ.EI – specifies EI values for each Y-coordinate in lb sq. in. The number of points must be the same as in the General.structures.EIGJ.Y.

General.structures.EIGJ.GJ – specifies GJ values for each Y-coordinate in lb sq. in. The number of points must be the same as in the General.structures.EIGJ.Y.

General.structures.EIGJ.GK – specifies GK values for each Y-coordinate in lb sq. in. The number of points must be the same as in the General.structures.EIGJ.Y.

**Note**: General structural parameters are not required if a rigid body aerodynamic analysis is performed

General.aerodynamics.altitude – specifies a flight altitude in ft.

General.aerodynamics.Mach – specifies the free-stream Mach number at a particular altitude. This parameter may be ignored if divergence and flutter speed analyses are performed

1. **Aerodynamic analysis input**. This section specifies input parameters required for rigid body aerodynamic analysis only.

Aerodynamics.Cla – specifies an airfoil lift-curve slope. It is assumed that deviatons in lift-curve slope do not significantly affect aeroelastic performance, so a constant value based on average lift curve slopes along the span may be used as an input.

Aerodynamics.a0 – defines zero-lift angles of attack for each wing section in degrees. The number of elements must be same as in the Wing.chords.Y.

Aerodynamics.alpha – defines angles of attack for the alpha-sweep in degrees. Any number of elements can be used.

1. **Static aeroelastic analysis input**. This section specifies input parameters required for static aeroelastic analysis only.

Static.structures.mg – defines lumped masses between each Y-station in pounds. This input is required only for aeroelastic lift distribution analysis under symmetric maneuver. The number of elements must be one less than in the Wing.chords.Y.

Static.structures.distances.to\_AC – distance from the elastic axis to the aerodynamic center at each chord section. Positive value is defined if the CG is ahead of the EA. The number of elements must be same as in the Wing.chords.Y.

Static.structures.distances.to\_CG - distance from the elastic axis to the center of gravity at each chord section. Positive value is defined if the CG is ahead of the EA. This input is not necessary for the divergence speed analysis. The number of elements must be same as in the Wing.chords.Y.

Static.aerodynamics.load\_factor – specifies the load factor for the symmetric maneuver analysis.

Static.aerodynamics.wing.CmAC – defines local moment coefficient at each chord section. Divergence speed analysis does not require this parameter. The number of elements must be same as in the Wing.chords.Y.

1. **Dynamic aeroelastic analysis input**. This section specifies input parameters required for dynamic aeroelastic analysis only.

Dynamic.structures.mg – defines lumped masses between each Y-station in pounds. This input is required only for aeroelastic lift distribution analysis under symmetric maneuver. The number of elements must be one less than in the Wing.chords.Y.

Dynamic.structures.distances.to\_AC – distance from the elastic axis to the aerodynamic center at each chord section. Positive value is defined if the CG is ahead of the EA. The number of elements must be same as in the Wing.chords.Y.

Dynamic.structures.distances.to\_CG - distance from the elastic axis to the center of gravity at each chord section. Positive value is defined if the CG is ahead of the EA. This input is not necessary for the divergence speed analysis. The number of elements must be same as in the Wing.chords.Y.

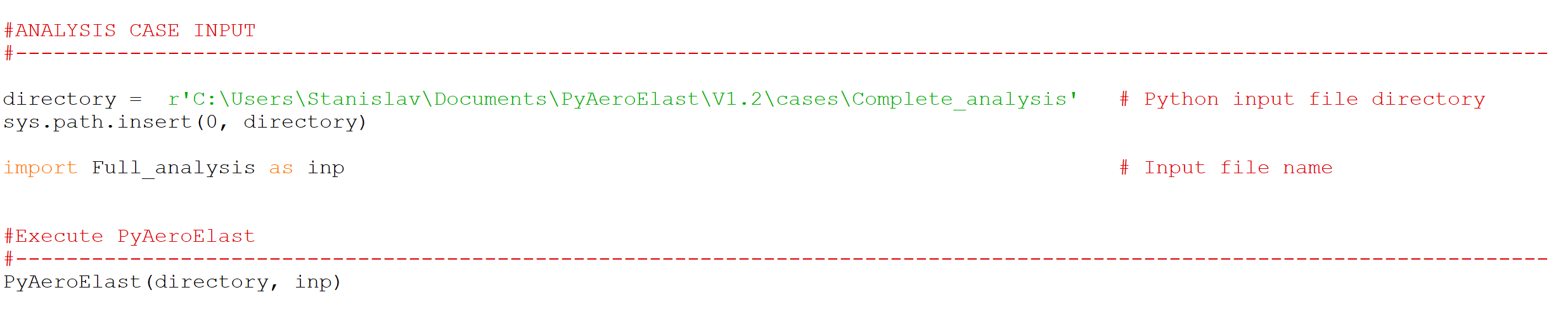
Dynamic.structures.Iyy – mass moment of inertia for each lumped mass along the wingspan in lb sq. in. The number of elements must be one less than in the Static.structures.mg.

For the user convenience, a set of example cases was created to demonstrate required inputs for each analysis and how they can be integrated in one big input script.

# Running the Code

To run PyAeroElast, two steps need to be completed.

1. The user needs to modify the RunPyAeroElast script to have the directory of the input file and the name of the input file, as shown in Figure 3.



**Specify the input file directory**

**Specify the input file name**

Figure 3. Run\_PyAeroElast script.

1. The script can be run either from the python editor or from the command window typing

> **python Run\_PyAeroElast**

# Post-processing

The PyAeroElast output consists of the **Output\_summary.dat** file which summarizes key parameters of the analysis and a set of plots. Typical plots, depending on analysis, include:

* A 2D view of the wing planform with the AC, CG, and EA lines if applicable
* A plot of structural properties along the wing span (EI, GJ, and GK)
* Lift and induced drag coefficients vs angle-of-attack
* Rigid and elastic wing lift distributions
* First natural modes and frequencies of the wing structure
* Structural damping coefficient vs Mach number

All files and plots are saved in the directory specified in the **Run\_PyAeroElast.py** file.

# Function description

This section briefly describes functions in each code module, their functionality, and a set of used assumptions.

## Rigid Body Aerodynamics

This section performs an aerodynamic alpha-sweep of the wing at particular free-stream conditions. Classical semi-empirical method of Diederich was used to calculate the wing lift-curve slope and an approximate lift distribution [3].

Assumptions:

1. Differences in airfoil lift-curve slopes are assumed to be small, so only one number equivalent to average airfoil lift-curve slope along the wing is used.
2. Aerodynamic center line is assumed to be nearly straight. Different sweep angles are permitted if the total change in sweep angle from the root to the tip is not excessive. Classical Yehudi wings should work well.
3. The range of Aspect Ratios is limited to 1.5 and the free-stream Mach number to 0.7. The code will automatically calculate the lift-curve slope for M=0.7 if the input parameter is exceeded.
4. For the wing with variable sweep angle along the span, an equivalent sweep angle estimation method is used [4].

### Divergence Speed

This section estimates the wing divergence speed based on the strip theory. The equation to determine the divergence dynamic pressure is

|  |  |  |
| --- | --- | --- |
|  |  | (1.1) |

where

– matrix of aerodynamic influence coefficients. With absence of more accurate and flexible methods for variable wing shapes, this simplified expression is good first-guess estimation

– weighted coefficient at each chord section. In PyAeroElst, the Multhopp’s quadrature method was used

Solution of the eigenvalue problem gives a divergence dynamic pressure that can be transformed to divergence speed and Mach number.

For more details regarding derivations and definitions, refer to Ref 1.

### Aeroelastic lift distribution to a symmetric maneuver

This section estimates the modified lift distribution due to a symmetric maneuver based on the strip theory. Given free-stream conditions, mass properties of the wing along the wing span and the load factor at which the maneuver is performed, the modified lift distribution if found as follows:

The angle of attack due to a symmetric maneuver is

|  |  |  |
| --- | --- | --- |
|  |  | (1.2) |

This information is then used in the system of equation to obtain elastic lift distribution and elastic angle-of-attack at the root chord:

|  |  |  |
| --- | --- | --- |
|  |  | (1.3) |

where

So, the total lift distribution becomes

|  |  |  |
| --- | --- | --- |
|  |  | (1.4) |

Rigid body lift distribution is found using the approximate method of Diederich [3]. For more details regarding the derivation of aeroelastic equations, refer to Ref 1.

### Control Power and Control Reversal Speed

This section estimates the aeroelastic effects on the wing due to elastic behavior based on the strip theory for a steady rolling maneuver and a control reversal speed.

Aileron effectiveness at a specific free-stream condition is computed by

|  |  |  |
| --- | --- | --- |
|  |  | (1.5) |

where

The control reversal speed is iteratively computed by

|  |  |  |
| --- | --- | --- |
|  |  | (1.6) |

Lift and moment coefficients due to the aileron deflections are estimated using methods described in Torenbeek [2]

|  |  |  |
| --- | --- | --- |
|  |  | (1.7) |
|  |  | (1.8) |

where

The constant 0.575 was obtained by multiplying the initial expression from the reference by 1.15 to reduce the tendency to underpredict the value.

### Flutter Speed

This section estimates the flutter speed using the U-g method. The analysis estimates the flutter speed at the ¾ of the wing span as a typical section where the flutter starts and does not include the aileron effects. Expansion of the function to include aileron effects will be available in the next version.

Assumptions and considerations:

1. Structural damping coefficients for bending and torsion are assumed to be almost the same
2. Bending and torsion first modes used for the method are computed using a lumped mass method
3. Uncoupled modes are used for the flutter calculations
4. Unsteady aerodynamic coefficients are calculated inside the integral, so analysis of planforms with variable spanwise sweeps is possible

Governing equations of the flutter determinant and more details about the solution process is demonstrated in 1. Estimation of unsteady aerodynamic coefficients is presented in 5.

# Appendix A. Validation of the Diederich method applied to the modified lift distribution analysis

The original derivation described in REF uses symmetric lift distribution using a Weissinger’s L-method. However, the classical Weissinger’s has several assumptions:

1. The line of aerodynamic center must be rectilinear.
2. The lift-curve slope is assumed to be equal to .
3. Weissinger’s method is used for incompressible flow.
4. Corrections for the lift-curve slope and compressibility are possible, but, from REF, are limited a specific number of wing sections.

For more flexibility in planform definitions, the strip theory and Diederich’s lift distribution method were used. However, validation of the method is required. Diederich’s lift distribution was compared an OpenVSP solution for the same wing. Several cases with different geometric and aerodynamic twists. The summary of test cases and a wing planform are shown in Table 1 and Figure 4 respectively.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Case 1** | **Case 2** | **Case 3** | **Case 4** |
| AR | 6.15 | | | |
| Root chord (in) | 225 | | | |
| Tip chord (in) | 100 | | | |
| Root airfoil | NACA 0012 | | NACA 2412 | |
| Tip airfoil | NACA 0012 | | | |
| Root incidence (deg) | 3 | 3 | 3 | 3 |
| Geometric washout (deg) | 0 | 3 | 0 | 3 |
| Aerodynamic washout (deg) | 0 | 0 | 3 | 3 |

Table 1. Summary of aerodynamic test cases.

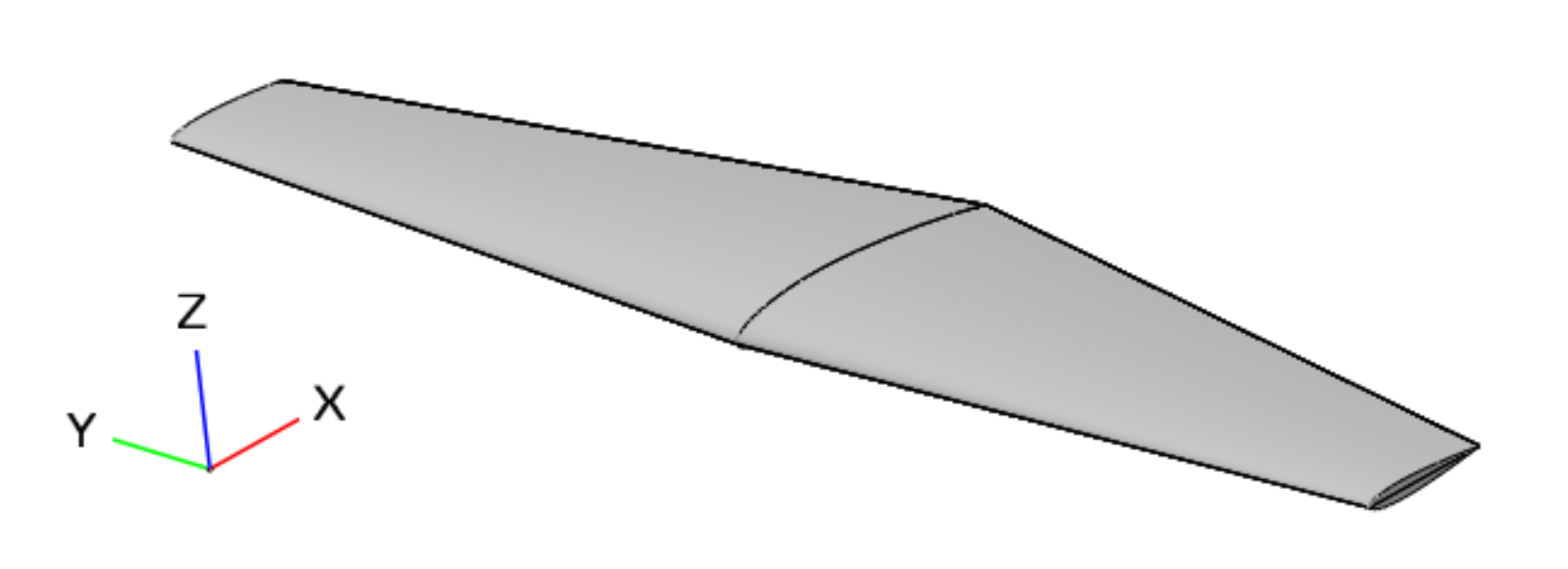


Figure 4. OpenVSP wing model.

Alpha-sweep for each case was performed for both the PyAeroElast and OpenVSP wings. For the aerodynamic washout case, PyAeroElast assumed the lift-curve slope to be an average of the two. Two parameters were compared: the lift curve and the lift distribution at each angle-of-attack. Results of the simulations are shown in Figure 5.

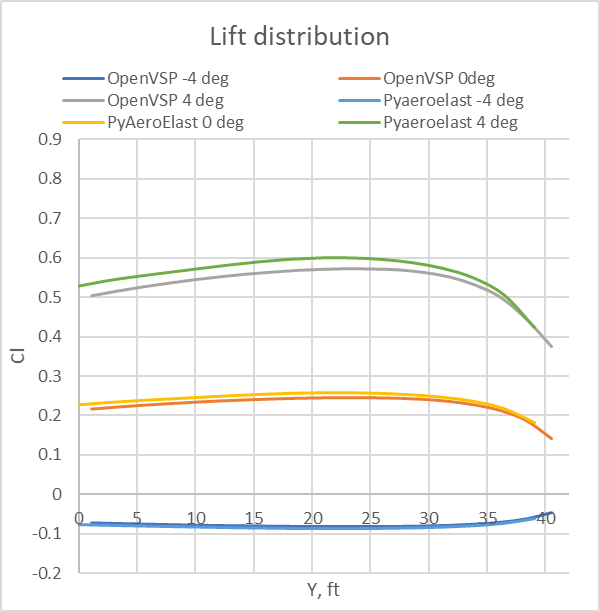
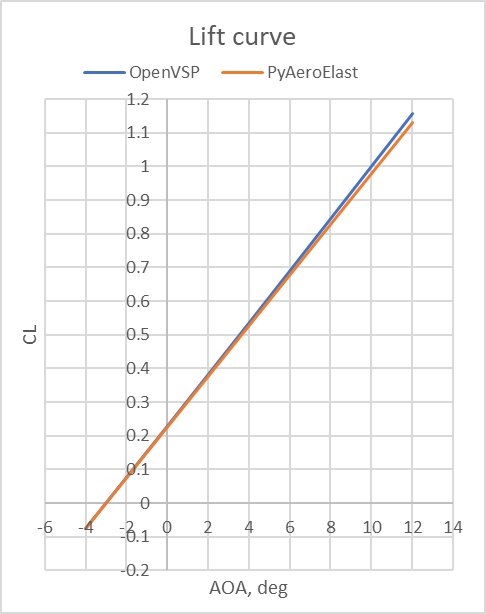


Figure 5. Lift curve and Lift distribution for Case 1.

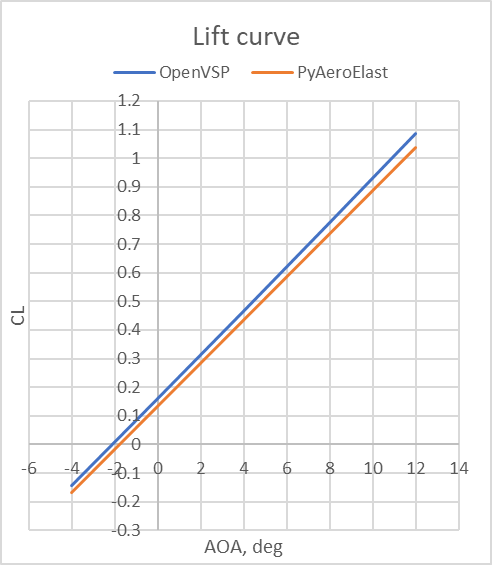
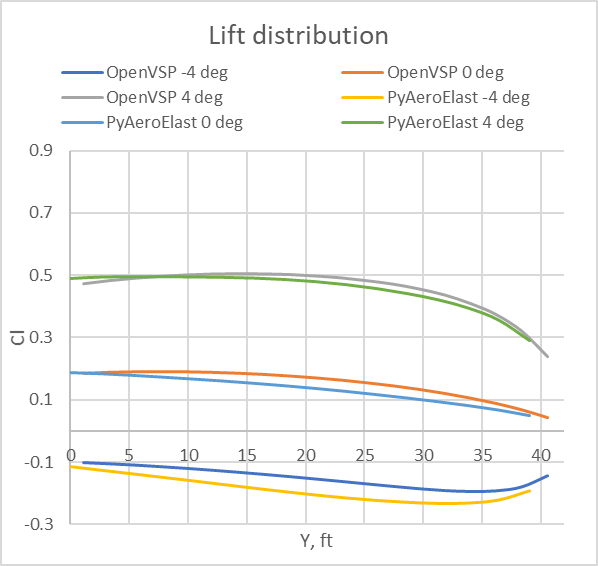


Figure 6. Lift curve and Lift distribution for Case 2.

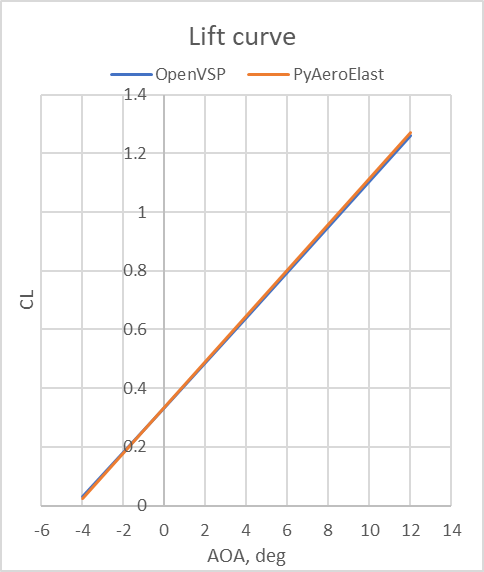
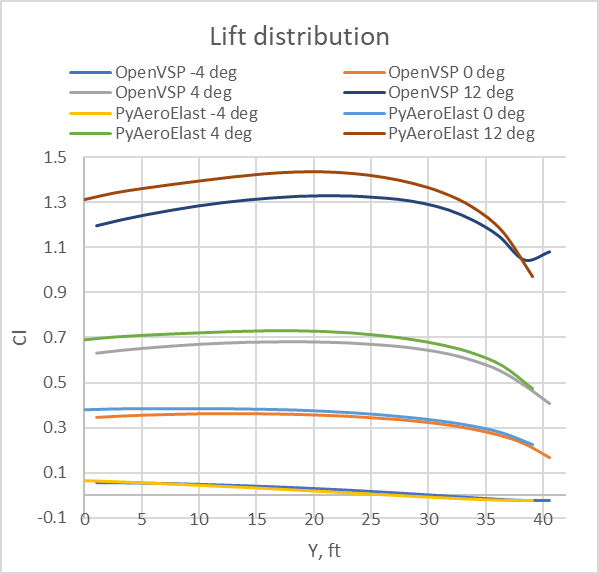


Figure 7. Lift curve and Lift distribution for Case 3.

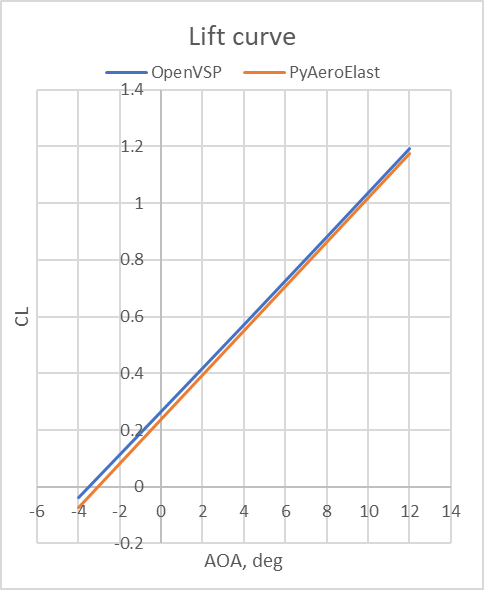
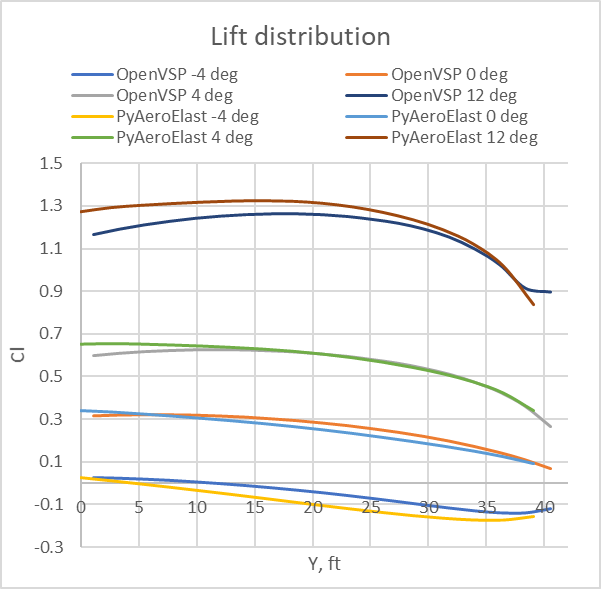


Figure 8. Lift curve and Lift distribution for Case 4.

Results for all cases show several important trends:

1. Lift curves matching is reasonably good
2. Differences between lift distributions with OpenVSP and PyAeroElast are minor for low angles of attack and become more significant at higher angles of attack.
3. Both geometric and aerodynamic twists work well for an assumption of an average lift-curve slope.

The comparison of lift distributions for the wing featuring symmetric airfoils for similar flight conditions and maneuvers between the Ref 1 and PyAeroElast is shown in Figure 9.

Figure 9. Comparison of modified lift distributions between Ref and PyAeroElast.

Results show similar reduction of the load at the root and its increase at the tip due to elastic twist. However, the magnitudes of both rigid and elastic lift distribution at the root are substantially different. The difference in rigid-body lift distribution is based on the differences in the estimation methods (Weissinger’s L-method vs Diederich’s estimation) while the elastic distribution may be a combination of the contribution of the aerodynamic and structural influence coefficients. However, the differences still allow the method to be used for the wing lift distribution estimation.

# References

1. Bisplinghoff, Ashley, Halfman, "Principles of Aeroelasticity", Dover Publications; First Dover Edition edition (August 14, 1996)
2. Torenbeek, "Synthesis of Subsonic Aircraft Design", Springer; 1982 edition (September 30, 1982)
3. Diederich, “A Simple Approximate Method for calculating spanwise lift distributions and aerodynamic influence coefficients at subsonic speeds”, NACA TN 2751, Washington, August 1952
4. Gudmundsson, “General Aviation Aircraft Design: Applied methods and Procedures”, Butterworth-Heinemann,1st ed, 2013
5. Scanlan, Rosenbaum “Outline of an acceptable nethod of vibration and flutter analysis for a conventional airplane”, Airframe and Engineering Report #43, Washington, October 1948

# Appendix B. Derivation of the angle-of-attack due to a symmetric maneuver

The load doe to a symmetric maneuver can be described by

|  |  |  |
| --- | --- | --- |
|  |  | (B1) |

From the Diederich’s method, the average angle-of-attack is

|  |  |  |
| --- | --- | --- |
|  |  | (B2) |

Substituting equation (B2) into (B1) gives

|  |  |  |
| --- | --- | --- |
|  |  | (B3) |

Solving for obtain

|  |  |  |
| --- | --- | --- |
|  |  | (B4) |