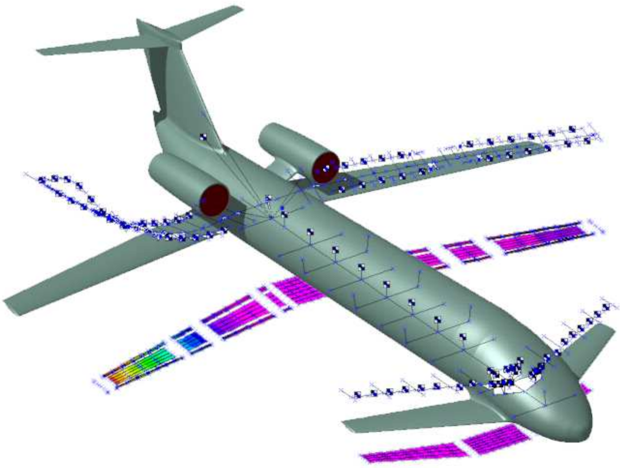
NeoCASS

Next generation Conceptual Aero Structural Sizing



Dipartimento di Scienze e Tecnologie Aerospaziali
Politecnico di Milano

Report name	NeoCASS Next generation Conceptual Aero Structural Sizing
Date	June 13, 2013
Partner Politecnico di Milano	Dipartimento di Scienze e Tecnologie Aerospaziali
Author	L. Cavagna, S. Ricci
Delivery number	D.XXXX
Dissemination level	Public

Contents

List of Figures

List of Tables

Chapter 1

NeoCASS : Next generation Conceptual Aero Structural Sizing

1.1 Introduction

This manual introduces the software NeoCASS developed by Dipartimento di Scienze e Tecnologie Aerospaziali
Politecnico di Milano(POLIMI) under the 6^{th} European Framework SimSAC for preliminary structural sizing and aeroelastic analysis for fixed-wing aircrafts.
The features of the design oriented analysis tool, together with its layout and its numerical key-strategies for reducing design cycle time and improving product design, are indicated. Deeper theoretical details can be found in the references included at the end of the report.

1.1.1 What is NeoCASS

NeoCASS is a numeric analysis tool particularly suited to conceptual and preliminary design of aicrafts. Its main purpose is to enhance these early design phases with details regarding bearing airframe which is usually poorly represented by the sole structural weight coming from empirical formulas.
The software helps indeed the designer to size structure of the aircraft under development and to investigate its aeroelastic behaviour by means of structural and aerodynamic numerical methods having physical basis. Thus, statistical formulas, usually characterized by poorness of details and accuracy are overcome so that the code can even be to design innovative and uncommon aircraft layouts.
The code is completely written in MATLAB© environment and can be run both in batch or interactive mode by means of its Graphical User Interface (GUI).
To accomplish the introduced tasks, NeoCASS provides two modules, GUESS (Chapter 2) and SMARTCAD (Chapter 5) with the following features:

- analytical structural sizing at different maneuvering flights and ground taxing coming from NASA-PDCYL code [3]; this first solution is merely used as *guess* and starting point for the different numerical aeroelastic solvers;
- structural optimization loop to satisfy user defined criteria (to be developed) ;

- linear/non-linear beam elements with distributed and lumped masses for efficient structural analyses of high-aspect ratio airframes [16, 10];
- linear equivalent plate elements [12, 6] with distributed and lumped masses for efficient structural analyses of low-aspect ratio airframes (not present in the current release) ;
- Vortex Lattice Method (VLM) [13] for steady subsonic aeroelastic analyses developed starting from Tornado code [18];
- Doublet Lattice Method (DLM) [2] for oscillatory subsonic stability derivatives and Aerodynamic Influece Coefficients (AIC) matrix calculation required for flutter assessment;
- Moving Least Squares (MLS) [21] and Radial Basis Functions (RBF) [22] spatial coupling methods for data transfer between non-matching structural and aerodyanamic meshes
- vibration modes solver (results are exported in FFA-format files for Edge [8] CFD solver when high-fidelity aeroelasticity is required);
- flutter solver to efficiently assess flutter (and divergence) instabilities in the whole flight envelope
- static aeroelastic analyses for deformed flight shape calculation.

NeoCASS can be used as a stand-alone application or inside a multi-disciplinary design environment. SimSAC 's core is indeed based on the CEASIOM system, a mixture of analytical and numerical tools assisting the designer in the conceptual and preliminary design of the machine. In this case NeoCASS will be used to provide preliminary details (such as stiffness and mass distribution, global weight) about the aiframe and its static and dynamic aeroelastic behaviour.

1.1.2 Installing the software

The installation for NeoCASS is extremely easy since the whole code is developed and runs in MATLAB© environment. The package is composed of several folders, one for each module. Since for aeroelastic analyses several modules are recursively called, in order to succesfully run all the solvers, the whole NeoCASS folder with its sub-folders must be included in MATLAB© `PATH` variable. This is accomplished by running first `set_neocass_path.m` from the home directory of the whole program where the function lies. This function appends to the current path definition, all the subdirectories where `.m` file are found. If the Graphical User interface (GUI) is used, the user needs to specify the launch command line for his own favourite XML editor and text editor respectively in the `neocass_xml_editor_path.m` and `neocass_text_editor_path.m` files. The GUI will directly call for these two editors for data visualization and managing.

External dependencies

NeoCASS has at the moment only one external dependency, which requires different strategies according to the operative system (OS) being used.
For Unix OS, a wrapper to link to the open source library *Approximate Nearest Neighbours (ANN)* (`http://www.cs.umd.edu/~mount/ANN`) needs to be compiled. This library provides optimized nearest neighbours searching tools which are required by the Moving Least Squares (MLS) spatial coupling method. A copy of the library is available in `NEOCASS_HOME/interface/ann_1.1`. Compilation process is started by typing `make` at the command prompt. After the library is succesfully compiled, a `Makefile` is provided in `NEOCASS_HOME/interface` to enable the linking of MATLAB© functions to the external libraries. The user needs to type `make` in this last folder to start the compilation of the wrapper. This process required both the open-source `g++` compiler and MATLAB© `mex` compiler.
For Windows OS, no user's intervention is required, since a pre-compiled executable `interface.exe` is already available and fully working.
By the way, project files for Microsoft© compiler to regenerate the executable are provided in `NEOCASS_HOME\interface\interface-win`.
NeoCASS automatically detects the OS being used and calls for the compiled MATLAB© function or for the executable.
To check everything is sucessfully set, the user can run a simple test from the folder `NEOCASS_HOME\interface\examples`.

1.2 Motivations and targets

Aircraft manufacturers consequently to the continous increase in the world-wide passenger air traffic, have been pushing new generation passenger vehicles more and more beyond the limits of existing flying machines. Competitiveness has been raising the necessity of the development of economical, ecological aircrafts with augmented sizes and payload capabilities, low operational costs and increasing performances. These requirements

- make aeroelastic analysis extremely important; negative aeroelastic effects like divergence, control surfaces reversal, flutter, increased drag at cruise speed due to structural deformability may require such considerable changes to the structural design, limitations in flight envelope or weight penalties that a project can be considered as a bad expensive project and has necessairly to be closed in some cases;

- can no longer be succesfully met by traditional design methodologies where aerodynamic and structural designs are split into two consecutive tasks; tipically aerodynamic shape is optimized supposing a rigid internal structure which is then designed with the constraint of assuming the optimum configuration when deformed. This approach has some drawbacks because for example there is no guarantee to design a structure having this behaviour, flutter and divergence are phenomena which need to be studied with a coupled fluid-structure approach and the mass required for the structural design may lead to unexpected lift requirements.

The solution to the raised issues consists in trying to face the influences of deformability on flight and handling performances, on structural weight and on design costs with rapid

and reliable methods starting from the fist steps of the conceptual design process when many parameters have not been estabilished yet. The enhancement of this phase with explicit multi-level design oriented numerical tools having physical basis rather than relying on implicit statistics of existing aicrafts, prevents the design from being excessively modified during the detailed design phase if deficiencies are found, guarantees the study of unconventional architectures (joined wings and blended-wing-body aircrafts) and new attractive technologies like composite materials. This justifies the development of integrated simulation environment and of Multidisciplinary Design Environments (MDO) as reported in [7, 11]
As pointed out by Giles [11] these tools need to:

1. guarantee easiness in coupling with other codes since multidisciplinary analyses are required;

2. be relatively accurate for the conceptual phase and give correct trend data to enable the design to progress in the correct direction;

3. be computationally efficient since several configurations need to be examined;

4. give the capability to trade accuracy for speed to the designer who is left free to decide the level of discretisation and to rule accuracy of modelling by the adoption of different solvers available in his toolbox;

5. require minimal time for model preparation and modification; the development of automatic procedures and the exploitation of geometry parametrisation which can be used as design variables, guarantees to easily reflect the changes of a desing variable in all the numerical models;

6. provide sensitivity derivatives of the design variables.

These features have also been recognized by the SimSAC project and are specifically introduced in NeoCASS tool.
The following sections introduce a critical overview of the available methodologies and describe in detail the layout and the chosen numerical key-strategies for reducing design cycle time and improving product design. Special care is taken to discuss how the above indicated features are fulfilled by the selected approach to the problem.

Chapter 2

GUESS : Generic Unknowns Estimator in Structural Sizing

This chapter introduces the GUESS module for analytical sizing of a complete airframe. GUESS is used to improve the prediction of structural weight relying on a method with somewhat physical basis. Furthermore, a stick beam model is determined from the calculated analytical solution to be used by the numerical aero-structural module SMARTCAD .

2.1 Airframe analytical guess solution

NeoCASS provides a method based on fundamental structural principle for estimating the load-bearing airframe for fuselage and lifting surfaces.
This method is particularly useful in the preliminary weight estimation of aircraft since it represents a compromise between the rapid assessment of component weight using empirical methods, based on actual weights of existing aircraft, and detailed but time-consuming finite-element analysis. Both methods have particular advantages but also limitations which make them not completely suitable for the preliminary design phase.
The empirical approach is the simplest weight estimation tool, which requires the knowledge of fuselage and wing weights from a number of similar existing aircraft in order to produce a linear regression useful to derive the data required for the aircraft to be designed. Obviously, the accuracy of this method depends upon the quality and quantity of available data for existing aircraft and how much closely the aircraft under investigation matches them. Thus, this approach is inappropriate for studies of unconventional aircraft concepts for two reasons:

- since the weight estimating formulas are based on existing aircraft, their application to unconventional configuration (i.e., canard aircraft) is suspect;
- the impact of advanced technologies and materials (i.e., advanced composite laminates) cannot be assess in a straightforward way.

Continuum structures, as fuselages or wings, which would use some combination of solid, flat plate or shell finite elements, are not easily discretizable and the solution of even a moderately complex model like an aircraft is computationally intensive and can become a bottleneck in the vehicle synthesis.

Finite-element methods, commonly used in aircraft detailed design, are not appropriate for conceptual design, as the idealized structure model must be built off-line and many details are missing in such a premature phase. The following two approaches which may simplify the finite-element model also have drawbacks. The first aims at creating detailed analysis models at a few critical locations on the fuselage and wing, to successively extrapolate the results to the entire aircraft. This approach can be misleading because of the great variety of structural, load and geometric characteristics in a typical design.
The second approach aims instead to creating a coarse model of the aircraft, but this scheme may miss key loading and stress concentrations.
An alternative approach exists and it is based on beam theory. The work was originally performed by Ardema and al. and presented as structural sizing tool for fuselage and wing (briefly described in Reference [3]). This results in a weight estimate which is directly driven by material properties, load conditions, and vehicle size and shape, thus being not confined to an existing data base.
NeoCASS starts from this last approach and extends it to the sizing of horizontal and vertical tail planes to have a complete view of the aiframe for the whole aircraft. All the improvements included in the present version of GUESS , indeed not defined in the already given reference [3], will be outlined in the following sections.
The distribution of loads and vehicle geometry is accounted for, since the analysis is done station-by-station along the vehicle longitudinal axis and along the lifting surface structural chord, giving an integrated weight which depends on local conditions. Nevertheless, an analysis based solely on fundamental principles will give an accurate estimate of structural weight only. Thus, weights for fuselage and lifting surfaces secondary structure (including control surfaces and leading and trailing edges) and items from primary structure (such as doublers, cutouts and fasteners) must be estimated by a correlation to existing aircraft.
In the following sections, all the steps required to obtain fuselage and lifting surfaces weight estimations are presented. References to some of the parameters which drive the execution of the tool are also made.

2.2 Weight and Balance Module

In order to start the structural sizing module explained in the following section, GUESS is provided with a Weight and Balance Module to have an estimate of the overall weight and inertias for the aircraft under design. This module allows determining non-structural masses such as payload (passengers and baggages), fuel, paint and systems (landing gears, APU, navigation systems) and the position of their center of gravity. Further, different configurations by varying the number of carried passengers and the amount of fuel stored in the tanks. This module also gives a first empirical estimation of the structural weight.
When it concerns structural sizing, the Weight and Balance module is useful to have a rough estimate of the position of center of gravity, moments of inertias and total weight of the aircraft required to correctly identify inertial loads to be applied when different kinds of maneuver are used for sizing.
This mudule is derived from QCARD code and can of course be used for other purposes different from GUESS goals, such as flight mechanics performances and stability analysis. The interested reader is referred to the reference given to have an overview of the analyt-

ical formulas used to estimate the weight and balance of each item.
The original program is composed of four scripts, grouped in the main *weight_ xml* function which receives as an input a string storing an *.xml file name and returns a MATLAB© struct.
The first script *wb_ weight* calculates all the weights with formulas based on statistical data or empirically determined. The *rcogs* routine computes each mass center of gravity coordinates with respect to a coordinate system whose origin is on the foremost airplane point, i.e. nose apex. The x axis points towards tail and z axis is directed upwards. It is pointed out all the calculations only consider these two axes, regarding the symmetry plane as a reference for weight arms computation.
The *riner* routine calculates body moments of inertia and the inertia matrix of the aircraft about its center of gravity; two ways to get to this result are available: setting to zero all the "guess" values leads to a "coarse" approximation, based on statistical formulas meanwhile, if minus one (-1) value is imposed, the code is forced to a "refined" inertia estimate which relies on geometrical considerations.
The last function *rweig*, computes aircraft global center of gravity for two significant mass distribution scenarios, MTOW (Maximum Take-off Weight) and MEW (Mass Empty Weight); this last refers to a "dry" condition in which fuel, passengers and baggage masses are not considered in the calculation.
All the calculated quantities are finally saved in a struct named *weight_ balance* which is added to the one derived from the input file which is introduced below. A summary of principal data is stored in *weight_ balance.COG* field, an $30 \times 4 \times 15$ matrix (an example is given at the end of the section), where each row refers to an item of the aircraft, the four columns store x,y,z coordinates and mass value (respectively in this order) and the latter index indicates which airplane is under analysis. This last information enables to pick the differences in the weight and balance global values, when one or several design parameters are varied by the designer.

Input/Output

As said above, the core of the weight and balance module is the *weight_ xml* function, that can be run independently once input data are provided.
The syntax to launch the function from MATLAB© environment is simply:

```
aircraft = weight_xml(filename);
```

where

- aircraft: output struct, which after the call has the new field *weight_ balance*;
- filename: input string referring to an *.xml file (see Appendix B for an example of this file).

The input *.xml file mainly contains geometric parameters and additional quantities required for weights estimation. In particular, these additional parameters are:

- internal design specification, i.e. cabin and baggage layout and data (`<Cabin>` and `<Baggage>` in the *.xml input file);

- fuel mass and its distribution in each tank (primary and auxiliaries);
- all weights which are independent of aircraft external geometry (volume), such as APU's or electrical/hydraulic systems.

Though not strictly necessary and perhaps often difficult to obtain or even guess, the latter information could improve weight estimation, but usually they do not lead to dramatic variation in weight and balance estimations.

Test case: DLR-F12 (full scale)

This section briefly summarizes some of the outputs from the weight and balance module applied to the DLR-F12 wind tunnel model. Since the F12 is very similar to an Airbus A340, all the geometry parameters are simply recovered by a proper amplifying factor. A real comparison can be made with the Airbus A340 itself or with a Boeing 747, both of them sharing nearly the same general layout and payload.
The MTOW results in 233910.70 kilograms, slightly less than 257000 kilograms for the A340, but also with 100 passengers less. A summary of weights and arms is reported in table 2.3.
As for the inertia matrix (see table 2.1), a significant comparison can be made with Boeing 747 data given in table 2.2 (source R.C. Nelson); the values herein reported have been calculated using the "refined" option.

23825362	0	0
0	30406362	0
0	0	53910982

Table 2.1: Inertia matrix for DLR-F12 with respect center of gravity $[kg \cdot m^2]$

24670000	0	0
0	44880000	0
0	0	67400000

Table 2.2: Inertia matrix for Boeing 747 with respect center of gravity $[kg \cdot m^2]$

2.3 Sizing module layout

The structural sizing procedure implemented in GUESS is briefly outlined in figure 2.1, depicting each single module as a sort of black box connected to other functional boxes. The flow chart underlines two main inputs are required to run the computer program. The

N	PART NAME	X [m]	Y [m]	Z [m]	MASS [kg]
1	WING1	31.33	0.000	-1.56	39881.19
2	WING2	0.000	0.000	0.000	0.000000
3	HT	61.03	0.000	2.450	2818.855
4	VT	60.15	0.000	6.600	1535.289
5	FUSELAGE	31.43	0.000	-0.62	24307.26
6	LANDING GEAR	0.000	0.000	0.000	10425.63
7	POWERPLANT1	26.31	0.000	-0.30	593.7820
8	POWERPLANT2	22.48	0.000	-0.23	890.6730
9	*unused*	0.000	0.000	0.000	0.000000
10	*unused*	0.000	0.000	0.000	0.000000
11	*unused*	0.000	0.000	0.000	0.000000
12	*unused*	0.000	0.000	0.000	0.000000
13	*unused*	0.000	0.000	0.000	0.000000
14	*unused*	0.000	0.000	0.000	0.000000
15	*unused*	0.000	0.000	0.000	0.000000
16	*unused*	0.000	0.000	0.000	0.000000
17	FURNITURE	31.43	0.000	-0.62	10414.02
18	WING TANKS	29.09	0.000	-1.63	72886.00
19	CENTRE FUEL TANKS	27.37	0.000	0.000	33300.00
20	AUXILIARY TANKS	60.39	0.000	0.000	3000.000
21	INTERIOR	25.49	0.000	-1.09	5960.401
22	PILOTS	9.890	0.000	-1.85	680.0000
23	CREW	0.000	0.000	0.000	75.00000
24	PASSENGERS	25.49	0.000	-1.85	24072.00
25	BAGGAGE	41.46	0.000	1.820	3070.608
26	*unused*	0.000	0.000	0.000	0.000000
27	CG at MTOW	28.77	0.000	-1.00	0.000000
28	*unused*	0.000	0.000	0.000	0.000000
29	CG at MEW	28.53	0.000	-0.78	0.000000
30	*unused*	0.000	0.000	0.000	0.000000

Table 2.3: Table summary of weights and arms for DLR-F12

procedure to run GUESS , from the structural sizing tool through the beam model generation and the definition of output file in NASTRAN format to run aeroelastic analysis, from MATLAB© environment is slightly different if it is used as a stand-alone application or included in the NeoCASS softwere, since in the latest case a user-friendly GUI interface has been developed.
Independently the way to lunch GUESS , two input files are used by all the subsections:

- *geometry* `.xml` input file, generated by a CAD softwere, containing detailed informations about the overall airframe geometry; this file is never modified within the current computer program;

- *technology* `.xml` input file collects all the other informations that are not geometric parameters (i.e. material properties for the airframe, informations to properly define the beam model and the aerodynamic model, and data to setup the aero/structural analysis); it has been defined in a user-friendly version since the user is allowed, at first, to change the values of all the parameters, according his own experience to run GUESS analysis;

More informations about the *geometry* input file are defined in Section 2.4, while an example of *technology* input file is shown in Appendix A. As much as it is useful for the NeoCASS beginner to learn and understand the meaning of the *technology* input file and consciously modify all the parameters, each single entry of the `.xml` file will be lately explained.

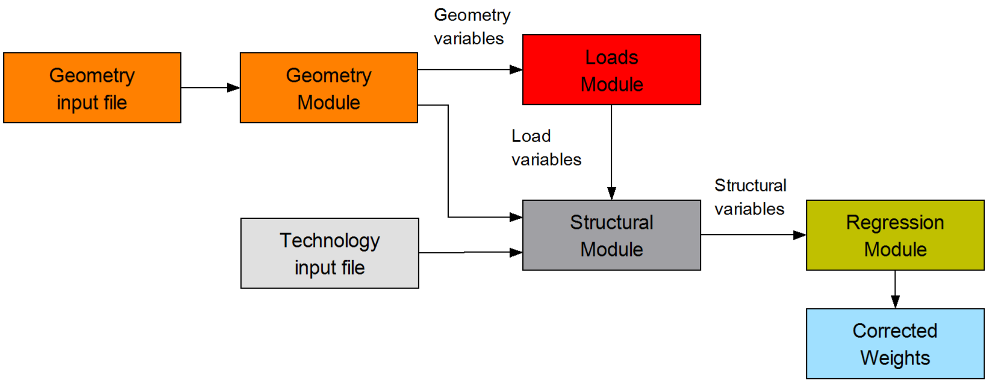


Figure 2.1: Layout of the analytical structural sizing tool

Once the geometric airframe description has been performed within the *Geometry Module* (Section 2.4) using the two `.xml` input files, the *Load Module* is called to determine the loads distributed along vehicle longitudinal axis and lifting surfaces structural span (Section 2.5). Several load cases are analyzed for both fuselage and lifting surfaces, and the envelope of the maximum load is then transferred to the *Structural Module* (Section 2.6) which computes the amount of structural material required to preclude failure in the most critical condition for each station.
As already introduced above, because the theory used in the present computer program only predicts the load-carrying structure of the aircraft components, a correlation between the predicted weight and the actual load-carrying structural weight and primary weight,

as well as the total weight of fuselage, wing, horizontal and vertical tail, is considered by means of the *Regression Analysis Module* (Section 2.7).
The term *lifting surfaces* is extensively used through the following sections and it gathers together the wings, horizontal and vertical tail, since a unified analysis is possible for all the aerodynamic surfaces as lately described.

2.4 Geometry module

For all aircraft design proposals, a good deal of importance must be placed on an accurate definition and subsequent analysis of geometric attributes. This statement is supported by the basic fact the entire scope of intermediary and final objective function evaluations, i.e. design weights, aerodynamics, performance, etc., stem from a fundamental geometric description leading to a physically tangible outcome.
Within the *Geometry module*, a detailed description of fuselage and lifting surfaces is obtained starting from the provided `.xml` input files. GUESS is able to extend the analytical structural sizing, as originally intended in Reference [3], to all lifting surfaces including horizontal and vertical tails. Moreover new geometrical features are available within the procedure, and no simplifications are used within the *Geometry module* since the correspondent geometry is directly generated from the *geometry* `.xml` input file. This input file is particularly rich in details (several tens) and drives itself a CAD software for the creation of a solid model.
The aircraft is assumed to be composed of different entities like fuselage, wing, horizontal and vertical tail planes. Each aerodynamic surface is possibly further composed by:

- 3 sectors within the wing semispan by means of 2 kinks;
- 2 sectors within vertical tail span by means of 1 kink;
- 2 sectors within horizontal tail semispan by means of 1 kink.

2.4.1 Fuselage

The original fuselage geometric description, as defined in the given reference, is roughly simple respect the real layout for actual aircrafts and more suitable for hypersonic vehicle than subsonic/transonic since it is assumed that:

- fuselage is composed of a nose section and a tail section, which are described by two power-law bodies of revolution placed back-to-back, with an optional cylindrical midsection between them, as shown schematically in Figure 2.2.

To obviate the loss of precision thus introduced analyzing subsonic/transonic transport aircrafts using power-law bodies of revolution, one solution is to employ a quasi-analytical method for the fuselage geometric description.
With the necessary geometric informations stored in the *geometry* `.xml` input file, the fuselage is taken to be a four-segment body: the nose; the fore and aft section; the tail. As an example of the enhancement achieved by the *Geometry modulus* employing the quasi-analytical method, the fuselage layout for $Boeing747 - 100$ and TCR are shown in Figure 2.3. In the former example, the hump over the nose represents the passengers compartment located on the upper floor.

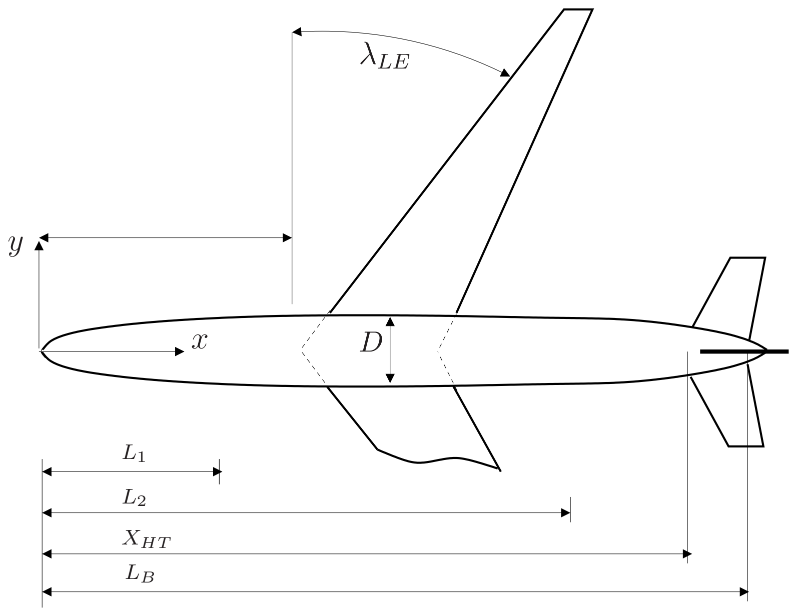
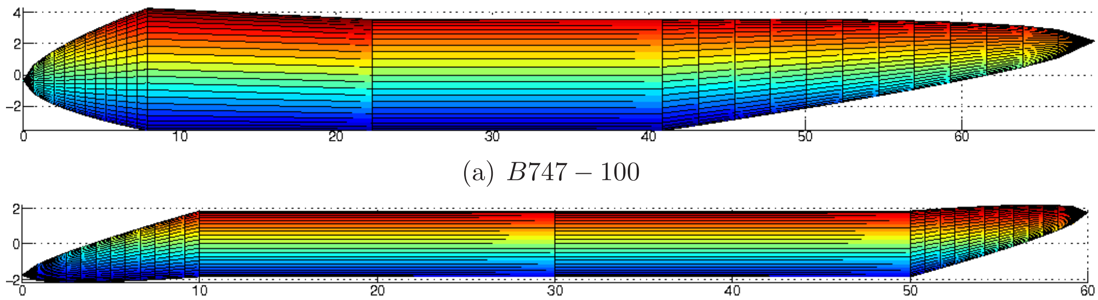


Figure 2.2: Lifting surface structural planform geometry



(a) $B747 - 100$

(b) TCR

Figure 2.3: Fuselage geometric description by means of GUESS .

2.4.2 Lifting surfaces

The reference [3] defines the assumptions the wings geometric description is based on:

1. wings are assumed to be tapered, swept with straight leading and trailing edges; the shape of the planform is trapezoidal as the root and tip chord are parallel;

2. specified portions of the streamwise (i.e. aerodynamic) chord are required for controls and high lift devices, leaving the remainder for the structural wing box; fuel can be carried optionally within structural wing box;

3. the intersection of the structural box with the fuselage boundary determines the location of the rectangular carrythrough structure; the width W_C of the carrythrough structure is defined by the corresponding fuselage diameter.

The trapezoidal wing planform area, as above described, is shown in Figure 2.4. The assumption 1. clearly defines a limitation in the geometric description because actual aircrafts do not have a pure trapezoidal wing.

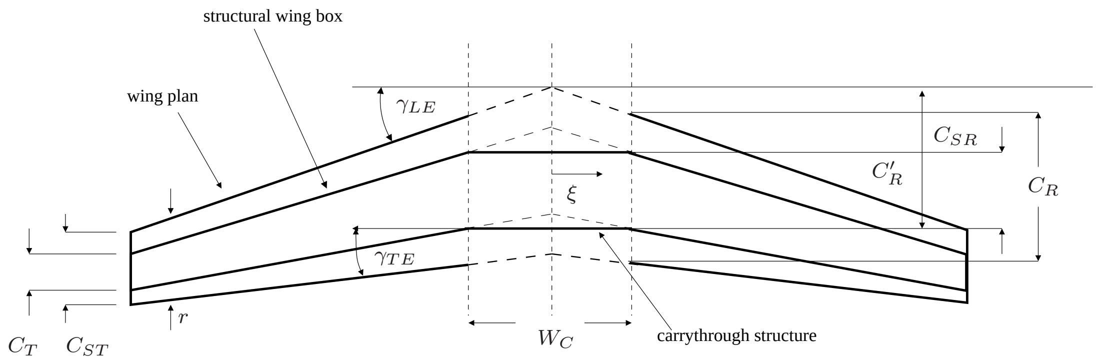


Figure 2.4: Trapezoidal wing body configuration

The current *Geometry module* developed within GUESS is indeed able to extend the wing geometric description to a cranked planform area (Figure 2.5), thus giving a more faithfull lifting surface geometric description for actual aircrafts. The procedure to deal with a generic number of sectors within the planform area is extended to all the lifting surfaces.

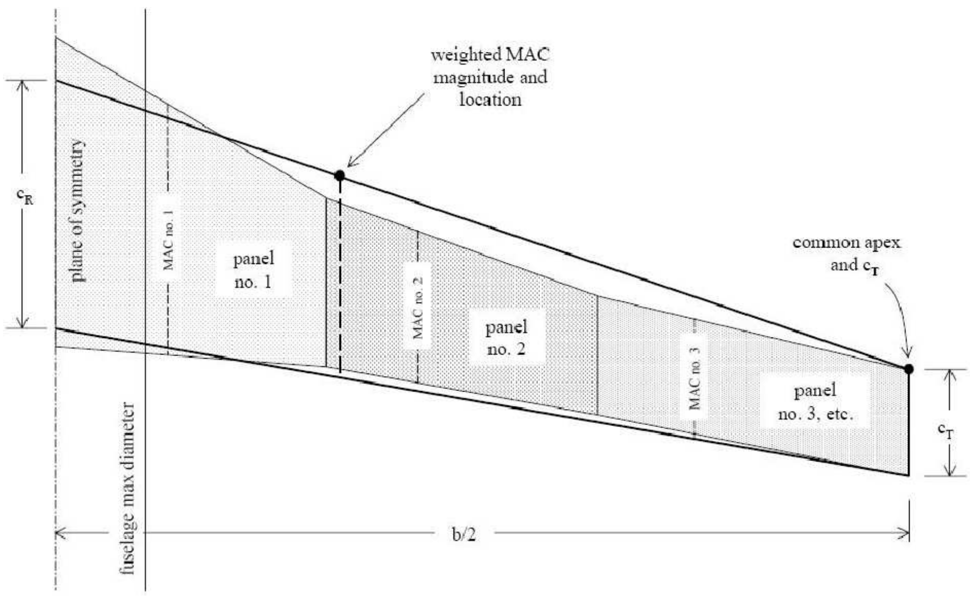


Figure 2.5: Cranked lifting surface geometric description by means of GUESS

In Table 2.4 the maximum number of segments, or aerodynamic panels, that can be used within each lifting surface is restricted by the informations contained in the *geometry* `.xml` input file.
As already mentioned in the present chapter and deduced in particular from Figure 2.4, lifting surfaces may be divided into two sections: *structural wing box* and *carrythorugh structure*. The following descriptions might be useful to fix the concepts:

- that part of the wing which transfers net aerodynamic and inertia loads to the fuselage is referred to as the *wing box*; it is essentially a box beam which resists these applied wing loads by shear, bending and torsion in the box; in addition, the

Component	Maximum number of sectors	Notes
Wing	3	over the semispan
Vertical tail	2	
Horizontal tail	2	over the semispan

Table 2.4: Maximum number of sectors for each component.

box supports the control surfaces, leading and trailing edges, secondary structure and other possible wing-mounted items such as landing gears and engines;

- on aircraft configurations with adequate fuselage volume (that is most transport aircraft), wing boxes are extended across the fuselage for a continuous box from tip to tip; this part is referred to as *carrythorugh structure*; this is most efficient from a structural aspect, since symmetrical spanwise bending loads do not enter the fuselage structure; moreover, wing fuel capacity is much greater since the section with the greatest depth is within the fuselage limits.

2.5 Loads module

Loads determination is based on simplified vehicle loading models, assuming that:

- fuselage lift forces are nominally zero for subsonic transports; in future developments of the code fuselage aerodynamic forces will be accounted for by means of simplified theories;
- wing loading, determined independently, is transferred by a couple of vertical forces and torque through the wing carrythrough structure;
- fuselage weight, thus inertia forces, is uniformly distributed over fuselage volume;
- landing gears loads are point loads;
- the propulsion system, if mounted on fuselage, is uniformly distributed.

GUESS determines the loading conditions for fuselage and wing, as originally intended. Moreover, the computer program extends the analysis to the vertical and horizontal tail planes, enhancing the prediction of weight and stiffeness distribution all over the complete airframe. In the following sections, a brief summary of the implemented procedures is given ([?], [?]).

2.5.1 Fuselage load condition

Fuselage loading is determined on a station-by-station basis along the length of the vehicle. Three types of loads are considered, as reported in Table 2.5:

1. longitudinal acceleration; in the original formulation it is applicable to high-thrust propulsion system only (thus neglecting the contribution for common transport aircrafts); in the current version, this feature is indeed extended to all propulsion system and included in the procedure since engines trust might be available from the given specification in the *geometry* input file;

2. tank or internal cabin pressure caused by pressure differential in passenger compartment; the value is computed in *Weight and Balance module*;

3. bending moments.

Load type	**Source**
Longitudinal acceleration	axial acceleration
Tank or internal cabin pressure	pressure difference
Longitudinal bending moment	quasi-static pull-up maneuver landing runway bump

Table 2.5: Summary of different load conditions for fuselage

These load cases occur at user-specified fractions of Maximum TakeOff Weight ($MTOW$) which are defined by dedicated parameters in the *technology* input file. Table 2.6 briefly reports the parameters which need to be defined in order to successfully carry out the load prediction for the fuselage.

Entry – user_input.loading.*	**Note**	**Unit**
`normal_load_factor`	normal load factor in $g's$	–
`weight_fraction.cman`	$MTOW$ weight fraction at maneuver	–
`weight_fraction.cbum`	$MTOW$ weight fraction at bump	–
`weight_fraction.clan`	$MTOW$ weight fraction at landing	–

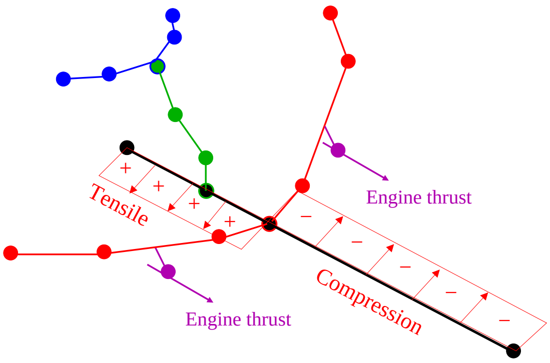
Table 2.6: Entries in the *technology* input file.

Load cases are superimposed simultaneously to determine maximum compressive and tensile loads at the outer shell fibers at each section and determine the most critical load that fuselage is designed for. A factor of safety, nominally 1.5, is applied to each load case.

Longitudinal acceleration

NeoCASS is able to deal with high-thrust propulsion system and take into account the longitudinal stresses produced by axial acceleration. The analysis computes stresses on a station-by-station basis along fuselage length.
These stresses are compressive ahead of the propulsion system and tensile behind the

propulsion system. If the aircraft is provided by wing-mounted propulsion pod, acceleration loads are introduced in the connection point between wing structural span and fuselage; otherwise they are introduced where fuselage-mounted propulsion system is located (see Figure 2.6). When the aircraft is provided with both wing and fuselage pods, principle of superposition is applied to compute the distribution of stresses.



(a) Underwing thrust (b) Tail thrust

Figure 2.6: Dependence of axial forces with thrust position

Tank or internal cabin pressure

For internal pressure loads, the longitudinal distribution of longitudinal and circumferential (hoop) stress resultants is computed at each fuselage station (see Figure 2.7). Within *technology* input file, it is possible to establish if pressure will be stabilized or not, editing the entry

```
user_input.analysis_setup.pressure_stabilization
```

NeoCASS allows to select among different fuselage structural concepts and consequently it is able to account for the fact a certain percentage of shell material (for example, the core material in sandwich design) is supposed to resist hoop stresses. More details are given in Section 2.6.1.

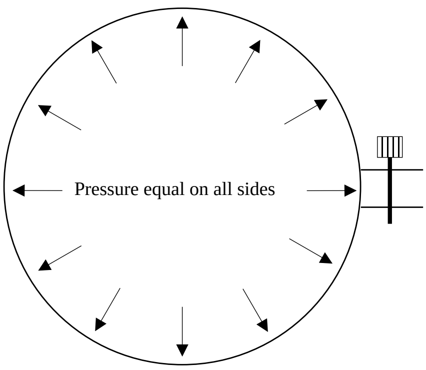


Figure 2.7: Internal cabin pressure distribution

Longitudinal bending moment

Bending loads are obtained by simulating:

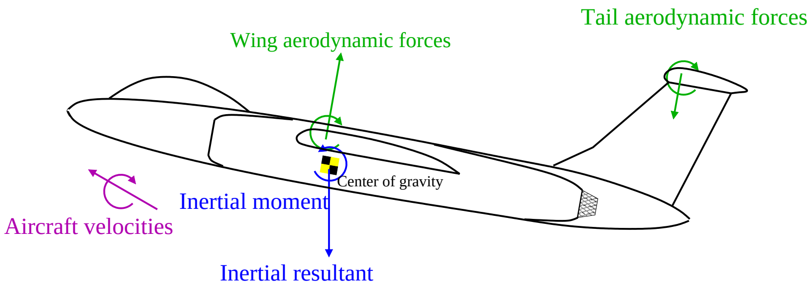


Figure 2.8: Resulting forces for pull-up maneuver.

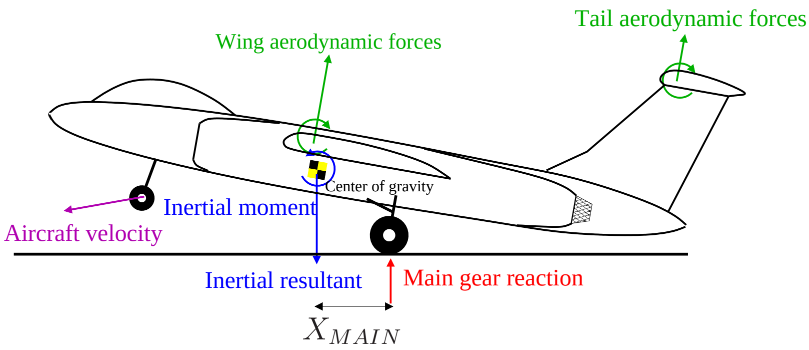


Figure 2.9: Resulting forces for a tail-down landing.

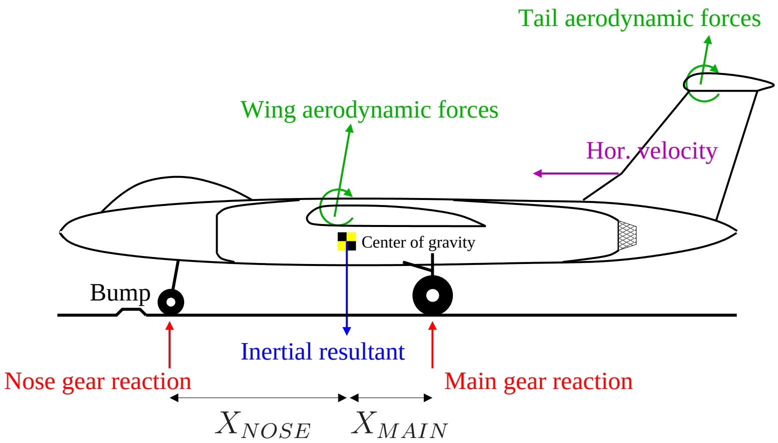


Figure 2.10: Resulting forces for a hit with a runaway bump.

- vehicle pitch-plane motion during a quasi-static pull-up maneuver at a given normal load factor (normally 2.5 for transport aircraft) as sketched in Figure 2.8; the user might update the normal load factor, editing the correspondent entry in *technology* input file (Appendix A); NeoCASS computes station-by-station fuselage mass distribution (already available from the *Geometry* module) and determines wing and tail lift loads to equilibrate airplane at current maneuver; at this stage, it is assumed that wing and tail lift forces act through the point where, respectively, wing and tail structural spans are connected to fuselage; further, the procedure distributes these concentrated loads over the correspondent structural root chord computing a system of forces and torques equivalent to the lift loads previously calculated by equilibrium. These calculations allow lift forces to be introduced gradually over wing and tail structural root chord;

- landing touchdown with a given sink velocity as sketched in Figure 2.9; it is assumed to analyze a tail-down landing where the landing load is introduced only in correspondence of the main landing gear; if the main landing gear is located in fuselage, the introduced load is a concentrated force; otherwise, if main landing gear is located under wings, the landing force is transmitted to fuselage through a distributed system of forces and torques, as previously done with the pull-up maneuver;

- a hit with a runway bump during taxing as sketched in Figure 2.10; nose and main landing gear loads are computed from equilibrium and longitudinal bending moment may be calculated; again, if the main landing gear is located under wings, the procedure can redefine its load as a distributed equivalent system of forces and torques.

When each single load contribution is computed, the net stress resultants in the axial direction (caused by longitudinal bending, axial acceleration and pressure) and in the circumferential (hoop) direction (caused by pressure only) may be calculated. It is assumed that acceleration loads never decrease stress resultants, while pressure loads may relieve stress if pressure stabilization is chosen as an option. This feature is outlined in detail in Section 2.6.1.

2.5.2 Lifting surfaces load condition

As originally described in [3], the load case used for the main lifting surface weight analysis is the quasi-static pull-up maneuver. Nevertheless, NeoCASS provides in addition a structural sizing tool for tail surfaces; horizontal and vertical planes are sized using appropriated load cases, as pointed out in the Sections 2.5.4 and 2.5.5, respectively.

2.5.3 Wing load condition

The wing load case is determined considering a quasi-static pull-up maneuver condition at a given normal load factor n. Load factor for the current maneuver is defined within the *technology* input file by the parameter

`user_input.loading.normal_load_factor`

and it can be edited manually by the user for different conditions.

The applied loads to the wing include the distributed lift and inertia forces, the point

loads of landing gear and propulsion, if placed on the wing. Moreover, wings can carry fuel within structural box. It is assumed that fuel is uniformly distributed with respect to the structural wing box volume.
Within the *Load module*, two lift distributions are available to compute lift loads for lifting surfaces along the wingspan, as shown in Figure 2.11. It is possible to select:

- a trapezoidal lift distribution [23] representing a uniform lift over the exposed area of trapezoidal wing panel, editing the following entry in the *technology* input file
  `user_input.analysis_setup.lift_distribution~=1;`

- Schrenk distribution representing an average between the trapezoidal and the elliptical distributions, where the lift is zero at the wingtip and maximum at the wing-fuselage intersection; the option is activated by setting the entry
  `user_input.analysis_setup.lift_distribution=1`

The adoption of Schrenk lift load distribution is suggested since it provides a better prediction of the actual aircraft loading [17].

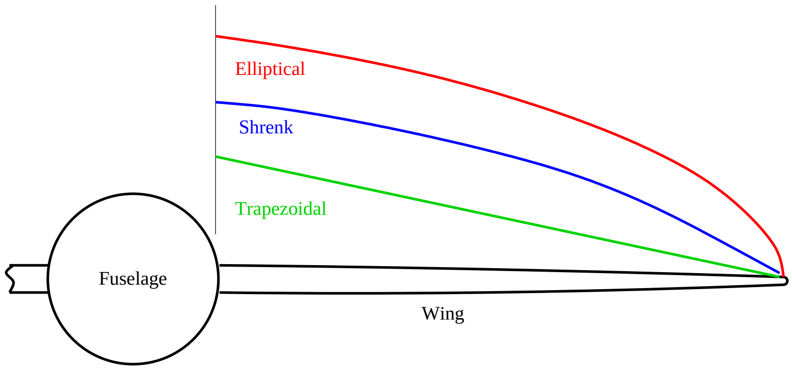


Figure 2.11: Lift distribution models available for aerodynamic load prediction

At each spanwise station along the elastic line, positioned in the middle of the wing box, from tip to wing-fuselage intersection, the lift load, center of pressure, inertia load, center of gravity, shear force and bending moment are computed.

Shear force and bending moment

The generic relations expressing the shear force and bending moment over the wing structural semispan at the user-specified loading condition are given by Equation 2.1 and 2.2, respectively:

$$F_S(y) = n\,K_S\left(L_{lift}(y) - L_{fuel}(y) - L_{eng}(y) - L_{lg}(y)\right) \tag{2.1}$$

$$M(y) = n\,K_S\left(M_{lift}(y) - M_{fuel}(y) - M_{eng}(y) - M_{lg}(y)\right) \tag{2.2}$$

The total shear force $F_S(y)$ and bending moment $M(y)$ are defined on a station-by-station basis along the structural semispan. Total shear force at the given normal load factor is determined by the sum of lift forces $L_{lift}(y)$ and inertia forces, caused by fuel $L_{fuel}(y)$, wing-mounted propulsion pods $L_{eng}(y)$ and landing gears $L_{lg}(y)$ force distribution.
Similarly, total bending moment is computed as the sum of bending moment due to lift

forces $M_{lift}(y)$ and inertia forces, accounting for fuel, wing-mounted propulsion pods and landing gears terms (denoted $M_{fuel}(y)$, $M_{eng}(y)$ and $M_{lg}(y)$, respectively).
The constant K_S in both equations is defined as *constant for shear stress in wing* and it is automatically set by the computer program.
In the following subsections, every single contribution appearing in Equation 2.1 and 2.2 is discussed in detail and more informations about the available computations are given.

Lift load distribution

The lift load is assumed to be distributed over the wing semispan. Lift load is zero at wing tip and maximum at wing-fuselage intersection. At the generic y station, the lift load distribution of the sector outboard of y is given as:

$$L_{lift}(y) = \left(\frac{W}{S}\right) A(y) \tag{2.3}$$

and the correspondent bending moment as:

$$M_{lift}(y) = \left(\frac{W}{S}\right) A(y)\, C_p(y) \tag{2.4}$$

where W represents the weight of the aircraft, S the exposed-aerodynamic area of the trapezoidal wing, $A(y)$ is the area outboard of y station and $C_P(y)$ is the centroid of area $A(y)$ measured respect to the actual y station.
As already anticipated, a good match between the predicted lift load and the actual aircraft lift load is obtained when Schrenk lift distribution is chosen as an option, setting
`user_input.analysis_setup.lift_distribution=1`.
Considering the *elliptical* lift load distribution as an example, the lift load matches the contour of an ellipse with the end of its major axis on the tip and the end of the minor axis above the wing/fuselage intersection. The values of the area of the ellipse outboard of y, $A(y)$, may be given in the form

$$A_{ELL}(y) = S_{ELL} - \left\{\frac{2\, S_{ELL}}{\pi\, b_S^2}\left[y\, \left(b_S^2 - y^2\right)^{\frac{1}{2}} + b_S^2\left(\sin\frac{y}{b_S}\right)^{-1}\right]\right\} \tag{2.5}$$

and, center of pressure of lift outboard of y, $C_P(y)$, for y measured along the structural semispan in the form

$$C_{P_{ELL}}(y) = \frac{4}{3\,\pi}\,(b_S - y) \tag{2.6}$$

where b_S represents the structural semispan running over the semiwing.

Fuel load distribution

As already introduced, forces and moments are calculated from wing tip to the wing/fuselage intersection for each generic section y. Referring to Figure 2.12, GUESS calculates

1. the volume of fuel $V_F(y)$ outboard of y section till wing tip, to compute the resulting shear force presented in Equation 2.7;

2. the centroid $C_g(y)$ of the volume $V_F(y)$ measured respect to y section, to compute bending moment presented in Equation 2.8.

These calculations are easily carried out directly in the *Geometry module* when all the geometry of the aircraft is considered. Data are stored during the geometry-processing and are then available for the load module.

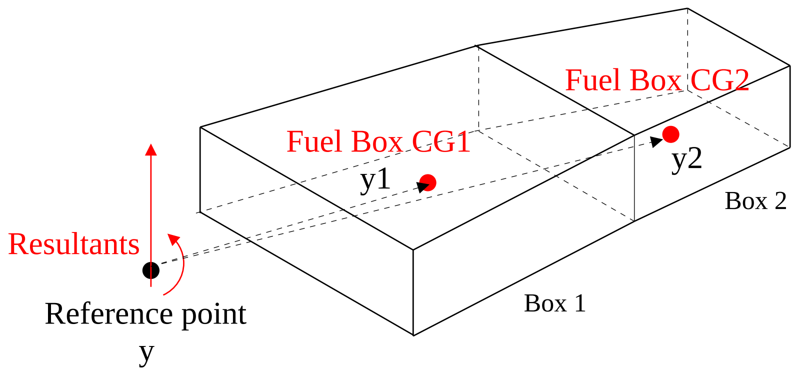


Figure 2.12: Fuel volume outboard of y-station and centroid with respect to y-station

At the generic y station, the shear force and bending moment caused by fuel weight is given, respectively, by:

$$L_{fuel}(y) = \left(\frac{W_{FT}}{V_W}\right) V_F(y) \tag{2.7}$$

and

$$M_{fuel}(y) = \left(\frac{W_{FT}}{V_W}\right) V_F(y)\, C_g(y) \tag{2.8}$$

where W_{FT} is the weight of fuel carried in the wings and V_W is the total volume of wing structural box where fuel is supposed to be stored, including both semiwings. Estimation of the fuel weight in the wing tanks is supplied by *Weight and Balance module*, reading in the *geometry* `.xml` input file the field:

`weight_balance.COG(18,4,1)`

selecting in the above matrix the row 18 for the correspondent entry *wing tanks* and column 4 for the mass value ($[kg]$).

Engines load distribution

In case engines are located under wings, the correspondent shear force and bending moment caused by propulsion pods along structural span are given by:

$$L_{eng}(y) = \sum_{i=1}^{n_e} h_e(y_{e_i} - y)\, W_{e_i} \tag{2.9}$$

$$M_{eng}(y) = \sum_{i=1}^{n_e} h_e(y_{e_i} - y)\, W_{e_i} \left(y_{e_i} - y\right) \tag{2.10}$$

where n_e is the number of engines mounted on the semispan, W_{e_i} and y_{e_i} are respectively the weight of the i^{th} engine and its location along the structural wing semispan. Force and moment are of course null outboard of the outer engine nacelle. Engine nacelles are considered as application point for loads as:

$$h_e(y_{e_i} - y) = \begin{cases} 1 & \text{if } y_{e_i} >= y \\ 0 & \text{if } y_{e_i} < y \end{cases} \tag{2.11}$$

The propulsion system is classified within the *geometry* input file in five different configurations, whose meaning is given in Table 2.7. The user is responsible to choose the better option representing the actual engine architecture. The options $0-3$ define the engine type and clearly specifying the attachment, either to the wing $(0-2)$ or to the fuselage (3). The remaining options $(4-5)$ do not define the way the engine is attached to the aircraft, considering the duct as a *floating* propulsion system.

Entry – **aircraft.engines1.*** **aircraft.engines2.***	**Note**
`Layout_and_config`	0 → attached by a pylon under or over the *wing* 1 → on *wing* nacelle with an inverted U section 2 → on *wing* nacelle with an elongated O section 3 → *fuselage* mounted engines attached by pylons 4 → straight duct without any attachment 5 → S duct without attachment

Table 2.7: Available engines configurations in *geometry* input file.

Since from the location of the propulsion pod is not known a priori the attachment to the airframe, which might depend on several technical reasons, GUESS asks the user to define the attachment to the wing or to the fuselage and correctly estimate the correspondent loads.
The weight of the propulsion system ($[kg]$) is estimated from *Weight and Balance module* and it is stored in the *geometry* input file, in the following entries,

```
aircraft.weight_balance.COG(7,4,1)
aircraft.weight_balance.COG(8,4,1)
```

while the number of engines is specified in

```
aircraft.engines1.Number_of_engines
aircraft.engines2.Number_of_engines
```

Landing gears load distribution

Main landing gear system can be located either on fuselage or wings. In case they are located under wings, the shear force and bending moment they introduce are computed by:

$$L_{lg}(y) = \sum_{i=1}^{n_{lg}} h_{lg}(y_{lg_i} - y)\, W_{lg_i} \tag{2.12}$$

$$M_{lg}(y) = \sum_{i=1}^{n_{lg}} h_{lg}(y_{lg_i} - y)\, W_{lg_i}\left(y_{lg_i} - y\right) \tag{2.13}$$

where n_{lg} is the number of landing gears mounted on the semispan, W_{lg_i} and y_{lg_i} are respectively the weight of the i^{th} landing gear and its location along the structural wing semispan. As it is done for propulsion system, landing gears are considered concentrated loads:

$$h_{lg}(y_{lg_i} - y) = \begin{cases} 1 & \text{if } y_{lg_i} >= y \\ 0 & \text{if } y_{lg_i} < y \end{cases} \tag{2.14}$$

The weight of the landing gear system ($[kg]$) is estimated by *Weight and Balance module* and it is stored in

`aircraft.weight_balance.COG(6,4,1)`

selecting in the above matrix the row 6 for the correspondent entry *landing gears* and column 4 for the mass value ($[kg]$).

2.5.4 Horizontal tail load condition

Horizontal tail loads affect the design of a significant part of the aircraft structure and hence require careful consideration of the various design requirements and resulting conditions. In general the structures that are designed by horizontal tail loads are:

1. the horizontal tail stabilizer and elevator;
2. the body structure aft of the pressure bulkhead and horizontal tail support structure;
3. the aft fuselage monocoque structure;
4. the fuselage center section (overwing) structure;
5. the stabilizer actuator (jackscrew mechanism).

Parameters required for horizontal tail structural load analysis

Parameters required for horizontal tail structural load analysis are reported in Table 2.8. Several different load conditions will be considered in the following sections and relationship between horizontal tail loads, pitching moment, stabilizer angle of attack and elevator angle will be pointed out. GUESS implements these load conditions for horizontal tail structural analysis.

As pointed out in Appendix A, the user might define the required parameters for the current analysis if accurate values are available by means of more dedicated tools. In case the user does not own further details, GUESS can compute the above parameters using a Vortex-Lattice Method.

Balanced maneuver analysis

Horizontal tail load and pitching moment may be defined as functions of stabilizer angle of attack, α_s, and elevator angle, δ_e, as shown in Equations (2.15) and (2.16).

$$L_t = L_{\alpha_s}\,\alpha_s + L_{\delta_e}\,\delta_e + L_c \tag{2.15}$$

$$M_t = M_{\alpha_s}\,\alpha_s + M_{\delta_e}\,\delta_e + M_c \tag{2.16}$$

Engineering symbol	Variable name	Notes	Unit
L_{α_s}	`L_alpha_s`	horizontal tail load due to unit α_s	[N/rad]
M_{α_s}	`M_alpha_s`	horizontal pitching moment due to unit α_s	[Nm/rad]
L_{δ_e}	`L_delta_e`	horizontal tail load due to unit δ_e	[N/rad]
M_{δ_e}	`M_delta_e`	horizontal pitching moment due to unit δ_e	[Nm/rad]
L_c	`L_c`	horizontal tail load due to unit built-in chamber	[N]
M_c	`M_c`	horizontal pitching moment due to unit built-in chamber	[Nm]
$d\,\alpha_s\,/\,d\,n_z$	`dalphas_dnz`	fuselage flexibility due to n_z	[rad]
$d\,\alpha_s\,/\,d\,L_t$	`dalphas_dLt`	fuselage flexibility due to L_t	[rad/N]
$d\,\alpha_s\,/\,d\,M_t$	`dalphas_dMt`	fuselage flexibility due to M_t	[rad/Nm]
$C_{M\,0.25}$	`CM_025`	pitching moment coefficient about 0.25 *mac* wing	[−]

Table 2.8: Parameters for horizontal tail load analysis.

where L_t is the horizontal tail load ([N]), M_t is the horizontal tail pitching moment about $0.25\,mac_t$ ([Nm]); L_{α_s} and M_{α_s} are the tail load and pitching moment due to unit α_s ([N/rad] and [Nm/rad], respectively); L_{δ_e} and M_{δ_e} are the tail load and pitching moment due to unit δ_e ([N/rad] and [Nm/rad], respectively); L_c and M_c are the tail load and pitching moment due to unit built-in camber ([N] and [Nm], respectively).
The horizontal stabilizer angle of attack, α_s, is calculated considering different incremental contributions, as pointed out in Equation (2.17)

$$\alpha_s \;=\; s \,+\, \triangle\alpha_{s_1} \,+\, \triangle\alpha_{s_2} \,+\, \triangle\alpha_{s_3} \tag{2.17}$$

where s is the initial trim setting of the stabilizer with respect to a body reference plane. The increment due to wing angle of attack is modified by the wing downwash at the horizontal stabilizer in Equation (2.18)

$$\triangle\alpha_{s_1} \;=\; (1 \,-\, \epsilon_{\alpha\,w})\;\alpha_w \,-\, \epsilon_0 \tag{2.18}$$

where $\epsilon_{\alpha\,w}$ is the change in wing angle of attack (denoted α_w) due to downwash at the horizontal tail reference point (the quarter-chord of the horizontal tail mean aerodynamic chord) and ϵ_0 is the downwash angle at the horizontal tail at $\alpha_w \,=\, 0$.
The increment due to pitching velocity is derived for a steady-state maneuver in Equation (2.19)

$$\triangle\alpha_{s_2} \;=\; 57.3\,l_t\,g\;(n_z \,-\, 1)\;/\;V_t^2 \tag{2.19}$$

where l_t is the distance between horizontal tail mean aerodynamic center and center of mass (Figure 2.13). The stabilizer angle of attack is affected moreover from body flexibility (Equation (2.20)) due to tail loads, normal load factor and pitching acceleration with respect the center of mass, as defined in Figure 2.14-2.15.

$$\triangle\alpha_{s_3} \;=\; \left(\frac{d\,\alpha_s}{d\,n_z}\right)\,n_z \,+\, \left(\frac{d\,\alpha_s}{d\,L_t}\right)\,L_t \,+\, \left(\frac{d\,\alpha_s}{d\,M_t}\right)\,M_t \tag{2.20}$$

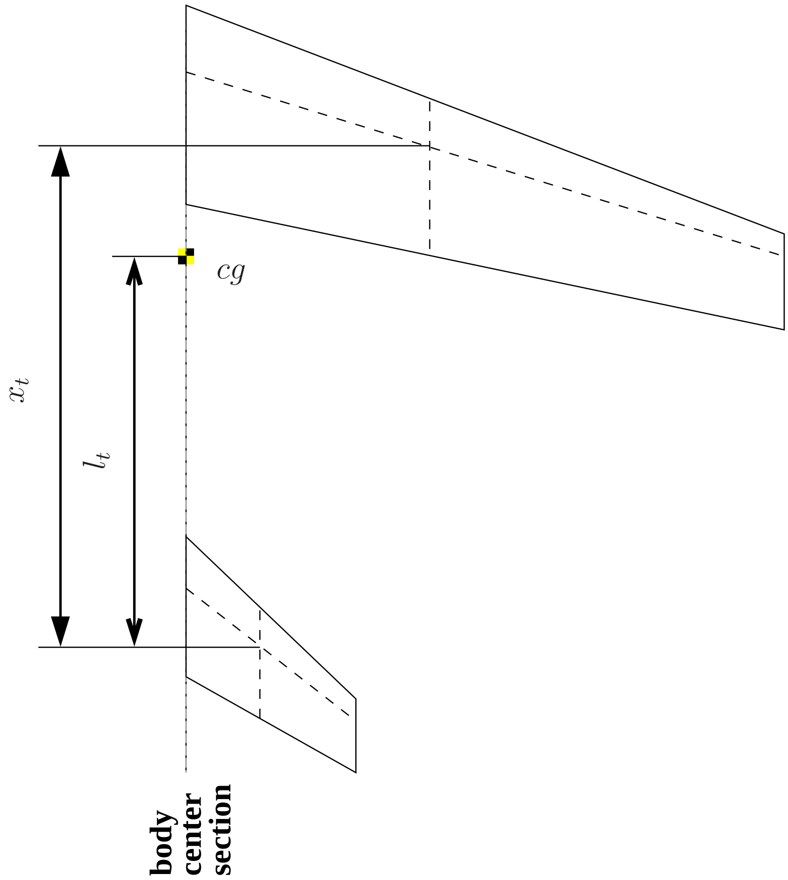


Figure 2.13: Geometric parameters for horizontal tail.

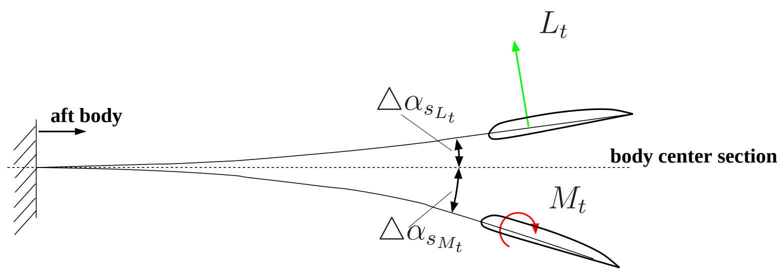


Figure 2.14: Vertical bending due to horizontal tail loads.

Balanced maneuver analysis: equations for conditions with unknown elevator angle

Equations 2.15 through 2.17 may be arranged in matrix notation as shown in Equation 2.21 for conditions where the elevator angle is an unknow. Solution of this set of equations will give the tail load, pitching moment, stabilizer angle of attack, and the elevator required to accomplish the maneuver

$$\begin{bmatrix} 1 & 0 & -L_{\alpha_s} & -L_{\delta_e} \\ 0 & 1 & -M_{\alpha_s} & -M_{\delta_e} \\ x_t & -1 & 0 & 0 \\ b_1 & b_2 & 1 & 0 \end{bmatrix} \begin{Bmatrix} L_t \\ M_t \\ \alpha_s \\ \delta_e \end{Bmatrix} = \begin{Bmatrix} L_c \\ M_c \\ BTL\, x_t \\ c_1 \end{Bmatrix} \tag{2.21}$$

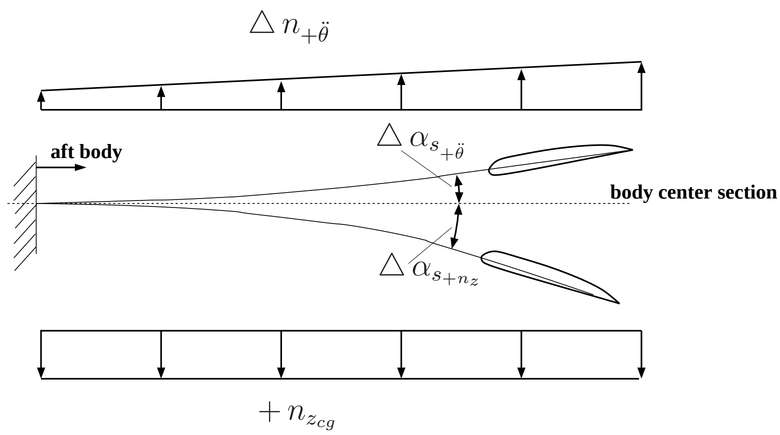


Figure 2.15: Vertical bending due to load factor and pitching acceleration.

where

$$b_1 \;=\; -\left(\frac{d\,\alpha_s}{d\,L_t}\right) \tag{2.22}$$

$$b_2 \;=\; -\left(\frac{d\,\alpha_s}{d\,M_t}\right) \tag{2.23}$$

$$c_1 \;=\; (1 \,-\, \epsilon_{\alpha\,w})\; \alpha_w \,-\, \epsilon_0 \,+\, 57.3\, l_t\, g\; (n_z \,-\, 1)\; /\; V_t^2 \,+\, s_{trim} \,+\, \left(\frac{d\,\alpha_s}{d\,n_z}\right)\, n_z \tag{2.24}$$

The term BTL is referred to as the balancing tail load and it might be computed from rotational equilibrium assuming the tail moment over the length x_t to be neglegible respect the other contributions in Equation 2.25

$$BTL \;=\; n_z\, W\; (x_t \,-\, l_t) \,+\, M_{0.25\,mac}\, /\, x_t \tag{2.25}$$

Balanced maneuver analysis: equations for conditions with known elevator angle

For steady-state maneuvers where the elevator angle is known, Equation 2.21 can be reduced to three unknowns as herein reported. Simultaneous solution of this set of equations will give the tail load, pitching moment, and stabilizer angle of attack.

$$\begin{bmatrix} 1 & 0 & -L_{\alpha_s} \\ 0 & 1 & -M_{\alpha_s} \\ x_t & -1 & 0 \end{bmatrix} \begin{Bmatrix} L_t \\ M_t \\ \alpha_s \end{Bmatrix} = \begin{Bmatrix} L_c \,+\, L_{\delta_e}\,\delta_e \\ M_c \,+\, M_{\delta_e}\,\delta_e \\ BTL\, x_t \end{Bmatrix} \tag{2.26}$$

The stabilizer geometric position required to attain the load factor n_z may be calculated from Equation 2.27

$$s \;=\; \alpha_s \,-\, (\triangle\alpha_{s_1} \,+\, \triangle\alpha_{s_2} \,+\, \triangle\alpha_{s_3}) \tag{2.27}$$

The user might set the known elevator deflection by editing the entries in the *technology* input file

```
user_input.loading.Elevator_limit_deflection_up
user_input.loading.Elevator_limit_deflection_down
```

Abrupt unchecked elevator conditions

Horizontal tail loads may be modified respect the equations shown in 2.15 and 2.16 and rearranged in the following form

$$L_t \;=\; L_{t\,nz=1} \;+\; \underbrace{L_{\alpha_s}\,\triangle\alpha_s \;+\; L_{\delta_e}\,\triangle\delta_{e\,max}}_{\triangle L_{t\,\theta}} \tag{2.28}$$

$$M_t \;=\; M_{t\,nz=1} \;+\; \underbrace{M_{\alpha_s}\,\triangle\alpha_s \;+\; M_{\delta_e}\,\triangle\delta_{e\,max}}_{\triangle M_{t\,\theta}} \tag{2.29}$$

where $L_{t\,nz=1}$ and $M_{t\,nz=1}$ are horizontal tail load and pitching moment calculated for $1-g$ flight condition ([N] and [Nm], respectively), L_{α_s} and M_{α_s} are horizontal tail load and pitching moment due to change in α_s ([N/rad] and [Nm/rad], respectively), L_{δ_e} and M_{δ_e} are horizontal tail load and pitching moment due to change in δ_e ([N/rad] and [Nm/rad], respectively), $\triangle\alpha_s$ is the change in horizontal tail angle of attack due to airplane response and body flexibility, and $\triangle\delta_e$ is the change in elevator angle.
Defining the airplane response factor for an abrupt unchecked maneuver as

$$K_r \;=\; \frac{\triangle L_{t\,\theta}}{L_{\delta_e}\,\triangle\delta_{e\,max}}$$

combining with Equation 2.28, the change in stabilizer angle of attack becomes

$$\triangle\alpha_s \;=\; (K_r \,-\, 1)\;(L_{\delta_e}\,/\,L_{alpha_s})\;\triangle\delta_{e_{max}} \tag{2.30}$$

Thus, the net horizontal tail load and pitching moment due to an abrupt unchecked elevator in terms of the response parameter are

$$L_t \;=\; L_{t\,nz=1} \;+\; K_r\,L_{\delta_e}\,\triangle\delta_{e_{max}} \tag{2.31}$$

$$M_t \;=\; M_{t\,nz=1} \;+\; [M_{\alpha_s}\;(K_r \,-\, 1)\;L_{\delta_e}\,/\,L_{\alpha_s} \;+\; L_{\delta_e}]\;\triangle\delta_{e_{max}} \tag{2.32}$$

For analysis load surveys to determine the critical abrupt elevator horizontal tail load condition, a conservative response factor of $K_r = 0.90$ may be used.

Checked maneuver conditions

Using a simplified approach, the total horizontal tail load and pitching moment due to checked maneuver can be written in terms of the balance increment plus the increment due to pitching acceleration caused by returning the elevator to neutral or overchecked position:

$$L_{t\,cm} \;=\; L_{t\,bal} \;+\; L_{t\,\delta_e}\,\triangle_{e\,cm} \tag{2.33}$$

$$M_{t\,cm} \;=\; M_{t\,bal} \;+\; M_{t\,\delta_e}\,\triangle_{e\,cm} \tag{2.34}$$

where $L_{t\,cm}$ and $M_{t\,cm}$ are the check maneuver tail load and pitching moment ([N] and [Nm], respectively), $L_{t\,bal}$ and $M_{t\,bal}$ are the balancing tail load and pitching moment at normal load factor n_z ([N] and [Nm], respectively), $L_{t\,\delta_e}$ and $M_{t\,\delta_e}$ are the tail load and pitching moment due to $\triangle_{e\,cm}$ ([N] and [Nm], respectively), and $\triangle_{e\,cm}$ is the incremental check maneuver elevator angle, defined as

$$\triangle_{e\,cm} \;=\; -\;(1 \,+\, CBF)\;(\delta_{e\,bal} \,-\, \delta_{e\,trim}) \tag{2.35}$$

where CBF the checkback factor equal zero (0) for elevator returned to the original trim position and equal to 0.50 for overchecked conditions, $\delta_{e\,bal}$ is the elevator angle required for steady-state maneuver at n_z, and $\delta_{e\,trim}$ is the elevator angle at the beginning of the maneuver, $n_z = 1$.

Following the above guidelines, GUESS computes horizontal tail lift and pitching moment. Assuming the horizontal plane to be similar to the main wing, it is possible to distribute the loads, applied to the mean aerodynamic center as concentrated force and moment, over the lifting surface semispan. Thus, horizontal tail loads may be recomputed applying the two available types of lift load distributions, namely trapezoidal and Schrenk distribution, in the very similar way already described in Section 2.5.3.
Horizontal tail loads are thus defined on a station-by-station basis along the structural span defined in *Geometry module* and shear force and bending moment might be easily computed for the next *Structural module* analysis.

2.5.5 Vertical tail load condition

Vertical tail loads affect the design of a significant part of the aircraft structure and thus require careful consideration of the various design requirements and resulting conditions. The structures affected by vertical tail loads are:

1. the vertical tail and rudder;

2. the aft body structure;

3. the horizontal tail structure if the tail is mounted up to the fin, as on the $DC-10$ or on top of the fin like 727 configurations;

4. the fuselage center section (underwing) structure.

In general, the basic conditions that will determine the maximum loads for the vertical tail and related structure are:

- yawing maneuver conditions (pilot induced and engine out);

- lateral guest.

Other conditions such as rolling maneuvers usually are not as critical for design of the vertical tail structure except possibly for structural configurations with horizontal tails mounted up to fin.
Within the current GUESS version, yawing maneuvers are considered as load conditions for the vertical tail while lateral guest is left for future enhancement of the structural sizing modulus.
Yawing maneuvers when applied to structural load analysis are maneuvers involving the abrupt application of the rudder in producing a sideslip condition or during engine-out conditions. Two types of yawing maneuvers must be considered for structural design:

1. rudder maneuvers as used for structural design are essentially flat maneuvers whereby the rudder is abruptly applied in a wings-level attitude (see 2.16). This maneuver

is difficult to do in flight because large amounts of lateral control must be applied to maintain wings level. The purpose of holding the wings level is to maximize the resulting sideslip;

2. engine-out maneuvers, as used for structural design, are essentially flat maneuvers whereby abrupt application of the rudder is made in conjunction with the resulting sideslip due to unsymmetrical engine thrust (see 2.17).

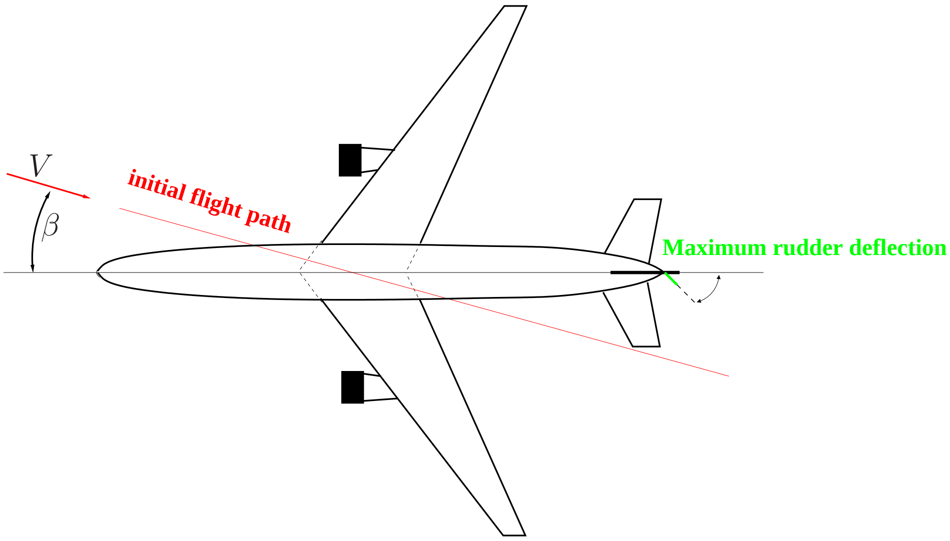


Figure 2.16: Yawing maneuvers: pilot-induced rudder maneuvers.

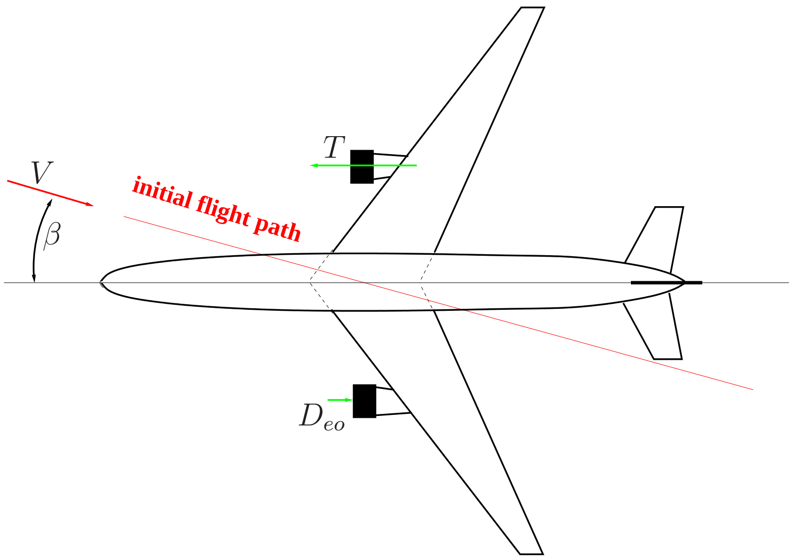


Figure 2.17: Yawing maneuvers: asymmetrical thurst (engine-out) maneuvers.

Parameters required for vertical tail structural load analysis

The parameters required for structural load analysis for yawing maneuvers conditions are shown in 2.9. The relationship between the applied rudder and sideslip for rudder maneuvers, and between unsymmetrical thrust, sideslip and corrective rudder for engine-out maneuvers, may be determined using the methods developed in the following sections.

Engineering symbol	Variable name	Notes	Unit
CY_{δ_r}	`CY_dr`	side force coefficient due to rudder	$[\mathrm{rad}^{-1}]$
Cn_{δ_r}	`Cn_dr`	yawing moment coefficient due to rudder	$[\mathrm{rad}^{-1}]$
Cl_{δ_r}	`Cl_dr`	rolling moment coefficient due to rudder	$[\mathrm{rad}^{-1}]$
CY_{δ_w}	`CY_dw`	side force coefficient due to aileron/spoilers	$[\mathrm{rad}^{-1}]$
Cn_{δ_w}	`Cn_dw`	yawing moment coefficient due to aileron/spoilers	$[\mathrm{rad}^{-1}]$
Cl_{δ_w}	`Cl_dw`	rolling moment coefficient due to aileron/spoilers	$[\mathrm{rad}^{-1}]$
Cn_{β}	`Cn_b`	yawing moment coefficient due to sideslip	$[\mathrm{rad}^{-1}]$
Cl_{β}	`Cl_b`	rolling moment coefficient due to sideslip	$[\mathrm{rad}^{-1}]$
CY_{β}	`CY_b`	side force coefficient due to sideslip	$[\mathrm{rad}^{-1}]$
CL	`CL`	airplane lift coefficient	$[-]$

Table 2.9: Parameters for vertical tail load analysis.

As pointed out in Appendix A, the aero-data required for the present loading analysis might be evaluated by a dedicated program implemented within GUESS in case the user does not provide further detailed values obtained from more complex calculations. Otherwise the user can edit and update every single entry, as explained in Appendix A.
GUESS contains a program for the evaluation of all the aero-data collected in Table 2.9 and a short comparison against the Boeing $747-100$ is reported in Appendix D.

Rudder maneuver requirements-FAR 25 Criteria

A similar procedure to the one described in $FAR\,25.351(a)$ is considered for the load condition due to rudder deflection. The use of the commercial regulations has proven adequate for structural design over the past years and the criteria applied are based on simplistic maneuvers that are difficult to achieve in actual service operation of commercial aircraft.
With the airplane in unaccelerated flight at zero yaw, it is assumed that the rudder control is suddenly displaced to the maximum deflection as limited by the namelist in the *technology* `.xml` input file

`user_input.loading.Rudder_limit_deflection`

With the rudder deflected, it is assumed that the airplane yaws to the resulting sideslip. Assuming that the airplane is in level flight at a constant airspeed and the aerodynamic coefficients are linear, the three equations for side force, yawing and rolling moments may be written in coefficient form, using notation form:

$$\begin{bmatrix} CY_\beta & CY_{\delta_w} & -CL \\ Cn_\beta & Cn_{\delta_w} & 0 \\ Cl_\beta & Cl_{\delta_w} & 0 \end{bmatrix} \begin{Bmatrix} \beta_{SS} \\ \delta_w \\ \phi \end{Bmatrix} = - \begin{Bmatrix} CY_{\delta_r} \\ Cn_{\delta_r} \\ Cl_{\delta_r} \end{Bmatrix} \delta_r \tag{2.36}$$

where β, δ_r and δ_w represent sideslip, rudder and control wheel angles, respectively; CY_β, CY_{δ_r} and CY_{δ_w} are the side force coefficient due to sideslip, rudder and aileron/spoiler, respectively; Cn_β, Cn_{δ_r} and Cn_{δ_w} are the yawing moment coefficient due to sideslip, rudder

and aileron/spoiler, respectively; Cl_β, Cl_{δ_r} and Cl_{δ_w} are the rolling moment coefficient due to sideslip, rudder and aileron/spoiler, respectively; CL is the airplane lift coefficient and ϕ the airplane bank angle.
Since the rudder available for the current condition is known, solution of equation (2.36) may be performed for the sideslip (Eq. (2.37)), wheel (Eq. (2.38)) and bank (Eq. (2.39)) angles for steady sideslip:

$$\beta_{SS} = \left(\frac{-Cn_{\delta_r} + Cl_{\delta_r}\, Cn_{\delta_w} \,/\, Cl_{\delta_w}}{Cn_\beta - Cl_\beta\, Cn_{\delta_w} \,/\, Cl_{\delta_w}} \right) \delta_r \tag{2.37}$$

$$\delta_w = (- Cl_{\delta_r}\, \delta_r - Cl_\beta\, \beta_{SS}) \,/\, Cl_{\delta_w} \tag{2.38}$$

$$\phi = (CY_\beta\, \beta_{SS} + CY_{\delta_r}\, \delta_r + CY_{\delta_w}\, \delta_w) \,/\, CL \tag{2.39}$$

Engine-out maneuver requirements-FAR 25 Criteria

The engine-out maneuver is defined by $FAR\,25.367$ and basically two design conditions are considered: engine-out conditions due to fuel flow interruption; engine-out conditions due to mechanical failure of the engine or propeller system. The difference between these two conditions is the time of engine thrust decay. Fuel flow interruption may occur from $1s$ to as long as several seconds, whereas a mechanical failure happens very abruptly.
Structural sizing developed within GUESS considers only the case of engine-out condition due to mechanical failure and a factor of safety of 1.0 is applied. The following considerations are helpful to figure out the resulting sideslip due to unsymmetrical engine thrust. The amount of yawing moment due to engine-out may be determined from (2.40), considering the thrust on the remaining engine and the drag of the dead engine:

$$Cn_{EO} = (T + D_{EO})\, a_{EO} \,/\, (q\, S_w\, b_w) \tag{2.40}$$

where T is the engine net thrust, D_{EO} is the drag of the dead engine and a_{EO} is the arm of the dead engine; q is the dynamic pressure at flight condition, S_w and b_w are, respectively, wing reference area and span. Geometrical parameters are sketched in Figure 2.18.
Equation (2.40) represents the case of single failure in which the thrust and dead engine drag are acting on opposite engines; in general the equation may be applied for a configuration with more than two engines.
The steady-state equations for engine-out conditions are represented in matrix notation in Equation (2.41), determined in a similar way to the system of equations obtained for rudder maneuvers.

$$\begin{bmatrix} CY_\beta & CY_{\delta_w} & CY_{\delta_r} \\ Cn_\beta & Cn_{\delta_w} & Cn_{\delta_r} \\ Cl_\beta & Cl_{\delta_w} & Cl_{\delta_r} \end{bmatrix} \begin{Bmatrix} \beta_{EO} \\ \delta_w \\ \delta_r \end{Bmatrix} = \begin{Bmatrix} CL\,\phi \\ -Cn_{EO} \\ 0 \end{Bmatrix} \tag{2.41}$$

When it is assumed that rudder is held neutral, the engine-out steady sideslip with zero rudder may be determined from solution of Equations (2.42) through (2.44).

$$\beta_{EO} = \frac{-Cn_{EO}}{Cn_\beta - Cl_\beta\, Cn_{\delta_w} \,/\, Cl_{\delta_w}} \tag{2.42}$$

$$\delta_w = -Cl_\beta\, \beta_{EO} \,/\, Cl_{\delta_w} \tag{2.43}$$

$$\phi_{EO} = \left(CY_\beta \, \beta_{EO} + CY_{\delta_w} \, \delta_w\right) / CL \tag{2.44}$$

From the steady sideslip solutions as above determined, the steady condition with zero sideslip may be accomplished by use of rudder to balance the unsymmetrical engine-out condition, solving Equations (2.45) through (2.47).

$$\delta_{r_{EO}} = \frac{-Cn_{EO}}{Cn_{\delta_r} - Cl_\beta \, Cn_{\delta_w} / Cl_{\delta_w}} \tag{2.45}$$

$$\delta_w = -Cl_{\delta_r} \, \delta_{r_{EO}} / Cl_{\delta_w} \tag{2.46}$$

$$\phi_{EO} = \left(CY_{\delta_r} \, \delta_{r_{EO}} + CY_{\delta_w} \, \delta_w\right) / CL \tag{2.47}$$

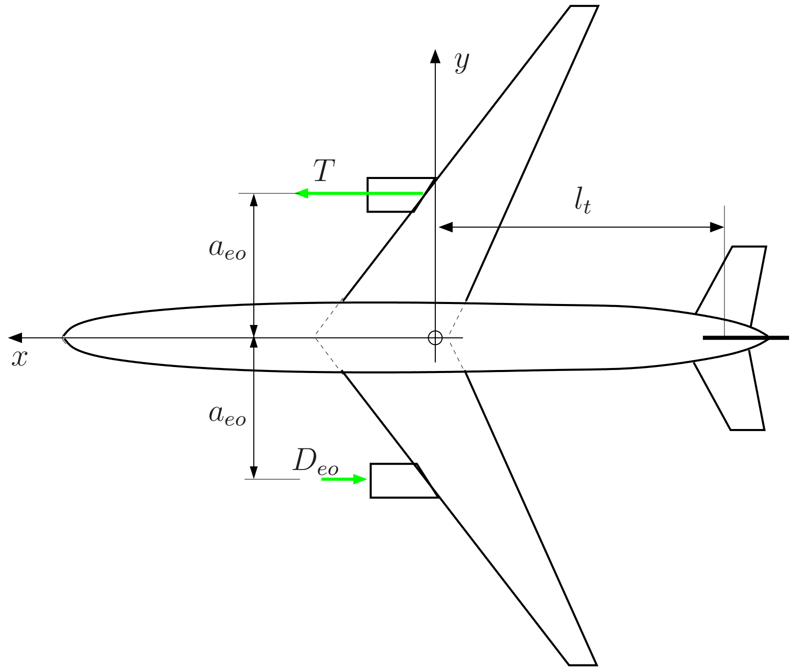


Figure 2.18: Engine-out geometry.

Since from the outlined procedures, it is possible to determine the angle of sideslip, the *Strip Theory* is used to predict the distributed lift load over the vertical tail structural semi-span. Shear force and bending moment might be easily computed on a station-by-station basis for the next *Structural module* analysis.

2.6 Structural module

Once total shear force and bending moment along fuselage length and lifting surfaces structural span have been computed, GUESS applies the sizing tool developed within the *Structural module*. Net stress resultants are calculated on a station-by-station basis for the components identified by the user in the stick model. Then, the minimum amount of structural material required to preclude failure in the most critical conditions for fuselage and lifting surfaces is determined.

In the following pages, a brief description of the structural sizing procedure is given and few informations to correctly setup the *technology* `.xml` input file in relation to the available structural concepts are proposed.

2.6.1 Fuselage structural analysis

Weight estimates are developed for the load-carrying fuselage structure, which are ruled by the user by specifying in the input file one of the seven different structural geometry concepts reported in Table 2.10.

Entry `kcon`	**Structural concept**
1	Simply stiffened shell, frames, sized for minimum weight in buckling
2	Z-stiffened shell, frames, best buckling
3	Z-stiffened shell, frames, buckling-minimum gage compromise
4	Z-stiffened shell, frames, buckling-pressure compromise
5	Truss-core sandwich, frames, best buckling
6	Truss-core sandwich, no frames, best buckling
7	Truss-core sandwich, no frames, buckling-minimum gage-pressure compromise

Table 2.10: Available fuselage structural geometry concepts

GUESS provides a simply stiffened shell concept using longitudinal frames, three concepts with Z-stiffened shells and longitudinal frames, one with structural material proportioned to give minimum weight in buckling, one with buckling efficiency compromise to result in lighter weight in minimum gage, and one a buckling-pressure compromise. Similarly, there are three truss-core sandwich designs, two for minimal weight in buckling with and without frames, and one a buckling-minimum gage compromise.
To correctly update the *technology* `.xml` input file for different aircraft configurations, the user shall set the entry

`user_input.material_property.fus.kcon`

choosing one of the seven available fuselage structural concepts (from $1-7$) with the physical/structural meaning as illustrated in Table 2.10.
According to the specific fuselage structural concept, GUESS loads a ASCII file where all the fuselage structural geometric parameters are stored and sorted by structural concepts (Appendix C). It is then possible to perform the structural sizing for the different available fuselage concepts.
Failure modes considered for fuselage sizing are:

1. compressive yield strength;
2. ultimate tensile strength;
3. local buckling and gross buckling of the entire structure;
4. a minimum gage restriction on the thickness.

In order to carry out the sizing, the following assumptions are made :

- maximum stress failure theory is used for predicting yield failure;
- buckling calculations assume stiffened shells behave as wide columns and sandwich shells behave as cylinders;
- the frames used for stiffened shells are sized according to *Shanley* Criterion, which considers the frames as elastic support for the wide column.

Stress resultants in axial and hoop direction

Once fuselage loads have been determined as presented in Section 2.5.1, and considering first the circular shell, the stress resultants in the axial direction (dimensionally $[N/m]$) caused by longitudinal bending, axial acceleration, and pressure at a generic fuselage station x are

$$N_{xB} = \frac{M\,r}{I_y'} \tag{2.48}$$

$$N_{xA} = \frac{W_s}{P} \tag{2.49}$$

$$N_{xP} = \frac{A\,P_g}{P} \tag{2.50}$$

respectively, where r is the fuselage radius, $A = \pi\,r^2$ is the fuselage cross-sectional area, and $P = 2\,\pi\,r$ is the fuselage perimeter. In Equation 2.48, $I_y' = \pi\,r^3$ is the moment of inertia of the shell divided by the shell thickness. In Equation 2.49, accounting for the engines thrust, W_s is the portion of vehicle inertia ahead of station x if x is ahead of the inlet entrance, or the portion of vehicle inertia behind x if x is behind the nozzle exit. In Equation 2.50, P_g is the pressure differential for the passenger compartment during cruise.
The total tension resultant is thus

$$N_x^+ = N_{xB} + N_{xP} \tag{2.51}$$

if x is ahead of the nozzle exit, and

$$N_x^+ = N_{xB} + N_{xP} + N_{xA} \tag{2.52}$$

if x is behind it. Similarly, the total compressive stress resultant is

$$N_x^- = N_{xB} + N_{xA} - \begin{cases} 0 & \text{if not pressure stabilized} \\ N_{xP} & \text{if stabilized} \end{cases} \tag{2.53}$$

if x is ahead of the nozzle exit, and

$$N_x^- = N_{xB} - \begin{cases} 0 & \text{if not pressure stabilized} \\ N_{xP} & \text{if stabilized} \end{cases} \tag{2.54}$$

if x is behind it. These relations are based on the premise that acceleration loads never decrease stress resultants, but pressure loads may relieve stress, if pressure stabilization is chosen as an option. The user shall set the entry

```
user_input.analysis_setup.pressure_stabilization=1
```

if pressure differential for passengers compartment is stabilized at cruise altitude, or

`user_input.analysis_setup.pressure_stabilization=0`

if pressure is not stabilized. More informations about options ruled by the user may be found in Appendix A.

The stress resultant in the hoop direction, dimensionally $[N/m]$, is

$$N_y = r\, P_g\, K_P \tag{2.55}$$

where K_P accounts for the fact that not all of the shell material (i.e., the core material in sandwich designs) is available for resisting hoop stress. The value of the coefficient K_P is given in Table C.2, Appendix C, for each fuselage structural concept.

Equivalent isotropic thickness for shell and frames

The equivalent isotropic thicknesses of the shell are given by

$$t_{S_C} = \frac{N_x^-}{F_{cy}} \tag{2.56}$$

$$t_{S_T} = \frac{1}{F_{tu}} \max\left(N_x^+\,,\, N_y\right) \tag{2.57}$$

$$t_{S_G} = K_{mg}\, t_{mg} \tag{2.58}$$

for designs limited by compressive yield strength (F_{cy}), ultimate tensile strength (F_{tu}), and minimum gage, respectively. A fourth thickness that must be considered is that for buckling critical design, t_{S_B}. The following relations are derived for the equivalent isotropic thickness of the shell required to preclude buckling, t_{S_B}, and for the smeared equivalent isotropic thickness of the ring frames required to preclude general instability, t_F.

Stiffened shell with frames concept: `user_input.material_property.fus.kcon`, 1 to 5. Assuming the shell to be a wide column, denoting the frame spacing as d and Young's modulus for the shell material as E, the buckling equation is

$$\frac{N_x^-}{d\,E} = \varepsilon \left(\frac{t_{S_B}}{d}\right)^2 \tag{2.59}$$

or, solving for the shell thickness t_{S_B}

$$t_{S_B} = \sqrt{\frac{N_x^-\, d}{E\,\varepsilon}} \tag{2.60}$$

The frames are next sized to prevent general instability failure. Assuming that frames act as elastic support for the wide column (*Shanley* Criterion), the smeared equivalent thickness of the frames is

$$t_{F_B} = 2\, r^2 \sqrt{\frac{\pi\, C_F\, N_x^-}{K_{F1}\, d^3\, E_F}} \tag{2.61}$$

where C_F is Shanley's constant, K_{F1} is a frame geometry parameter, and E_F is Young's modulus for the frame material. Assuming the structure to be buckling critical, the equivalent thickness of the structure, $t = t_{S_B} + t_{F_B}$, must be minimized with respect d, the frame spacing, leading to the following relations

$$t = \frac{4}{27^{1/4}} \left(\frac{\pi\, C_F}{K_{F1}\, \varepsilon^3\, E_F\, E^3} \right)^{\frac{1}{8}} \left(\frac{2\, r^2\, \rho_F\, (N_x^-)^2}{\rho} \right)^{\frac{1}{4}} \tag{2.62}$$

$$t_{S_B} = \frac{3}{4}\, t \tag{2.63}$$

$$t_{F_B} = \frac{1}{4}\, t \tag{2.64}$$

$$d = \left(6\, r^2\, \frac{\rho_F}{\rho} \sqrt{\frac{\pi\, C_F\, \varepsilon\, E}{K_{F1}\, E_F}} \right)^{\frac{1}{2}} \tag{2.65}$$

where, obviously, the density of the frame material is denoted by ρ_F and the density of the shell material denoted by ρ.

Frameless sandwich shell: `user_input.material_property.fus.kcon`, 6 and 7.
Assuming that the elliptical shell buckles at the load determined by the maximum compressive stress resultant N_x^-, the buckling equation for the frameless sandwich concepts is

$$\frac{N_x^-}{r\, E} = \varepsilon \left(\frac{t_{S_B}}{r} \right)^m \tag{2.66}$$

or, solving for the shell thickness t_{S_B}

$$t_{S_B} = r \left(\frac{N_x^-}{r\, E\, \varepsilon} \right)^{\frac{1}{m}} \tag{2.67}$$

Several values of the coefficients involved in Equations 2.59 to 2.67 are given in the Appendix C.

For each fuselage station, the shell thickness t_S is selected according to compression t_{S_C}, tension t_{S_T}, minimum gage t_{S_G} and buckling criteria t_{S_B}. All failure criterias and geometrical constraints have to be satisfied

$$t_S = \max\left(t_{S_C}\,,\, t_{S_T}\,,\, t_{S_G}\,,\, t_{S_B}\right) \tag{2.68}$$

If the structure is buckling critical, $t_S = t_{S_B}$, the equivalent isotropic thickness of the frames, t_F, is computed from Equation 2.64; if the structure is indeed not buckling critical, $t_S > t_{S_B}$, the frames are re-sized to make $t_S = t_{S_B}$. In particular a new frame spacing is computed from Equation 2.61. The sizing procedure is outlined in Figure 2.19.

Ideal fuselage structural weight

Once the equivalent thickness of the frames and shells are computed for each fuselage station, the ideal fuselage structural weight can be easily determined

$$W_I = 2\,\pi \sum_{i=1}^{N} \left(\rho\, t_{S_i} + \rho_F\, t_{F_i}\right) r_i\, \Delta\, x_i \tag{2.69}$$

where the quantities subscripted i are defined station-by-station along the fuselage length.

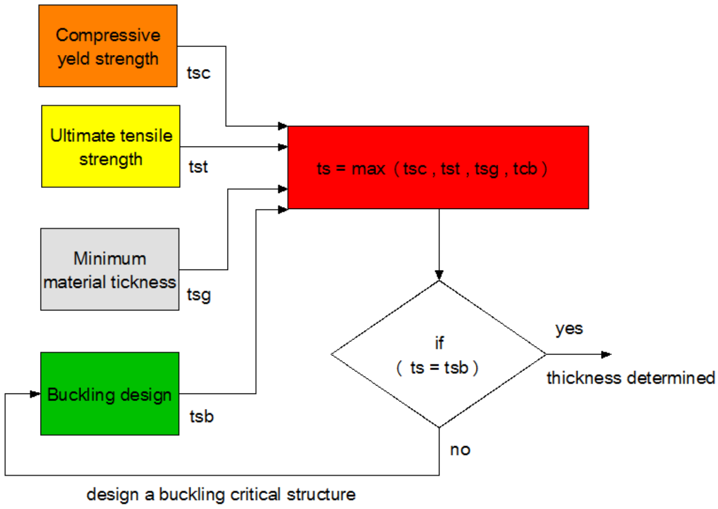


Figure 2.19: Station sizing criteria to determine minimum allowable thickness

2.6.2 Conceptual fuselage cross-section layout

For the purpose of weight estimation, it is convenient to size the structural elements in terms of equivalent isotropic thicknesses, as discussed in the previous section. These can be thought of as the thicknesses that would result if the structural material is *smeared* to a constant thickness on the shell surface.
More often in the preliminary design it is required a more detailed description of the cross-section for use in a Multidisciplinary Design Optimization (MDO). This section discusses a methodology to define detailed informations for the fuselage cross-section layout depending on the structural concept used in the analysis. The methodology is extensively treated in Reference [**?**].
In the following section the minimum-weight analyses are presented for wide columns. Wide columns are axial-compression members which are free along their unloaded edges; therefore, the general instability may be predicted by Euler column theory. It is assumed that wide columns are stiffened in the direction of loading and include the effects of coupling between adjacent elements in the local mode of instability. Local buckling is calculated from a single equation, where the buckling coefficient is dependent upon the relative proportions of the elements of the structure.
The analyses are defined on the assumption that the structure has a sufficiently large number of stiffeners to permit the geometric properties to be based on a unit repetitious width between stiffeners, namely b_s, even though the end bays may be of width b_s.

Unflanged, integrally stiffened wide columns

Considering the cross-sectional geometry of an unflanged, integrally stiffened wide column as depicted in Figure 2.20, 4 parameters might be computed to determine the structural design layout:

1. t_w, thickness of stiffener web element;
2. t_s, thickness of sheet or skin element between each stiffener;

3. b_w, height of stiffener web element;

4. b_s, width of sheet element between each stiffener.

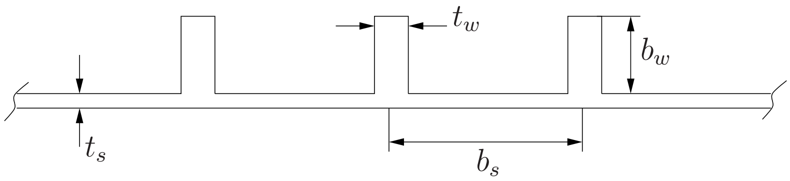


Figure 2.20: Typical unflanged, integrally stiffened shell geometry.

Combining the buckling stress for general instability determined by Euler theory with the methods of minimum-weight analysis, the following efficiency equation is determined

$$\frac{N_x^-}{l\,\overline{\eta}\,E} = \varepsilon \left(\frac{t_S}{l}\right)^2 \tag{2.70}$$

denoting by

- N_x^-, the maximum compressive stress resultant per unit width;
- l, the length of the width column;
- $\overline{\eta}$, the effective plasticity reduction factor defined as $\overline{\eta} = \eta_T^{3/4}$, where η_T is the ratio of tangent modulus to Young's modulus, E;
- t_S, the equivalent isotropic thickness of the shell computed by Equation 2.68;
- ε, the efficiency factor.

The efficiency factor, ε, is a non-linear function of $b_w\,/\,b_s$ and $t_w\,/\,t_s$. The relationship shows that the maximum value of efficiency factor, $\varepsilon_{max} = 0.656$, is obtained selecting the ratios

$$\left(\frac{t_w}{t_s}\right) = 2.25 \tag{2.71}$$

$$\left(\frac{b_w}{b_s}\right) = 0.65 \tag{2.72}$$

and corresponds to the fuselage structural concept number 1 presented in Table 2.10 and later in Appendix C.
The following additional equations necessary to determine the dimensions of the wide-column design are:

$$\frac{b_s}{l} = 1.1\left[1 + \frac{t_w}{t_s}\,\frac{b_w}{b_s}\right]\sqrt{\frac{\dfrac{N_x^-}{l\,\eta_T\,E}}{\dfrac{t_S}{l}\left[\dfrac{t_w}{t_s}\left(\dfrac{b_w}{b_s}\right)^3\right]\left[4 + \dfrac{t_w}{t_s}\,\dfrac{b_w}{b_s}\right]}} \tag{2.73}$$

$$t_s = \frac{t_S}{1 + \dfrac{t_w}{t_s}\,\dfrac{b_w}{b_s}} \tag{2.74}$$

Equations 2.71 through 2.74 represent a system of 4 equations in 4 unknowns and the detailed fuselage cross-sectional layout might be determined.

Z-stiffened wide columns

Considering the cross-sectional geometry of a Z-stiffened wide column as depicted in Figure 2.21, 6 parameters might be computed to determine the structural design layout:

1. t_w, thickness of stiffener web element;
2. t_s, thickness of sheet or skin element between each stiffener;
3. t_f, thickness of stiffener flange element;
4. b_w, height of stiffener web element;
5. b_s, width of sheet element between each stiffener.
6. b_f, width of stiffener flange element.

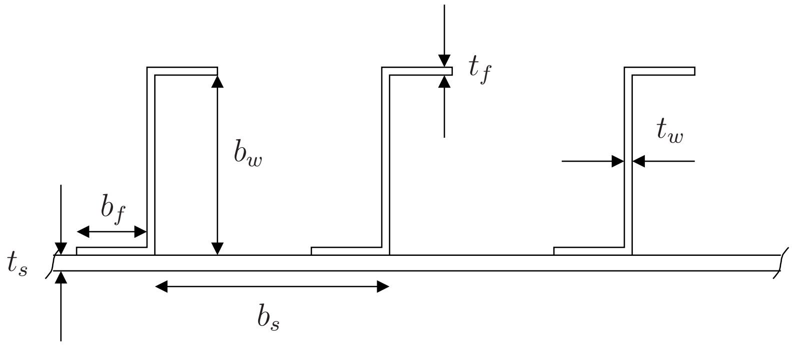


Figure 2.21: Typical Z-stiffened shell geometry.

The minimum-weight equation as function of the efficiency factor, ε, is determined by

$$\frac{N_x^-}{l\,\overline{\eta}\,E} = \varepsilon\left(\frac{t_S}{l}\right)^2 \tag{2.75}$$

with the same notation as presented above in the current section. The efficiency factor, ε, is a non-linear function of $b_w\,/\,b_s$ and $t_w\,/\,t_s$. The optimum values of $b_w\,/\,b_s$ and $t_w\,/\,t_s$ may vary considerably with little reduction in maximum efficiency.
Table 2.11 presents the values of ε_{max}, $b_w\,/\,b_s$ and $t_w\,/\,t_s$ considering that the typical design might be influenced by bending and pressure loads and by minimum gage constraint. Structural concepts (2 through 4) are summarized in Table 2.10 and later presented in Appendix C. The following equations may be used to complete a Z-stiffened wide-column

$b_w\,/\,b_s$	$t_w\,/\,t_s$	ε_{max}	Structural concept
0.87	1.06	0.911	2
0.58	0.90	0.76	3
0.60	0.60	0.76	4

Table 2.11: Efficiency factor and structural concepts.

design, as 6 geometry parameters have to be computed:

$$b_f = 0.3\, b_w \tag{2.76}$$

$$t_f = t_w \tag{2.77}$$

$$t_s = \frac{t_S}{1 + 1.6\,\dfrac{t_w}{t_s}\,\dfrac{b_w}{b_s}} \tag{2.78}$$

$$\frac{b_w}{l} = \frac{0.4\left(1 + 1.6\,\dfrac{t_w}{t_s}\,\dfrac{b_w}{b_s}\right)}{\sqrt{\dfrac{b_w}{b_s}\,\dfrac{t_w}{t_s}\left(1 + 0.59\,\dfrac{t_w}{t_s}\,\dfrac{b_w}{b_s}\right)}}\sqrt{\frac{\dfrac{N_x^-}{l\,\eta_T\,E}}{t_S\,/\,l}} \tag{2.79}$$

As the user shall specify the structural concept the fuselage is supposed to be designed, the 2 correspondent relations ($b_w\,/\,b_s$ and $t_w\,/\,t_s$) from Table 2.11 along with Equations 2.76 through 2.79 might be used to determine detailed fuselage cross-sectional layout.

Truss-core sandwich wide columns

Considering the cross-sectional geometry of a truss-core sandwich wide column as depicted in Figure 2.22, 4 parameters might be computed to determine the structural design layout:

1. t_f, thickness of facing sheet in sandwich panels;
2. t_c, thickness of core material in sandwich panels;
3. b_f, width of sandwich facing sheet element;
4. h, thickness of sandwich.

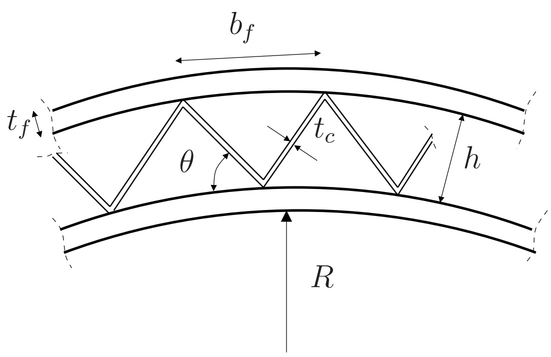


Figure 2.22: Truss-core sandwich geometry

The minimum-weight equation as function of the efficiency factor, ε, is determined by

$$\frac{N_x^-}{l\,\eta\,E} = \varepsilon\left(\frac{t_S}{l}\right)^2 \tag{2.80}$$

The efficiency factor, ε, is a non-linear function of $t_c\,/\,t_f$ and θ, the angle between facing and core elements. Table 2.12 presents the values of ε_{max}, θ and $t_c\,/\,t_f$ for the structural

θ	$t_c\,/\,t_f$	ε_{max}	Structural concept
62°	0.92	0.605	5
55°	0.65	0.4423	6
45°	1.0	0.3615	7

Table 2.12: Efficiency factor and structural concepts.

concepts (5 through 7) summarized in Table 2.10 and later presented in Appendix C. The following equations may be used to complete a truss-core sandwich column design, as 4 geometry parameters have to be computed:

$$t_f = \frac{t_S}{2 + \frac{t_c}{t_f}\frac{1}{cos\,(\theta)}} \tag{2.81}$$

$$b_f = 0.95\,t_f\sqrt{\frac{K_X\,\frac{t_S}{l}}{\frac{N_x^-}{l\,\eta_L\,E}}} \tag{2.82}$$

$$h = \tan\,(\theta)\,\frac{b_f}{2} \tag{2.83}$$

$$\overline{\eta} = \sqrt{\frac{2\,\eta_T^{3/2}}{1 + \eta_T}} \tag{2.84}$$

As the user shall specify the structural concept the fuselage is supposed to be designed, the correspondent relation $(t_c\,/\,t_f)$ from Table 2.12 along with Equations 2.81 through 2.84 might be used to determine detailed fuselage cross-sectional layout.

2.6.3 Lifting surfaces structural analysis

Structural sizing for both wing and tail is performed in very similar way. As it happens for fuselage sizing, the user can choose among six different structural concepts presented in Table 2.13. Three concepts feature unstiffened covers while the remaining feature truss-stiffened cover. Both cover configurations use webs which can be Z-stiffened, unflanged or trusses. Although wing and tail usually share the same structural concepts, it is possible to define, respectively, for wing, horizontal and vertical tail the entries

`user_input.material_property.user_input.wing.kcon`
`user_input.material_property.user_input.htail.kcon`
`user_input.material_property.user_input.vtail.kcon`

to independently define a structural concept for each lifting surface. Lifting surfaces structural parameters are stored in a ASCII file which is automatically loaded by GUESS (Table C.4).

By means of the geometry and loads distribution calculated from *Geometry module* and *Loads module*, the structural dimensions and weight of the structural box can now be calculated for either wing or tail empennages. It is assumed that the wing and tails are

Entry kcon	Covers	Webs
1	Unstiffened	Truss
2	Unstiffened	Unflanged
3	Unstiffened	Z-stiffened
4	Truss	Truss
5	Truss	Unflanged
6	Truss	Z-stiffened

Table 2.13: Available lifting surface structural geometry concepts

multi-web box beam with the webs running in the direction of structural semispan. The critical instability mode for multi-web box beams is characterized by simultaneous buckling of the covers and of the webs due respectively to local instability and flexure induced crushing.
Within wing structural box, the spanwise distribution of shear force and bending moment along structural span is used to size covers and webs thickness. The carrythrough structure, carrying the spanwise bending, shear and torsion loads is designed to resist the resulting forces and moments at wing-fuselage intersection. Finally, ideal weight of the wing box structure and ideal weight of the carrythrough structure are computed as sum of the weigths for covers and webs.
In a similar fashion, GUESS sizes the horizontal tail along the structural semispan and in correspondence of the carrythrough structure, which can be now connected to the fuselage or to the vertical tail if a T-tail is considered.

Equivalent isotropic thickness for covers and webs

With the above assumptions, the solidity, ratio of volume of structural material to total wing box volume, of the least weight multi-web box beams is given as

$$\Sigma = \varepsilon \left(\frac{M}{Z_S \, t^2 \, E} \right)^e \tag{2.85}$$

and

$$\Sigma = \frac{W^{'}_{bend}\,(y)}{\rho \, Z_S \, t} \tag{2.86}$$

where the structural coefficients, ε and e, are given in Table C.4 depending on the web and cover geometries, M is the applied bending moment, t is the thickness, E is Young's modulus, and Z_S is dimensionally a length. Many efforts have been spent defining correctly the last parameter.
Combining Equations 2.85 and 2.86, the weight of bending material per unit span, $W^{'}_{bend}$, dimensionally $[kg/m]$, may be computed on a station-by-station basis.
The weight of shear material per unit span, $W^{'}_{shear}$, is

$$W^{'}_{shear}\,(y) = \frac{\rho \, F_S}{\sigma_S} \tag{2.87}$$

where F_S is the applied shear load and σ_S is the allowable shear stress. The material density of the frames and webs is denoted by ρ. The optimim web spacing is computed from the relation

$$d_W = t \left[\frac{1 - 2\,e_C}{(1 - e_C)\,\sqrt{2\,\varepsilon_W}} \left(\frac{M}{Z_S\,t^2\,E} \right)^{\frac{2\,e_C - 3}{2\,e_C}} \varepsilon_C^{\frac{3}{2\,e_C}} \right]^{\frac{2\,e_C}{4\,e_C - 3}} \tag{2.88}$$

where the subscripts W and C refer to the webs and covers, respectively. As last, the equivalent isotropic thicknesses of the covers and webs may be determined form

$$t_C = d_W \left(\frac{M}{Z_S\,t\,E\,\varepsilon_C\,d_W} \right)^{\frac{1}{e_C}} \tag{2.89}$$

$$t_W = t \sqrt{ \left(\frac{M}{Z_S\,t^2\,E} \right)^{2-\frac{1}{e_C}} \left(\frac{\varepsilon_C\,d_W}{t} \right)^{\frac{1}{e_C}} \left(\frac{2}{\varepsilon_W} \right) } \tag{2.90}$$

In a similar fashion to the fuselage analysis, a minimum gage thickness for webs and covers is prescribed, namely t_{g_W} and t_{g_C}, respectively.

Ideal weight of the wing box structure

The ideal weight of the wing box structure, dimensionally $[kg]$, is therefore

$$W_{BOX} = \frac{2\,b_S}{N} \sum_{i=1}^{N} \left(W'_{bend_i} + W'_{shear_i} \right) \tag{2.91}$$

indicating the structural semispan running over the wing-semispan by b_S and the number of elements the span is divided into by N.

Ideal weight of the carrythrough structure

The carrythrough structure must resist torsion in addition to bending and shear loads. The torsion material is required to carry on the twist induced due to the sweep of the wing. The principle of superposition is applied when more than one sector is defined for each single lifting surface, since each sector has generally its own geometric charateristics. Defining with a subscript 0 the correspondent parameters at the root (i.e. wing/fuselage intersection, fuselage/horizontal tail intersection for configurations similar to $B747-100$, or vertical/horizontal tails for configurations with a T-tail), the solidity is

$$\Sigma_C = \varepsilon \left(\frac{M_0\,\cos\left(\Lambda_S\right)}{t_0^2\,C_{SR}\,E} \right)^e \tag{2.92}$$

where the only longitudinal component of the bending moment contribuites to the bending material estimation. The weight of the bending material is therefore

$$W_{bend_C} = \rho\,\Sigma_C\,C_{SR}\,t_0\,W_C \tag{2.93}$$

indicating by W_C the width of the carrythrough structure. The weight of the shear material is

$$W_{shear_C} = \rho \frac{F_{S_0}}{\sigma_S} W_C \tag{2.94}$$

The torque on the carythrough structure is calculated as the chordwise proiection of the bending moment at the root and the weight of the torsion material is

$$W_{torsion_C} = \frac{\rho T \, (t_0 + C_{SR}) \, W_C}{t_0 \, C_{SR} \, \sigma_S} \tag{2.95}$$

At last, the weight of the carrythrough structure is computed from summation of the bending, shear and torsion material

$$W_C = W_{bend_C} + W_{shear_C} + W_{torsion_C} \tag{2.96}$$

Remark. The ideal structural weight for fuselage and lifting surfaces have been computed using equations 2.69, 2.91 and 2.96. Both the fuselage length and lifting surfaces structural span have been divided into a generic number of segments, namely N. The user is recommended to refer to the Appendix A and notice the relation between the above discretization and the one which is available in the entry

`experienced_user_input.geometry.guess`

of the *technology* `.xml` input file. Once again, the number of elements herein considered is not related to the beam elements in the stick model and precisely the former shall be greater than the latter.

2.7 Regression module

Statistical analysis techniques are used to determine a relation among the weight predicted by GUESS for the load-bearing structure computed, named W_{guess}, and the actual weight of load-bearing structure, W^S_{actual}, the weight of primary structure, W^P_{actual}, and the total weight W^T_{actual}.

Two different applications have been developed:

- a linear regressione equation;
- *power-intercept* regression equation.

The first basic application is the linear regression wherein the weights estimated by GUESS procedure are related to actual weigths of the aircraft by means of a simple linear relation:

$$W_{actual} = m \, W_{guess} \tag{2.97}$$

In order to estimate the slope coefficient m for the linear relation, the resulting residual ε is minimized with respect to m:

$$\varepsilon = \sum_{i=1}^{n} (W_{actual_i} - W_{guess_i})^2 \tag{2.98}$$

where n is the number of aircraft whose data are to be used in the fit.
The second form of regression equation used is non linear:

$$W_{actual} = m\, W_{guess}^{a} \tag{2.99}$$

where m and a are two unknowns to be determined by minimization of the residual (defined in Equation (2.98)). In order to formulate the resulting *power-intercept* regression equation, an iterative approach is utilized since the formulation is basically not linear [15]. A *correlation coefficient* R is introduced as a measure of the accuracy the actual aircraft weight is predicted. R represents the reduction in residual error due to the regression technique. It is defined as

$$R = \sqrt{\frac{\varepsilon_t - \varepsilon_r}{\varepsilon_r}} \tag{2.100}$$

where ε_t and ε_r refer to the residual errors associated with the regression before and after analysis is performed, respectively. R is limited between 0 and 1: a perfect fitting of the data leads to $R = 1$ while no improvement introduced by the regression leads to $R = 0$

2.7.1 Fuselage regression analysis

The user can choose the desired statistical technique for fuselage correlation setting in the *technology* `.xml` input file the following entry:
`user_input.analysis_setup.regression.analf`
If it is set to `'linear'`, char type, a linear regression equation is utilized; otherwise, the *Regression module* assumes a *power-intercept* regression equation.
With a slightly different notation than that used in Section 2.6.1, the ideal fuselage structural weight computed by means of GUESS is herein denoted as W_{guess}, instead of W_I. Fuselage structural, primary and total weights are computed by applying the relations shown in Table 2.14 to the ideal fuselage weight obtained from the developed method. In the table, R represents the correlation coefficient given by Eq. (2.100) for each case.

	Linear-intercept eq.	**Power-intercept eq.**
Structural weight	$W^S_{actual} = 1.305\, W_{guess}$ $R = 0.9946$	$W^S_{actual} = 1.1304\, (W_{guess})^{1.0179}$ $R = 0.9946$
Primary weight	$W^P_{actual} = 1.8872\, W_{guess}$ $R = 0.9917$	$W^P_{actual} = 1.6399\, (W_{guess})^{1.0141}$ $R = 0.9917$
Total weight	$W^T_{actual} = 2.5686\, W_{guess}$ $R = 0.9944$	$W^T_{actual} = 3.9089\, (W_{guess})^{0.9578}$ $R = 0.9949$

Table 2.14: Regression analysis applied to the predicted fuselage weight

Fuselage structural weight consists of all load-carrying members including bulkheads, major and minor frames, coverings, covering stiffeners and longerons.
Fuselage primary weight consists of all load-carrying members as well as any secondary structural item such as joints fasteners, keel beam, fail-safe straps, flooring, flooring structural supplies, and pressure web. It also includes the lavatory structure, gallery support, partitions, shear ties, tie rods, structural firewall, torque boxes and attachment fittings.

Finally, *fuselage total weight* accounts for all members of the body, including the structural and primary weights. It does not include passenger accommodations, such as seats, lavatories, kitchens, stowage and lightning, the electrical system, flight and navigation system, alighting gear, fuel and propulsion system, hydraulic and pneumatic system, communication system, cargo accommodations, flight deck accommodations, air conditioning equipment, the auxiliary power system and emergency system.

2.7.2 Lifting surfaces regression analysis

Similarly to the fuselage regression analysis, GUESS offers the option to set the desired statistical technique in the *technology* `.xml` input file for each single lifting surface component, editing the entries

`user_input.analysis_setup.regression.analw`
`user_input.analysis_setup.regression.analv`
`user_input.analysis_setup.regression.analh`

for wing, vertical and horizontal tail, respectively.
With a slightly different notation than that used in Section 2.6.3, the ideal weight of the wing box structure, W_{BOX}, and the ideal weight of the carrythrough structure, W_C, are generically denoted by W_{guess} in the following of the present section.
Lifting surfaces structural, primary and total weights are computed by applying the relations shown in Table 2.15 to the ideal weights obtained from the developed method. Again, in the table, R represents the correlation coefficient (Eq. (2.100)) for each case.

	Linear-intercept eq.	**Power-intercept eq.**
Structural weight	$W^S_{actual} = 0.9843\, W_{guess}$ $R = 0.9898$	$W^S_{actual} = 1.3342\, (W_{guess})^{0.9701}$ $R = 0.9946$
Primary weight	$W^P_{actual} = 1.3442\, W_{guess}$ $R = 0.9958$	$W^P_{actual} = 1.6399\, (W_{guess})^{0.9534}$ $R = 0.9969$
Total weight	$W^T_{actual} = 1.7372\, W_{guess}$ $R = 0.9925$	$W^T_{actual} = 3.9089\, (W_{guess})^{0.9268}$ $R = 0.9946$

Table 2.15: Regression analysis applied to the predicted wing weight

Wing structural weight consists of spar caps, interspar coverings, spanwise stiffeners, spar webs, spar stiffeners, and interspar ribs.
Wing primary weight includes all wing box items in addition to auxiliary spar caps and spar webs, joints and fasteners, landing gear support beam, leading and trailing edges, tips, structural firewall, bulkheads, jacket fittings and attachments.
Wing total weight includes wing box and primary weight items in addition to high-lift devices, control surfaces and access items. It does not include the propulsion system, fuel system, and thrust reverses; the electrical system; alighting gears; hydraulic and pneumatic system; anti-icing devices; emergency system.

2.8 GUESS validation for six aircrafts

Several test cases have been considered, through the development of the computer program, to evaluate the analytical solutions offered by means of GUESS . In the current section, the structural sizing tool has been tested against six different aircrafts, very different in weigths, geometrical dimensions and usage purpose. The correspondent input files, *technology* and *geometry* `.xml` files, have been edited for each case-study. A comparison between the analytical estimation for the load-carrying weight and the actual load-carrying weight for the fuselage and wing, considering six different aircrafts, is herein proposed.
The present comparisons have been obtained using gross geometric informations for fuselage and wing, representing the wing as a trapezoidal flat panel and the fuselage with the two power-law bodies, since more accurate informations were not available at that time. The analysis is limited to the fuselage and wings, lacking data for the tail planes. The following aircrafts have been included in the case-study:

- Boeing 720
- Boeing 737
- Boeing 747-100
- Douglas DC-8
- McDonnell Douglas MD-11
- Lockheed L-1011

2.8.1 Analitycal GUESS solution for fuselage

The comparison against the six analyzed aircrafts is given in Table 2.16, where the fuselage analytical weight estimation computed by means of GUESS is refered to the actual load-carrying structure weight. With same notation used in Section 2.7.1, the former is denoted by W_{guess} and the latter by W^S_{actual}.
Weights are expressed in US customary system since the available original data are provided in these units.

Aircraft	**Load-carrying structure** [lb]	**GUESS** [lb]
B-720	9013	9476
B-737	5089	5190
B-747	39936	43428
DC-8	13312	14337
MD-11	25970	30171
L-1011	28352	31837

Table 2.16: Fuselage weight breakdown for eight transport aircraft

2.8.2 Analytical GUESS solution for wing

The comparison against the six analyzed aircrafts is given in Table 2.17, where the wing analytical weight estimation computed by means of GUESS is refered to the actual load-carrying structure weight. With same notation used in Section 2.7.1, the former is denoted by W_{guess} and the latter by W^S_{actual}. Weights include left and right wing box and the carrythrough structure.
Weights are expressed in US customary system since the available original data are provided in these units.

Aircraft	Load-carrying structure [lb]	GUESS [lb]
B-720	11747	10602
B-737	5414	4439
B-747	50395	62449
DC-8	19130	19786
MD-11	35157	35393
L-1011	28355	27155

Table 2.17: Wing weight breakdown for eight transport aircraft

2.8.3 First conclusions

The available geometric description provided within the *geometry* `.xml` input file for the six case-studies is rather semplicistic, not considering the actual shape for fuselage and wings, and consequently the loads determination could be influenced by the inaccuracy introduced in the geometric modelling. As shown schematically in Tables 2.16 and 2.17, GUESS analytical results are close, sometimes matching, the actual values for the aircrafts.
The vehicle, that most is far from the analytical solution, is Boeing 747−100. The reader, scrolling the present paper, will have opportunity to notice further analysis and comparisons for the same aircraft, that is Boeing 747−100. At this point, it can be predated that the analytical results may be precise, matching within some point fractions the actual values, thanks to a more detailed and faithfull geometric modelling.

2.9 Generation of a stick model from the analytical solution

As explained in the previous chapters, GUESS provides an analitycal estimation of fuselage and lifting surfaces weight by means of methods having a somewhat physical basis. This is of course a first goal toward the enchancement of details for the structural system in the conceptual design phase of the aircraft under development. For sure now the designer is more independent on solely statistical methods, with the inaccuracies and loss of generality they may provide, and can rely on an improved structural weight for at least

stability and flight performances analyses.
Now, the further step consists in analyzing the airframe to predict its static, dynamic, aeroelastic behaviour in a very simplified way (since many details are still missing and are not even important in this early design phase) starting from a suitable stiffness distribution. The SMARTCAD module will be used for these purposes. The common practice in aerospace industry is to start with simplified structural model such as a stick beam model with lumped masses. This kind of model guarantees:

- easiness in modelling the structural system since not so many details are required to be set up the numerical model;
- low computational costs, especially if it is considered a large amount of simulations are required to assess the aircraft with different inertial configurations and flight conditions.

Of course, the solution determined by GUESS may not be free of unwanted and inadmissible phenomena, thus the necessity to numerically assess its safety and performances for the operating conditions the machine will work. Optimization procedures may then be adopted to improve structural material distribution under some constraints, such as minimum weight, stiffness, absence of flutter instabilities, defined by the designer. Future developments will be enhanced the code with this feature.

2.10 Stick module layout

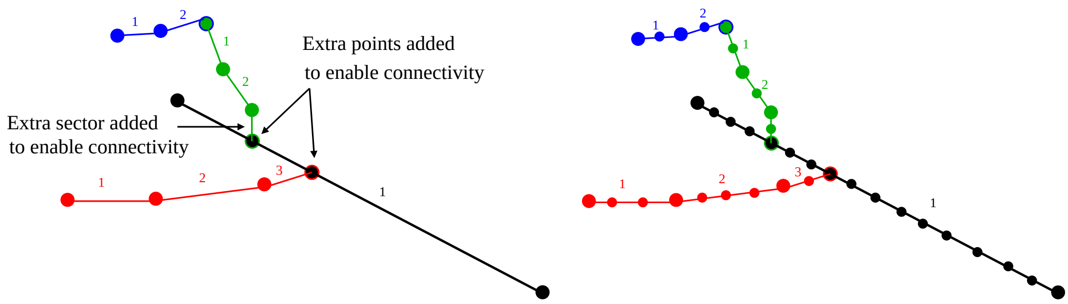
GUESS generates for the aiframe under development a stick model by converting the informations available in its internal database. This tool can be considered as a sort of pre-processor for the numerical aeroelastic tool SMARTCAD .
The preprocessor has the following tasks:

- obtain a mass distribution over fuselage length and lifting surfaces structural span, presented from Sections 2.4 to 2.7;
- generate a stick model for the aircraft using beam elements (see Section 2.10.2);
- extract stiffeness distribution useful to define beam mechanical propertis (see Section 2.11);
- write an ASCII file containing the model in a suitable format for SMARTCAD (see Section 5.1).

In the following sections, the procedure to generate the stick model from the *geometry* and *technology* input files will be outlined and the most important features will be pointed out.

2.10.1 Elements connectivity

The first task for the pre-processor is the one to create a close cruxiform scheme for the aircraft, defined as *geometric stick model*. The primary purposes of this first model which is shown in Figure 2.23(a) is to give a brief idea of the aircraft overall shape, dimensions



(a) Geometric stick model

(b) Analytical mesh

Figure 2.23: Analytical structural model generation from the geometric stick model

and connectivity among the different components and to create a closed circuit for the structural model able to transfer loads from each neighbouring component. The aircraft is indeed considered to be made of single independent *components* which need to be assembled together by sharing some points in common. Each component is composed of different pieces named *sectors* which allow to define different properties along the axis of the component (aerodynamic, inertial structural). The number of allowed sector is defined a-priori in the input file; default values for the admissible number of sectors for each component are reported in Table 2.18. Sussessively, the created geometric-stick is converted into the classic structural stick model; each single line becomes one or more beams with associated structural and inertial properties as shown in Figure 2.23(b). The pre-processor allows to user to have full control for generation of the stick model by:

1. choosing and selecting the total number of components to be used among fuselage, wing, vertical and horizontal tail up to assemblying a complet aircraft; it is possible to consider whatever combination, from a simple single wing up to a complete airframe model;

2. specifying the number of sectors used for each component;

3. specifying the number of beam elements used within each sector;

4. applying symmetry with respect to vertical symmetry plane; in this case, special care needs to be taken into account correctly defining the structural constraints (see Section 5.3.9) and aerodynamic boundary conditions (see Section 5.4);

Component	**Maximum number of sectors**	**Notes**
Wing	3	over the semispan
Vertical tail	2	
Horizontal tail	2	over the semispan

Table 2.18: Maximum number of sectors for each component.

To successully accomplish the generation of the geometric stick model, the following guidelines are followed:

1. fuselage is the first component to be assemblied within the airframe model; at the beginning of the procedure, fuselage is defined by the minimum number of points required to correctly describe its own geometric shape; more then two points, one located at the nose and one at the tail, are generally used to account for the actual fuselage shape, as clearly shown in Figure 2.3;

2. the wing is the second component to be assembled; the procedure calculates the position of the points defining each sector from wing root to wing tip (referring to Figure 2.23(a), four points have been calculated). At this stage, the procedure can estabilish a relation of connectivity between fuselage and wing. One point, coincident with the first point on the wing root, is added to the fuselage in order to enable connectivity between the wing and the fuselage;

3. the vertical tail is the third component to be assembled; the procedure calculates the position of points defining each sector from root to tip (in Figure 2.23(a), three points bordering sector 1 and 2 are depicted); the relation of connectivity between fuselage and vertical tail is performed in the following way:

   - if the first point for vertical tail does not lie on reference fuselage line, one more vertical sector is added to vertical tail; if the new point on reference fuselage line does not coincide with any existing point, it is added to the points defining the fuselage;
   - if the first point for vertical tail lies on reference fuselage line, no points and sector are added to vertical tail but one more point can be added to the fuselage if it is not coincident with any existing point;

4. the horizontal tail is the last component assembled; depending on the vertical position of the horizontal plane, relation of connectivity can be estabilished with either fuselage or vertical tail; Figure 2.23(a) shows one exaple of T-tail empennages.

Within the implemented routines, it is assured that no coincident points have been defined twice for the same component. In a nutshell, the procedure herein depicted adds additional points for each component in order to define correct relations of connectivity, which play an important role when the *Output* file is written in a suitable format for SMARTCAD .

2.10.2 Beam mesh generation

The geometrical stick model generated can be considered as the main reference frame along which the structure is laid and built. All structural and inertial properties are referred to its axes. This model can be easily converted into a beam model by a simplified mesh-generation procedure which preserves mesh smoothness and uniformity as shown in Figure 2.23(b). Thus more lines (beams) and points (nodes) are added.
Within the *technology* input file, appropriate entries are dedicated to define the number of beam elements to use in the mesh-generation routine. For a complete airframe model the corrispondent entries are shown in Table 2.19.
The mesh-generation process assures consistency between the sectors defined in the *geomtery* input file and the entries herein presented to link univocally for each generic component, the number of beam elements with the corrispondent sector. If a generic

Physical component	**Entry – user_input.geometry.***
Fuselage	`beam_model.nfuse`

Wing	`beam_model.nwing_inboard`
	`beam_model.nwing_midboard`
	`beam_model.nwing_outboard`
	`beam_model.nwing_carryth`

Vertical tail	`beam_model.nvtail_inboard` `beam_model.nvtail_outboard`

Horizontal tail	`beam_model.nhtail_inboard`
	`beam_model.nhtail_midboard`
	`beam_model.nhtail_carryth`

Table 2.19: Entries to set the number of beam elements within each sector

component (fuselage, wing or tail for example) is not assembled, there is obviously no need to define the corrispondent entry to set the number of beam elements. The procedure does not consider at all any component not directly defined in the *geometry* input file.
One short example is useful to understand this feature. Table 2.20 collects the entries only for wing component. Two cases are considered, the first case where the mid-board sector is not defined and the second case where the out-board sector is not defined. Supposing to have the same geometrical data in the input file and choosing the number of beam elements as reported, the mesh-generation procedure will not consider the second entry in the first case and the third entry in the second case. The result will be consequently identical.

Sector	**Defined**	**Entry – user_input.geometry.***	**Nr of beams**
Wing-inboard	✓	`beam_model.nwing_inboard`	2
Wing-midboard		`beam_model.nwing_midboard`	4
Wing-outboard	✓	`beam_model.nwing_outboard`	3
Wing-inboard	✓	`beam_model.nwing_inboard`	2
Wing-midboard	✓	`beam_model.nwing_midboard`	3
Wing-outboard		`beam_model.nwing_outboard`	4

Table 2.20: Sectors within wing and number of beam elements

As presented in Section 2.10.1, it is necessary to add additional points to estabilish relations of connectivity among components. It is obvious that the connection points will introduce new sectors within a generic component. Refering back to Figure 2.23(a), the following considerations have to be pointed out:

1. the fuselage contains two connection points, for a total of three sectors, even only one entry has been defined;

2. the vertical tail has a total of three sectors even two entries have been defined;

Consequently, the mesh-generation procedure implements some features to face these additions.
Regarding the first item pointed out, no more inputs have to be supplied to the procedure but automatically the routines tries to mesh in the most uniform and homogeneous way the sector. The procedure first calculates the distance among points as originally intended. Starting from the first point (point on the nose in the case of fuselage), the position of further points are computed until one connection point is reached. Depending on how much the latest generated point is close to connection point, the latest point can be left on that position or merged into the connection point. The procedure continues to locate further points untill the last point (point on the tail in the case of fuselage) is reached.
For the second point outlined, no more inputs have to be supplied because the procedure makes the assumption the number of beam elements to be used within the additional sector is set equal to the number of beam elements used within the sector which is firstly defined. Back to Figure 2.23(a) the procedure will use for the additional sector the entry

`user_input.geometry.beam_model.nwing_inboard`

if the inboard sector is defined (denoted number 2); otherwise it will use the entry

`user_input.geometry.beam_model.nwing_outboard`

2.11 Structural properties definition

Since at this stage a geometric stick model for aircraft is available and weight estimation has been previously performed, it is necessary to extract mass and stiffeness distribution over the nodes of the built beam mesh. As it usually happens, the mesh used in GUESS analytical solution, differs from newly defined beam mesh. Thus freedom is given to the user to come out with a desidered beam model to trade accuracies in some parts of the model for computational costs. To solve the issue of non-matching meshes, an interpolation procedure is required.

2.11.1 Mass and stiffeness distribution

GUESS estimates the mass distribution for both fuselage and lifting surfaces along the length and the structural span, respectively. Since from the *Geometry module* introduced in Section 2.4, all geometrical dimensions and structural concepts used within the estimation procedure are available, it is possible to compute separately for fuselage and lifting surfaces the following properties useful to define the stiffness distribution:

- local bearing area distribution;
- transverse shear stiffnesses by default are considered infinite (that is the transverse shear flexibilities are set to zero); they can be computed if the user selects the monocoque method as option;
- local second moment of inertia respect to section axis;
- local torsional constant, evaluated by Bredt analytical formula or by monocoque method, depending on the selected option.

As presented in Appendix A, the *technology* input file contains the entries

`user_input.analysis_setup.torsion_stiffness.fus`
`user_input.analysis_setup.torsion_stiffness.wing`
`user_input.analysis_setup.torsion_stiffness.vtail`
`user_input.analysis_setup.torsion_stiffness.htail`

which are used to select the preferred method for the evaluation of the torsional constant and transverse shear stiffnesses, choosing between Bredt analytical formula (if set to one) and monocoque method (otherwise).
The automatic procedure implemented in GUESS is then able to transfer these informations into the correspondent NASTRAN-format card to define the properties of a simple beam element (reading PBAR card). In the following sections, more details about the exported informations in NASTRAN format are given.

Fuselage

GUESS has estimated the thickness of shell t_S and frames t_F, according procedures and methods presented in Section 2.6, as functions of the coordinate running along the fuselage length. Assuming the fuselage cross-section to be approximated by a circular shape with radius r, the local bearing area may be computed by

$$A = 2\pi \, r \, t_B \tag{2.101}$$

and, the second moment of inertia respect to two perpendicular axis defining the cross section by the following relations

$$I1 = \pi \, t_B \, r^3 \tag{2.102}$$

$$I2 = \pi \, t_B \, r^3 \tag{2.103}$$

The torsional constant may be computed using Bredt analytical formula, as shown in Equation 2.104, since the fuselage is assumed to be a single-cell structure. For a single-cell structure, the monocoque method predicts exactly the same value of the torsional constant evaluated by the analytical formula.

$$J = \frac{4 \, (\pi \, r^2)^2}{2 \, \pi \, r / t_B} \tag{2.104}$$

where t_B is the sum of t_F and t_S. Since within GUESS it is assumed to approximate fuselage cross-section with a circular shape with a constant radius r for each station along fuselage length, second moment of inertia $I1$ and $I2$ are computed by the same expression.

Lifting surfaces

The previous sections have demonstrated the similarities for the lifting surfaces. At this point, GUESS has estimated the thickness of cover t_C, thickness of webs t_W and the number of webs n_W for each spanwise station.
Local bearing area is computed by Equation (2.105) as summation of the area for covers and webs.

$$A = 2 \, t_C \, Z_S + n_W \, n_W \, t_B \tag{2.105}$$

Second moment of inertia $I1$ respect to section axis 1 in chordwise direction is expressed by Equation (2.106); second moment of inertia $I2$ respect to section axis 2 in direction of thickness is obtained from Equation (2.107).

$$I1 = 2\, t_C\, Z_S \left(\frac{t_B}{2}\right)^2 + \frac{1}{12}\, n_W\, t_W\, t_B^3 \tag{2.106}$$

$$I2 = 2\, \frac{1}{12}\, t_C\, Z_S^3 + \sum_{i=1}^{n_W} t_W\, t_B\, d_i^2 \tag{2.107}$$

Torsional constant is estimated from Equation (2.108) if Bredt analytical formula has been selected. It is assumed that stresses due to torsion circulate within covers and webs carry only bending moment,

$$J = \frac{4\,(Z_S\, t_B)^2}{2\,(Z_S + t_B)\,/t_B} \tag{2.108}$$

where Z_S is the structural box chord and d_i is the distance between the i^{th} web and axis 2.

Lifting surfaces are intended to be multi-cells structure and hence the application of Bredt formula (valid for single-cell) might introduce unacceptable errors, leading to inaccurate evaluations of the torsional constant. In order to compute more accurately the torsional constant, the monocoque method might be selected as option for all the lifting surfaces.

2.11.2 Interpolation over beam model mesh

Structural sizing estimations, including the mechanical properties distribution herein presented, have been performed over a detailed mesh. The mesh used within GUESS procedure is intended to resolve the functions with smoothness and continuity and not to export the beam model mesh. The user, mainly the experienced user, has control over the GUESS mesh since the *technology* input file contains dedicated entry

`experienced_user_input.geometry.guess`

On the other side, the beam model mesh is fully controlled by the user editing the entry (Appendix A)

`user_input.geometry.beam_model`

Mainly for two reasons, interpolation via spline functions is carried out, since:

1. the beam model mesh does not overlap the mesh GUESS uses for the analytical sizing; the beam mesh responds to the user's needs and it might resolve in detail few critical regions, while using a caorse mesh where no critical conditions are expected to be;

2. the distribution of mechanical properties in the GUESS mesh are collocated pointwise; to define the output file for the beam model, beam properties might be defined along each beam-line (Figure 2.24);

The second point outlined is of foundamental importance when a stick beam model for the aircraft is built up. Distribution of mass and inertia originally refered to points have to be interpolated over elements in order to define beam element structural properties, as described in more detail in Chapter 5.

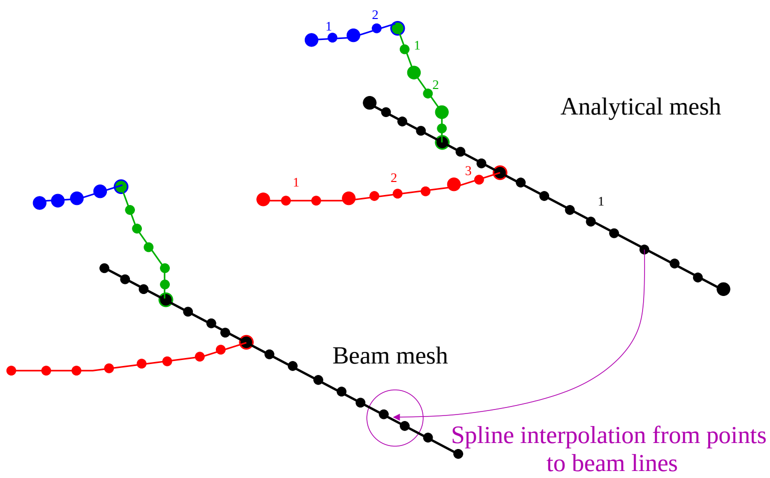


Figure 2.24: Interpolation of structural properties from GUESS to the beam model

Chapter 3

GUESS validation over the semi-wing of a Boeing 747−100

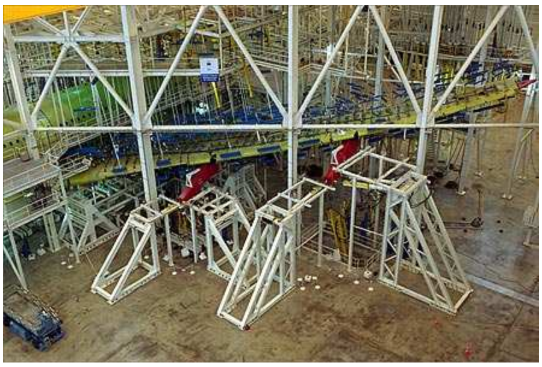
A case study is proposed to validate the semi-analytical tool and estimate mass and stiffeness distribution over the semi-wing of a Boeing 747−100 by means of GUESS solver. *Geometry* and *technology* `.xml` input files have been accordingly defined for the current analysis.
The geometric description of the wing planform is shortly reported in Appendix B. It is worthy noting that the wing planform is described by one single sector running over the semi-span, i.e. using a trapezoidal wing from the wing/fuselage intersection to the wing tip. The first kink defining the outboard limit of the inboard sector is set to one, since it is given as percentage of the semi-span.
The semi-wing is modelled as a cantilever wing and the exposed area, being of some importance for the following calculations as already shown in Section 2.5.3, is set equal to the wetted aerodynamic area, from wing/fuselage intersection to wing tip not considering the carrythorough portion of wing.
Even the analysis is exclusively restricted over the right semi-wing, setting the correspondent entry in the *technology* file
`user_input.geometry.analysis_setup.beam_model.winr=1`,
the structural sizing might be performed for both the cantilever and carrythrough portion of the wing if the carrythrough dimension is specified. The carrythrough dimension is selected to be equal to the fuselage diameter in correspondace to wing/fuselage intersection.
The present survey over the semi-wing of a Boeing 747−100, as shown in Figure 3.1, assumes that

- main landing gears are located on wings and their position is known;
- the propulsion system is located under wings; two engine nacelles are defined for each semiwing;
- the fuel is stored within the wing structural box;

To give a better interpretation of the actual lift load distribution, the approach based on Schrenk method is used, setting the entry
`user_input.geometry.analysis_setup.lift_distribution=1`.

(a) Airframe assemblying

(b) Upper fuselage section



(c) Structural test

(d) Root carrythrough chord

Figure 3.1: Pictures of Boeing 747 structure

3.1 Loads distribution

Design loads conditions for the semi-wing under investigation are computed for a quasi-static pull-up maneuver at a given normal load factor, $n = 2.5$. Correspondent shear force and bending moment occuring at this load condition are determined by Equation (2.1) and (2.2), reported for easiness

$$F_S(y) = n\,K_S\left(L_{lift}(y) - L_{fuel}(y) - L_{eng}(y) - L_{lg}(y)\right) \tag{3.1}$$

$$M(y) = n\,K_S\left(M_{lift}(y) - M_{fuel}(y) - M_{eng}(y) - M_{lg}(y)\right) \tag{3.2}$$

The total shear force, $F_S(y)$, in Equation (3.1) and bending moment, $M(y)$, in Equation (3.2) are defined on a station-by-station basis along the structural semispan. Total shear force at the given normal load factor is determined by the sum of lift forces, $L_{lift}(y)$, and inertia forces, caused by fuel, $L_{fuel}(y)$, wing-mounted propulsion pods, $L_{eng}(y)$, and landing gears, $L_{lg}(y)$, force distribution.
Similarly, total bending moment is computed as the sum of bending moment due to lift forces, $M_{lift}(y)$, and inertia forces, accounting for fuel, wing-mounted propulsion pods and landing gears terms (respectively $M_{fuel}(y)$, $M_{eng}(y)$ and $M_{lg}(y)$).
Details on the calculations performed for the semi-wing are reported in the current section and some graphical interpretations are also given.

3.1.1 Lift load distribution

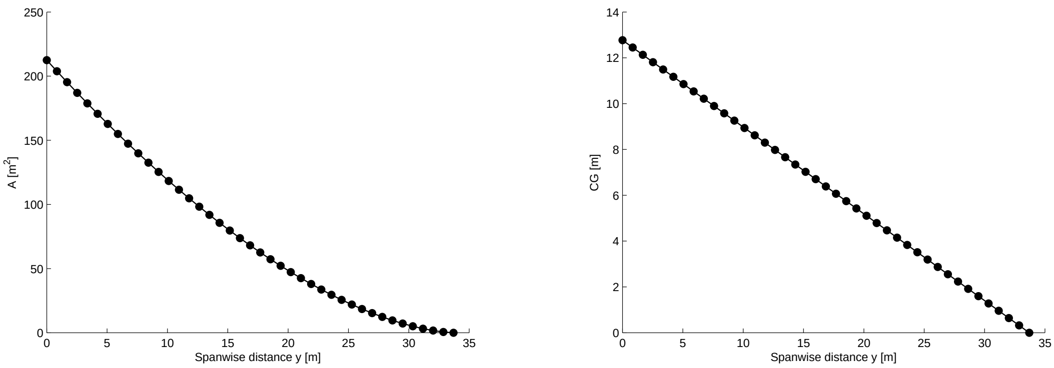
The lift load is assumed to be distribuited over the wing semispan. Lift load is zero at wing tip and maximum at wing/fuselage intersection. At the generic y station, the lift load distribution of the sector outboard of y is given as:

$$L_{lift}(y) = \left(\frac{W}{S}\right) A(y) \tag{3.3}$$

while the correspondent bending moment as:

$$M_{lift}(y) = \left(\frac{W}{S}\right) A(y)\,C_p(y) \tag{3.4}$$

where W represents the weight of the aircraft, S the exposed-aerodynamic area of the trapezoidal wing, $A(y)$ is the area outboard of y station and $C_P(y)$ is the centroid of area $A(y)$ measured respect to the actual y station.
The weight of the aircraft, W, is set equal to the Maximum Take-Off Weight (MTOW), whose averaged value is approximately around 364.000kg. Schrenk lift distribution is chosen as an option to gain a better match between the predicted lift load and the actual aircraft lift load.
The distribution of area $A(y)$ predicted by Schrenk model is shown in Figure 3.2(a); the centroid of this area, $C_P(y)$, measured respect to the actual y station, is reported in Figure 3.2(b).



(a) Outboard area value

(b) Outboard area centroid

Figure 3.2: Spanwise distribution of aerodynamic wetted area

3.1.2 Fuel load distribution

As already introduced in Section 2.5.3, forces and moments are calculated from wing tip to the wing/fuselage intersection for each generic section y. Thus, it is necessary to calculate the volume of fuel, $V_F(y)$, outboard of y section and its centroid, $C_g(y)$, measured respect to y section. These calculations are easily carried out directly in the *Geometry* module when the geometric description of the aircraft is considered. Data are stored during the geometry-processing and are then available for the load module.
At the generic y station, the shear force and bending moment caused by fuel weight carried from the actual y station to the wing tip is given, respectively, by:

$$L_{fuel}(y) = \left(\frac{W_{FT}}{V_W}\right) V_F(y) \tag{3.5}$$

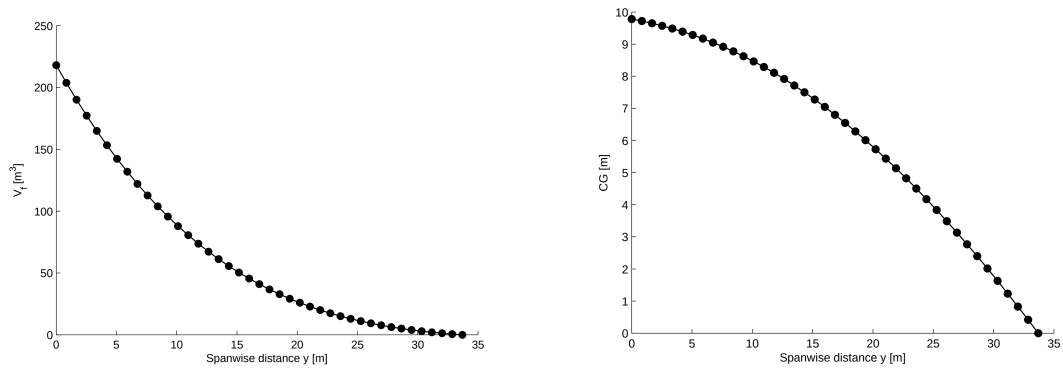
$$M_{fuel}(y) = \left(\frac{W_{FT}}{V_W}\right) V_F(y)\, C_g(y) \tag{3.6}$$

where W_{FT} is the weight of fuel carried in the wings and V_W is the total volume of wing structural box where fuel is supposed to be stored, including both semiwings.
Boeing $747-100$ is equipped with wing fuel tanks, carring a mass of fuel of about 40.000kg on both semi-wings; one additional fuel tank, namely central fuel tank, is located on the wing carrythrough structure but it does not inflence the calculations for the cantilever beam herein analyzed. Dimensions of the wing fuel tanks, as percentage of the aerodynamic chord, are given in the *geometry* input file.
Figure 3.3(a) shows the available fuel volume, which is maximum at wing/fuselage intersection and null at the wing-tip. Figure 3.3(b) shows the centroid of the outboard fuel volume, measured respect to y station.

3.1.3 Engines load distribution

Since the propulsion system is located under wings, the correspondent shear force and bending moment might be computed along structural span by:

$$L_{eng}(y) = \sum_{i=1}^{n_e} h_e(y_{e_i} - y)\, W_{e_i} \tag{3.7}$$



(a) Fuel volume (b) Fuel centroid

Figure 3.3: Calculation of fuel storage for inertial loads

$$M_{eng}(y) = \sum_{i=1}^{n_e} h_e(y_{e_i} - y) W_{e_i} \left(y_{e_i} - y\right) \tag{3.8}$$

where n_e is the number of engines mounted on the semispan, W_{e_i} and y_{e_i} are respectively the weight of the i^{th} engine and its location along the structural wing semispan. Force and moment are of course null outboard of the outer engine nacelle. Engine nacelles are considered as application point for loads as:

$$h_e(y_{e_i} - y) = \begin{cases} 1 & \text{if } y_{e_i} >= y \\ 0 & \text{if } y_{e_i} < y \end{cases} \tag{3.9}$$

Boeing 747−100 is equipped with four engine nacelles located under both semi-wings, whose total mass is around 20.000kg. For the cantilever wing analyzed in the study case, two wing-mounted nacelles are modelled. Location over the wing semi-span is specified in the *geometry* input file.

3.1.4 Landing gears load distribution

Since the main landing gear system is located under wings, the shear force and bending moment they introduce are computed by

$$L_{lg}(y) = \sum_{i=1}^{n_{lg}} h_{lg}(y_{lg_i} - y) W_{lg_i} \tag{3.10}$$

$$M_{lg}(y) = \sum_{i=1}^{n_{lg}} h_{lg}(y_{lg_i} - y) W_{lg_i} \left(y_{lg_i} - y\right) \tag{3.11}$$

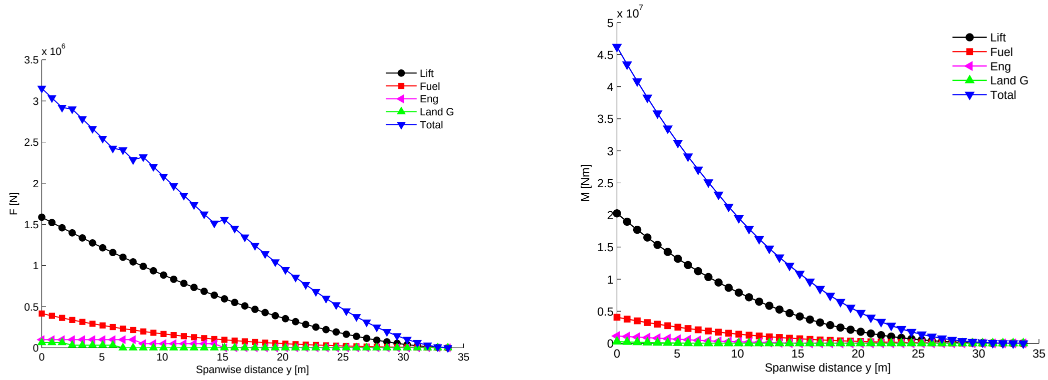
where n_{lg} is the number of landing gears mounted on the semispan, W_{lg_i} and y_{lg_i} are respectively the weight of the i^{th} landing gear and its location along the structural wing semispan. As it is done for propulsion system, landing gears are considered concentrated loads:

$$h_{lg}(y_{lg_i} - y) = \begin{cases} 1 & \text{if } y_{lg_i} >= y \\ 0 & \text{if } y_{lg_i} < y \end{cases} \tag{3.12}$$

Location of main landing gears along the structural semi-span and the correspondent weight is known for the present study case. The mass value for the main gears is approximately 15.000kg.

3.1.5 Shear force and bending moment

For wing structural sizing, shear force and bending moment are computed using the Equations (3.1) and (3.2). The shear force caused by lift load distribution is defined by Equation (3.3), the shear force caused by fuel stored in the wing by Equation (3.5) and the total shear force by Equation (3.1). Their trend for the case under consideration are shown in Figure 3.4(a) while their correspondent bending moment are reported in Figure 3.4(b). Bending moment due to lift load distribution is computed by Equation (3.4), bending moment caused by fuel by Equation (3.6) and the total bending moment by Equation (3.2). It is worthy noting the moderate relief in shear force, thus in bending



(a) Vertical shear force (b) Out-of plane bending

Figure 3.4: Spanwise resultant loads

moment, introduced by engines and main landing gears attached to the wing.

3.2 Structural sizing

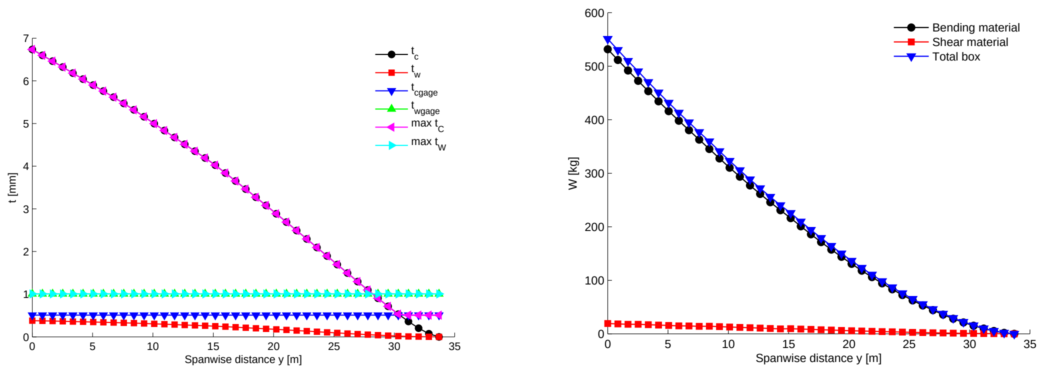
Once the load condition is known, it is possible to carry out the wing structural sizing. Depending on the geometric and structural concept the wing is supposed to be built, the correspondent minimum gage thickness for covers t_{gC} and webs t_{gW} can be easily calculated. Besides the minimum amount of required material, analytical calculations are performed within GUESS .
The following points are defined in the *Structural* module for this study case:

- the structural concept in the entry `wing.kcon` $=$ 2 defines a wing with *unstiffened* covers and *unflanged* webs (see Table 2.13). The minimum gage thickness for covers and webs is computed, respectively, as $t_{gC} = t_g/K_{gC}$ and $t_{gW} = t_g/K_{gW}$ where K_{gC} and K_{gW} are constants depending on the wing structural concept;

- GUESS estimates the thickness t_c of covers and the thickness t_w of webs according to analytical formulas;

- the maximum thickness values for covers $t_C = max(t_{gC}, t_c)$ and webs $t_W = max(t_{gW}, t_w)$ are considered for the wing structural sizing.

In Figure 3.5(a), minimum gage thickness, analytical thickness and maximum thickness for covers and webs are depicted.
Weight distribution of bending and shear material is computed separately (W_{bend} and W_{shear}); the total *ideal* weight of the wing box structure is simply computed as the sum of these values (see 3.5(b)).
For the carrythrough structure, weight estimation is straightforward because the same procedure as for structural wing box is utilized. The wing carrythrough structure consists of torsion material in addition to bending and shear material. The torsion material is required to resist the twist induced due to the sweep of the wing. In the following, torsion, bending and shear material weight for the wing carrythrough structure are indicated, respectively, as $W_{torsion_C}$, W_{bend_C} and W_{shear_C}.



(a) Thickness parameters (b) Weights distribution

Figure 3.5: Spanwise wing properties

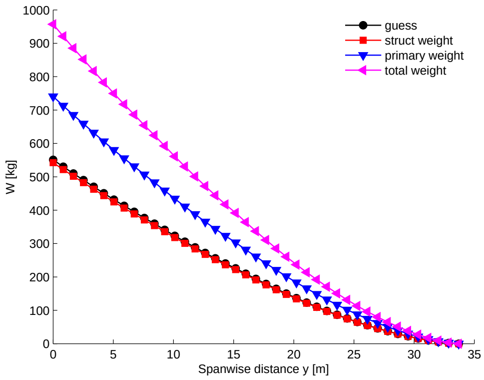
3.3 Sectional properties distribution

Corrected weight distributions are obtained when regression analysis is applied. A linear regression equation is defined in the entry `regression.analw` and primary, secondary and total weight distributions can be obtained as reported in Figure 3.6(a).
In the following Figures 3.6(b), 3.7(a), 3.7(b), the distribution of area, second moment of inertia and torsional constant are reported. Estimates are done firstly on the analytical sizing mesh and then are interpolated on the beam model mesh.

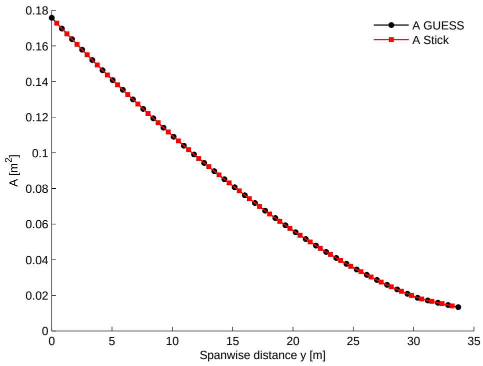
3.4 Validation of GUESS estimation

Weight estimation provided by GUESS is not simply limited to the evaluation of the total structural weight for the cantilever wing herein presented. Indeed, the procedure provides:

- for the structural semiwing, weight distribution of shear and bending material, separately;

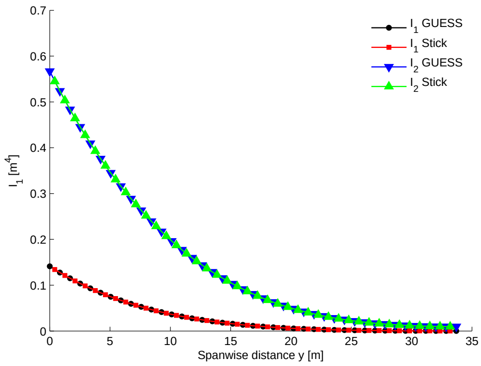


(a) Weights

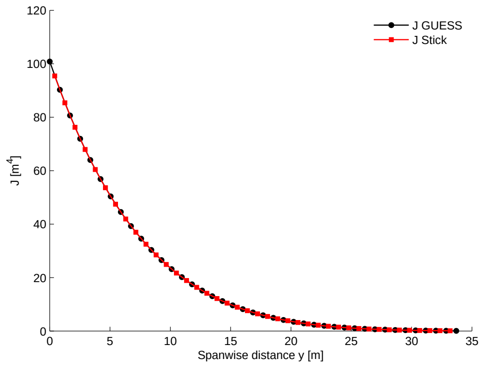


(b) Area distribution

Figure 3.6: Spanwise wing properties



(a) Inertias distribution



(b) Torsional constant distribution

Figure 3.7: Spanwise wing properties

- for the carrythrough structure, weight of shear, torsion and bending material, separately.

Since a detailed weight estimation is available in NASA-PDCYL code [3] accounting for each single weight contribution, GUESS may be compared with this reference. It is assumed that the comparison is performed for the load-carrying weights, which are directly computed from the analytical procedure and not corrected using regression analysis.
Table 3.1 reports the weight of shear material, W_{shear}, and bending material, W_{bend}, for the cantilever wing; load-carrying weight of structural semiwing box is the sum of shear and bending weight, $W_{tot} = W_{shear} + W_{bend}$.
Again in Table 3.1, weight comparison for the carrythrough structure is reported. Weight of shear material, W_{shear_C}, bending material, W_{bend_C} and torsion material, $W_{torsion_C}$, are summed to obtain the load-carrying weight of the carrythrough structure W_{tot_C}.
It is worthy noting that the carrythrough weight represents the weight of the complete carrythrough structure. Thus, the load-carrying weight of the wings is the sum of both semi-wings and the carrythrough structure $W_{wings} = 2 \cdot W_{tot} + W_{tot_C}$.

	NASA-PDCYLIN	**GUESS**
structural wing box		
W_{shear} [kg]	401	345
W_{bend} [kg]	8564	8700
W_{tot} [kg]	8965	9045
carrythrough structure		
W_{shear_C} [kg]	127	140
W_{bend_C} [kg]	3625	3787
$W_{torsion_C}$ [kg]	673	720
W_{tot_C} [kg]	4425	4648

Table 3.1: Boeing 747−100: load-carrying weights.

3.4.1 Second conclusions

Although the first conclusions given in Section 2.8.3 for six different aircrafts presented the Boeing 747−100 as the vehicle whose structural estimation was the most far from the actual weights, the comparison presented in Table 3.1 underlines a good overall agreement between the structural sizing performed by GUESS and the reference values provided in reference [3], which is the source for the reference values indicated as NASA-PDCYLIN in the above comparison.

Both carrythrough structure and structural wing box have been analyzed for the semi-wing of a Boeing 747−100. The biggest error in the above comparison concerns the weight estimation for shear material in the cantilever wing, which is around 13%. Indeed the estimation of the load-carrying weight for the cantilever wing estimated by GUESS is less than 1% far from the reference value.

Similarly for the carry-through structure. The biggest difference in the shear material weight estimation consists of an error around 10%. Due to the small fraction of shear material required in the carry-through structure, the load-carrying weight is 5% greater then the reference values.

Even not modelling the actual wing by means of a detailed geometric description, remember the planform is still described by a trapezoidal wing, the predicted weights overlap the actual weights. The overall good agreement now achived is performed by a revisited gemetric description, which is slightly different in few parameters respect the ones presented in the reference. The aircraft description has been updated to the *geometric* input file layout adopted by SimSAC and misleading parameters given in the reference have been easily adjusted.

At this stage, GUESS has shown to give accurate and fast semi-analytical prediction of weights and stiffness distribution. Additional routins are used to create an output file in ASCII format, represanting the beam model structural and aerodynamic mesh. Thus, GUESS is easily integrated in a more complex environment and actually is successfully adapted to give a first structural sizing estimations, later refined by multi-disciplinary optimisation tools.

3.5 Generation of the beam model for aeroelastic analysis

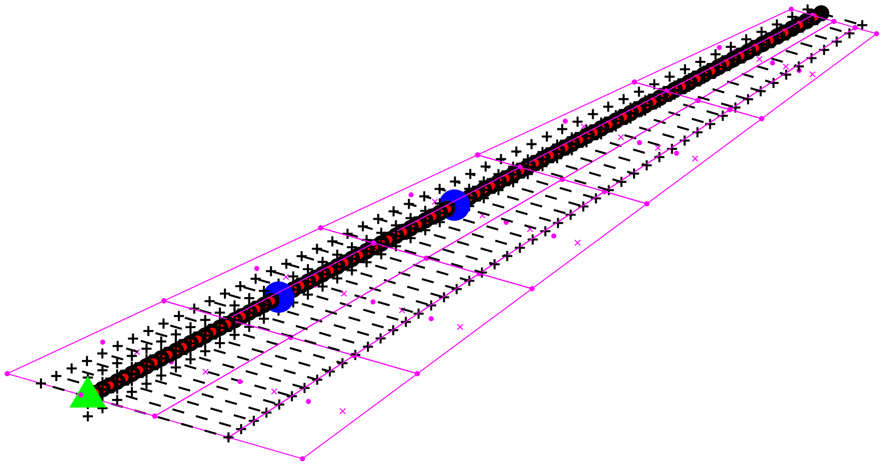


Figure 3.8: Beam model for the B747 semiwing

Figure 3.8 gives a brief overview of the beam model generated for the numeric aeroelastic solver. It is possible to node all the elements which are required to carry out the aeroelastic analysis:

- beam elements (black segments);
- nodes (red points);
- lifting surfaces for Vortex Lattice or Doublet Lattice Method (magenta);
- lumped masses (blue points);
- constraints (green triangle);
- rigidly linked point to structural beam nodes for aeroelastic spatial coupling (black crosses)

Of course, structural data such as material and beam mechanical properties are available in the generated output file. All the parameters for the aeroelastic spatial coupling method are defined to succesfully couple the aerodynamic to the structural beam model. Some of parameters required to rule the creation of the beam model can be modified in the `Stick_model` section of the *technology* file (see Appendix A). Many other parameters are set by default by the software and can be manually modified in the output file. Freedom is left to the user to add/modify the output file with further parameters and solver-settings for the aero-structural module SMARTCAD available in NeoCASS or the commercial code NASTRAN© .

Chapter 4

Practical demonstration of Structural Sizing by means of GUESS

4.1 TransCRuiser – TCR

The application of the structural solver is performed over the conceptual design of a target aircraft, named TransCRuiser (TCR), as most of the work packages in SimSAC project is focused on this test-case.
The present study-case shows the reliability to use GUESS computer code as structural solver for the first structural estimation in preliminary design phases and pre-processor for the generation of suitable finite elements model in order to successfully perform further analysis. The mission profile for TCR is sketched in Figure 4.1 and the design specifications are given in Table 4.1.

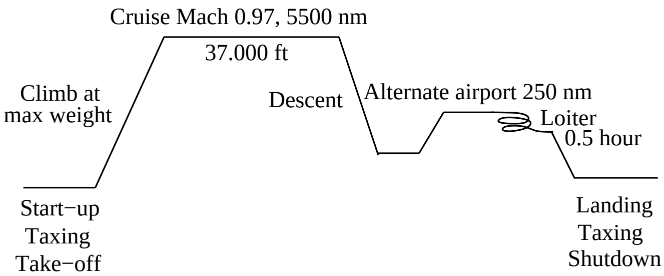


Figure 4.1: Mission profile for TCR.

The main purpose of the present section is to provide a structural sizing estimation of the TCR and compare weight and balance estimations using classical methodologies with the new structural tools developed within GUESS computer code. As the tool performs a great variety of calculations, in the following pages some key points will be given to better understand the procedured implemented within the tool.
A survey on the influence of the available structural concepts provided by the analytical tool is performed over fuselage and main lifting surface and different structural configurations are obtained. The structural design options are compared by means of the total weight, either for fuselage and for wing. Using the concepts introduced in Section 2.6.2,

Cruise Mach	0.97 at altitude $\geq$ 37.000 ft
Range	5.500 nm + 250 nm to alternate airport + 0.5 hour loiter at 1.500 ft
Max payload	22.000 Kg
Passengers	200
Crew	2 pilots, 6 cabin attendants
Take-off distance	2.700 m at max W_{TO} altitude 2.000 ft
Landing distance	2.000 m at max W_L altitude 2.000 ft max payload and normal reserves
Powerplant	2 turbofans
Certification	JAR25
Maneuvering load factors	2.5, -1
Max load factors	3.1, -1.7

Table 4.1: Design specifications for TCR.

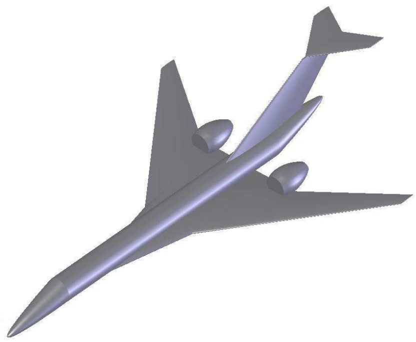
the solver is able to determine a detailed description of the fuselage cross-sectional geometry, not only providing a smeared equivalent thickness for the shell, as later detailed.

4.1.1 Structural sizing and comparison

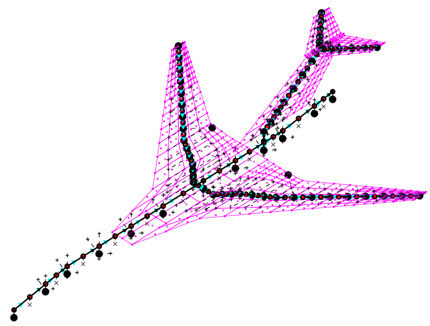
After the first structural sizing performed by *Weight and Balance* module (W&B) on the TCR airplane, GUESS is applied to refine the weight prediction and to give a first solution of stiffness and load bearing material distribution. Once this last is available, an airframe model using beam elements is automatically generated to be used by the numerical aero-structural module SMARTCAD.
Figure 4.2(a) shows the generated CAD model. The stick model generated by GUESS is depicted in Figure 4.2(b). It is worthy noting the representation of fuselage by means of beam elements and the absence of aerodynamic panels to simulate the aerodynamics.
Starting from an estimation of a total fuel amount of 92.000kg, distributed between integrated wing tanks and fuselage central tank, the MTOW prediction perfomed by W&B module is 220.035 kg. For the purpose of GUESS structural solver, the user shall define the structural concepts the fuselage and lifting surfaces are supposed to be designed. For the current analysis the following have been selected:

- the structural concept for fuselage have been set to `kcon=4`, namely Z-stiffened shell;
- the structural concepts for lifting surfaces have been set to `kcon=4`, namely truss-stiffened cover;

After the structural sizing obtained by means of GUESS , the new MTOW weight for the TCR is 225.220 kg. As detailed in Section 2.7, this weight is determined considering two regressions on load-bearing material which is numerically determined for the considered aircraft under physical basis. Furthermore, the position of the center of gravity and the aircraft total weight for the MTOW configuration are reported in Table 4.2 and compared with the previous estimations of W&B.
Comparison of the second moment of inertia measured respect the center of gravity for



(a) Generated CAD model.



(b) Generated GUESS model.

Figure 4.2: Transonic cruiser.

	x_{MTOW} [m]	$MTOW$ [kg]
W&B	36.39	220.035
GUESS	36.53	225.220

Table 4.2: TCR weight survey

the MTOW configuration is reported in Table 4.3.
Table 4.4 compares the results by W&B with the new estimation of the structural components determined by GUESS , respectively the weight of the load-bearing material and the two steps of regression to account for secondary airframe components as introduced above.
As expected, the results computed by GUESS show an overall good agreement with the first estimations performed by W&B. Thus, the new structural sizing tool performs a consistent estimation along with extracting the mechanical properties that are not available in W&B module. To define the weights and inertias for the TCR airplane, the load-bearing material weights for the structural components, as reported in the above table, and the non-structural masses, as recovered by W&B module, have been considered.

	I_{xx} [kg m^2]	I_{yy} [kg m^2]	I_{zz} [kg m^2]
W&B	$9.768 \cdot 10^6$	$2.041 \cdot 10^7$	$2.824 \cdot 10^7$
GUESS	$8.315 \cdot 10^6$	$2.243 \cdot 10^7$	$2.909 \cdot 10^7$

Table 4.3: TCR inertia survey

Component	**GUESS [kg] bearing material**	**GUESS [kg] primary weight**	**GUESS [kg] total**	**W&B [kg]**
Wing	15.494	21.158	27.345	24.904
Horizontal tail	722	986	1.275	1.034
Vertical tail	1.378	1.882	2.432	2.899
Fuselage	11.148	15.580	21.206	20.931

Table 4.4: TCR weight survey for each single component.

Non-structural masses distribution

The main difficulty in this phase is the creation of a stick model which can also faithfully represent the assumed mass distribution of the payload and non-structural mass. Representing concentrated items like engines, auxiliary tanks and landing gears by means of lumped masses rigidly attached to the main structure is trivial. Introducing distributed masses like the fuel in the wing-box, paint, passengers, furniture and especially the secondary structural weight which is only predicted as a value by the regression curve, is more complicated.
The following guidelines have been adopted within the solver:

- secondary structural masses (obtained by regression analysis within GUESS) are simply placed as lumped masses on the nodes of the mesh proportionally to the volume of the beams;
- non-structural masses (estimated by W&B module) are considered to be distributed masses per unit length.

Distribution of non-structural masses along fuselage lenght have been reported in Figure 4.3. Each single item is supposed to be distribuited over a specific portion of the total fuselage length and it is distribuited using an appropriate criteria. It is worthy noting that non-structural masses distribuited over fuselage lenght are:

1. interior, over fuselage length in proportion to beam volume;
2. furniture, over fuselage length in proportion to beam volume;
3. baggage, over passenger compartment length in proportion to beam volume;
4. crew, over passenger compartment length in proportion to beam volume;
5. passengers, over passenger compartment length in proportion to beam volume;
6. pilots, over cockpit length in proportion to beam volume;

7. paint, over fuselage length in proportion to beam wetted area;

Non-structural mass for all lifting surfaces, wing and tails, include paint weight which is distributed in proportion to the beam wetted area. No additional masses are considered for the tail planes.
For the main lifting surface, wing fuel tanks and central fuel tank are accounted for, recovering the estimations performed by W&B module. The following are adopted:

- fuel stored in wing tanks is distributed in proportion to beam volume over the cantilever wing, from wing-fuselage intersection to wing tip;

- fuel stored in central tank is uniformely distributed over the wing carrythorugh structure.

Figure 4.4 represents the distribution of non-structural masses over the structural beam for the semiwing, from body center-line to wing tip. It is worthy noting that fuel in central tank is located in correspondence of the carrythrough structure, while the wing fuel tanks are smeared on the structural span, from wing-fuselage intersection to wing tip. Paint weight is distributed in proportion to beam wetted area over the structural span excluding the contribution from the carrythrough structure as it is internal to the fuselage and the more significant contribution from fuselage have been already accounted for.

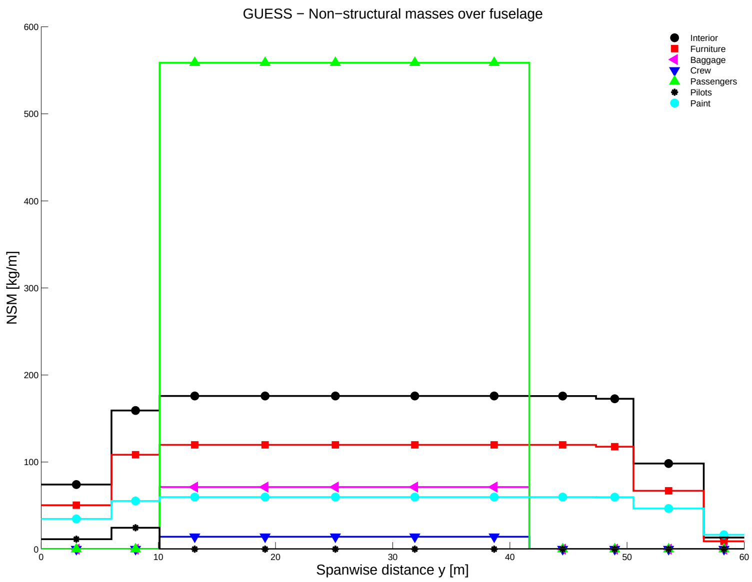


Figure 4.3: Non-structural masses per unit length distribuited over fuselage.

Stress resultant along fuselage length

The present section gathers togheter the results of the loads used to perform the structural sizing for the fuselage.
The stress resultant in the axial direction caused by longitudinal moment is computed on

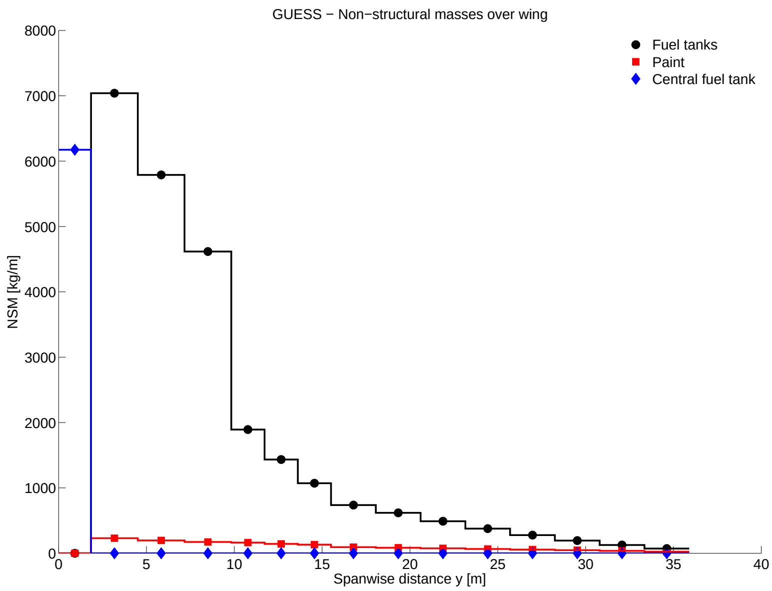


Figure 4.4: Non-structural masses per unit length distribuited over wing semispan.

a station-by-station basis and it is reported in Figure 4.5. Inertia loads distribution and lifting forces from wing and tails are accounted for.
Figure 4.6 represents the stress resultant in the axial direction caused by the propulsion system. The TCR aircraft is equipped with wing-mounted propulsion pods, as clearly shown in Figure 4.2(a). Inertia loads due to the propulsion pods are compressive ahead of the intersection between the structural beams representing the fuselage and the wing structural box, and tensile behind it.

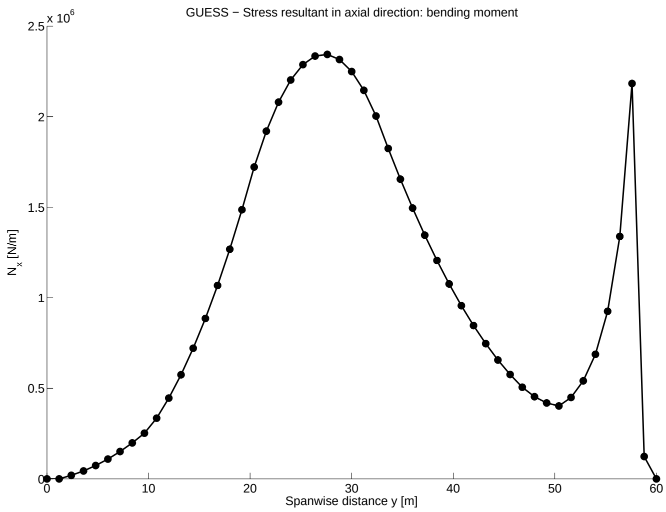


Figure 4.5: Stress resultant in axial direction caused by bending moment.

The effects of internal pressure in passenger compartment is shown in Figures 4.7 through

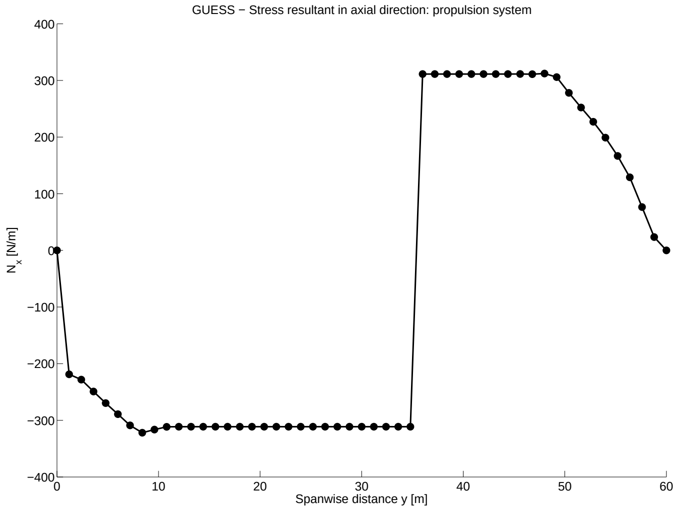


Figure 4.6: Stress resultant in axial direction caused by propulsion system.

4.8. The former represents the stress resultant in the axial direction and the latter in the radial (hoop) direction.

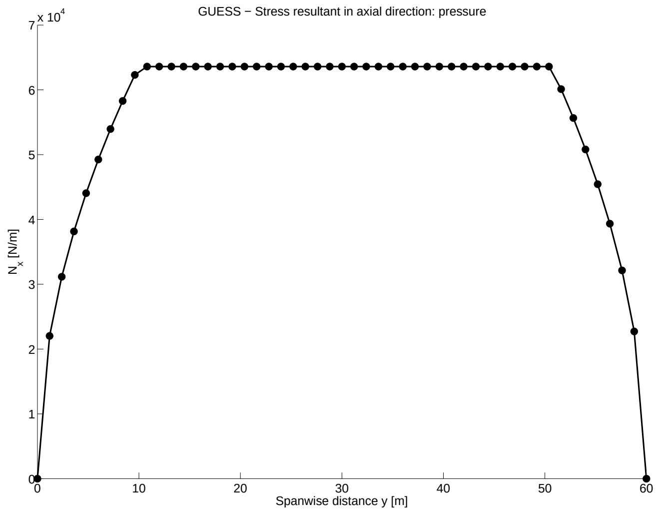


Figure 4.7: Stress resultant in axial direction caused by internal pressure.

Combining the loads in the axial direction using the methodology discussed in Section 2.5.1, the total compressive and tension stress resultant in the axial direction is computed and reported in Figure 4.9.

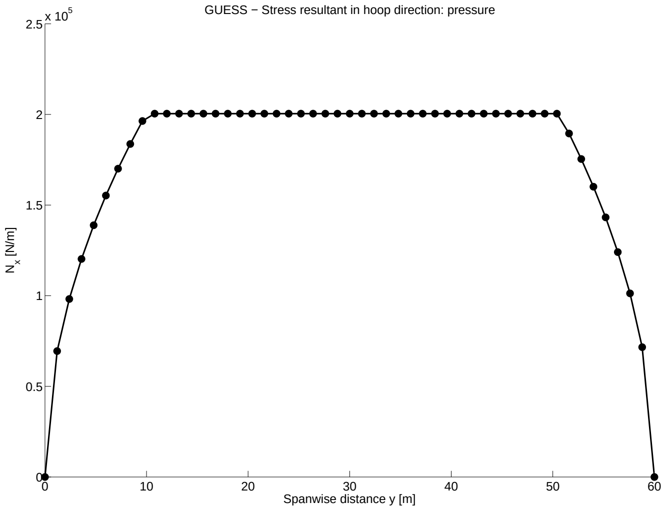


Figure 4.8: Stress resultant in hoop direction caused by internal pressure.

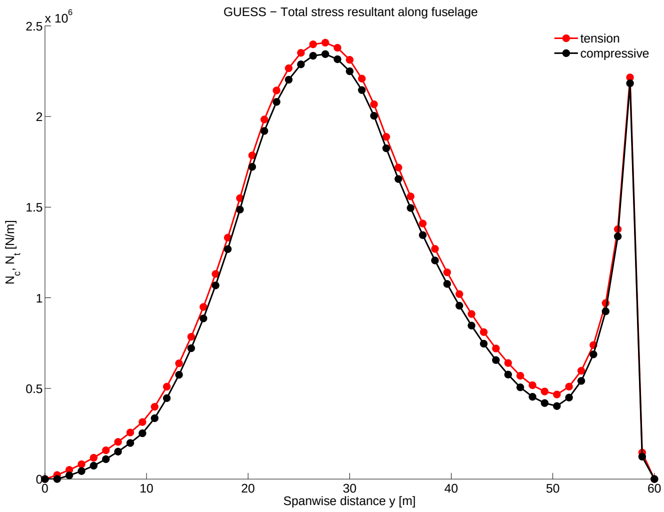


Figure 4.9: Total stress resultant along fuselage structural span.

4.1.2 Aeroelastic model

GUESS generates the beam model for the airframe as the structural sizing procedure has been previously performed. The stick mesh model is automatically created within the solver and exported in the output file in ASCII format. A 3-dimensional representation is shown in Figure 4.2(b) beside the CAD-model. It represents the aero-structural mesh created for the aeroelastic and MDO procedures. Since GUESS has a complete knowledge of the parameters which rule the geometry of the aircraft, it is also possible to create a Vortex/Doublet Lattice mesh. Also, GUESS creates stress-recovery points on the boundary limits of every beam (along lifting surfaces wing-box or fuselage diameter) and transfers to SMARTCAD all its database in terms of material and beam properties, nodes, connectivities, aerodynamic mesh and control surfaces position. A planar view of the TCR aircraft is depicted in Figure 4.10. As pointed out in the Appendix A and in more detail in Reference [19], beams may have offset.

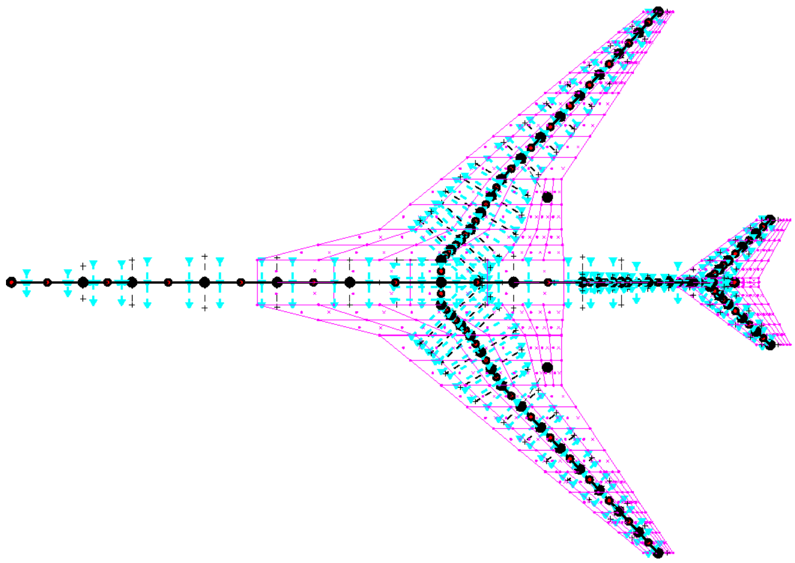


Figure 4.10: Planar view of TCR.

Figure 4.11 shows the use of offset for the fuselage beams in order to get a closed cruxiform scheme for the airframe.

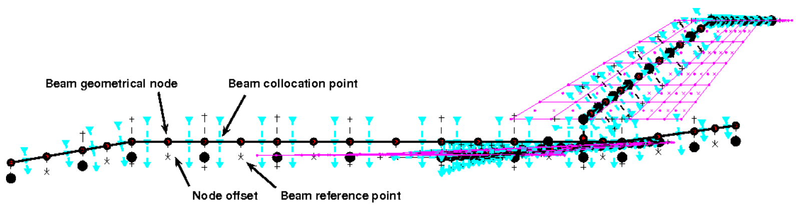


Figure 4.11: Fuselage beam model offset for TCR.

A detailed view of the main wing for the TCR is given in Figure 4.12. The aileron is attached to the tip and expands on the outboard panel. It is worthy noting the dimensions of the wing box as limited by the front and rear spar location. For the TCR, fuel is stored within the wing box.

4.1.3 Fuselage structural concepts survey

This section gathers the results of a structural survey performed over the fuselage of the TCR airplane. For each single analysis, it is assumed that fuselage is designed according different criteria and all the seven structural concepts available within the structural solver have been considered. As discussed in detail in Section 2.6.1, each structural concept is designed to be optimum for a particular loading conditions, thus a variety of structural

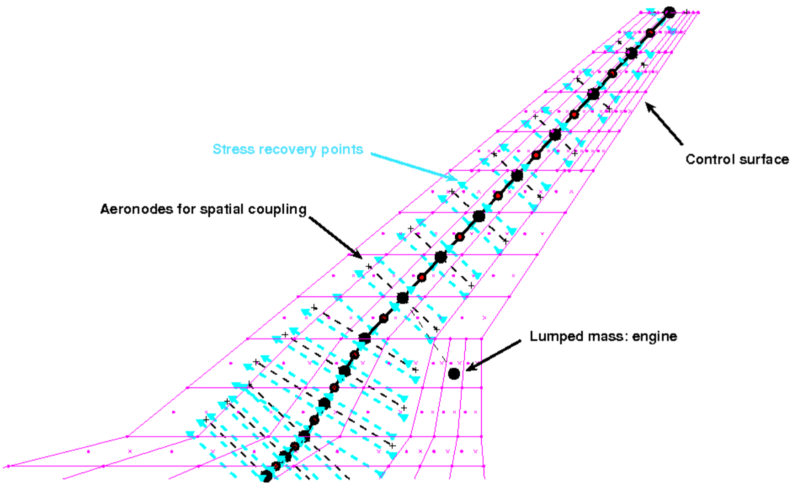


Figure 4.12: Main wing for TCR.

design solutions are expected.
Fuselage is designed using the concepts summarized in Table C.1 and C.2, Appendix C. The results herein presented are summarized considering the following three cross-sectional geometry:

1. simply stiffened shell, for structural concept 1;
2. Z-stiffened shell, for structural concept 2 through 4;
3. Truss-core sandwich shell, for structural concept 5 through 7.

The geometric parameters defining the cross-section design have been computed for the section in which the loading condition is the most critical. As the loads are independent on the structural concepts, the geometric parameters are referred to the same cross-section. In the *technology* input file, the entry

`user_input.material_property.fus.kcon`

is edited selecting the correspondent structural option available (1 through 7). The fuselage total weight is reported for each single structural concepts. It is worthy noting that fuselage total weight accounts for all members of the body, including the structural and primary weights. It does not include passenger accommodations, such as seats, lavatories, kitchens, stowage and lightning, the electrical system, flight and navigation system, alighting gear, fuel and propulsion system, hydraulic and pneumatic system, communication system, cargo accommodations, flight deck accommodations, air conditioning equipment, the auxiliary power system and emergency system.

Simply stiffened shell

Fuselage structural concept 1 defines a simply stiffened shell. The typical geometry layout and parameters are defined in Figure 4.13. The geometry parameters computed in the section in which the loading condition is the most critical and the total fuselage weight are summarized in Table 4.5.

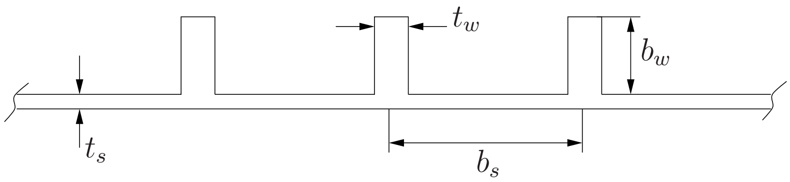


Figure 4.13: Typical unflanged, integrally stiffened shell geometry.

Structural concept	t_s	t_w	b_s	b_w	weight
`kcon`	[mm]	[mm]	[mm]	[mm]	[kg]
1	2.6	5.8	104.7	68.1	20696

Table 4.5: Fuselage design based on simply stiffened shell geometry.

Z-stiffened shell

Fuselage structural concept 2 through 4 defines a Z-stiffened shell. The typical geometry layout and parameters are defined in Figure 4.14. The geometry parameters computed in the section in which the loading condition is the most critical and the total fuselage weight are summarized in Table 4.6. The comparison of total weight and inertias above

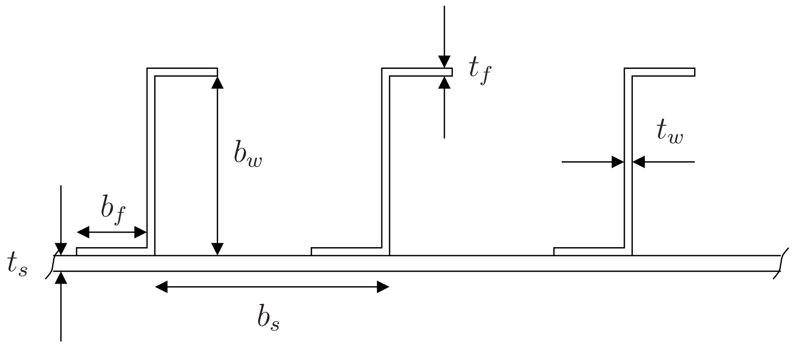


Figure 4.14: Typical Z-stiffened shell geometry.

Structural concept	t_s	t_f	t_w	b_s	b_f	b_w	weight
`kcon`	[mm]	[mm]	[mm]	[mm]	[mm]	[mm]	[kg]
2	2.5	2.7	2.7	77.0	20.1	67.0	20299
3	3.4	3.1	3.1	103.2	18.0	59.9	18820
4	4.0	2.4	2.4	107.2	193.0	64.3	21206

Table 4.6: Fuselage design based on Z-stiffened shell geometry.

presented have been performed assuming the fuselage to be designed with Z-stiffened shell geometry, namely structural concept 4.

Truss-core sandwich shell

Fuselage structural concept 5 through 7 defines a truss-core sandwich shell. The typical geometry layout and parameters are defined in Figure 4.15. The geometry parameters computed in the section in which the loading condition is the most critical and the total fuselage weight are summarized in Table 4.7.

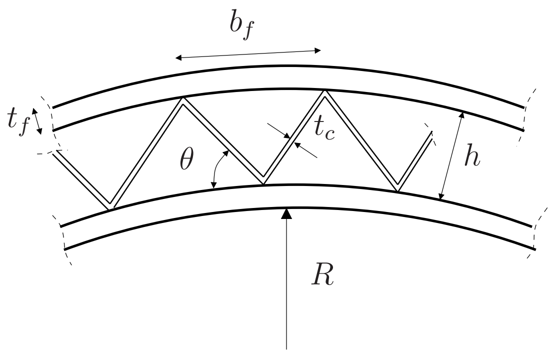


Figure 4.15: Truss-core sandwich geometry

Structural concept	t_f	t_c	b_f	b_c	h	θ	weight
`kcon`	[mm]	[mm]	[mm]	[mm]	[mm]	[°]	[kg]
5	1.6	1.5	43.1	45.9	40.5	62	30191
6	2.1	1.4	55.6	48.5	39.7	55	34235
7	1.8	1.8	56.8	40.2	28.4	45	26161

Table 4.7: Fuselage design based on truss-core sandwich shell geometry.

4.1.4 Wing structural concepts survey

GUESS provides six different structural concepts for lifting surfaces, summarized in Table C.3 and C.4, Appendix C. This section presents the influence of each structural option on wing structural design. Three concepts feature unstiffened covers while the remaining feature truss-stiffened covers. Both cover configurations use webs which can be Z-stiffened, unflanged or trusses. It is then possible to include

1. structural concept 1 through 3 into unstiffened covers category;
2. structural concept 4 through 6 into truss-stiffened covers category;

The survey is performed exclusively over the main lifting surface, namely the wing, and not over the horizontal and vertical tails. In the *technology* input file, the entry

`user_input.material_property.wing.kcon`

is edited selecting the correspondent structural option available (1 through 6). The wing total weight is reported for each single structural concepts. It is worthy noting that wing total weight includes wing box and primary weight items in addition to high-lift devices, control surfaces and access items. It does not include the propulsion system, fuel system, and thrust reverses; the electrical system; alighting gears; hydraulic and pneumatic system; anti-icing devices; emergency system.

Unstiffened covers

The results of the analysis for the wing design based on unstiffened covers and different webs layout are summarized in Table 4.8.

Structural concept	Total weight
`kcon`	[kg]
1	31104
2	30122
3	30962

Table 4.8: Wing design based on unstiffened covers.

Truss-stiffened covers

The results of the analysis for the wing design based on truss-stiffened covers and different webs layout are summarized in Table 4.9. The comparison above presented have been performed assuming the lifting surfaces to be designed using truss-stiffened covers, namely structural concept 4.

Structural concept	Total weight
`kcon`	[kg]
4	27345
5	22742
6	23167

Table 4.9: Wing design based on truss-stiffened covers.

Chapter 5

SMARTCAD: Simplified Models for Aeroelasticity in Conceptual Aircraft Design

This chapter introduces the SMARTCAD module for numeric aero-structural analysis. This module is based on simplified structural and aerodynamic model such as beam models and Vortex/Doublet Lattice aerodynamics to allow rapid aeroelastic in the conceptual design phase. SMARTCAD can used as a stand-alone solver or can be considered as the second tool in the chain to be used after the preliminary sized airframe by GUESS .

5.1 Input file format for the numeric tools

NeoCASS file format for structural, aerodynamic and aeroelastic solvers, are ASCII files derived from NASTRAN© formats, *de-facto* a standard in aerospace industry. Its input files are characterized by a serie of dedicated cards, each of them requires a variable but precise number of inputs. The user not familiar with these formats, is referred to the Quick Reference Guide in [19].
The following motivations justifies the choice made:

- an ASCII file can be easily read, manipulated and is platform independent;
- avoid wasting time to define and learn a new format; the user familiar with the code can easiliy understand NeoCASS input file;
- commercial structural pre-processor and post-processors can be used to visualize the model and the results coming from NeoCASS ;
- NeoCASS can be almost easily bypassed in favour of a NASTRAN© without precluding the overall functionality of CEASIOM design tool;
- the comparison with the validated commercial code is then straighforward.

No modification has been required for the structural cards (grid definition, material properties, beam connectivity and its structural properties) which are actually used as they are originally meant

Nevertheless, minor modifications have been added to the definition of aerodynamic surfaces, in order to:

- agree with the requirements of the aerodynamic mesh generator derived from Tornado;
- provide all the data required for a steady vortex lattice analysis which is missing in NASTRAN© (for example airfoil used, twist given to the mean lifting plane) since steady aerodynamic load is specified by the user.

The following section describe in detail each card which can be used in the input file. Few guidelines are here reported:

- the name of the entry is given in the first field;
- the subsequent fields are card-dependent as explained in each section;
- each field is 8-characters long using NASTRAN© fixed format.
- each card as a maximum of 9 entries;
- when the card is continued in the next line an initial blank space of 8 characters is used.
- blanks entries are usually interpreted as zeros and set to the default values as specified in each card description.

SMARTCAD provides an input file parser which just collects the entries of the allowed cards to define the model. Thus, if an input file comes directly from a NASTRAN© model, the unrecognized cards are simply skipped. For this reason, it is possible to include comments almost everywhere in the file with the following two exceptions:

- among the entris of a card;
- among the lines of a card which is continued.

5.2 Structural models

5.2.1 Beam model validation

Axial buckling load

In linear static analysis, a structure is normally considered to be in a state of stable equilibrium. As the applied load is removed, the structure is assumed to return to its original position. However, under certain combinations of loadings, the structure may become unstable with consequent buckling. Hence, the term “instability” is often used interchangeably with the term “buckling”.

$$P_{cr} = \left(\frac{\pi}{2}\right)^2 \left(\frac{E\, J_{min}}{L^2}\right) \tag{5.1}$$

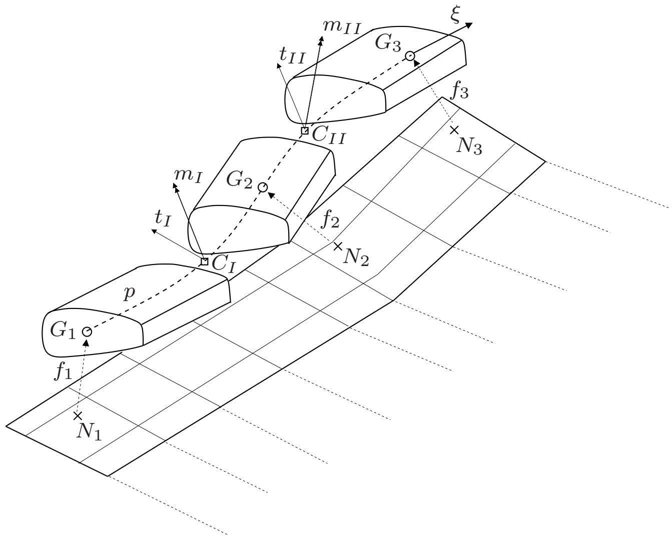
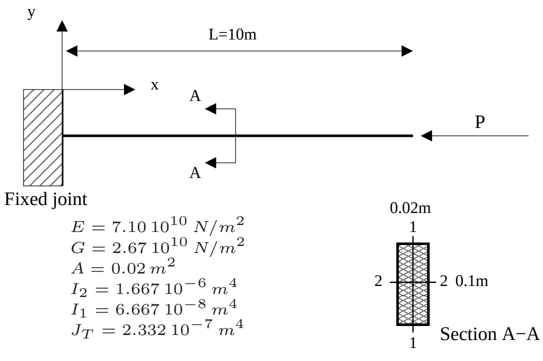
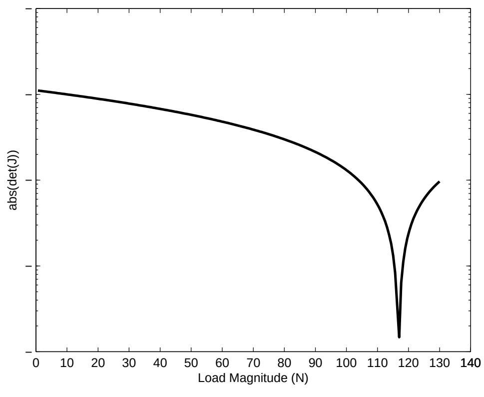


Figure 5.1: Finite Volume three-node beam coupled to an aerodynamic lifting surface method



(a) Axial buckling case sketch

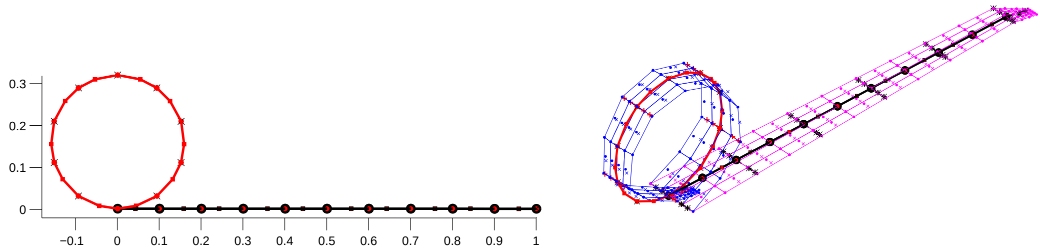


(b) Jacobian determinant plot

Figure 5.2: Axial buckling load determination of a clamped beam (four beam elements used)

N. elements	Buckling load (N)
1	118.554
2	116.915
4	116.803
Theory	116.796

Table 5.1: Comparison of the calculated buckling axial load with linear elasticity theory



(a) Side-view (b) Circle beam coupling validation

Figure 5.3: Creation of a circle beam through a tip moment

Circle beam

$$M_f = E\,J\,y'' \tag{5.2}$$

5.3 Structural model input

5.3.1 Reference frames

A numeric structural model is composed by a set of grid points (usually defined as *nodes*) which, together with structural elements, define the size and shape of the model under investigation. When the location of a node is defined in space, it is implicitly assumed its coordinates are defined relatively to a coordinate system. As it happens with NASTRAN© , the basic absolute coordinate system is indicated with the zero (0) identifier.

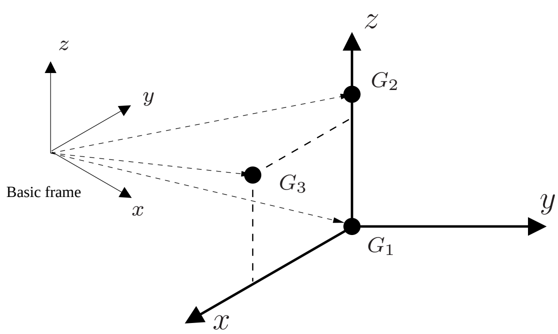


Figure 5.4: Construction of a relative coordinate system

CORD1R	CIA	N1A	N2A	N3A	CIB	N1B	N2B	N3B	

CIA,CIB	frame identifiers
N_iA, N_iB	nodes identifiers

Table 5.2: CORD1R coordinate system card

CORD2R	CID	0	A1	A2	A3	B1	B2	B3	
	C1	C2	C3						

CID	frame identifier
A_i, B_i, C_i	coordinates of three points in the 0 frame

Table 5.3: CORD2R coordinate system card

SMARTCAD uses the two cards CORD1R and CORD2R to define a generic coordinate system respect to the absolute one as sketched in Figure 5.4. No relative coordinate system inter-dependency is allowed in the code. The card CORD1R (see Table 5.2) defines one or two coordinate systems by means of absolute coordinates of three nodes. The first one is the origin, the second lies on the z-axis, and the third lies in the x-z plane. The three nodes points must be noncolinear and not coincident. The card CORD2R (see Table 5.3) defines one coordinate system by means of the coordinates of three points which must be unique and noncolinear. Noncolinearity is checked by the geometry processor. The first point defines the origin. The second defines the direction of the z-axis. The third point defines a vector which, with the z-axis, defines the x-z plane.

5.3.2 Nodes

When a finite element model of a structure is created, an equivalent mathematical model in matrix form is determined. The unknowns appearing in the matrix equation ruling the static or dynamic behaviour of the structure, are the generalized displacements of the model. These displacements generally consist of the six components for each of the nodes (three displacements and three rotations) and are referred to as the degrees of freedom of the model. The GRID card (see Table 5.4) specifies the location of the node ID in space with respect to a reference coordinate system CS in terms of its coordinates CX, CY, CZ. The coordinates given in a relative coordinate system will always be reported to the absolute basic frame.

5.3.3 Material properties

The card MAT1 (see Table 5.5) defines a linear isotropic material properties by means of its Young's modulus E, shear modulus G, Poisson's ratio and density. The values of E, G and NU are involved in the formulation of the sriffness of the elements available, while the material mass density RHO is used in the generation of the mass amtrix of the numeric model.
The following rules apply for the mechanical material properties:

- E and G must not be given as zeros (or blank);
- if NU and E, or NU and G, are both zeros (blank), then both will be set to 0.0;

GRID	ID	CS	CX	CY	CZ	CD			

ID	node identifier
CS	coordinate system identifier to define node coordinates (0 is the main absolute frame)
CX	x coordinate in CS frame
CY	y coordinate in CS frame
CZ	z coordinate in CS frame
CD	Coordinate system identifier in which the displacements, degrees-of-freedom, constraints, and solution vectors are defined at the grid point (at the moment this field is used only by CBAR card defined in Section 5.3.4)

Table 5.4: GRID node coordinates definition

- if only one among E, G, or NU then it will be computed by means of linear elastic theory as $E = 2(1 + NU) \cdot G$.

MAT1	MID	E	G	NU	RHO				

MID	material identifier
E	Young's modulus
G	Shear modulus
NU	Poisson's ratio
RHO	mass density

Table 5.5: MAT1 linear isotropic material definition

5.3.4 Beams

SMARTCAD provides linear/non-linear one-dimensional beam element which can be used to represent structural members that have stiffness along a line two grid points. Typical aerospace application is a beam stick model to easiliy represent the airframe for fuselages and lifting surfaces. Usually these elements are characterized by high aspect-ratio and a predominant axis along which each section extends.
The beam model developed consists of three nodes (the midlength node is automatically generated by the solver) with the following features:

- extensional stiffness along the neutral axis and torsional stiffness about the neutral axis may be defined;
- bending and transverse shear stiffness can be defined in the two perpendicular directions to the axial direction of the element;
- the properties must be constant along the length of the element (this limitation will be removed soon with the adoption of CBEAM/PBEAM cards)
- the shear center and the neutral axis must coincide (this limitation will be removed soon);

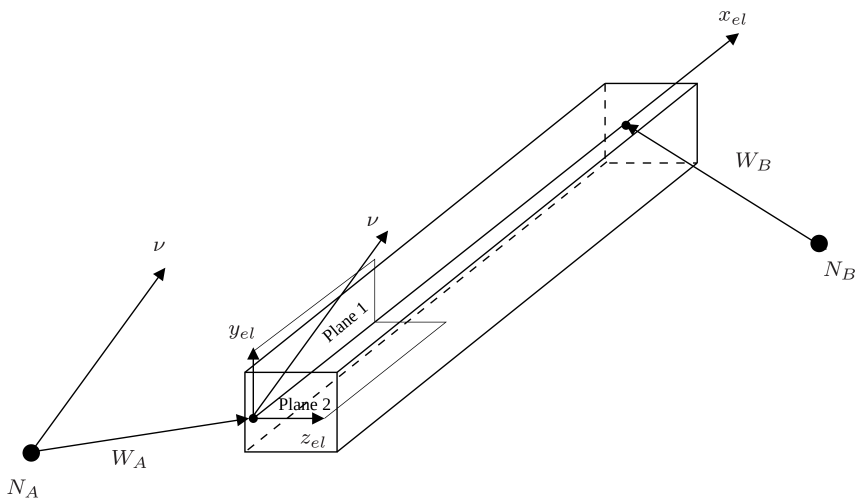


Figure 5.5: CBAR beam geometric parameters

- the ends of the element may be offset from the nodes;
- transverse shear stiffness along the length of the element can be included.

Beam geometry definition

The card CBAR (see Table 5.6) is used to specify the nodes connected by the beam, its structural properties and how the section is laid along the beam axis. Figure 5.5 shows the beam model together with its geometric parameters for the definition of nodal offsets and element coordinate system.

Beam structural properties definition

The PBAR card (see Table 5.8) is used to provide structural informations for the CBAR beam. The following parameters nust be defined:

- total resistant area A;
- material used;
- moments of inertia respect to section axes y_{el}, z_{el};
- torsional constant J;
- non structural mass density;
- shear factors.

All these parameters are require to correctly define the stifness matrix for each element which is the assembled into the global stifness matrix for the structure under investigation. It is important to stress the parameters given are related to the section considered, which is the correctly laid along beam axis by correctly defining the nu vector (see Section 5.3.4) to construct the element reference frame.

CBAR	ID	PID	N1	N2	X1	X2	X3	OFT	
	0	0	W1A	W2A	W3A	W1B	W2B	W3B	
or									
CBAR	ID	PID	NA	NB	G0			OFT	
	0	0	W1A	W2A	W3A	W1B	W2B	W3B	

ID	beam identifier
PID	beam property identifier
NA,NB	Connected nodes identifier
X_i	Components of orientation vector ν, from NA, in the displacement coordinate system at NA (default, see CD field in GRID card presented in Section 5.3.2), or in the basic coordinate system
G0	Alternate method to supply the orientation vector ν using grid point G0. The direction of ν is from NA to G0. ν is then translated to NA
W_iA	Offset components from NA
W_iB	Offset components from NB
OFT	Offset identification (see Table 5.7) OFT is a character string code that describes how the offset and orientation vector components are to be interpreted. By default (string input is GGG or blank), the offset vectors are measured in the displacement coordinate systems at grid points A and B and the orientation vector is measured in the displacement coordinate system of grid point A. At user option, the offset vectors can be measured in an offset coordinate system relative to grid points A and B, and the orientation vector can be measured in the basic system. Any attempt to specify invalid combinations results in an input error message. For example, a value of OOO (indicating offset and orientation vectors are specified in an offset reference system) results in a fatal error since the orientation vector cannot be specified in an offset system. The offset system x-axis is defined from NA to NB. The orientation vector ν and the offset system x-axis are then used to define the z and y axes of the offset system. Structural properties are then referred to the element coordinate system

Table 5.6: CBAR beam definition

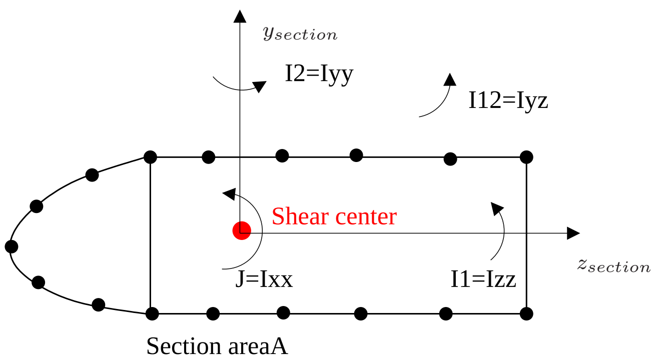


Figure 5.6: PBAR beam structural properties for a *semi-monocoque* section

Label	**Orient. Vector**	**Offset A**	**Offset B**
GGG	Global	Global	Global
BGG	Basic	Global	Global
GGO	Global	Global	Offset
BGO	Basic	Global	Offset
GOG	Global	Offset	Global
BOG	Basic	Offset	Global
GOO	Global	Offset	Offset
BOO	Basic	Offset	Offset

Table 5.7: Beam orientation and offset definition

PBAR	PID	MID	A	I1	I2	J	NSM		
unused line	0	0	0	0	0	0	0	0	
	K1	K2	I12						

PID	Beam property identifier
MID	Material identifier
A	Cross section area
I1,I2,I12	Area moments of inertia ($I1 \cdot I2 > I1^2$)
J	Torsional constant
NSM	Non-structural mass per unit length
K1,K2	Area factor for shear The transverse shear stiffnesses per unit length in Planes 1 and 2 (see Figure 5.5) are K1*A*G and K2*A*G, respectively, where G is the shear modulus. The default values for K1 and K2 are infinite; in other words, the transverse shear flexibilities are set equal to zero

Table 5.8: PBAR beam property definition

5.3.5 Lumped masses

Lumped masses are used to introduce inertial loads due to non-structural masses (engines, on-board systems, payload). Structural mass is automatically accounted for when assembling elements mass matrix due to material density and non-structural mass density as specified through the card PBAR (see Section 5.3.4). SMARTCAD adopts two cards to provide lumped mass informations. The card CONM1 (see Table 5.9) provides the 6x6 mass matrix in a node G. The card CONM2 (see Table 5.10) provides the data for a lumped mass in a node G exploiting the rigid body form by introducing an offset from the reference node. This last is particularly useful when the mass matrix for an engine is known and its mass matrix needs to be reported directly on the beam model if the pylon is not modelled (the pylon is assumed to be infinetely rigid). For the CONM2 card, the

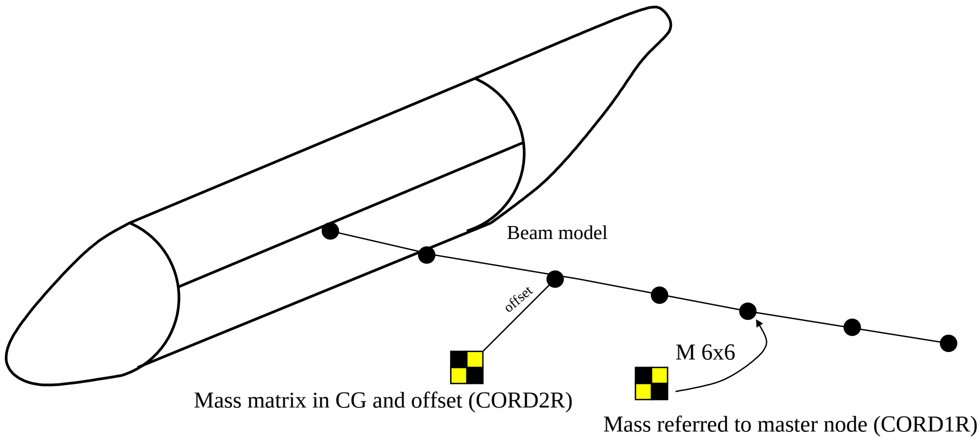


Figure 5.7: Different strategies for lumped masses

CONM1	ID	G	CID	M11	M21	M22	M31	M32	
	M33	M41	M42	M43	M44	M51	M52	M53	
	M54	M55	M61	M62	M63	M64	M65	M66	

ID	mass identifier
G	reference node
CID	coordinate system for the 6x6 mass matrix The mass matrix is then reported into the basic coordinate system
M_{ij}	mass matrix elements value

Table 5.9: CONM1 lumped mass matrix

CONM2	ID	G	CID	M	X1	X2	X3		
	I11	I21	I31	I32	I33				

ID	mass identifier
G	reference node
CID	coordinate system to define mass matrix (maybe equal to -1)
M	mass value
X_i	offset array in the frame CID from node G to mass center of gravity if CID = -1, X_i are treated as coordinates, not offsets, of the center of gravity in the basic frame Offset is determined with the difference of the point X_i and node G
I_{ij}	mass moments of inertia measured at the mass center of gravity in the frame CID. If CID = -1, the basic coordinate system is implied and inertias are implicitly assumed to be defined in a frame parallel to the basic one

Table 5.10: CONM2 lumped rigid body mass matrix

form of the mass matrix about its center of gravity is assumed to be the following:

$$\begin{bmatrix} M & & & & & \\ & M & & & & \\ & & M & & & \\ & & & I11 & -I21 & -I31 \\ & & & -I21 & I22 & -I32 \\ & & & -I31 & -I32 & I33 \end{bmatrix}$$

where:

$$
\begin{aligned}
M &= \int \rho \, dV \\
I11 &= \int \rho \, (x_2^2 + x_3^2) \, dV \\
I22 &= \int \rho \, (x_1^2 + x_3^2) \, dV \\
I33 &= \int \rho \, (x_1^2 + x_2^2) \, dV \\
I21 &= \int \rho \, x_1 \, x_2 \, dV \\
I31 &= \int \rho \, x_1 \, x_3 \, dV \\
I32 &= \int \rho \, x_2 \, x_3 \, dV
\end{aligned}
$$

and x_1, x_2, x_3 are components of distance from the center of gravity in the frame CID, ρ is the density of the element. The negative signs for the off-diagonal terms are supplied automatically.

5.3.6 External concentrated generalized forces

Concentrated forces can be applied directly to strctural nodes with the FORCE, FORCE1, and FORCE2 entries. The cards MOMENT, MOMENT1, MOMENT2 are equivalent but are used to define applied moments (loads on rotational degrees of freedom).
The FORCE card (see Table 5.11) specifies the force direction and magnitude on a node G in a coordinate system CID.
The two remaining cards maybe used to easily prescribe the direction of the load. The FORCE1 (see Table 5.12) entry allows to define a force by specifying a magnitude and two grid points (not necessarily the loaded grid point) to determine force-direction while the FORCE2 (see Table 5.13) entry specifies a magnitude with the direction defined by the vector product of two other vectors.
The load set is then applied to the structural model provided a LOAD= CID card is defined, where CID is the load set identifier. The remaining load sets with other identifiers are then discarded and not used.

5.3.7 External concentrated generalized forces

For static structural analysis, it is possible to define an acceleration vector for gravity or an external acceleration loading. The card GRAV (see Table 5.14) is used to define the direction and magnitude of an acceleration vector in any user-defined coordinate system. The components of the gravity vector are multiplied by the mass matrix to obtain the components of the gravity force at each grid point. Since the mass matrix is used to compute the forces, a non-null material density or lumped masses must be given. Each GRAV entry must have a unique identifier and must be selected by a LOAD as it happens

FORCE	LID	G	CID	F	CX	CY	CZ		
or									
MOMENT	LID	G	CID	F	CX	CY	CZ		

LID	Load set identifier
G	Node identifier
CID	Reference frame identifier
F	scale magnitude factor
C_i	Force/moment components in the frame CID
	The norm of this array gives the force/moment magnitude
O_i	Offset components of the application point from node G in the frame CID

Table 5.11: FORCE/MOMENT load definition

FORCE1	LID	G	F	G1	G2				
or									
MOMENT1	LID	G	F	G1	G2				

LID	Load set identifier
G	Node identifier
F	Force/moment magnitude
G_i	Node identifiers; the force/moment goes from G1 to G2

Table 5.12: FORCE1/MOMENT1 load definition

FORCE2	LID	G	F	G1	G2	G3	G4		
or									
MOMENT2	LID	G	F	G1	G2	G3	G4		

LID	Load set identifier
G	Node identifier
F	Force/moment magnitude
G_i	Node identifiers; the force/moment is parallel to the cross product of vectors from G1 to G2 and G3 to G4

Table 5.13: FORCE2/MOMENT2 load definition

for external generalized forces. When external forces and acceleration share the same load-set identifier, superimposition is simply applied.

GRAV	LID	CID	A	N1	N2	N3			

LID	Load set identifier
CID	Coordinate system identifier
A	Acceleration multiply scale factor
N_i	Acceleration components in coodinate system CID

Table 5.14: GRAV acceleration load

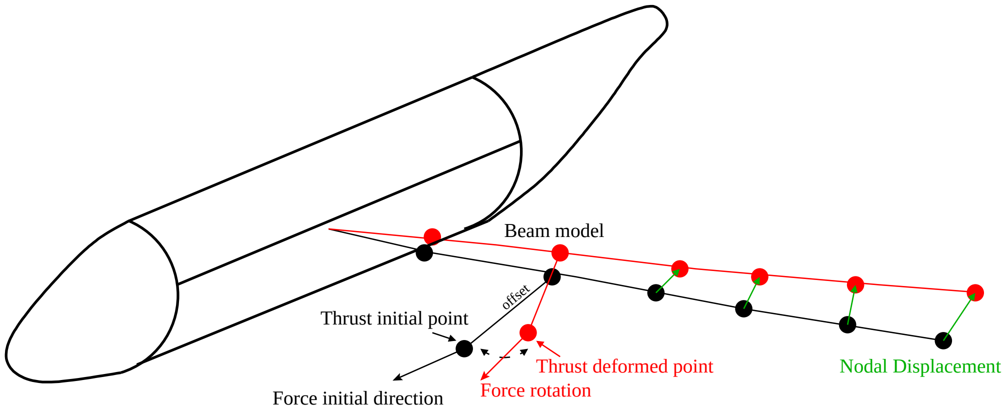


Figure 5.8: Follower force application example

5.3.8 Follower forces

The follower force can be used with the non-linear structural solvers. In this case the force *follows* the deformed structure and the displacements and rotations of its application point. If a linear solver is used, the force will be treated as usual, i.e. the application point is always the same and still in space. The card THRUST provides all the required data in order to correctly account for a follower force (see Table 5.15). The user specifies a reference node G and an offset vector O and force components C in a frame CID (usually the basic 0 frame). During the simulation the force follows the rotations of the master node G keeping fixed the offset from it (rigid arm).
The load set is then applied to the structural model provided a LOAD= CID card is defined, where CID is the load set identifier. The remaining load sets with other identifiers are then discarded and not used.
An example of follower force may be the thrust generated by underwing engines which changes direction of application when the wing deforms as shown in Figure 5.8.

THRUST	LID	G	CID	CX	CY	CZ	OX	OY	OZ
LID	Load set identifier								
G	Node identifier								
CID	Reference frame identifier								
C_i	Initial force components in the frame CID The norm of this array gives the force magnitude								
O_i	Offset components of the application point from node G in the frame CID								

Table 5.15: THRUST follower definition

5.3.9 Constraints

Presently, SMARTCAD performs static analyses provided all rigid body displacements are removed prior to solving the static equilibrium equations. This process involves specifying the appropriate boundary conditions for the model. Boundary conditions are imposed in the form of constraints on selected degrees of freedom on the model. As for the flutter solution (see Section 5.6.5), the code is able to extract vibration modes for the free-free aircraft, thus considering in the flutter solution the effects related to the rigid dynamics. The Single-Point Constraint SPC card provides constraints that are applied to a single degree of freedom. The primary applications for single-point constraints are as follows:

- to tie a structure to ground;
- to apply symmetric or antisymmetric boundary conditions by restraining the degrees of freedom that must have a zero value to satisfy symmetry or antisymmetry;
- to remove degrees of freedom that are not used in the structural analysis (i.e., are not connected to any structural elements or otherwise joined to the structure); at the moment this capability is automatically provided since the code checks if every node is connected to the structure by means of beam elements. If it does not happen, the node is automatically discarded from the structural solution.

The SPC1 card (see Table 5.17) is used to define a list of node sharing the same contraints on their degrees of freedom. The degrees of freedom are expresses by means of their index as shown in Table 5.16. Multiple SPC1 cards are of course allowed and are simply merged all together if the set identifier is the same.
The constraint set is then applied to the structural model provided a SET= CID card is defined, where CID is the constraint set identifier. The remaining sets with other identifiers are then discarded and not used.

Index	**Saturated DOF**
1	x-axis translation
2	y-axis translation
3	z-axis translation
4	x-axis rotation
5	y-axis rotation
6	z-axis rotation

Table 5.16: Correspondence among indeces and saturated degrees of freedom

SPC1	CID	C	G1	G2	G3	G4	G5	G6	
	G7	G8	etc...						
CID	Constraint set identifier								
C	Components to be constrained								
	Use a combination of 1,2,3,4,5,6 (for example 1234 to model a pin with x-axis)								
G_i	List of nodes to be constrained								

Table 5.17: SPC1 nodal contraint definition

5.3.10 Extra parameters

SMARTCAD allows the definition of some optional parameters to rule the management of the internal database.

The parameter STRIM (see Table 5.18) requires symmetric trim search. In this case only two equations will be used, respectively force equilibrium along z-body axis and torque along pitch y body axis. Thus, flight condition is considered to be symmetric. When the parameter is set to one, symmetric trim is required. The default setting is null value and the user need not to include this parameter. The parameter RES_TOL (see Table 5.19) is

PARAM	STRIM	FLAG							
FLAG	0 (default), 1 to require symmetric trim								

Table 5.18: STRIM symmetric trim

used to rule convergence criteria in several solver available in SMARTCAD . By default this value is set to $1.0e-3$. The parameter REL_FAC (see Table 5.20) is used to as under-

PARAM	RES_TOL	VALUE							
VALUE	residual convergence value								

Table 5.19: RES_TOL residual convergence criteria

relaxation factor in coupled aeroelastic analysis when aerodynamic forces are transferred to the structural model. Since aeroelastic analysis consists in a coupling process between the two systems, aerodynamic forces applied on the structure at i iteration are:

$$F_a \;=\; (1-\alpha)F_a^{i-1} + \alpha\, F_a^i$$

where α is the relaxation-factor. This way, a sort of fictitious damping is introduced to the system which generally prevents from over-shoot during the iterative process. The

PARAM	REL_FAC	VALUE							
VALUE	under-relaxation factor (default value is 0.5)								

Table 5.20: REL_FAC residual convergence criteria

parameter NSTEP (see Table 5.21) is used to rule the number of external iterations during aeroelastic trim process or static analysis with non-linear beams.

PARAM	NSTEP	VALUE							
VALUE	number of outer iterations (default value is 10)								

Table 5.21: NSTEP outer iterations

The parameter NITER is used to rule the number of inner iterations for each outer NSTEP iteration.

The parameter GRDPNT (see Table 5.24) specifies the identification number of the node to be used as a reference point for rigid body mass matrix. If GRDPNT=0, the reference

PARAM	NITER	VALUE							
VALUE	number of innter iterations (default value is 5)								

Table 5.22: NITER residual convergence criteria

PARAM	GRDPNT	ID							
ID	Node identifier								

Table 5.23: GRDPNT reference point definition

point is taken as the origin of the basic coordinate system. If this point is not defined, the rigid body mass matrix will be determined The parameter WTMASS is used to scale all the terms of the structural mass matrix. This is sometimes required to correctly scale dynamics problem using consistent units. Everything regarding structural inertias is scaled: lumped masses, material densities, non-structural densities. Aerodynamic flight densities are not scaled. The parameter EIG_FILE is used to set the header name of

PARAM	WTMASS	VALUE							
VALUE	density and mass scale factor								

Table 5.24: WTMASS structural inertias scaling factor

three structural files storing vibration modes data. This feature can be used to use SMARTCAD for flutter calculation on an external modal coming from:

- experimental Ground Vibration Tests (GVT);
- external structural codes with different element formulation and types;
- a user-defined mode sets.

SMARTCAD will look for three ASCII files with the following features:

- EIG_FILE.mas with the generalized mass matrix;
- EIG_FILE.stf with the generalized stiffness matrix;
- EIG_FILE.mod with the modal shapes;

PARAM	EIG_FILE	ID	'FILE_NAME'
ID 'FILE_NAME'	card identifier global head filename (path included). Note the apeces		

Table 5.25: EIG_FILE external vibration modes data

An example of .stf file is here reported for the AGARD 445.6 wing [25]:

```
3637.721    0.000000   0.000000   0.000000    0.000000    0.000000
0.000000    57501.128  0.000000   0.000000    0.000000    0.000000
0.000000    0.000000   92275.890  0.000000    0.000000    0.000000
0.000000    0.000000   0.000000   330845.600  0.000000    0.000000
0.000000    0.000000   0.000000   0.000000    550800.200  0.000000
0.000000    0.000000   0.000000   0.000000    0.000000    776008.200
```

The same applies for the mass matrix. As for the .mod file, modal shapes displacements (columns from 2 to 4) and rotations (from 5 to 7) are simply reported consequently together with node identifiers (1 column):

```
1 0.000000 0.000000 -0.003098 0.000000 0.000000 0.000000
2 0.000000 0.000000 -0.001133 0.000000 0.000000 0.000000
3 0.000000 0.000000  0.000000 0.000000 0.000000 0.000000
. .        .         .        .        .        .
. .        .         .        .        .        .
. .        .         .        .        .        .
1 0.000000 0.000000 -0.023803 0.000000 0.000000 0.000000
2 0.000000 0.000000 -0.009672 0.000000 0.000000 0.000000
3 0.000000 0.000000  0.000000 0.000000 0.000000 0.000000
```

Actually, nodal rotations are not used by the solver and need not to be given in the file. Modal shapes must be given for all the grid nodes defined in the model by the GRID card. The loaded database can be restricted just to a selection of modes by means of the MSELECT card where a list of modal indeces is given:

MSELECT	ID	M1	M1	M2	M4	M5	M6	M7	
	M9	M9	etc...						

CID	Card identifier
M_i	List of mode indeces to be extracted from an external normal modes database

Table 5.26: MSELECT external normal modes selection

The parameter DIVERG (see Table 5.27) is used to enable the calculation of static divergence when aeroelastic trim is carried out. When this parameter is given, the solver determines the dynamic pressure of divergence together with its eigen-mode for the Mach number specified in the card AEROS (see Table 5.48).

PARAM	DIVERG								
	Require divergence calculation								

Table 5.27: DIVERG static divergence calculation

5.4 Aerodynamic modelling

SMARTCAD provides the following two aerodynamic methods:

- a new steady solver for the Vortex Lattice Method (VLM) for TORNADO. This new solver directly recovers aerodynamic loads on each panel given the downwash boundary condition and supports vertical symmetry aerodynamic condition, thus lowering computational costs for these cases;
- a new Doublet Lattice solver used for generalized forces prediction for discrete values of reduced frequency $jk = \frac{\omega L_a}{V_\infty}$ and unsteady stability derivatives.

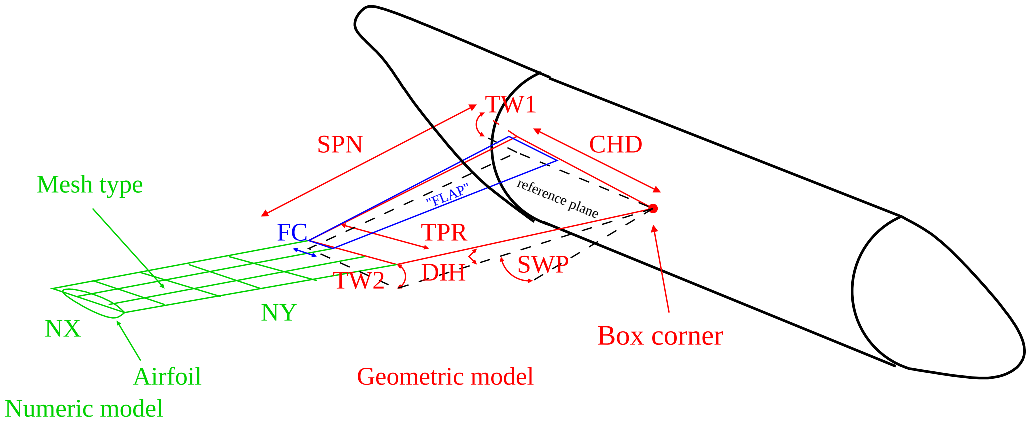


Figure 5.9: Aerodynamic model input parameters

Description	Label
Starting box corner	C
Root chord	CHD
Dihedral angle	DIH
Root twist angle	TW1
Tip twist angle	TW2
Sweep angle	SWP
Taper	TPR
Span	SPN
Control surface chord fraction	FC

Table 5.28: Aerodynamic geometry parameters

The former method is used for static aeroelastic analyses (see Section 5.6.6) or for simple steady rigid aerodynamic solutions (see Section 5.6.7). The latter for the creation of Reduced Order Models (ROM) of the generalized forces to be used in flutter analysis (see Section 5.6.5). As for aerodynamic geometry, both methods share the same data structure, since they are represented by a mean lifting surface. Minor differences are present in the mesh generation since the former is based on vortex filaments, while the latter on simple chordwise doublets.

5.4.1 Lifting surface geometry and mesh definition

The CAERO1 card (see Table 5.30) is used to define all the geometric (see Table 5.28) and mesh parameters (see Table 5.29) for each box of the aerodynamic model which are required by the meshing tool coming from TORNADO Vortex Lattice code. Figure 5.9 briefly illustrates the required information set. For each box a reference lifting surface is constructed which is then dicretized into a user-defined number of panels. A specific mesh-generator is called depending on if Vortex Lattice or Doublet Lattice is selected; each method requires indeed a different number and type of mesh data. The Doublet Lattice for example, does not require wake generation and needs the definition of the axis, length and midpoint for each doublet. Each surface is treated as a flat plate aligned with the flow-field. For the Vortex Lattice it is possible to define the airfoil used at the

Description	Label
Chordwise numberof panels	NX
Chordwise control number of panels	FNX
Spanwise panel number	NY
Mesh type	TYP
Root airfoil	RFO
Tip airfoil	TFO
Control name	NAM

Table 5.29: Aerodynamic mesh parameters

root and the tip of each box, with linear interpolation of the mean line normals between the two extremes. The user specifies one of the airfoils available in TORNADO library.
For each box it is possible to define a trailing-edge control surface in the second input line by means of the ratio of its chord and the one of the boc. Futhermore, a name for the control is required to identify the surface during the steady trim process. The next input lines are used to define successive neighbouring partitions sharing one side with the previous one. It is of course possible to define control surfaces in the same way as done with the first partition (the second line the input card is exactly the same for each partition).

5.5 Spatial coupling methods

In order to use a staggered approach as the one adopted in SMARTCAD , where two independent codes, each one optimal for its purpose, are used for each field and must interact, a spatial coupling scheme is required. The adoption of a partitioned approach [9] requires the definition of an interface scheme to exchange displacements, velocities and loads between the structural grid and the CFD boundary surfaces. The two models are typically discretised in very different, often incompatible, ways. Structural models usually present complex geometries, including many discontinuities. They are often based on schematic models, which have a long tradition in the aerospace industry, using elements with very different topologies, such as beams and plates, which usually hide the real structural geometry up to the point of making the aircraft external shape partially or completely disappear. Despite the computational power available nowadays, these simplified models will be used for some time to come in aerospace industry, especially in early design stages where NeoCASS is supposed to be adopted, so it is essential to be able to cope with them as well. On the contrary, CFD meshes require an accurate description of boundary surfaces. As a consequence, these two representations of the same aircraft must be made compatible in order to transfer information between them. This is a well known problem, already tackled in the literature. For general references on the problem see [24, 4].
An innovative scheme, based on a "mesh-free" Moving Least Square (MLS) method [21], is proposed in this paper to cope with incompatible situations, when the two meshes do not share a common surface.
A second approach is the Radial Basis Function (RBF) method [22], which is the state of the art in spatial coupling methods.

CAERO1	AID	DIH	CID	NY	NX	RFO	TFO	TYP	
	CX	CY	CZ	CHD	SPN	TPR	SWP	TW1	TW2
	FLP	FC	FNX	NAM					
Optional	DIH	NY	SPN	TPR	TFO	SWP	TW2	TYP	
Optional	FLP	FC	FNX	NAM					

AID	Aerodynamic box identifier
DIH	Dihedral angle (degrees)
CID	Coordinate system identifier to define box corner coordinates
NY	Spanwise panel number
NX	Chordwise panel number
RFO,TFO	Root/Tip airfoil name (for VLM only)
	The .DAT extension will be automatically appended
	Blank or 0 for a flat plate (or DLM used)
TYP	Mesh type (see Table 5.31)
C_i	Box corner coordinates in CID coordinate system
CHD	Box root chord
SPN	Box span
TPR	Box taper
SWP	Box sweep (degrees)
TW1,TW2	Box root/tip twist (degrees)
FLP	Specify if partion has a control surface (0=NO, 1=YES)
	If zero value, the data on the same line will be discarded
FC	Control surface chord fraction respect to CHD
FNX	Control surface chordwise panel number
NAM	Name associated to the control surface

Table 5.30: CAERO1 aerodynamic box definition

Both methods ensure the conservation of momentum and energy transfer between the fluid and the structure and they are suitable for the treatment of complex configurations. To guarantee the conservation between the two models, the correct strategy consists in enforcing the coupling conditions only in a weak sense, through the use of simple variational principles. Using the Virtual Works principle the energy exchange can be investigated as reported in [14].

5.5.1 Aeronodes

SMARTCAD easy the interpolation process for the spatial coupling between structural and aerodynamic meshes in the case of beam elements. The main problem relies in the way to transfer nodal rotations from the structural one-dimensional model to the two dimensional lifting surfaces which represent the aerodynamic mesh. Of course, if only displacements are trasferred an important piece of information is lost with poor interpolation results (variation of angle of attack due to deformability is not taken into account). The reconstruction of rotations or torsions is well known in beam-spline structural models. The reader interested in the topic of recovering rotations for aeroelastic analysis is

Index	Chord	Span
1	Linear	Linear
2	Linear	Half cosine
3	Cosine	Half cosine
4	Cosine	Cosine

Table 5.31: Available aerodynamic mesh types

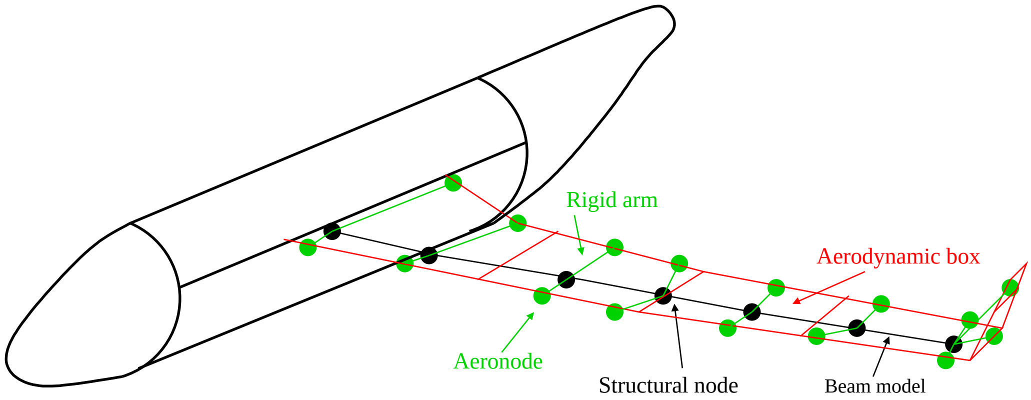


Figure 5.10: Aeronodes for aeroelastic interpolation

referred to [1] and its references.
In order to use the general purpose methods for all the types of elements available in the code, SMARTCAD introduces slave *aeronodes* as sketched in Figure 5.10.
Each aeronode has the following features:

- it is defined by means of the GRID card introduced in Section 5.3.2.
- no degree of freedom is associated;
- a master node must be defined in order to recover the motion of the slave through the RBE0 card (see Section 5.5.1);
- each slave undergo a rigid-body motion from its master, exploiting the assumption of initively rigid section made when beam model is used;
- it is only used for spatial coupling purposes through the SET1 card.

Master node definition for aeronodes

The card RBE0 provides the dependency information for each aeronode to its master (see Table 5.32). For each master node M a list of its slave nodes S which undergo rigid body motion during the numeric analysis. When the code determines from the structural solution the displacements u_M and the variation Δg_M of the Gibbs-Rodriguez parameters of the master node, the displacement u_S of each its aeronode, is recovered by the following relation:

$$u_S = u_M + \mathbf{R}(\Delta g_M)\, O_{SM}$$

where O_{SM} is the offset between the aeronode and its master and $\mathbf{R}$ the rotation matrix.

RBE0	ID	M	S1	S2	S3	S4	S5	S6	
	S7	S8	etc...						

ID	Element identifier
M	Master node identifier
S_i	Slave node identifiers

Table 5.32: RBE0 master definition for aeronodes

5.5.2 Interpolation node sets

The card SET1 (see Table 5.33) is used to defined an interpolation set of nodes which is used by one of the available spatial coupling method to transfer data between structural and aerodynamic meshes. The set defined can be composed of common structural nodes and aeronodes (see Section 5.5.1) if beam elements are used. It is up to the user to reasonably define a structural set which can faithfully guarantee a correct and physical interpolation process. An interpolation set can be used multiple times for the many aerodynamic boxes. Considering Figure 5.10, a simple example would consist in defining a structural set composed by all the structural nodes and aeronodes for the semi-wing and the use it for the interpolation of all the spanwise aerodynamic boxes.

SET1	ID	N1	N2	N3	N4	N5	N6	N7	
	N8	N9	etc...						

ID	Set identifier
N_i	Nodes defining the interpolation set (structural nodes and aeronodes)

Table 5.33: SET1 interpolation set definition

5.5.3 Interpolation parameters input

SMARTCAD provides three methods for the spatial coupling purposes: a Moving Least Squares (MLS) [21] a Radial Basis Function (RBF) [22] and a classic spline based on beam functions. All the methods lead to the construction of a sparse *interpolation matrix* **H** which is used to transfer a data field u_s from the structural mesh to the aerodynamic mesh, thus recovering an interpolated field u_a:

$$u_a = \mathbf{H}\, u_s$$

The reversed transfer is realized simply by the transpose of the interpolation matrix:

$$u_s = \mathbf{H}^T\, u_a$$

The interpolation process may be carried out several times for each aerodynamic panel belonging to the box under to be considered. On the contrary of what usually happens for CFD solvers, many elements need to be coupled to the structural mesh when lifting surface methods are adopted. Each box is composed of panels made by nodes, having a collocation point and vortex or doublet lines. The following elements are interpolated:

- panel nodes to update the mesh following structural displacements (only for Vortex Lattice, for the Doublet Lattice this step is carried out just for illustration purposes);

- collocation points to correctly set no-flow boundary conditions;
- vortex/doublet corners to follow correctly structural displacements (vortex corners in the wake are simply moved to follow free-stream conditions).

All these steps are automatically performed by the code, simply using the same interpolation parameters for each of the three components.

Moving Least Squares (MLS) input parameters

The SPLINE3 card (see Table 5.34) is used to specify spatial coupling parameters for a list of panel belonging to an aerodynamic box AID using MLS method. In order to correctly used this tool on Unix machines, an external library is required as discussed in Section 1.1.2. The card can be used to set interpolation parameter even for a restricted set of panels belonging to the patch. Usually all the box-panels are interpolated with the same settings.

SPLINE3	ID	AID	P1	P2	SET	POLY	W	NP	
	RMX	CND							

ID	Interpolation card identifier
AID	Aerodynamic box identifier
P1,P2	First and last panel index interpolated using this card Panels are numbered starting from inboard, with chordwise growing index Usually P1=1 and P2=NTOT, where NTOT is the total number of panels belonging to the box
SET	Structural SET1 card identifier
POLY	Polynomial order used (1=Linear, 2=Quadratic)
W	Weight function to be used (see Table 5.35)
NP	Number of structural nodes used to build the compact suport
RMX	(OPTIONAL) Maximum radius used for the compact suport
CND	(OPTIONAL) Condition number for Singular Value Decomposition algorithm. Useful to avoid ill-conditioned problems

Table 5.34: SPLINE3 MLS interpolation parameters definition

Radial Basis Function (RBF) input parameters

The SPLINE2 card (see Table 5.36) is used to specify spatial coupling parameters for a list of panel belonging to an aerodynamic box AID using RBF method. As it happens with

Weight	**Type**	$\Phi(\delta)$
1	Wendland C^0	$(1-\delta)^2$
2	Wendland C^2	$(1-\delta)^4\cdot(4\,\delta+1)$
3	Wendland C^4	$(1-\delta)^6\cdot(35/3\,\delta^2+18/3\,\delta+1)$
4	Wendland C^6	$(1-\delta)^8\cdot(32\,\delta^3+25\,\delta^2+8\,\delta+1)$

Table 5.35: Weight functions available for the MLS method.

SPLINE3 card, SPLINE2 can be used to set interpolation parameter even for a restricted set of panels belonging to the patch. Usually all the box-panels are interpolated with the same settings.

SPLINE2	ID	AID	P1	P2	SET	W	RMX	CND	

ID	Interpolation card identifier
AID	Aerodynamic box identifier
P1,P2	First and last panel index interpolated using this card Panels are numbered starting from inboard, with chordwise growing index Usually P1=1 and P2=NTOT, where NTOT is the total number of panels belonging to the box
SET	Structural SET1 card identifier
W	Weight function to be used (see Table 5.37)
RMX	Compact suport radius used. It is not used by weight 1, 2 (this method is the same available in NASTRAN©), 3
CND	(OPTIONAL) Condition number for Singular Value Decomposition algorithm. Useful to avoid ill-conditioned problems

Table 5.36: SPLINE2 RBF interpolation parameters definition

Beam spline

The SPLINE1 card (see Table 5.38) is used to specify spatial coupling parameters for a list of panel belonging to an aerodynamic box AID using the beam spline method. This is the default method used by GUESS when generating the file for SMARTCAD. SPLINE1 cards will be automatically created for each aerodynamic box. The beam spline is very robust, less computationally demanding and perfectly suits with the assumptions of the bar elements used which have no offset for the the elastic axis. Compared to MLS and RBF methods, deflections along aerodynamic mesh and forces and moments along the structural mesh can be directly recovered, respectively from structural displacements and aerodynamic panel forces.
The syntax is fairly similar to SPLINE2 and SPLINE3 cases (see Table 5.38). N1 and N2 are optional parameters. If N1 is not provided, the first node in SET1 will be used. If N2

Weight	**Type**	$\Phi(\delta)$, $\Phi(r)$
1	Volume Spline	r
2	Thin Plate Spline	$r\,log(r)$
3	Gaussian	e^{-r}
4	Euclid Hat	$\pi \cdot ((1/12r^3) - (r_{max}^2 \cdot r) + (4/3r_{max}^3))$
5	Wendland C^0	$(1 - \delta)^2$
6	Wendland C^2	$(1 - \delta)^4 \cdot (4\,\delta + 1)$
7	Wendland C^4	$(1 - \delta)^6 \cdot (35/3\,\delta^2 + 18/3\,\delta + 1)$
8	Wendland C^6	$(1 - \delta)^8 \cdot (32\,\delta^3 + 25\,\delta^2 + 8\,\delta + 1)$

Table 5.37: Weight functions available for the RBF method.

SPLINE1	ID	AID	P1	P2	SET	N1	N2	CND	

ID	Interpolation card identifier
AID	Aerodynamic box identifier
P1,P2	First and last panel index interpolated using this card Panels are numbered starting from inboard, with chordwise growing index Usually P1=1 and P2=NTOT, where NTOT is the total number of panels belonging to the box
SET	Structural SET1 card identifier
N1	(OPTIONAL) Identifier of one node in SET1 to be used as origin for the spline reference system
N2	(OPTIONAL) Identifier of one node in SET1 to be used to define the plane yz of the spline reference system
CND	(OPTIONAL) Condition number for Singular Value Decomposition algorithm. Useful to avoid ill-conditioned problems

Table 5.38: SPLINE1 RBF interpolation parameters definition

SOL	**Description**
101	Static linear
103	Vibration modes
144	Static linear aeroelastics
145	Flutter
600	Static nonlinear
644	Static non-linear aeroelastics
700	Rigid Vortex Lattice

Table 5.39: Available solvers in NeoCASS

is not provided, the last node in SET1 will be used. A reference frame $\bar{x}\bar{y}\bar{z}$ is required for the beam spline. N1 defines the origin. N2 is used to define the plane $\bar{y}\bar{z}$. The spline axis $\bar{y}$ goes from N1 to N2. The axis $\bar{z}$ is found through the cross product of the absolute x axis and the spline $\bar{y}$ axis. Finally, the spline $\bar{x}$ axis is determined through a cross product between $\bar{z}$ and $\bar{y}$ axes.

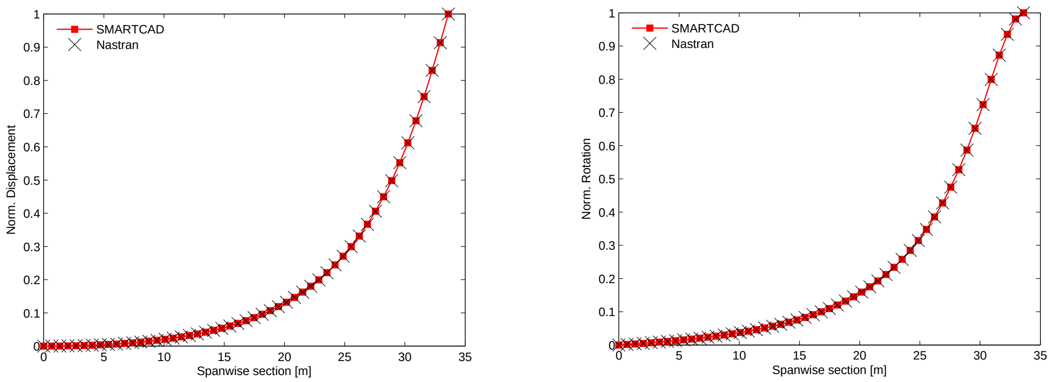
5.6 Solvers

SOL	SID								

SID	Solver identifier

Table 5.40: SOL solver definition

SMARTCAD provides different solvers based on the structural and aerodynamic components developed. Different kinds of analysis can be carried out from simple structural and aerodynamic solution up to steady aeroelastic analysis and flutter tracking. Each solver has its own identifier as reported in Table 5.39. The card SOL (see Table



(a) Normalized vertical displacement

(b) Normalized out of plane rotation

Figure 5.11: Static linear results for B-747 semiwing under concentrated tip force

5.40) specifies which solver the user is intented to use for the simulation. Moreover, SMARTCAD checks for all the cards given and states which other solver respect to the one defined by SOL can be used. Sometimes it is indeed possible to run different analyses even if the file is not specifically produced for that kind of simulation. For example, considering a flutter solution file (SOL=145), the user is allowed to run the vibration mode solver (SOL=103) since the parameters regarding this solver are available. Again, considering a static aeroelastic file (SOL=144 or 644) the user can only run a rigid Vortex Lattice analysis (SOL=700).

5.6.1 Static linear/nonlinear structural analysis

Treatment of rigid body modes

SUPORT

Static linear structural analysis validation

A simple case is considered to validate the static linear solver. The semi-wing generated by GUESS module for the Boeing-747 aircraft i s clamped at its root and subjected to a static vertical load at the tip node. The same discretization is used, despite SMARTCAD adopts a three-beam node model. The results are compared with NASTRAN© and are in excellent agreement. Figure 5.11 shows respectively the normalized vertical displacement and x-axis rotation of the beam-nodes. These two field of the generalized displacement are indeed predominant in this simple test-case. The displacements of the middle-side beam node for the three beam model are not plotted for clarity.

5.6.2 Eigenvalue analysis

The usual first step in performing a dynamic analysis is determining the natural frequencies and mode shapes of the structure with damping neglected (normal modes of vibration). These results characterize the basic dynamic behavior of the structure under small disturbances and are an indication of how the structure will respond to dynamic loading. An eigenvalue analisys determines:

- the *natural frequencies* of a structure, the frequencies at which the structure naturally tends to vibrate;
- the *modal shapes* of the structure at a specific natural frequency.

Natural frequencies and mode shapes are functions of the structural stiffness, inertial distribution and boundary conditions if the structure is constrained at some points. The computation of the two previous information is commonly represented by an eigenvalue problem, where the eigenvalues (real, since no damping is usually considered) represent the natual frequency and the associated eigenvector which represent the mode shape. An overall understanding of normal modes analysis as well as knowledge of the natural frequencies and mode shapes is important for all types of dynamic analysis, ranging from dynamic response to excitations to prevent resonances up to aeroelastic response and flutter calculation. Using vibration modes in Computational Structural Dynamics (CSD) is a common choice, since it usually turns out to be a good approximations for classic mechanical structures and it brings to simplified models with far fewer degress of freedom respect to using a full model. Specifically for the aeroelastic analysis, using structural normal modes as generalised coordinates is a common practise in aeroelasticity since the calculation of aerodynamic loads is enourmously simplified and represented by a transfer matrix for the generalized forces. Moreover, this simple approach is extremely convenient in the preliminary phases of the aircraft design, when many structural details are not defined yet and several calculations are required. Finally, modal analysis can determine the goodness and accuracy of a numerical model by a validation to experimental Ground Vibration Tests (GVT). The normal-modes solver export a structural modal

EIGR	ID		F1	F2		ND			
	NRM	G	C						

ID	Card identifier
G	Node identifier used to normalize modes with POINT normalization
C	Generalized displacement component used to normalize modes with POINT normalization
NRM	Normalization criterion: MASS (default value), MAX, POINT MASS: normalize to unit generalized mass matrix MAX: normalize to unit value of the largest component in the analysis set POINT: normalize to unit value of the C component for node G
F1,F2	Frequency (Hz) range of interest. All the eigenmodes in this range will be used Default values are: F1=0.0, F2=∞
ND	Desidered number of modes. If null, all the modes in the frequency range will be used

Table 5.41: EIGR eigenvalue solution parameters

parameters file written in FFA-format ready to be used by the CFD code Edge[8]. Edge has aeroelastic features which enable to carry out coupled dynamic response analysis which are presently based only on a modal structural model. SMARTCAD can be then easily used for high-fidelity aeroelasticity by exporting its structural eigendata to investigate aeroelastic phenomena governed by non-linear flow-field conditions [5]. When flutter solution is sought, the user can rule the list of modes to be included in the modal base by directly specifying the indeces of the modes in the card MSELECT (see Table 5.42. The index refers to the mode extracted by a call to the previously presented EIGR card.

MSELECT	ID	M1	M2	M3	M4	M5	M6	M7	
	M8	etc							

ID	Card identifier
Mi	Mode index used

Table 5.42: MSELECT mode selection for flutter analysis

5.6.3 Validation on a cantilever beam model

A simple case of a fixed free aluminium cantilever beam with circular section is used to validate the normal-modes solver. This example is taken from NASTRAN© guide [19]. The structural properties are reported in Figure 5.12, where structural inertias are slightly modified respect to their nominal value $I_1 = I_2 = 3.0E^{-8}$ to account for manufacturing tolerances. Besides the volumetric density of the material ρ_w, a second density per unit length ρ_{NS} for non-structural mass is considered. Since weight densities are considered (respectively N/m^2 and N/m), the WTMASS parameter (see table 5.24) is required to convert their values to mass densities. Its value is $1/g = 1/9.81 = 0.102 sec^2/m$. Two families of modes are determined: displacements along y-axis are controlled by the inertial term I_1 and, since the beam is free to displace along z-axis, similar modes, controlled by the term I_2, occour in the that direction. Table 5.43 briefly summarizes the results obtained by SMARTCAD compared to NASTRAN© for different mesh discretizazions. Good agreement between the two solvers is found, despite SMARTCAD adopts a three-node beam with parabolic shape-functions and NASTRAN© a two-node beam with linear shape functions. Figure 5.13 shows respectively the first and second bending modes in the plane xy plane. Similar mode shapes are obtained in the xz plane.
A second comparison is carried out for the same cantilever beam with nominal structural inertias. Again, good agreement between the two solvers is found. Table 5.44 compares SMARTCAD with analytical results and with the results obtained by NASTRAN© with a consistent C or lumped L mass matrix (SMARTCAD only adopts a lumped mass matrix formulation). For this second case, only even modes are reported since due to the symmetry of the section, repeated roots are determined.

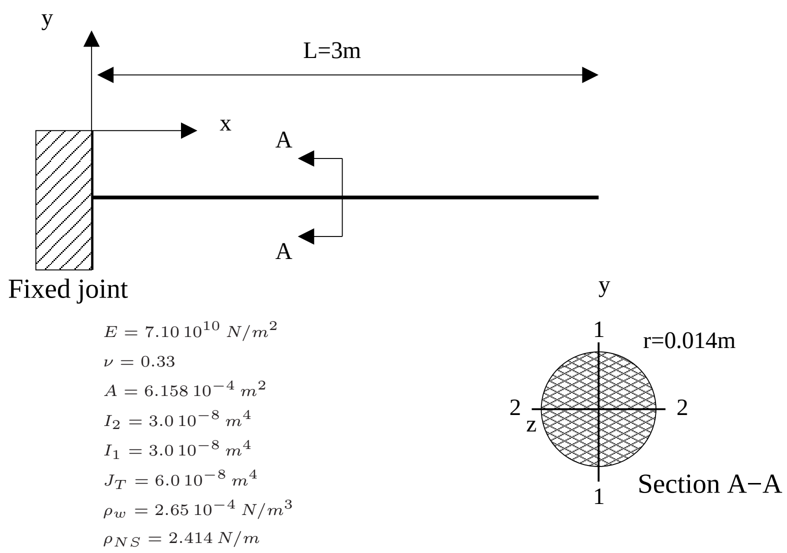
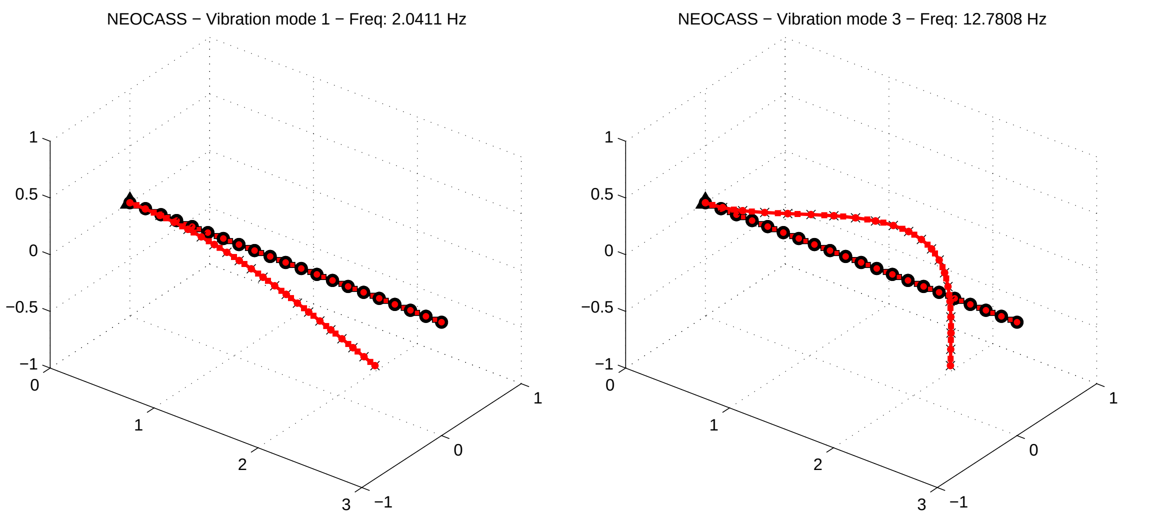


Figure 5.12: Cantilever beam model for normal-modes validation

M	**NAS** [Hz] 10 beams	**SMA** [Hz] 3 beams	**SMA** [Hz] 5 beams	**SMA** [Hz] 10 beams
1	2.032	2.042	2.041	2.041
2	2.101	2.111	2.111	2.110
3	12.591	12.702	12.756	12.781
4	13.018	13.132	13.188	13.214
5	34.902	34.529	35.508	35.722
6	36.096	35.699	36.712	36.933

Table 5.43: Validation results for the cantilever beam



(a) First bending mode

(b) Second bending mode

Figure 5.13: Validation of modes for a cantilever beam (10 beam elements) with manifacturing tolerances: modes in xy plane

M	**Theory** [Hz]	**NAS C** [Hz] 10 beams	**NAS L** [Hz] 10 beams	**SMA** [Hz] 3 beams	**SMA** [Hz] 5 beams	**SMA** [Hz] 10 beams
1	2.076	2.076	2.066	2.077	2.076	2.076
3	13.010	13.010	12.806	12.919	12.974	12.999
5	36.428	36.437	35.499	35.118	36.115	36.333
7	71.384	71.451	68.838	76.969	70.101	71.004

Table 5.44: Validation results for the cantilever beam

5.6.4 Flutter analysis

MKAERO1	M1	M2	M3	M4	M5	M6	M7	M8	
	k1	k2	k3	k4	k5	k6	k7	k8	

M_i	Mach number list
k_i	Reduced frequency number list The maximum value of k should be less than one quarter of the number of boxes on a representative chord. $MAX(k) < \frac{CHD}{4\Delta x\, MAX(k)} < \frac{CHD}{4\Delta x}$, where CHD is the reference chord and Δx a typical box chord length

Table 5.45: MKAERO1 aerodynamic transfer matrix sample points

AERO	SREF	MXS	CHD	RHO	SXZ	SXY	ORD	NS	BREF

SREF	Reference surface
MXS	Max speed used in flutter tracking algorithm
CHD	Reference length for reduced frequency k Actually half of its value is used as classically done in aeroelasticity: $k = \frac{\omega CHD}{2V_\infty}$
RHO	Density for flutter tracking solver If RHO=1.225, flutter envelope is produced instead of v-g diagrams at constant altitude
SXZ	Symmetry key for the aero coordinate x-z (vertical) plane SXZ = 1 for symmetry, 0 for no symmetry, and -1 for antisymmetry
SXY	Symmetry key for the aero coordinate x-y (horizontal) plane. Useful for ground effect modelling. Not available yet
ORD	Doublet lattice kernel order used: ORD=1 for linear kernel, ORD=2 for quadratic kernel (default)
NS	Number of velocity steps taken in flutter tracking in [0:MXS] range
BREF	Reference span

Table 5.46: AERO flutter solver parameters

5.6.5 Flutter validation: AGARD445.6 wing

A well-known three-dimensional standard aeroelastic configuration is considered here to validate the whole procedure: the AGARD 445.6 weakened wing which was tested in the

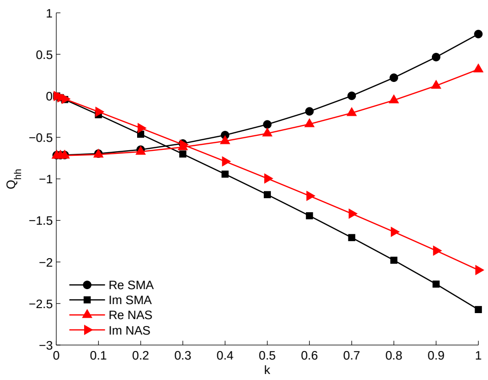
Mach	$V_{F_{\mathrm{Exp}}}$ m/s	$V_{F_{\mathrm{NAS}}}$ m/s	$V_{F_{\mathrm{SMA}}}$ m/s	ω_{Exp} Hz	ω_{NAS} Hz	ω_{SMA} Hz
0.678	231.37	237.94	235.86	13.89	14.82	14.30
0.960	309.00	337.07	324.23	17.98	20.53	20.06

Table 5.47: Comparison of the flutter velocity V_F and frequency ω_F for the AGARD 445.6 aeroelastic benchmark

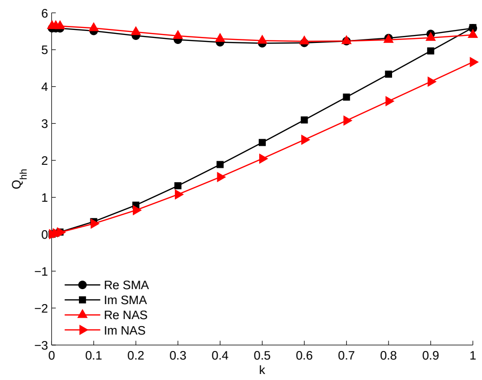
transonic wind-tunnel at NASA Langley. The wing semispan model is made of laminated mahogany, with NACA 65A004 airfoil, a quarter-chord sweep angle of 45 deg. , an aspect ratio of 1.65 and a taper ratio of 0.66. To reduce the stiffness, the wing was weakened by holes drilled through it and filled with foam. In this section we refer to the weakened model number 3 since all flow conditions (subsonic, transonic and supersonic) were tested. Further data concerning the models, test setups and conditions are reported in Ref. [25]. The structural model is represented by the first four normal undamped modes which are provided to SMARTCAD by altering common procedures (see Section 5.3.10 for how to use MSELECT and the parameter EIG_FILE). The first two modes are primarily involved in the flutter mechanism, identified respectively as first bending and first torsional modes. Since no structural g-damping (defined as twice the critical damping ratio of the mode) is available for the wing model, flutter analysis are carried out without any damping matrix. Two flight conditions are investigated by means of SMARTCAD and are compared, as summarized in Table 5.47, with both experimental and NASTRAN© results. The first condition at a Mach number M_∞ equal to 0.678 is subsonic, while the second one at a Mach number M_∞ equal to 0.960 is transonic. This condition is investigated despite the linearized potential theories overestimate the flutter velocity in this regime. The user interested in the improvements for transonic flutter by means of CFD methods is referred to [5]. The same aerodynamic discretization (10 chordwise and 10 spanwise evenly spaced panels) and reduced frequencies for the aerodynamic transfer matrix are used within the two solvers which predict almost similar results. For this test-case, SMARTCAD provides better agreement with the experiment respect to NASTRAN© . The same aerodynamic discretization and reduced frequencies for the aerodynamic transfer matrix are used of-course within the two solvers. The difference may rely:

- in the spatial coupling method; nevertheless the results have proved to be almost independent of the the adopted scheme;
- in minor differences for the Doublet Lattice implementation as shown in Figures 5.14 and 5.15.
- in the flutter-tracking method;

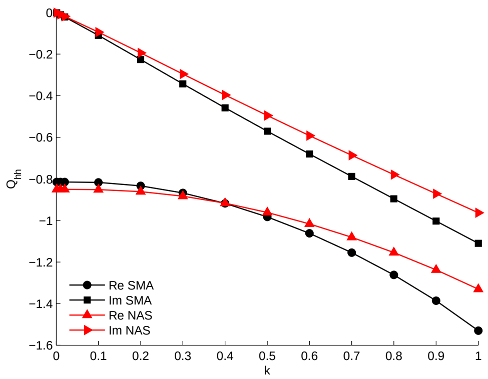
The major discrepancy between the two codes is found for the second condition. Since the slope of the damping diagram is not abrupt close to the flutter condition, a wrong evaluation of its pendence may lead to considerably wrong predictions. Figures 5.16 and 5.17 shows the flutter diagrams predicted by SMARTCAD for the two flight points investigated. The flutter onset point corresponds to the zero-crossing velocity in the g-damping plot of Figures 5.16(b) and 5.17(b). At $M_\infty = 0.678$ there is a sudden increment of the g-damping of the first bending mode which tend to coalesce with the first torsional



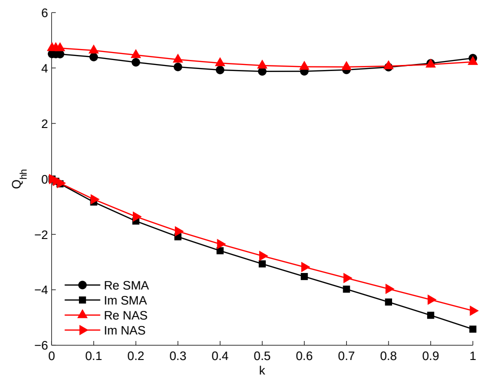
(a) Qhh_{11}



(b) Qhh_{12}



(c) Qhh_{21}



(d) Qhh_{22}

Figure 5.14: Aerodynamic transfer matrix for the AGARD 445.6 wing, M_∞=0.678

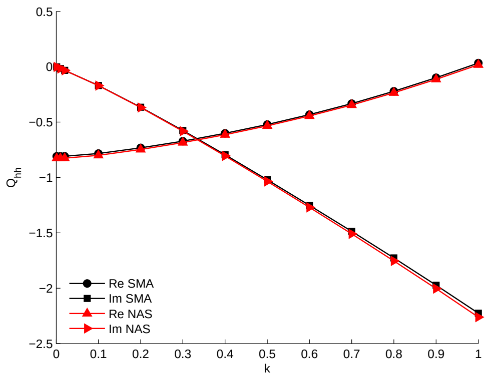
mode. At $M_\infty = 0.960$ the increment of the g-damping is more regular and no coalescence manifests.

5.6.6 Static linear/nonlinear aeroelastic analysis

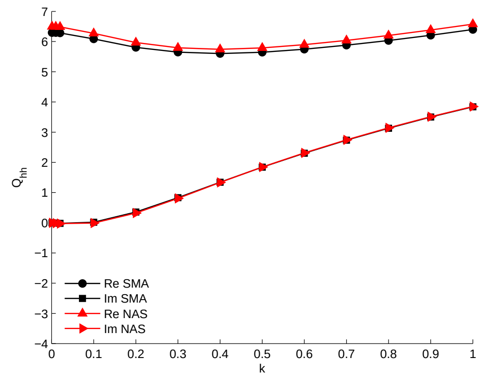
AEROS			CHD	SPN	SRF	SXZ	SXY	HGT	

CHD	Reference length
SPN	Reference span
SRF	Reference surface
SXZ	Symmetry key for the aero coordinate x-z (vertical) plane SXZ = 1 for symmetry, 0 for no symmetry. No antisymmetry is allowed.
SXY	Symmetry key for the aero coordinate x-y (horizontal) plane SXY = 1 for ground effect, 0 for no simmetry. No antisymmetry is allowed
HGT	Height from the ground if horizontal simmetry enabled

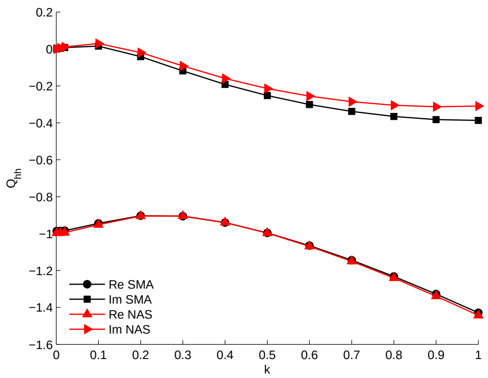
Table 5.48: AEROS static aeroelastic and rigid aerodynamics parameters



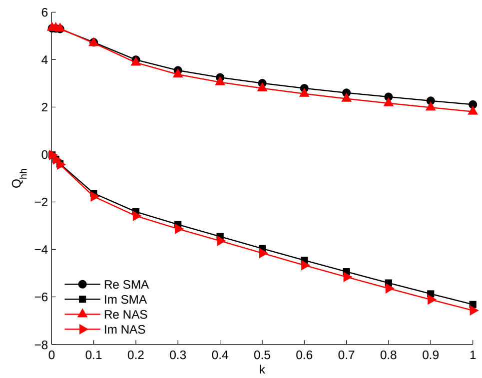
(a) Qhh_{11}



(b) Qhh_{12}

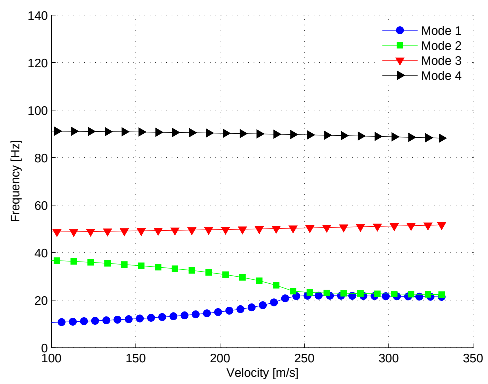


(c) Qhh_{21}

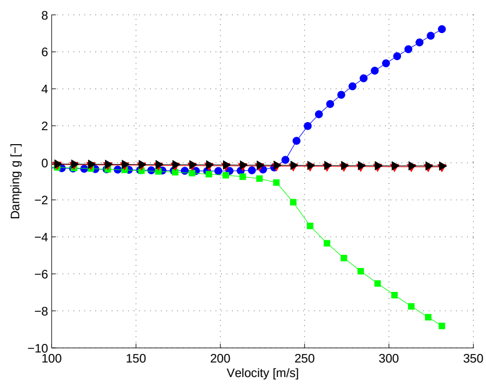


(d) Qhh_{22}

Figure 5.15: Aerodynamic transfer matrix for the AGARD 445.6 wing, M_{∞}=0.960

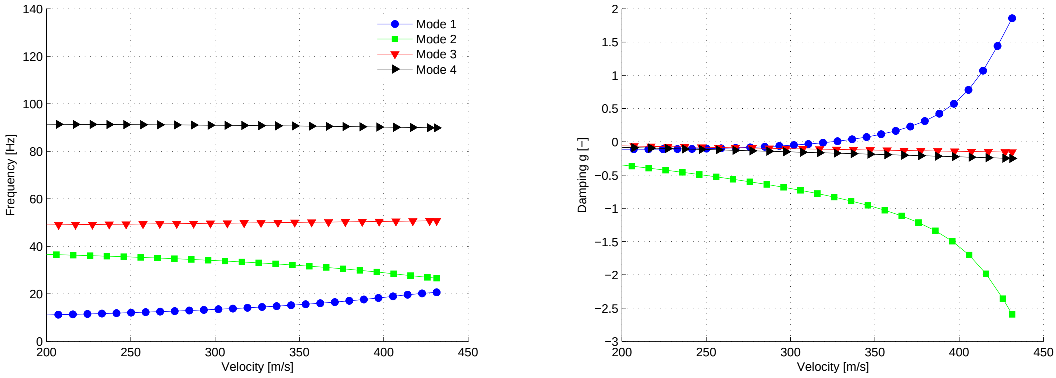


(a) V-f



(b) V-g

Figure 5.16: Flutter diagrams for the AGARD 445.6 wing, M_{∞}=0.678



(a) V-f

(b) V-g

Figure 5.17: Flutter diagrams for the AGARD 445.6 wing, M_∞=0.960

TRIM	ID	MCH	ALT	P1	V1	P2	V2		
	P3	V3	etc...						

ID	Trim set identifier
MCH	Flight Mach number
ALT	Flight altitude (meters)
P_i	Trim parameters label (see Table 5.50)
V_i	Trim parameters value

Table 5.49: TRIM flight condition parameters

5.6.7 Aerodynamic rigid analysis

SMARTCAD provides an aerodynamic solver based on the Vortex Lattice Method (VLM) derived from TORNADO. Aerodynamic performances for a rigid configuration can be evaluated by means of this solver. The same solver is used for aeroelastic analysis and it is called automatically by the different structural solvers in order to get aerodynamic forces. Since the structural model available are non-linear and large displacements can be modelled, the mesh is simply update to correctly follow the new structural deformed shape. The aerodynamic solver developed enables to exploit geometry simmetry to reduce the algebraic system for γ circulation to the minimum possible size. The general problem for the left L and right R side of the aircraft reads:

$$\begin{bmatrix} \mathbf{A}_{LL} & \mathbf{A}_{LR} \\ \mathbf{A}_{RL} & \mathbf{A}_{RR} \end{bmatrix} \begin{Bmatrix} \gamma_L \\ \gamma_R \end{Bmatrix} = \begin{Bmatrix} \alpha_L \\ \alpha_R \end{Bmatrix} \tag{5.3}$$

where $\mathbf{A}_{ij}$ represents the induced velocity matrix due to horse-shoe vorticies of the side j on the side i, α_i the downwash boundary condition on the side i and γ_i the circulation on the panels of side i. Thus, if the problem has simmetry along the vertical plane, $\gamma_L = \gamma_R$ and only half of the model is required. For example if the right side od the model is used:

$$(\mathbf{A}_{RR} + \mathbf{A}_{LR})\,\gamma_R = \mathbf{A}_{F_V} = \alpha_R \tag{5.4}$$

By doing this way, the resulting algebraic system for the full model is governed by the matrix $\mathbf{A}_{Vsimm}$ which is characterized by half of the total panels which should be used.

Label	Description
ANGLEA	Angle of attack (deg)
SIDES	Sideslip angle (deg)
ROLL	Roll rate (rad/s)
PITCH	Pitch rate (rad/s)
YAW	Yaw rate (rad/s)
URDD1	Acceleration along x-axis (m/s^2)
URDD2	Acceleration along y-axis (m/s^2)
URDD3	Acceleration along z-axis (m/s^2)
URDD4	Angular acceleration around x-axis (rad/s^2)
URDD5	Angular acceleration around y-axis (rad/s^2)
URDD6	Angular acceleration around z-axis (rad/s^2)

Table 5.50: Flight mechanics trim parameters

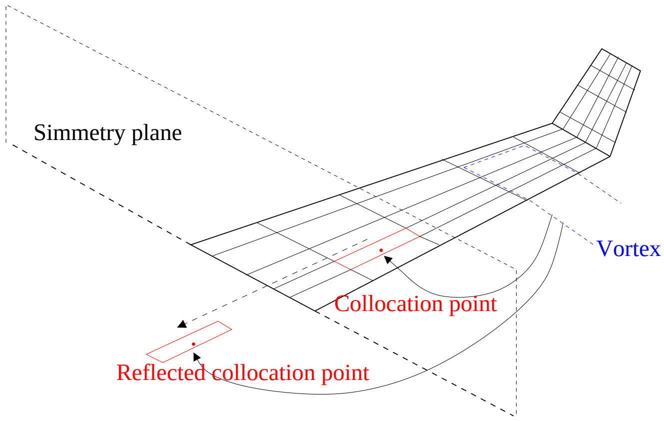


Figure 5.18: Application of vertical aerodynamic simmetry

Figure 5.18 shows how the simmetry condition works, by simply considering the influence of the modelled vortices on the mirrored collocation points. This term is exactly the same as considering the influence of the mirrored vorticies on the modelled collocation points. The same principle can be applied for the case of both horizontal and vertical simmetry where the modelled upper aerodynamic geometry U is reflected along the simmetry plane to create a lower model L. In this case if the problem has also vertical simmetry it is possible to model an aircraft in ground effect with only one quarth of the degrees of freedom. The general problem statement:

$$\begin{bmatrix} \mathbf{A}_{UL-UL} & \mathbf{A}_{UL-UR} & \mathbf{A}_{UL-DL} & \mathbf{A}_{UL-DR} \\ \mathbf{A}_{UR-UL} & \mathbf{A}_{UR-UR} & \mathbf{A}_{UR-DL} & \mathbf{A}_{UR-DR} \\ \mathbf{A}_{DL-UL} & \mathbf{A}_{DL-UR} & \mathbf{A}_{DL-DL} & \mathbf{A}_{DL-DR} \\ \mathbf{A}_{DR-UL} & \mathbf{A}_{DR-UR} & \mathbf{A}_{DR-DL} & \mathbf{A}_{DR-DR} \end{bmatrix} \begin{Bmatrix} \gamma_{UL} \\ \gamma_{UR} \\ \gamma_{DL} \\ \gamma_{DR} \end{Bmatrix} = \begin{Bmatrix} \alpha_{UL} \\ \alpha_{UR} \\ \alpha_{DL} \\ \alpha_{DR} \end{Bmatrix} \tag{5.5}$$

can be simplified, by virtue of horizontal and vertical simmetry, to:

$$(\mathbf{A}_{UR-UL} + A_{UR-UR} + A_{UR-DL} + A_{UR-DR})\,\gamma_{UR} = \mathbf{A}_{F_{VH}} = \alpha_{UR} \tag{5.6}$$

thus considerably reducing computation costs for system solution. As application, the

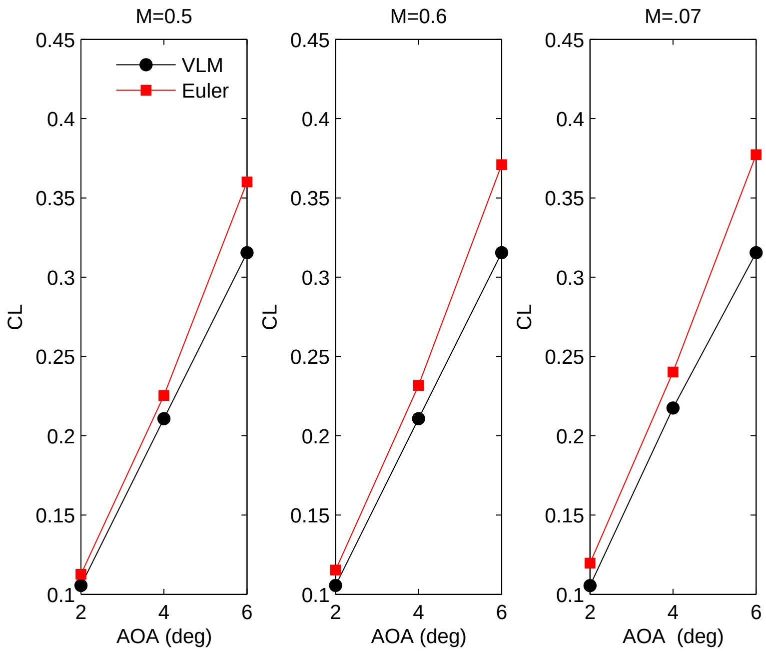
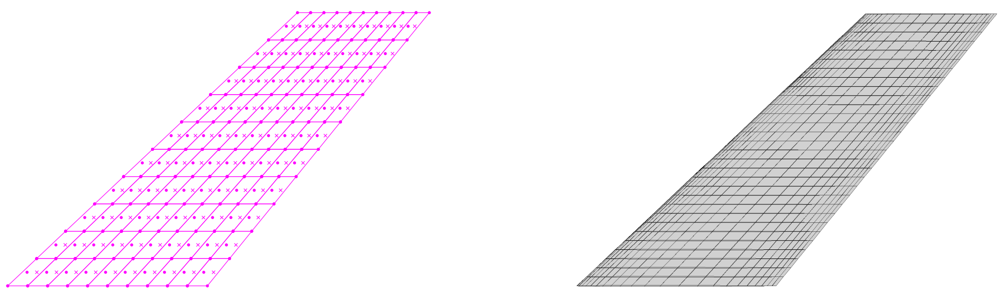


Figure 5.19: Comparison of steady aerodynamic results for the AGARD 445.6 wing



(a) VLM mesh

(b) CFD mesh

Figure 5.20: Mesh discretization adopted for the AGARD 445.6 wing

AGARD 445.6 wing is considered for three subsonic flight Mach numbers $M_\infty = 0.5, 0.6, 0.7$. Compressibility effects are simply accounted for in the VLM by means of the *Prantl-Glauert* correction $\beta = \sqrt{1 - M_\infty^2}$. As reported in Figure 5.19, lift coefficients are compared to CFD inviscid computations on a rather coarse structured mesh. Figure 5.20 shows the two meshes adopted. Vertical simmetry boundary condition is applied both in the VLM and the CFD model. The result obtained agree quite well, considering the simplifications introduced by the VLM:

- no thickness is accounted for and the wing is modelled as a flat plate since the airfoil used is simmetric;
- the wake is considered as fixed;
- the wing has a considerable sweep angle (45 deg).

Chapter 6

NeoCASS Graphical User Interface (GUI)

GUI interface to NeoCASS code is based on four main panels, i.e.:

- File;
- Settings;
- Run;
- Results;

and two sub-panels, i.e.:

- REFERENCE_Settings;
- ANALYSIS_Settings.

Using these user-friendly menu, the user could introduce all parameters requested by different analysis modules, while the order of the GUI panels well reproduces the typical analysis sequence. In the following, the role of each GUI panel will be briefly explained.

6.1 NeoCASS GUI Panel FILE

The GUI Panel File is the first one that appears when NeoCASS code is invoked. By means of this panel it is possible to perform the following actions:

- Read input files requested by GUESS , run GUESS code and generate the aircraft stick model;
- Add Reference Values for geometrical and aerodynamic parameters;
- Select type of analysis to be run and related input/output data;
- Open a previously saved NeoCASS data base (Matlab format);

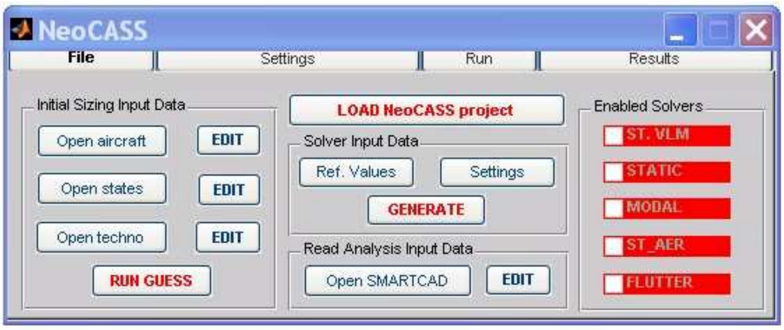


Figure 6.1: GUI interface: File panel

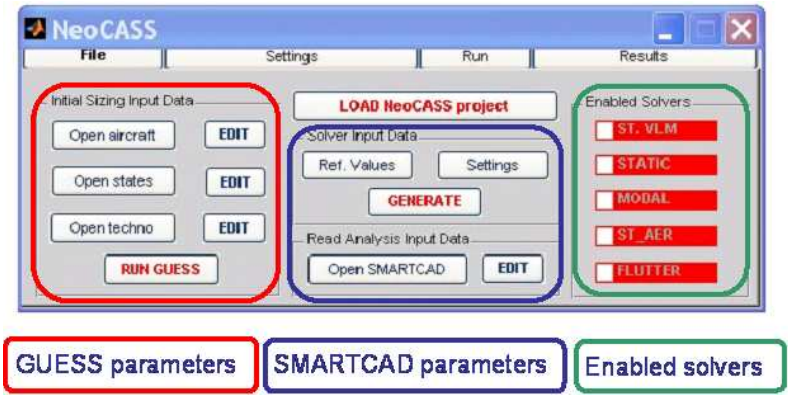


Figure 6.2: GUI interface: File panel description

- Save all input data into a new SMARTCAD input file (ascii .dat file);
- Open an already available SMARTCAD input file (ascii .dat file);
- Select among enabled solvers which one must be actually executed.

The GUI Panel File, shown in the following figure 6.11 can be divided into three parts, located on the left, middle and right side of panel, respectively. The first one (on the left) is related to the GUESS code, the second one (in the middle) is related to the definition of SMARTCAD parameters and files, while the third one (on the right) summarizes the type of analyses can be run on the basis of the available input parameters.

Three input files buttons are available in GUESS section panel: the first one must be used to open the `aircraft_geo.xml` file, the second one to open the `states.xml` file while the third to open an already available technological.xml input file.

On the right side of each aforementioned button, three EDIT buttons are available. Clicking on these buttons it is possible to edit the input files and save them. The EDIT programs invoked by these buttons are ones already defined into the `neocass_xml_editor_path.m` and `neocass_text_editor_path.m` setup files. Once completed the input files phase it is possible to run GUESS, simply pressing the *RUN GUESS* button: before GUESS code is started, an output file where GUESS code will save the stick model in SMARTCAD format (ascii .dat file) is requested.

The SMARTCAD section of GUI panel File allows to perform two different kind of operations. In case the output file resulting from GUESS code, including the aircraft stick

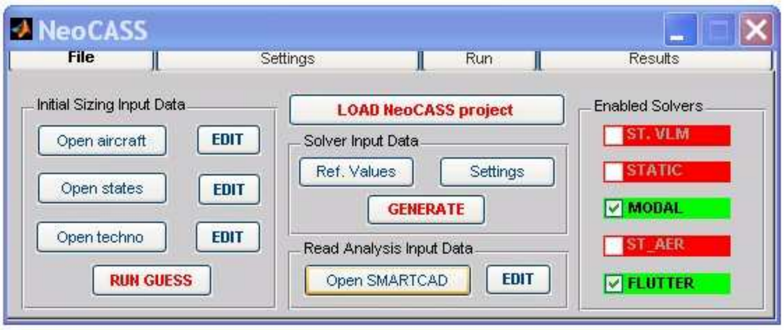


Figure 6.3: GUI interface: File Panel with enabled solvers

model (.dat file) is available, it is possible to use the subpanels REFERENCE_Settings and ANALYSIS_Settings to add to the .dat file all cards related to the analysis that must be performed. Otherwise, if a complete SMARTCAD file is already available, it is possible to simply open it, pressing the button *Open SMARTCAD*. Even in this case, an EDIT button is available to edit the SMARTCAD .dat file. After reading of SMARTCAD file, the Enabled solvers section of GUI Panel File summarizes the kind of analysis that can be performed, on the basis of cards included into the file. Figure 6.3 shows one example where Modal and Flutter analyses are enabled.

In case a previous NeoCASS analysis has been run and all data have been saved into the ad hoc MATLAB© binary file (.mat), it is possible to bypass all the input phases and simply read the complete MATLAB© data base pressing the *LOAD NeoCASS project* button.

The GUI Subpanel REFERENCE_Settings, shown in 6.4, must be used to input reference parameters used for the aerodynamic calculations, i.e.:

- Reference Chord (CREF);
- Reference Span (BREF);
- Reference Surface (SREF);
- Vertical Simmetry (0 Full model, 1 half model);
- Horizontal Simmetry (0 No Ground, 1 Ground effect);
- Height (active if Ground Effect is selected);
- Kernel order for DLM solver (1 Linear, 2 Quadratic).

The GUI Subpanel ANALYSIS_Settings, shown in figure 6.5, must be used to select which kind of analysis must be run and to input the requested parameters. The GUI Subpanel could be divided into three small panels, related to the the following type of analysis:

- Modal Analysis;
- Static Aeroelastic Analysis;

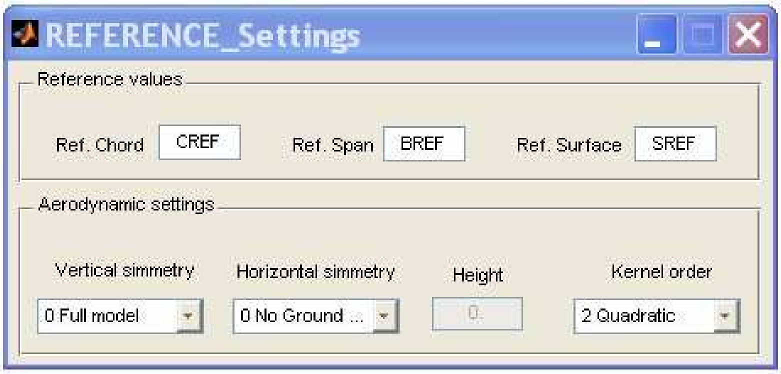


Figure 6.4: GUI interface: Reference Settings panel

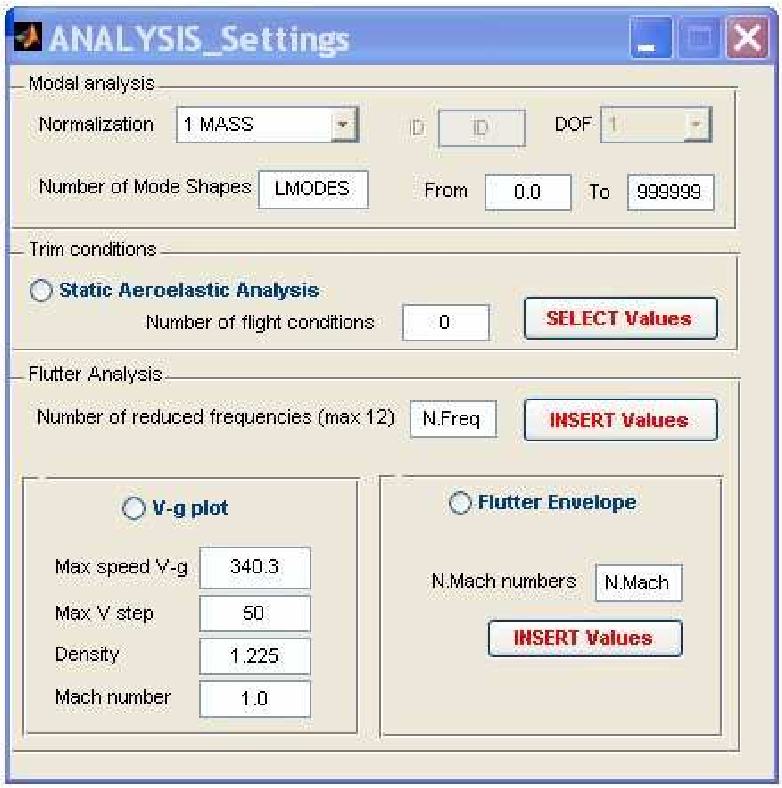


Figure 6.5: GUI interface: Analysis Settings panel

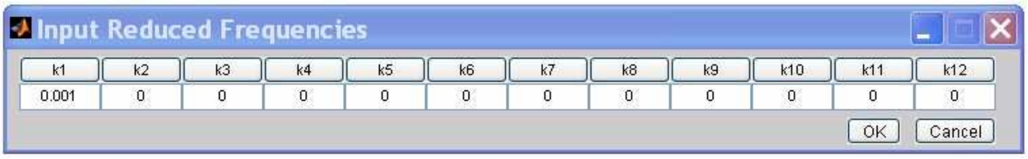


Figure 6.6: GUI interface: input reduced frequencies

- Flutter Analysis.

The parameters that must be provided to run a Modal Analysis are the following:

- Normalization (1 MASS, 2 MAX, 3 POINT): in case a POINT normalization is chosen the user must provide the Grid Point ID and DOF with respect to the normalization is done;
- ID: Grid Identification Number;
- DOF (1,2,3,4,5,6);
- LMODES: Number of modes retained during flutter calculations;
- From - To: an alternative way to define the bandwidth of interest (lower and upper frequencies).

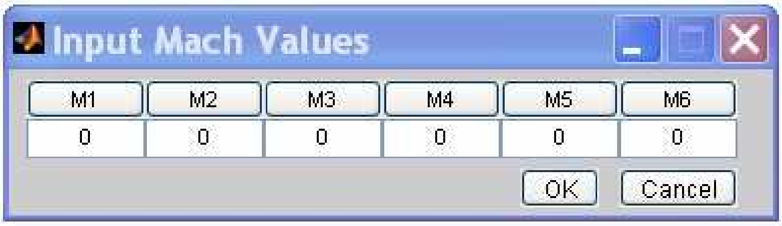
In the actual version of NeoCASS (1.1) the user must choose between Static Aeroelastic Analysis and Flutter Analysis since the two solvers cannot be executed in the same run. This is due to the fact that two different geometry meshes are generated for each solver and different data are needed. Concerning Flutter Analysis, two possibilities are available: V-g plot analysis for a single specified flight condition or flutter envelope for an assigned number of Mach values. The three Analysis Methods radio buttons are exclusively selected: when the user chooses one of them the other two are automatically switched off. When Static Aeroelastic Analysis is selected, two are the requested input parameters:

- Number of Flight Conditions;
- Values for Flight Conditions: when the *Select Values* button is pressed a table appears (see figure 6.7) where the user must insert all flight parameters requested to completely define the flight conditions. In the following version of NeoCASS the flight conditions will be directly recovered from `states.xml` file and the user will be requested just to select ones to be analyzed.

When Flutter Analysis is selected, the user first of all must specify the number of reduced frequencies (max. 12) and insert their values in the Table that appears (see figure 6.6) after pressing the button *INSERT Values*. The minimum reduced frequency is automatically set to 0.001 but can be modified by the user. Then, user must choose between two possibilities: Flutter analysis for a single assigned flight condition, to calculate V-g plot, or Flutter Envelope for an assigned number of Mach values. In the first case (V-g plot) the requested input parameters are:

	ID	MACH	ALT	ANGLEA	SIDES	ROLL	PITCH	YAW	URDD1	URDD2	URDD3	URDD4	URDD5	URDD6
☑	1	0	0	0	0	0	0	0	0	0	0	0	0	0
☑	2	0	0	0	0	0	0	0	0	0	0	0	0	0
☑	3	0	0	0	0	0	0	0	0	0	0	0	0	0
☑	4	0	0	0	0	0	0	0	0	0	0	0	0	0
☑	5	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6.7: GUI interface: select states table



M1	M2	M3	M4	M5	M6
0	0	0	0	0	0

Figure 6.8: GUI interface: input of Mach values for flutter envelope

- Max Speed for flutter calculation;
- Max V step (number of steps used during iterative mode tracking);
- Air Density;
- Mach Number.

When Flutter Envelope is selected, the requested input parameters are:

- Number of Mach values for which flutter envelope is computed;
- Values of Mach numbers: when the *Insert Values* button is pressed a table appears (see figure 6.8) where the user must insert Mach number values.

6.2 NeoCASS GUI Panel Settings

The GUI Panel Settings, shown in figure 6.9 is used to input parameters for analysis solvers. In particular, the following parameters must be selected by the user:

- Structural Model (1 Linear Beam, 2 Equivalent Plate, 3 Non-Linear Beam). In the version 1.1 of NeoCASS the Equivalent Plate structural model is not active even if already included;
- Aspect Ratio: when Equivalent Plate is selected the user can control the size (and number) of Plate elements automatically generated by means of this parameter, which control the aspect ratio of Plate. It is set by default equal to 1;
- Sub-Iter: when Non-Linear beam is selected, Sub-Iter defines the number steps needed to reach convergence with an assigned load;

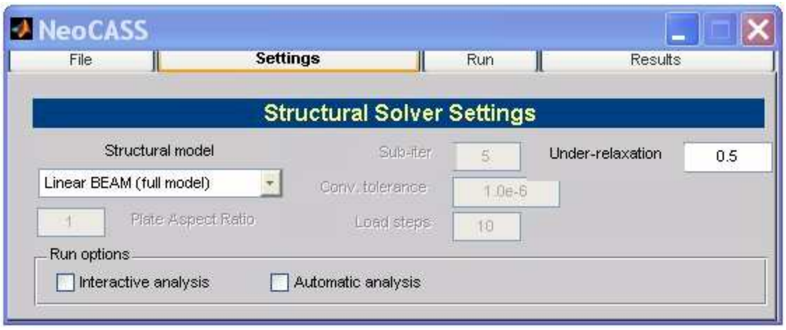


Figure 6.9: GUI interface: Settings panel

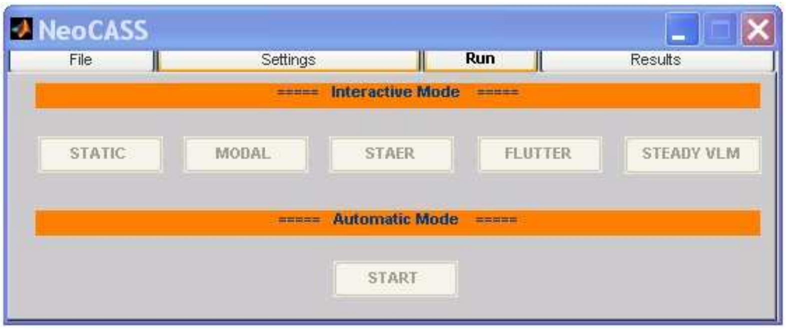


Figure 6.10: GUI interface: Run panel

- Conv.Tolerance (convergence error on the residue during non-linear analysis);
- Load Steps: number of load steps during static non-linear analysis or maximum number od coupled iterations during Static Aeroelastic Analysis;
- Under-relaxation (relaxation factor adopted transferring loads from aerodynamic to structural mesh): 0.5 is the default value.

GUI Panel Settings include also two checkboxes, usable but inactive in Version 1.1: when Interactive checkbox is selected, the user has to run each requested solver separately by pressing the specific button; when Automatic Analysis checkbox is selected NeoCASS automatically runs all the solvers selected into GUI Panel File.

6.3 NeoCASS GUI Panel Run

The GUI Panel Run, shown in figure 6.10, simply collects all buttons related to each solvers. Only buttons related to solvers for which all the requested data have been input are active (clickable). In the same panel is located the button named *Start* used to start all solvers in the automatic analysis mode, when selected (option inactive for Version 1.1).

6.4 NeoCASS GUI Panel Results

GUI Panel Results is a collection of buttons and checkbox options allowing the user to analyze and post-processing the results of a NeoCASS run. Many of the buttons and selection fields available on this GUI Panel have a different meaning, depending on which kind of analysis has been performed. The post-processing options are the following:

GUESS By pressing the button *GUESS* it is possible to plot the results of a GUESS analysis. The selection of which kind of diagram has to be plotted is done by filling the Selected Set field, ranging in this case from 1 to 10;

Aerodynamic Matrix In case of Flutter analysis, by pressing the button *Plot Aero Matrix* it is possible to plot the component of Aerodynamic Generalized Forces (Q_{hh}): in this case the user must supply the ROW and COL indices. The Selected Set field in this case allows the user chose among the different Mach numbers for which Q_{hh} has been computed (Flutter Envelope option)

Plot Model When a simple structural analysis has been performed (Modal Analysis), by pressing the button *Plot Model* a new figure showing the structural model is created. Otherwise, in case of a Steady Rigid Aerodynamic Analysis (VLM) the same button allows to see both structural and aerodynamic panels;

Plot Deformed Model In case of a simple structural analysis (Modal Analysis), pressing the button *Plot Deformed Model* it is possible to visualize the mode shapes. The number of mode to be plotted is as usual controlled by the Selected Set field, while the Scale factor field determines the amplitude of the deformed shape. It is possible to generate an animation for each mode shape, choosing the number of mode and the number of frames. Pressing button *Export Mode Animation* an .AVI file is created containing the vibration mode animation. In case of Static Aeroelastic Analysis, by pressing the button *Plot Deformed Model* a new figure showing the superposition of deformed and undeformed structural and aerodynamic models is created. The scale factor of deformed shape is controlled by typing the right value into the Scale field, set by default equal to 1. Using this option it is possible for the user to check the correctness of structure-aerodynamics interface.

Plot Flutter Diagrams In case of Flutter Analysis pressing the button *Plot Flutter diagrams* the figures reporting V-g plot and Flutter envelope are created (if related output has been requested).

Selection Checkboxes Three selection checkboxes are available, all related to the plot of aerodynamic panels. They allow to include or exclude into the plot the wake elements, the panels normals and the contour visualization.

SAVE NeoCASS Project Pressing this button all intermediate results and data, organized into separated MATLAB© structures, are saved into an unique MATLAB© binary file (.MAT). In this way, it is possible in any moment to read it by pressing the related button into the GUI Panel File so recovering all the available data.

Close ALL To Exit from NeoCASS and delete all temporary files.

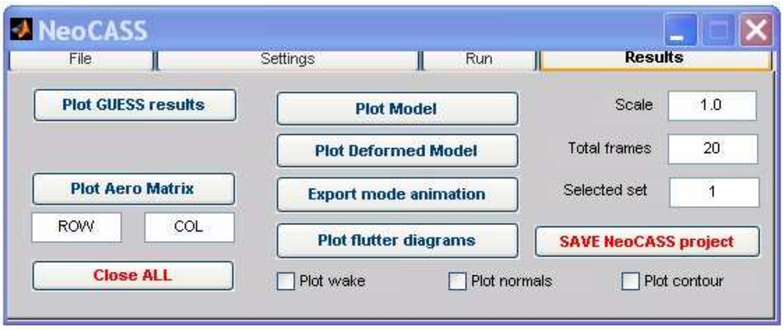


Figure 6.11: GUI interface: Results panel

6.5 NeoCASS Tutorial

This section includes a brief tutorial to help user in running NeoCASS modules. First of all it must be highlighted that NeoCASS is a framework running under MATLAB© that includes tools for the structural conceptual design of aircraft and for its aeroelastic behavior investigation. Basically, NeoCASS includes two main tools libraries, called GUESS and SMARTCAD . The goals of this two libraries are the following:

- GUESS is a library of tools aiming at the conceptual design of aircraft structure, starting from the `aircraft.xml` file coming from the CADaC module of CEASIOM. GUESS library requires three input files, i.e.

  1. `aircraft.xml`;
  2. `states.xml`;
  3. `technological.xml`.

  The first two files are already available within CEASIOM, while the third one contains some details related to the structural configuration, material and stick model parameters. An example of this file can be found in the Appendix A of this manual. The structural sizing is performed using a semi-analytical approach, and tacking into account three load conditions, i.e. a pull-up maneuver at maximum g, hard landing and taxying over bump. The results of GUESS are mainly the structural weight estimation and the stiffness trend along wing, fuselage and horizontal tail (vertical will be available soon). Once completed the initial structural sizing, GUESS generates the aircraft stick-model, i.e. a structural mesh in a NASTRAN© -like format saved into an ascii .dat file supplied by the user;

- SMARTCAD is a library of tools used to investigate the aeroelastic behavior of aircraft. The input file is a NASTRAN© -like .dat ASCII format that includes the stick model previously generated by GUESS (or already available as resulting from another pre-processor), enriched by the cards related to the specific aeroelastic analysis requested by the user.

An overview of NeoCASS is shown in figure 6.12. Two are then the possibilities to run NeoCASS modules, starting from GUESS input files or starting from SMARTCAD input file.

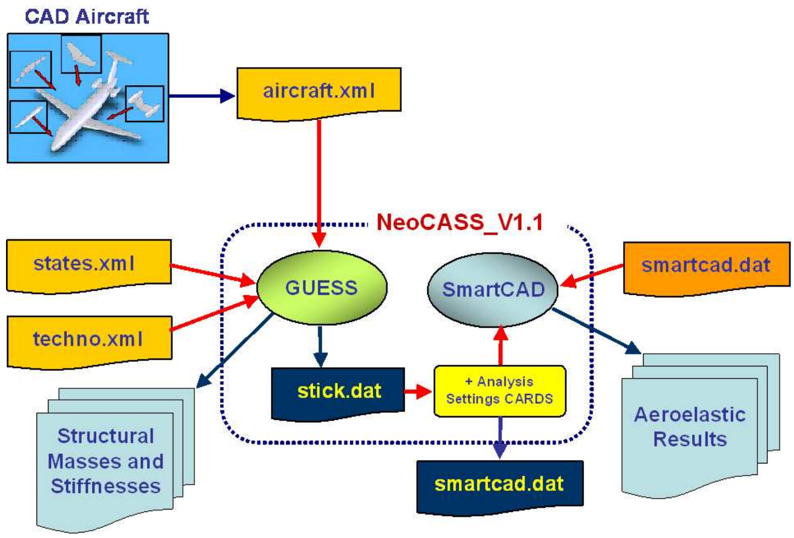


Figure 6.12: NeoCASS overview

6.5.1 Getting Started

In the following pages, it will assumed that NeoCASS V1.1 has been installed into the `NeoCASS_home` directory. The user must go into this directory and run the MATLAB© script:

```
set_neocass_path.m
```

that can be found in the same directory. This script automatically scans all the subdirectories building up the correct path variable. After that, it is possible to run all NeoCASS Modules. First of all, the user must run the GUI Interface Panel, from which it is possible to access all NeoCASS modules, simply by pressing:

```
NeoCASS
```

The Gui Panel File of NeoCASS GUI interface will appear.

6.5.2 Running GUESS

In this section will explained how to run GUESS tools, using as reference example the B747 aircraft included into the NeoCASS distribution tar file. Basically, the user must provide the names of three files requested by GUESS , i.e.:

1. `aircraft.xml`
2. `states.xml`
3. `technological.xml`

Press the *Open aircraft* button, an open file window will appear, go to the `example/guess/B747` directory and select the `B747_080304.xml` file. Press now the second button, named *Open states*, and select in the same directory the `states.xml` file. Finally, press the *Open* *techno* button and select in the same directory the `tech747_080304.xml` file. In the main MATLAB© windows an echo of the selected files is reported. At this time, the user can decide to edit one or more than one of these files, or can simply run the GUESS code. Remember that to edit the xml files pressing the related buttons *EDIT*, the correct path of Editor programs has to be setup filling the corresponding command lines into the editor setup files:

`neocass_xml_editor_path.m`

`neocass_xml_editor_path.m`

To run GUESS code, simply press the related *RUN GUESS* button: a new file window will appear asking to insert the name of ascii .dat file where the stick model will be saved. GUESS code will run very quickly and it will be possible to check the different computing phases looking at the MATLAB© window. After that, two kind of results are at disposal:

Structural Properties The structural properties of the aircraft are accessible through the Gui Panel Results of NeoCASS GUI interface (see the related chapter in the same manual). They are summarized in the following 10 Output sets (in this example only properties along wing span are considered):

1. Output set 1: Fuel volume
2. Output set 2: Fuel centroid
3. Output set 3: Lift load area
4. Output set 4: Center of pressure
5. Output set 5: Shear force
6. Output set 6: Bending moment
7. Output set 7: Thickness estimation
8. Output set 8: Weight estimation (bending-shear-total)
9. Output set 9: Weight estimation (Guess-structural weight-primary weight-total)
10. Output set 10: Guess vs. stick model properties

Stick model The stick model, saved into the .dat ASCII file provided by the user, is a beam-based representation of the aircraft structure. The syntax adopted is almost equivalent the one of Nastran. The stick model generated by GUESS contains the following information (please see the related chapter in the same manual):

1. Material definition (MAT1 cards)
2. Node definition (GRID cards)
3. Aeronode definition ((GRID cards)
4. Beam properties (PBAR cards)
5. Beam definition (CBAR cards)

6. Lifting surface definition (CAERO1 cards)
7. Structural interpolations set (SET1 cards)
8. Interpolation definition (SPLINE2 card)
9. Lumped masses (CONMN2 cards)

This file is not enough to run any kind of analysis, since includes just information about the model. The user must supply the input parameters requested by each kind of solver: this operation can be completed editing by hand the .dat file, or using the user friendly GUI interface panels.

6.5.3 Preparing the SMARTCAD file

As mentioned in the previous section, the .dat file generated by GUESS is not enough to run a SMARTCAD analysis. To do that, we have two possibilities:

1. Preparing the SMARTCAD adding all the information needed by the requested analysis runs;
2. Using and already available SMARTCAD file.

The two options are accessible using the available buttons grouped inside the two small panels named *Solver Input Data* and *Read Analysis Input Data* in the GUI Panel File. Two kind of input parameters are requested, related to reference quantities and to setting parameters for the solvers, respectively. Their input is managed by two GUI Subpanels named REFERENCE_Settings and ANALYSIS_Setting, accessible pressing the related buttons, and already described in the GUI chapter of this manual. When the *GENERATE* button is pressed, a window asking the name of the already available (from GUESS) .dat file will appear and, on the basis of input parameters supplied by the user, the related CARDS are automatically generated and appended to this .dat file.

6.5.4 Opening an already available SMARTCAD file

Once completed a GUESS output file with the necessary CARDS related to the requested analysis, or if one or more .dat SMARTCAD files including all necessary CARDS are already available, the user can skip all the previous phases and go directly to the *Open SMARCAD* button. Following the same B747 example used for GUESS code, pressing this button an open window will appear on the screen. Go to the `example/smartcad/B747` directory and select for example the `B747_flutter.dat` file. After reading this file one or more checkboxes on the GUI Panel File become checked and green colored: it means that the input file contains all the CARDS requested by these analyses. In this case, the green checkboxes are ones related to Modal Analysis (Eig) and Flutter Analysis (Flutter). In fact, if the CARDS requested by Flutter Analysis have been supplied also the Modal Analysis can be run.
If case for example the file `B747_aerosteady.dat` has been selected as input file, the checked and greened checkboxes will be ones related to Steady VLM and Static Aeroelastic Analysis (Staer). Once selected the input file, the user can go to the GUI Panel Settings to enter the solver settings parameters, going finally to the GUI Panel Run to run the requested analysis.

Appendix A

GUESS input: an introduction to *technology* file

The *technology* `.xml` file is herein presented in the most user-friendly version to run GUESS simulations. It contains several entries to cover a rather wide range of informations: from material properties definition, loading parameters setup, aero/structural mesh and aerodynamic modelling to the setup of the simulations.
The file has been conceived to be easily understood and modified by the user and it offers a great range of possible options. The *technology* file contains two main sections:

1. `user_input`;

2. `experienced_user_input`.

Although few parameters have been added in the latter section, the GUESS beginner is advised to modify exclusively the entries in the former section, that is `user_input`. The experienced GUESS user, able to understand the meaning of equations involved in the computer program and the overall structure of the code, may further set the values for the entries in the section `experienced_user_input`.
Within each main section, `user_input` and `experienced_user_input`, the parameters are further divided in four subsections, that are

1. `geometry`;

2. `material_property`;

3. `loading`;

4. `analysis_setup`.

In the following pages, the structure of *technology* `.xml` input file is shown and a brief explanation of the entries is given.

Entry – user_input.geometry.*	Note
beam_model.nwing_inboard	beam elements in inboard sector
beam_model.nwing_midboard	beam elements in midboard sector
beam_model.nwing_outboard	beam elements in outboard sector
beam_model.nwing_carryth	beam elements in carrythrough
beam_model.nfuse	beam elements along fuselage
beam_model.nvtail_inboard	beam elements in inboard sector
beam_model.nvtail_outboard	beam elements in outboard sector
beam_model.nhtail_inboard	beam elements in inboard sector
beam_model.nhtail_outboard	beam elements in outboard sector
beam_model.nhtail_carryth	beam elements in carrythrough
aero_panel.nx.wing_inboard	aero panels in inboard sector
aero_panel.nx.wing_midboard	aero panels in midboard sector
aero_panel.nx.wing_outboard	aero panels in outboard sector
aero_panel.nx.vert_inboard	aero panels in inboard sector
aero_panel.nx.vert_outboard	aero panels in outboard sector
aero_panel.nx.hori_inboard	aero panels in inboard sector
aero_panel.nx.hori_outboard	aero panels in outboard sector
aero_panel.nx.sup_control.wing_inboard	aero panels in inboard sector
aero_panel.nx.sup_control.wing_midboard	aero panels in midboard sector
aero_panel.nx.sup_control.wing_outboard	aero panels in outboard sector
aero_panel.nx.sup_control.vert_inboard	aero panels in inboard sector
aero_panel.nx.sup_control.vert_outboard	aero panels in outboard sector
aero_panel.nx.sup_control.hori_inboard	aero panels in inboard sector
aero_panel.nx.sup_control.hori_outboard	aero panels in outboard sector
aero_panel.ny.wing_inboard	aero panels in inboard sector
aero_panel.ny.wing_midboard	aero panels in midboard sector
aero_panel.ny.wing_outboard	aero panels in outboard sector
aero_panel.ny.vert_inboard	aero panels in inboard sector
aero_panel.ny.vert_outboard	aero panels in outboard sector
aero_panel.ny.hori_inboard	aero panels in inboard sector
aero_panel.ny.hori_outboard	aero panels in outboard sector

Remarks. The properties of the aero/structural mesh are defined in the current subsection. The user shall define the number of beam elements, used in the latter stick model, for all the single components. In addition to the beam elements along fuselage and within the admissible sectors for lifting surfaces (Table 2.4 in Section 2.4), two additional informations have to be defined: the number of beam elements in half the carrythrough structure for wing and horizontal tail.
The aerodynamic model is set by defining the number of chordwise and spanwise aerodynamic panels for all the lifting surfaces sectors. The fields `nx` and `ny` mean chordwise and spanwise, respectively. To account for control surface devices, the user shall set the number of chordwise aerodynamic panels used to discretize the movable surfaces.

Entry – user_input.material_property.*	**Note**	**Unit**
wing.kcon	wing structural concept	–
wing.esw	Young's modulus for wing material	[N/m^2]
wing.fcsw	ultimate compressive strength of wing	[N/m^2]
wing.dsw	density of the wing material	[kg/m^3]
fus.kcon	fuselage structural concept	–
fus.fts	tensile strength on top/bottom	[N/m^2]
fus.fcs	compressive strength	[N/m^2]
fus.es	Young's modulus for the shell material	[N/m^2]
fus.ef	Young's modulus for the frame material	[N/m^2]
fus.ds	density of shell material	[kg/m^3]
fus.df	density of frame material	[kg/m^3]
vtail.kcon	vertical tail structural concept	–
vtail.esw	Young's modulus for vertical tail material	[N/m^2]
vtail.dsw	density of the vertical tail material	[kg/m^3]
vtail.fcsw	ultimate compressive strength	[N/m^2]
htail.kcon	horizontal tail structural concept	–
htail.esw	Young's modulus for horizontal tail material	[N/m^2]
htail.dsw	density of the horizontal tail material	[kg/m^3]
htail.fcsw	ultimate compressive strength	[N/m^2]

Remarks. The `user_input.material_property` collects some material properties that the user may update. It is possible to define one different structural concept for each component analyzed. Different structural concepts are available within GUESS and they have been introduced in Section 2.6.1 and 2.6.3 for fuselage and lifting surfaces, respectively. Moreover, mechanical properties such as Young's modulus, tensile/compressive strength and density have dedicated entries. It is possible to identify separetely different properties for shell and frame.

Entry – user_input.loading.*	Note	Unit
normal_load_factor	normal load factor in $g's$	–
weight_fraction.cman	$MTOW$ weight fraction at maneuver	–
weight_fraction.cbum	$MTOW$ weight fraction at bump	–
weight_fraction.clan	$MTOW$ weight fraction at landing	–
aero_data.Cn_dr	yawing moment coefficient due to rudder	[rad^{-1}]
aero_data.Cl_dr	rolling moment coefficient due to rudder	[rad^{-1}]
aero_data.Cn_dw	yawing moment coefficient due due to aileron/spoilers	[rad^{-1}]
aero_data.Cl_dw	rolling moment coefficient due to aileron/spoilers	[rad^{-1}]
aero_data.Cn_b	yawing moment coefficient due to sideslip	[rad^{-1}]
aero_data.Cl_b	rolling moment coefficient due to sideslip	[rad^{-1}]
aero_data.CY_b	side force coefficient due to sideslip	[rad^{-1}]
aero_data.CY_dr	side force coefficient due to rudder	[rad^{-1}]
aero_data.CY_dw	side force coefficient due to aileron/spoilers	[rad^{-1}]
aero_data.L_alpha_s	horizontal tail load due to unit α_s	[N/rad]
aero_data.M_alpha_s	horizontal pitching moment due to unit α_s	[Nm/rad]
aero_data.L_delta_e	horizontal tail load due to unit δ_e	[N/rad]
aero_data.M_delta_e	horizontal pitching moment due to unit δ_e	[Nm/rad]
aero_data.Lc	horizontal tail load due to unit built-in chamber	[N]
aero_data.Mc	horizontal pitching moment due to unit built-in chamber	[Nm]
aero_data.dM_025dalpha	pitching moment coefficient about 0.25 mac wing	–
aero_data.CL	airplane lift coefficient	–
aero_data.CY_alpha	side force coefficient due to sideslip	[rad^{-1}]

Remarks. To setup the load analysis, few parameters need to be defined, such as the normal load factor and the weight fraction of $MTOW$ for different conditions. The user is not intended, at least in the very early stages of the desing process, to fill up all the list of `aero_data`. If no aerodynamic informations are available, the user needs to run the Vortex-Lattice method and a stability and control derivatives program, shortly described in Appendix D; then the aero data will be stored in the current *technology* input file. The user just needs to initialize the following two entris to run VLM and stability derivatives calculations:

```
user_input.analysis_setup.vlm_calculation.htail=1
user_input.analysis_setup.vlm_calculation.vtail=1
```

More informations for the analysis setup will follow in the next pages.

Entry – user_input.loading.*	Note	Unit
maximum_deflection. Rudder_limit_deflection	rudder limit deflection	[deg]
maximum_deflection. Elevator_limit_deflection_up	elevator limit deflection up	[deg]
maximum_deflection. Elevator_limit_deflection_down	elevator limit deflection down	[deg]
maximum_deflection. limit_tailplane_deflection_up	tailplane limit deflection up	[deg]
maximum_deflection. limit_tailplane_deflection_down	tailplane limit deflection down	[deg]

Remarks. Solving the trim solution for the tailplane and elevator deflection (described in Section 2.5), one needs to consider feasible solutions respect some physical constrains. The user can set the limit deflections, up and down, that must be fullfilled by the computed trim solution. Moreover the first entry, named `Rudder_limit_deflection`, is required to define the rudder input deflection for the vertical tail load analysis.

Entry – user_input.analysis_setup.*	Note
pressure_stabilization	1 $\rightarrow$ pressure is stabilized in cabin else $\rightarrow$ pressure is not stabilized
lift_distribution	1 $\rightarrow$ Schrenk load distribution on wing else $\rightarrow$ trapezoidal distribution
regression.analf	linear $\rightarrow$ linear regression equation for fuselage else $\rightarrow$ power-intercept regression equation
regression.analw	linear $\rightarrow$ linear regression equation for wing else $\rightarrow$ power-intercept regression equation
regression.analh	linear $\rightarrow$ linear regression equation for hor.tail else $\rightarrow$ power-intercept regression equation
regression.analv	linear $\rightarrow$ linear regression equation for vert.tail else $\rightarrow$ power-intercept regression equation
beam_model.fuse	1 $\rightarrow$ fuselage is defined in beam model else $\rightarrow$ it is not defined
beam_model.winr	1 $\rightarrow$ wing is defined in beam model else $\rightarrow$ it is not defined
beam_model.vert	1 $\rightarrow$ vertical tail is defined in beam model else $\rightarrow$ it is not defined
beam_model.horr	1 $\rightarrow$ horizontal tail is defined in beam model else $\rightarrow$ it is not defined
beam_model.symmXZ	1 $\rightarrow$ beam model is XZ symmetric else $\rightarrow$ it is not XZ symmetric

Entry – user_input.analysis_setup.*	Note
vlm_calculation.htail	1 → run VLM for horizontal tail else → do not run VLM
vlm_calculation.vtail	1 → run VLM for vertical tail else → do not run VLM
torsion_stiffness.fus	1 → Bredt formula for torsional constant else → monocoque method
torsion_stiffness.wing	1 → Bredt formula for torsional constant else → monocoque method
torsion_stiffness.vtail	1 → Bredt formula for torsional constant else → monocoque method
torsion_stiffness.htail	1 → Bredt formula for torsional constant else → monocoque method

Remarks. The section `user_input.analysis_setup` contains several options the user can modify according the needs.
`pressure_stabilization` takes into account the fact that pressure differential in passengers compartment may be stabilized or not. As already introduced in Section 2.5.1, pressure loads may relieve stresses on the fuselage if pressure stabilitation is chosen as an option (setting the entry `pressure_stabilization=1`).
`lift_distribution` offers the option to compute the load distribution over the lifting surfaces using either Schrenk method or a trapezoidal distribution, as described in Section 2.5.3.
The entry `regression` specifies for all the components eventually defined in the stick model the kind of regression equation to use. Two options are available: a linear regression equation or a power-intercept regression equation (introduces in Section 2.7).
The entry identified with `beam_model` let the user to choose each single component to include in the stick model. The flag set to one includes the correspondent component; and viceversa. It is possible to select all the components, activate the symmetry flag, and deal with the complete model for an aircraft; or to select just a single cantilver wing.
Since several aero data are required to run GUESS , the computer program may compute the necessary informations using a Vortex-Lattice method. In case the user owns more accurate aero data, for instance from time-consuming simulations (reading CFD), the correspondent entries may be directly defined and updated by the user.
The entry `torsion_stiffness` offer the option to compute the torsional constant for a multi-cell section (as lifting surfaces are intended to be) using the analytical Bredt formula or the monocoque method. More accurate results are obtained by using the latter option, which supply for more informations as described in the Section ??section?? .

Entry – experienced_user_input.geometry.*	Note
guess.wing.inboard	number of nodes in inboard sector
guess.wing.midboard	number of nodes in midboard sector
guess.wing.outboard	number of nodes in outboard sector
guess.fus	number of nodes along fuselage
guess.vert.inboard	number of nodes in inboard sector
guess.vert.outboard	number of nodes in outboard sector
guess.hori.inboard	number of nodes in inboard sector
guess.hori.outboard	number of nodes in outboard sector

Remarks. The section reserved for the experienced user contains first geometric details, named `geometry.guess`. The number of nodes along fuselage lenght and lifting surfaces structural span is required for the structural sizing developed within GUESS . These informations do not define the beam model which is indeed defined in the already discussed entry `user_input.geometry.beam_model`. The current parameters define a detailed mesh model over which GUESS performs the interpolation for the exported stick model. Thus the number of nodes herein considered is greater than the number of beam elements.

Entry – experienced_user_input. material_property.*	Note	Unit
wing.tmgw	minimum gage thickness for the wing	[m]
wing.effw	buckling efficiency of the web	–
wing.effc	buckling efficiency of the covers	–
wing.cf	Shanley's constant for frame bending	–
fus.ckf	frame stiffness coefficient	–
fus.ec	power in approximation equation for for buckling stability	–
fus.kgc	buckling coefficient for component general buckling of stiffener web panel	–
fus.kgw	buckling coefficient for component buckling of web panel	–
fus.tmg	minimum gage thickness for the fuselage	[m]
vtail.tmgw	minimum gage thickness for vertical tail	[m]
htail.tmgw	minimum gage thickness for horizontal tail	[m]

Remarks. The section named `experienced_user_input.material_property` contains few material property parameters, which do not belong to the `user_input` section since they require a deeper knowledge about materials and material manufacturing. The most important of them, common to all the components, is the minimum gage thickness (named `tmg` and `tmgw` for fuselage and lifting surfaces, respectively), defining the lower thickness that is physically significant from a manufacturing point of view.

Entry – experienced_user_input.loading.*	Note	Unit
aero_data.q	do not modify	[N/m^2]
aero_data.V	do not modify	[m/s]
flexibility.dalphas_dnz	fuselage flexibility due to n_z	[rad]
flexibility.dalphas_dLt	fuselage flexibility due to L_t	[rad/N]
flexibility.dalphas_dMt	fuselage flexibility due to M_t	[rad/Nm]

Remarks. In the section `experienced_user_input.loading`, the user is advised not to modify the entry `q`, dynamic pressure, and `V`, speed computed from Mach number and speed of sound at given altitude, because they are related to other parameters defined in the *state* `.xml` input file. Indeed the entry `flexibility` gathers together the rate of change in angle of attack at horizontal tail due to body flexibility, caused by normal load factor and horizontal tail loads. A pictorial view of these concepts is given in Figure 2.15 and 2.14.

Entry – experienced_user_input.analysis_setup.*	Note
BESTFIT	do not modify
iload	do not modify

Remarks. The last subsection contains two coefficients. The first, named `BESTFIT`, applies for the structural module and might be interpreted as a correction factor. The latter entry, named `iload`, applies for the loads module. These coefficients are not supposed to be modified, either by the beginner or the experienced user.

Appendix B

GUESS input: a very short introduction to *geometry* file

The aero-structural analysis of a new aircraft design covers the interaction of aerodynamics, weight, balance and loads, each of these requiring a peculiar description of the morphology, yet referring to the very same aircraft. In addition, with growing design maturity, the geometric description of the aircraft evolves substantially, offering more and more details. Such a complex multi-disciplinary and multi-fidelity problem calls for a geometric description that is flexible enough to suit all the separate study domains and levels of fidelity, yet remains simple enough to be intuitive to the user and to enable easy optimization and trade study analyzes.
The geometric description is obtained by an appropriate parameterization of the different aircraft components and of their relative positioning. For the wing for example, a two kinks description has been adopted with parameters such as aspect ratio, area, sweeps, dihedrals, twists and airfoil sections.
An aircraft design geometry is fully described in a unique `.xml` file to which all the different analysis module refer. In this `.xml` file appear in a structured way the different parts of the aircraft and the associated parameters.
The choice of the `.xml` format facilitates the sharing of data as well as the expansion of the dataset, i.e. the number of components of the aircraft can be expanded at will and it is possible to introduce new components as well as new parameters, thus enlarging the array of morphologies that can be modeled.

The present appendix collects few entries for the specific application of the semi-wing of a Boeing 747−100 presented in Chapter 3. The overview is purely inteded to give a basic insight of the geometric description that have been used for the intended analysis. For conciseness, actual aircraft description by means of `.xml` file generated by CADac (Computer Aided Design Aircraft) are not reported, but it is worthy noting the great flexibility offered by this morphology approach. Already in Section 2.4, it has been shown different fuselage layouts, while in Chapter 4 actual aircraft planform shapes are illustrated, correctly editing the very same entries herein reported in the following pages.

Entry – aircraft.wing1.*	**Value**	**Unit**
longitudinal_location	17.0891	[m]
area	508.0867	[m^2]
span	59.4667	[m]
spanwise_kink1	1.0	–
spanwise_kink2	1.0	–
taper_kink1	0.2646	–
taper_kink2	0.2646	–
taper_tip	0.2646	–
thickness_root	0.1794	[m]
thickness_kink1	0.078	[m]
thickness_kink2	0.078	[m]
thickness_tip	0.078	[m]
quarter_chord_sweep_inboard	37.17	[deg]
quarter_chord_sweep_midboard	37.17	[deg]
quarter_chord_sweep_outboard	37.17	[deg]
LE_sweep_inboard	40.0896	[deg]
LE_sweep_midboard	40.0896	[deg]
LE_sweep_outboard	40.0896	[deg]
dihedral_inboard	7.00	[deg]
dihedral_midboard	7.00	[deg]
dihedral_outboard	7.00	[deg]
root_incidence	1.00	[deg]
kink1_incidence	1.00	[deg]
kink2_incidence	1.00	[deg]
tip_incidence	1.00	[deg]

Entry – aircraft.fuel.*	**Value**	**Unit**
Fore_wing_spar_loc_root	0.088	–
Fore_wing_spar_loc_kik1	0.088	–
Fore_wing_spar_loc_kin2	0.088	–
Fore_wing_spar_loc_tip	0.088	–
Aft_wing_spar_loc_root	0.723	–
Aft_wing_spar_loc_kik1	0.723	–
Aft_wing_spar_loc_kink2	0.723	–
Aft_wing_spar_loc_tip	0.723	–

Entry – aircraft.fuselage.*	**Value**	**Unit**
Aftfuse_X_sect_vertical_diameter	6.157	[m]

Entry – aircraft.engines1.*	**Value**	**Unit**
Location_engines_nacelles_on_X	14.8927	[m]
Location_engines_nacelles_on_Y	7.1657	[m]
Location_engines_nacelles_on_Z	-0.6952	[m]
nacelle_length	4.572	[m]
Number_of_engines	2	–
d_max	1.8898	[m]

Entry – aircraft.engines2.*	**Value**	**Unit**
Location_engines_nacelles_on_X	20.2132	[m]
Location_engines_nacelles_on_Y	13.114	[m]
Location_engines_nacelles_on_Z	0.0295	[m]
nacelle_length	4.572	[m]
Number_of_engines	2	–
d_max	1.8898	[m]

Appendix C

Structural concept coefficients

The structural coefficients for fuselage and lifting surfaces are herein collected for different structural concepts. These parameters are mainly involved in the structural equations discussed in Section 2.6. Each concept has its own characteristic values.
The user defines the structural concept to analyze and GUESS automatically loads from the internal database the values correspondent to the selected concept. It is necessary to set in the *technology* input file, exclusively for the components defined in the stick model later exported in the ASCII format, the following entry for the fuselage

`user_input.material_property.fuse.kcon`

and for the wing, vertical and horizontal tail, respectively, the following

`user_input.material_property.wing.kcon`
`user_input.material_property.vtail.kcon`
`user_input.material_property.htail.kcon`

choosing for the first entry one of the seven options shown in Figure C.1, and for the lifting surfaces one of the six options shown in Figure C.3.

Entry `kcon`	**Structural concept**
1	Simply stiffened shell, frames, sized for minimum weight in buckling
2	Z-stiffened shell, frames, best buckling
3	Z-stiffened shell, frames, buckling-minimum gage compromise
4	Z-stiffened shell, frames, buckling-pressure compromise
5	Truss-core sandwich, frames, best buckling
6	Truss-core sandwich, no frames, best buckling
7	Truss-core sandwich, no frames, buckling-minimum gage-pressure compromise

Table C.1: Available fuselage structural geometry concepts

Entry kcon	m	ε	K_{mg}	K_p	K_{th}
1	2.000	0.6560	2.463	2.463	0.000
2	2.000	0.9110	2.475	2.475	0.000
3	2.000	0.7600	2.039	1.835	0.000
4	2.000	0.7600	2.628	1.576	0.000
5	2.000	0.6050	4.310	3.965	0.459
6	1.667	0.4423	4.820	3.132	0.405
7	1.667	0.3615	3.413	3.413	0.320

Table C.2: Fuselage structural geometry parameters.

Entry kcon	**Covers**	**Webs**
1	Unstiffened	Truss
2	Unstiffened	Unflanged
3	Unstiffened	Z-stiffened
4	Truss	Truss
5	Truss	Unflanged
6	Truss	Z-stiffened

Table C.3: Available lifting surface structural geometry concepts

Entry wing.kcon	ε	e	ε_c	e_c	ε_w	K_{g_c}	K_{g_w}
1	2.250	0.556	3.620	3.000	0.605	1.000	0.407
2	2.210	0.556	3.620	3.000	0.656	1.000	0.505
3	2.050	0.556	3.620	3.000	0.911	1.000	0.405
4	2.440	0.600	1.108	2.000	0.605	0.546	0.407
5	2.400	0.600	1.108	2.000	0.656	0.546	0.505
6	2.250	0.600	1.108	2.000	0.911	0.546	0.405

Table C.4: Wing structural coefficients and exponents.

Appendix D

Stability and control derivatives program

The stability and control derivatives program included in GUESS can be used to estimate some of the key lateral directional analysis, including stability and control derivatives for use in estimating aircraft characteristics. The code has been adapted and modified on the basis of LDstab code, developed by Joel Grasmeyer, and it is basically an implementation of the DATCOM method, with adjustments to match published B747 data.
The stability and control derivatives program is utilized by GUESS to compute the required derivatives for the vertical tail plane in case no more detailed data can be supplied by the user. More informations about the option GUESS offers are given in Appendix A.
The code is initialized with the corrispondent aircraft geometry stored in the *geometry* `.xml` input file and the output collects the computed derivatives. The code has been verified, for a Mach number of 0.25, against the Boeing $747-100$. The following table contains a comparison of the predicted stability and control derivatives with the flight test derivatives presented in Nelson [20].

Engineering symbol	flight test	computed by GUESS	Unit
CY_{β}	-0.96	-0.9601	[rad^{-1}]
Cl_{β}	-0.221	-0.2210	[rad^{-1}]
Cn_{β}	0.150	0.1500	[rad^{-1}]
CY_{δ_w}	–	0.0000	[rad^{-1}]
Cl_{δ_w}	0.0461	0.0461	[rad^{-1}]
Cn_{δ_w}	0.0064	0.0064	[rad^{-1}]
Cl_{δ_r}	0.175	0.1750	[rad^{-1}]
CY_{δ_r}	0.007	0.0070	[rad^{-1}]
Cn_{δ_r}	-0.109	0.1090	[rad^{-1}]

Appendix E

GUESS output file in ASCII format: one example

```
SOL 145
EIGR    1               0       20              20
        MASS    2050    4

AERO            631.37  8.778   0.85    1       0       1       50

MKAERO1 0.7
        0.0001  0.1     0.2     0.5     0.8     1.0


$-------2-------3-------4-------5-------6-------7-------8-------9-------10
 Material definition
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
MAT1    1       7.37e+10        0.3     2795.717
        4.032e+83.722e+82.584e+8
MAT1    2       7.37e+10        0.3     2795.717
        3.859e+83.859e+82.573e+8
MAT1    3       7.37e+10        0.3     2795.717
        3.859e+83.859e+82.573e+8
MAT1    4       7.37e+10        0.3     2795.717
        3.859e+83.859e+82.573e+8

$-------2-------3-------4-------5-------6-------7-------8-------9-------10
 Node definition
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
GRID    1000    0       0       0       -3.0349 0       0       0
GRID    1001    0       10      0       -1.295  0       0       0
GRID    1002    0       20.0406 0       -1.295  0       0       0
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
```

```
GRID    2000    0       29.7474 1.85    -1.295  0       0       0
GRID    2001    0       33.1614 3.1234  -1.2218 0       0       0
GRID    2002    0       36.5754 4.3967  -1.1487 0       0       0
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
GRID    3000    0       47.0409 0       1.85    0       0       0
GRID    3001    0       48.9245 0       3.2763  0       0       0
GRID    3002    0       50.8081 0       4.7025  0       0       0
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
GRID    4000    0       57.8043 0.35256 10      0       0       0
GRID    4001    0       59.3178 1.8126  10      0       0       0
GRID    4002    0       61.1908 3.4958  10      0       0       0

$-------2-------3-------4-------5-------6-------7-------8-------9-------10
 Beam definition
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
CBAR    1000    1000    1000    1001    -0.171410.0     0.9852  GGG
        0       0       0.0     0.0     1.295   0.0     0.0     1.295
CBAR    1001    1001    1001    1002    0.0     0.0     1.0     GGG
        0       0       0.0     0.0     1.295   0.0     0.0     1.295
CBAR    1002    1002    1002    1003    0.0     0.0     1.0     GGG
        0       0       0.0     0.0     1.295   0.0     0.0     1.295
CBAR    1003    1003    1003    1004    0.0     0.0     1.0     GGG
        0       0       0.0     0.0     1.295   0.0     0.0     1.295
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
CBAR    2000    2000    1003    2000    0.0     0.0     1.0     GGG
CBAR    2001    2001    2000    2001    -0.01880-0.007010.9998  GGG
CBAR    2002    2002    2001    2002    -0.01880-0.007010.9998  GGG
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
CBAR    3000    3000    1005    3000    -1.0    0.0     0.0     GGG
CBAR    3001    3001    3000    3001    -0.603660.0     0.79724 GGG
CBAR    3002    3002    3001    3002    -0.603660.0     0.79724 GGG
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
CBAR    4000    4000    3005    4000    0.0     0.0     1.0     GGG
CBAR    4001    4001    4000    4001    0.0     0.0     1.0     GGG
CBAR    4002    4002    4001    4002    0.0     0.0     1.0     GGG
```

```
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
 Aeronode definition
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
GRID    20000   0       24.7475 0       -1.295  0       0       0
GRID    20001   0       29.7474 0       -2.295  0       0       0
GRID    20002   0       34.7473 0       -1.295  0       0       0
GRID    20003   0       29.7474 0       -0.295030       0       0
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
RBE0    1003    1003    20000   20001   20002   20003
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
GRID    30000   0       44.6395 0       1.85    0       0       0
GRID    30001   0       47.0409 -0.705111.85    0       0       0
GRID    30002   0       49.4423 0       1.85    0       0       0
GRID    30003   0       47.0409 0.70511 1.85    0       0       0
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
RBE0    3000    3000    30000   30001   30002   30003
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
GRID    40000   0       56.1395 0       10      0       0       0
GRID    40001   0       57.8043 0       9.667   0       0       0
GRID    40002   0       59.4691 0       10      0       0       0
GRID    40003   0       57.8043 0       10.333  0       0       0
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
RBE0    3005    3005    40000   40001   40002   40003
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .

$-------2-------3-------4-------5-------6-------7-------8-------9-------10
 Beam properties
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
PBAR    1000    1       0.0240180.0217450.0217450.04349 255.8597
        1.1939  0       0       1.1939  -1.1939 0       0       -1.1939
PBAR    1001    1       0.0403240.0719250.0719250.14385 1059.002
        1.85    0       0       1.85    -1.85   0       0       -1.85
PBAR    1002    1       0.0769030.14264 0.14264 0.28528 1058.982
        1.85    0       0       1.85    -1.85   0       0       -1.85
```

```
.       .       .       .       .       .       .       .       .
.       .       .       .       .       .       .       .       .
.       .       .       .       .       .       .       .       .
PBAR    2000    2       0.33353 0.14149 0.78314 0.30917 6173.783
        0       2.1854  1.0191  0       0       -2.1854 -1.0191 0
PBAR    2001    2       0.17104 0.0895650.18399 0.16198 3218.927
        0       1.4994  0.93863 0       0       -1.4994 -0.938630
PBAR    2002    2       0.15162 0.0601170.12346 0.10901 2443.925
        0       1.3035  0.81593 0       0       -1.3035 -0.815930
PBAR    2003    2       0.16232 0.0478240.40295 0.10976 1974.177
        0       2.6154  0.71199 0       0       -2.6154 -0.711990
.       .       .       .       .       .       .       .       .
.       .       .       .       .       .       .       .       .
.       .       .       .       .       .       .       .       .
PBAR    3000    3       0.0557710.35085 0.0155860.0326230
        2.3026  0       0       0.67611 -2.3026 0       0       -0.67611
PBAR    3001    3       0.0443680.0821540.0095880.01324431.3961
        2.3026  0       0       0.67611 -2.3026 0       0       -0.67611
PBAR    3002    3       0.0340460.0607360.0070190.00925429.0894
        2.105   0       0       0.61809 -2.105  0       0       -0.61809
PBAR    3003    3       0.0211080.0435380.0050310.00663429.2925
        1.8839  0       0       0.55317 -1.8839 0       0       -0.55317
.       .       .       .       .       .       .       .       .
.       .       .       .       .       .       .       .       .
.       .       .       .       .       .       .       .       .
PBAR    4000    4       0.0265520.0012760.0130570.0015790
        0       1.2039  0.3358  0       0       -1.2039 -0.3358 0
PBAR    4001    4       0.0237580.0009240.0077280.00106133.9193
        0       1.0001  0.30503 0       0       -1.0001 -0.305030
PBAR    4002    4       0.0185860.0004660.0039020.00053524.7003
        0       0.79638 0.24289 0       0       -0.79638-0.242890
PBAR    4003    4       0.0121750.0001730.0014460.00019816.6583
        0       0.57213 0.1745  0       0       -0.57213-0.1745 0
.       .       .       .       .       .       .       .       .
.       .       .       .       .       .       .       .       .
.       .       .       .       .       .       .       .       .

$-------2-------3-------4-------5-------6-------7-------8-------9-------10
 Lifting surfaces definition
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
CAERO1  200     0       0       1       3       n64a206 n64a206 1
        21.7478 0       -1.2427 19.9994 1.85    1       0       1       1
CAERO1  201     3       0       2       3       n64a206 n64a206 1
        21.7478 1.8491  -1.2427 19.9994 2.5504  0.7546  71.3675 1       1
        1       0.2     0.2     2           flap1d
CAERO1  202     3       0       2       3       n64a206 n64a206 1
```

```
        30.5391 4.3952  -1.1092 15.0916 4.1732  0.66128 42.7191 1       1
        1       0.2     0.2     2         flap2d
CAERO1  203     3       0       4       3       n64a206 n64a206 1
        35.6705 8.5627  -0.890839.9797  13.9234 0.24442 46.9245 1       -2
        1       0.2     0.2     2       ailern1d
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
CAERO1  300     90      0       2       3       n64a006 n64a006 1
        41.4    0       1.85    14.1022 2.8525  0.825   55.4158 0       0
        1       0.3     0.3     2        rudder1
CAERO1  301     90      0       3       3       n64a006 n64a006 1
        46.1544 0       4.7025  11.6344 5.2975  0.60606 55.4158 0       0
        1       0.3     0.3     2        rudder2
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .

$-------2-------3-------4-------5-------6-------7-------8-------9-------10
 Structural interpolation set
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
SET1    2       1003    2000    2001    2002    2003    2004    2005
        2006    2007    2008    2009    2010    2011    2012    20000
        20001   20002   20003   20004   20005   20006   20007   20008
        20009   20010   20011   20012   20013   20014   20015   20016
        .       .       .       .       .       .       .        .
        .       .       .       .       .       .       .        .
        .       .       .       .       .       .       .        .


$-------2-------3-------4-------5-------6-------7-------8-------9-------10
 Interpolation definition
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
SPLINE2 200     200     1       3       2       2
SPLINE2 201     201     1       10      2       2
SPLINE2 202     202     1       10      2       2
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .
.       .       .       .       .       .       .       .        .

$-------2-------3-------4-------5-------6-------7-------8-------9-------10
 Lumped masses
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
CONM2   1       1000    0       752.94830       0       0
        0       0       0       0       0       0
CONM2   2       1001    0       1958.1090       0       0
```

```
        0       0       0       0       0       0
CONM2   3       1002    0       2370.2350       0       0
        0       0       0       0       0       0
CONM2   4       1003    0       2370.2170       0       0
        0       0       0       0       0       0
.       .       .       .       .       .       .       .       .
.       .       .       .       .       .       .       .       .
.       .       .       .       .       .       .       .       .
CONM2   109     2011    0       17772.943.4367  -2.8876 0.84017
        0       0       0       0       0       0
CONM2   110     2032    0       17772.943.4367  2.8876  0.84017
        0       0       0       0       0       0
.       .       .       .       .       .       .       .       .
.       .       .       .       .       .       .       .       .
.       .       .       .       .       .       .       .       .
```

Appendix F

Publications on NeoCASS

This appendix briefly collects all the papers regarding NeoCASS produced by the authors of the code together with external collaborations.

- L. Cavagna, L. Riccobene, S. Ricci, A. Bèrard, A. Rizzi, A.T. Isikveren *"Development and Validation of a Next-Generation Conceptual Aero-Structural Sizing Suite."* ICAS-International Council for the Aeronautical Sciences September 14-19, 2008, Anchorage, Alaska, USA.

- L. Cavagna, P. Masarati, P. Mantegazza, S. Ricci *"Development and Validation of an Investigation Tool for Nonlinear Aeroelastic Analysis."* ICAS-International Council for the Aeronautical Sciences September 14-19, 2008, Anchorage, Alaska, USA.

- R. von Kaenel, A. Rizzi, T. Grabowski, M. Ghoreyshi, L. Cavagna, S. Ricci, A. Bérard *"Bringing Adaptive-Fidelity CFD to Aircraft Conceptual Design: CEASIOM."* ICAS-International Council for the Aeronautical Sciences September 14-19, 2008, Anchorage, Alaska, USA.

- L. Cavagna, S. Ricci, L. Riccobene, A. Bérard, A. Rizzi. *"A Fast MDO tool for Aeroelastic Optimization in Aircraft Conceptual Design."* 12^{th} AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference 10-12 September 2008, Victoria, British Columbia, Canada.

Bibliography

[1] R. Ahrem, A. Beckert, and H. Wendland, *Recovering rotations in aeroelasticity*, Journal of Fluids and Structures **23** (2007), 874–884.

[2] E. Albano and W. P. Rodden, *A doublet–lattice method for calculating the lift distributions on oscillating surfaces in subsonic flow*, AIAA Journal **7** (1969), no. 2, 279–285.

[3] M.D. Ardema, A.P.P. Chambers, A.S. Hahn, H. Miura, and M.D. Moore, *Analytical Fuselage and Wing Weight Estimation of Transport Aircraft*, Tech. Report 110392, NASA, Ames Research Center, Moffett Field, California, May 1996.

[4] Klaus-Jürgen Bathe and Hou Zhang, *Finite element developements for general fluid flows with structural interactions*, International Journal for Numerical methods in Engineering **60** (2004), 213–232.

[5] Luca Cavagna, Giuseppe Quaranta, and Paolo Mantegazza, *Application of Navier-Stokes simulations for aeroelastic assessment in transonic regime*, Computers & Structures **85** (2007), no. 11-14, 818–832, Fourth MIT Conference on Computational Fluid and Solid Mechanics.

[6] L. Demasi and E. Livne, *Structural ritz-based simple-polynomial nonlinear equivalent plate approach: An assessment*, Journal of Aircraft **43** (2006), no. 6, 1685–1697.

[7] M. Drela, *Integrated simulation environment model for preliminary aerodynamic, structural and control-law design of aircraft*, Proceedings of the 40^{rmth} SDM (Cambridge,MA), no. AIAA Paper 99-1394, April 1999.

[8] P. Eliasson, *A Navier-Stokes Solver for Unstructured Grids.*, FOI/FFA report FOI-R-0298-SE, FOI, 2001.

[9] Charbel Farhat and Michael Lesoinne, *Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems*, Computer Methods in Applied Mechanics and Engineering **182** (2000), 499–515.

[10] Gian Luca Ghiringhelli, Pierangelo Masarati, and Paolo Mantegazza, *A multi-body implementation of finite volume beams*, AIAA Journal **38** (2000), no. 1, 131–138.

[11] G. L. Giles, *Design oriented structural analysis*, Tech. Report 109124, NASA, Langley, Hampton, Virginia, June 1994.

[12] G.L. Giles, *Design-oriented analysis of aircraft fuselage structures using equivalent plate methodology*, Journal of Aircraft **36** (1999), no. 1, 21–27.

[13] Joseph Katz and Allen Plotkin, *Low-Speed Aerodynamics From Wing Theory to Panel Methods*, McGraw-Hill series in Aeronautical and Aerospace Engineering, 1991.

[14] N. Maman and C. Farhat, *Matching fluid and structure meshes for aeroelastic computations: a parallel approach*, Computers & Structure **54** (1995), no. 4, 779–785.

[15] D. W. Marquardt, *Least squares estimation of nonlinear parameters, a computer program in fortran iv language, IBM SHARE library*, Tech. report, Distribution Number 309401, August 1966.

[16] Pierangelo Masarati and Paolo Mantegazza, *On the C^0 discretisation of beams by finite elements and finite volumes*, l'Aerotecnica Missili e Spazio **75** (1997), 77–86.

[17] B. W. McCormick, *Aerodynamics, aeronautics, and flight mechanics*, John Wiley & Sons, New York, NY, 1979.

[18] T. Melin, *User's guide Tornado 1.0 release 2.3 2001-01-31*, KTH report -, Royal Institute of Technology (KTH), 2003.

[19] MSC, *Msc/nastran encyclopedia v 70.5*, May 1998.

[20] R. C. Nelson, *Flight stability and automatic control*, McGraw-Hill, New York, 1989.

[21] Giuseppe Quaranta, Pierangelo Masarati, and Paolo Mantegazza, *A conservative mesh-free approach for fluid-structure interface problems*, International Conference on Computational Methods for Coupled Problems in Science and Engineering (Santorini, Greece) (M. Papadrakakis, E. Oñate, and B. Schrefler, eds.), CIMNE, 2005.

[22] R. Schaback and H. Wendland, *Characterization and construction of radial basis functions*, EILAT Proceedings (N. Dyn, D. Leviatan, and D. Levin, eds.), Cambridge University Press, 2000.

[23] O. Schrenk, *A simple approximation method for obtaining the spanwise lift distribution*, REPORT 948, NACA, 1940.

[24] Marilyn J. Smith, Dewey H. Hodges, and Carlos E. Cesnik, *Evaluation of computational algorithms for suitable fluid-structure interactions*, Journal of Aircraft **37** (2000), no. 2, 282–294.

[25] E. C. Yates, *AGARD standard aeroelastic configurations for dynamic response. I wing 445.6*, R 765, AGARD, 1985.