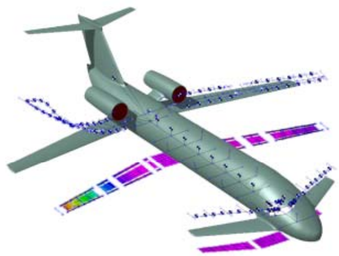
NeoCASS Tutorial

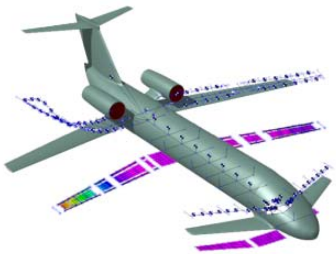
Running NeoCASS from the command window without the GUI “some guide lines”



Version 2.2(.790)

August 2017

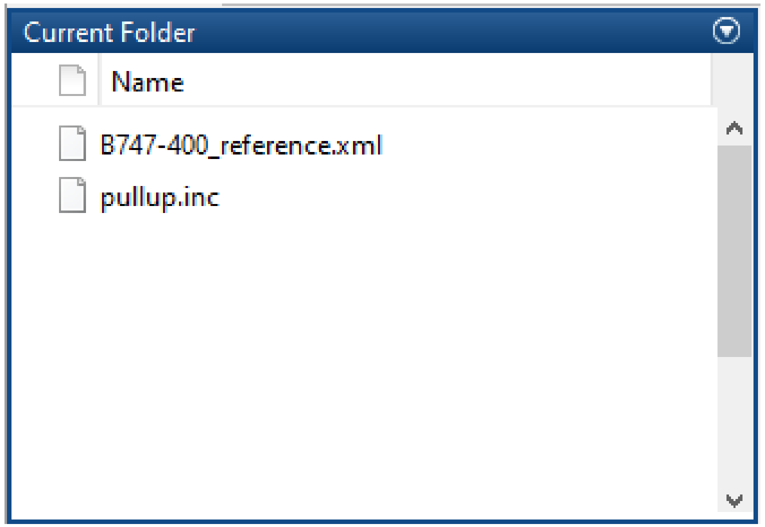
Outline

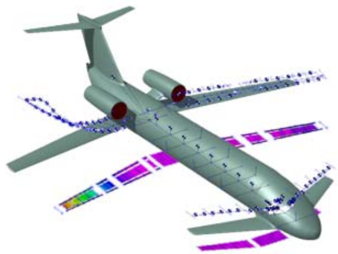


GUESS model

Run the script ***set_neocass_path.m*** in the installation directory. That allows to include the NeoCASS routines into the current path.

Then change directory that you will use for your analysis. Paste in this path the aircraft model (***B747-400_reference.xml***) and the sizing maneuvers file (***pullup.inc***). The first file is created through Acbuilder, and the last one could be created either automatically, using the NeoCASS GUI, or manually, knowing the actual file’s format.





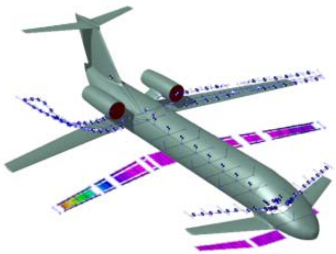
GUESS model

Impose the desired number of iteration till convergence respecting a certain tolerance:

global NMAX EPS	*(set as global)*
NMAX = 3;	*(number of iterations)*
EPS = 1e-3;	*(tolerance)*

Hypothesis: The *.xml* file contains all the a/c information (general geometry, stick model and technology). However, the 'guess' function deals the parts separately, so one should extract the stick model from the original file.

[~, stick_filename] = fileparts(xmlfilename);
stick = [stick_filename '.inc'];



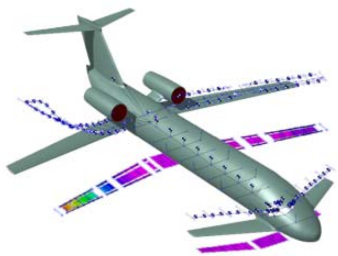
GUESS model

Moreover, some other model clarifications must be set. Considering the reference B747 aircraft, fill the following fields:

model.Joined_wing = false;
model.Strut_wing = false;
model.EnvPoints = [ ];
model.guessstd = true;

One last step, before starting the guess model generation, is initializing the process by creating a void structure:

init_guess_model;



GUESS model

Now it is time to process the raw aircraft using the ***guess.m*** function:

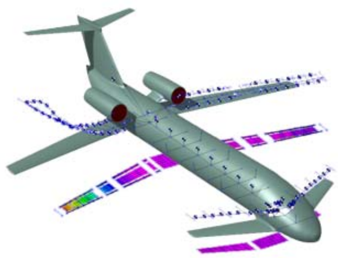
guess_model = guess(xmlfilename, xmlfilename, stick , trim_incfilename, model, false);

In this case:

guess_model = guess('B747-400_reference.xml', 'B747-400_reference.xml', stick, 'pullup.inc', model, false);

```
------------------------------------------ CONVERGENCE --------------------
- Refinement loop history:
    Iter    1: Total structural mass: 135299 Kg. Tolerance: 1.323e-02.
    Iter    2: Total structural mass: 133217 Kg. Tolerance: 1.518e-02.

- GUESS model saved in B747-400_reference_guess.mat file.
- GUESS summary saved in B747-400_reference_guess.txt file.
- SMARTCAD main file with OEW configuration saved in B747-400_reference.inc.
- SMARTCAD configuration file saved in B747-400_referenceCONM_CONF1.inc file.
```



SMARTCAD : TRIM problem

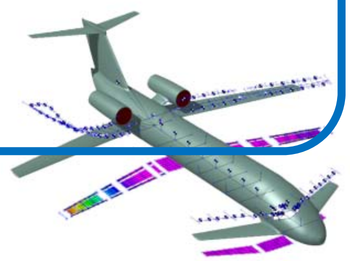
At the first phase one have to provide all the essential inputs for performing the trim analysis. Two inputs are, of course, the results of the previous step GUESS: the aircraft and the configuration file (both are '.inc').

Secondly, one have to provide a file containing the trim condition, that could differ from the dimensioning one (pullup.inc)

Another input file would be the one containing the **reference parameters** and the **solver ID**: **trim_param.inc**. It has the structure below:

```
SOL 144
$ Vortex lattice solver parameters
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
AEROS           4       25      125     0       0       0
```

Ref.chord = 4m;   ref.span = 25m;   ref.Surface = 125m^2



SMARTCAD : TRIM problem

One have to create and INCLUDE file named trim.dat that collect all the input required by the trim solver. The trim.dat file has the following structure:

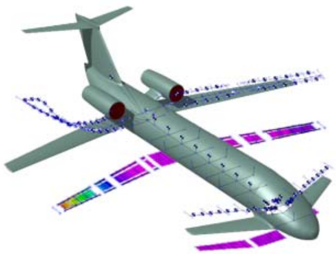
```
INCLUDE C:\directory\Nameac.inc
INCLUDE C:\directory\NameacCONM_CONF1.inc
INCLUDE C:\directory\pullup.inc
INCLUDE C:\directory\trim_param.inc
```

Now everything is ready to performing the trim analysis. Use these commands to load the model and start the solver:

```
global guess_model
guess_model = load_nastran_model('Nameac.inc');
solve_free_lin_trim();
```

At the end of the process the solver communicates where the results are saved:

Solution summary exported to *C:\directory\trim_man_1.txt* file.



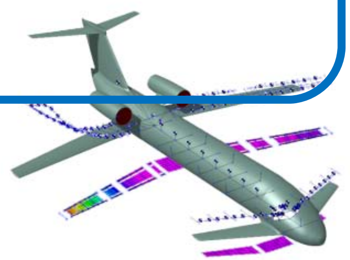
SMATCAD : FLUTTER problem

Initially one have to provide all the essential inputs for performing the MODAL analysis. Two inputs are, of course, the results of the previous step GUESS: the aircraft and the configuration file (both are '.inc').

Secondly, one have to provide a file containing the **reference parameters**, the **solver ID** and the modes driving information: **modal_param.inc**. It has the following structure:

```
SOL 103
$ Vortex lattice solver parameters
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
AEROS           4       25      125     0       0       0
$ Eigenvalue solver parameters
$-------2-------3-------4-------5-------6-------7-------8-------9-------10
EIGR    1               0       100             30
        MASS
```

frequency range [0,100] Hz; select the first 30 modes



SMATCAD : FLUTTER problem

One have to create and INCLUDE file named **modal.dat** that collect all the input required by the modal solver. The modal.dat file has the following structure:

```
INCLUDE C:\directory\Nameac.inc
INCLUDE C:\directory\NameacCONM_CONF1.inc
INCLUDE C:\directory\modal_param.inc
```

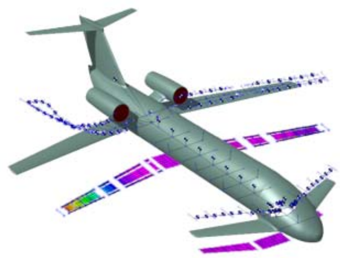
Now everything is ready to performing the modal analysis. Use these commands to load the model and start the solver:

global **guess_model**

guess_model = load_nastran_model(‘Nameac.inc’);

solve_eig();

At the end of the process the results will be displayed in the command window



SMATCAD : FLUTTER problem

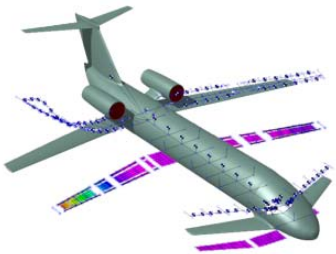
As usual, one have to create the INCLUDE file named **flutter.dat** that collect all the input required by the flutter solver. The flutter.dat file has the following structure:

```
INCLUDE C:\directory\Nameac.inc
INCLUDE C:\directory\NameacCONM_CONF1.inc
INCLUDE C:\directory\flutter_param.inc
```

… where the *flutter_param.inc* is shown in the following page. In this file one have to select the mode of interest for further analysis from the results' domain of the previous modal solver.

At this point one could use the ***run.flutter*** function, however using another, higher level, function is more straightforward: ***init_dyn_model(****flutter.dat****)***

The parameters decribed in flutter_param.inc are sufficient for flutter analysis, even if the init_dyn_model function may require more input for a further dynamic response analysis with NeoRESP.



SMATCAD : FLUTTER problem

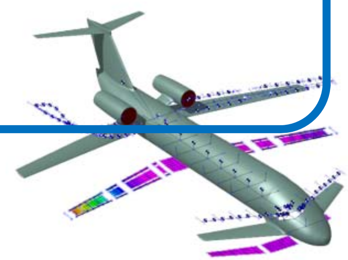
```
$-------2-------3-------4-------5-------6-------7-------8---
$ VLM parameters
$-------2-------3-------4-------5-------6-------7-------8-
AEROS           4     36.2   126.8  0     0     0
SOL 145
$-------2-------3-------4-------5-------6-------7-------8-
$ Flutter solver parameters
$-------2-------3-------4-------5-------6-------7-------8-
AERO   126.8  300    4     1.225  0     0     2     50
36.2
$-------2-------3-------4-------5-------6-------7-------8-
$ Reduced frequency and Mach number list
$-------2-------3-------4-------5-------6-------7-------8-
MKAERO1 0.5
        0.001  0.01   0.05   0.1    0.2    0.5    1.0    2.0
$-------2-------3-------4-------5-------6-------7-------8-
$ Eigen-analysis cards
$-------2-------3-------4-------5-------6-------7-------8-
EIGR    1             0     1000         30
        MASS
```

.......cont......

......cont......

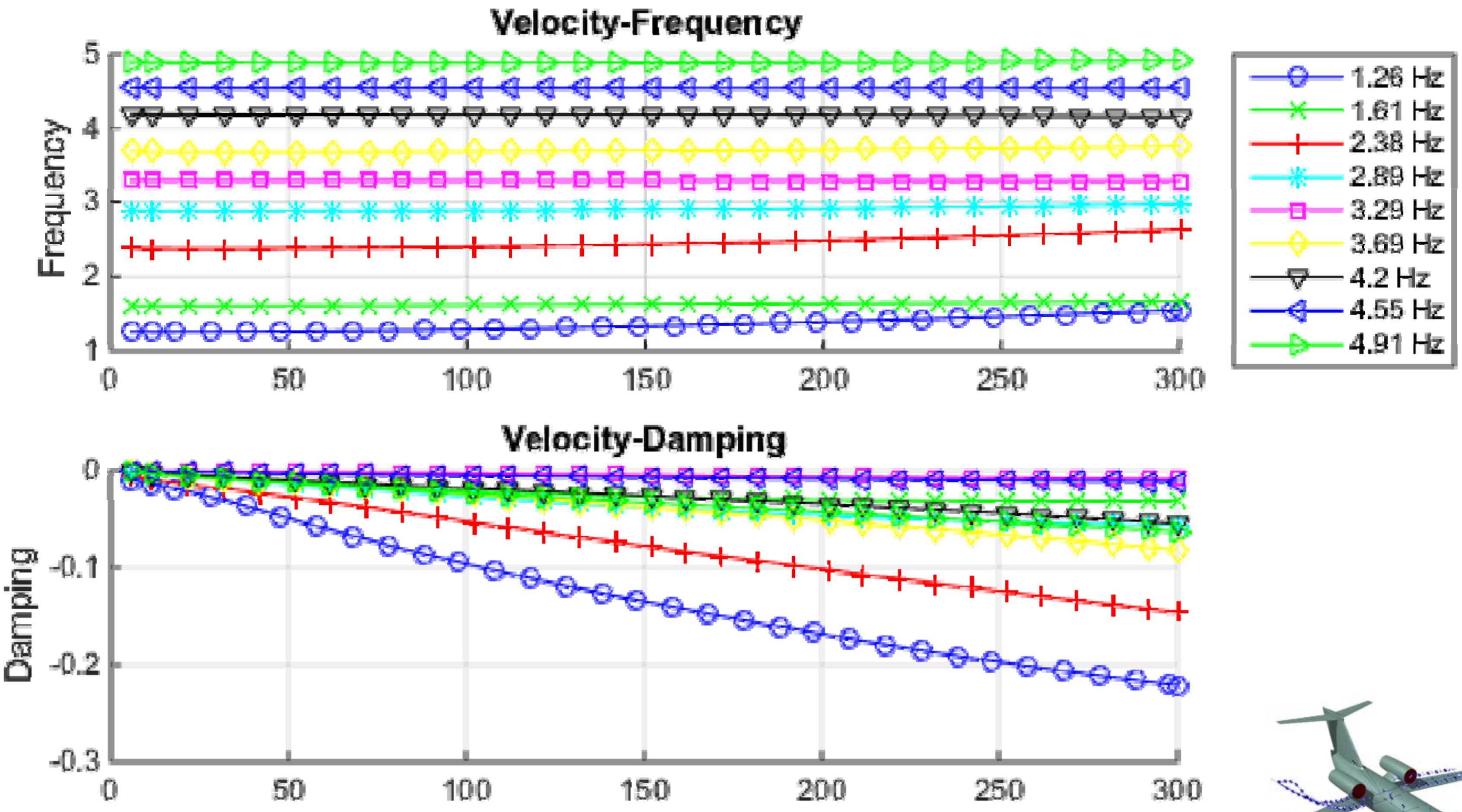
```
$-------2-------3-------4-------5-------6-------7-------8-
$ Select modes from an external database
$-------2-------3-------4-------5-------6-------7-------8-
MSELECT     1      2      3      4      5      6      7      8
            9      10     11     12     13     14     15     16
            17     18     19     20     21     22     23     24
            25     26     27     28
$-------2-------3-------4-------5-------6-------7-------8-
$ Modes to follow in flutter solution
$-------2-------3-------4-------5-------6-------7-------8-
FMODES      7      8      9      10     11     12     13
            14     15
```

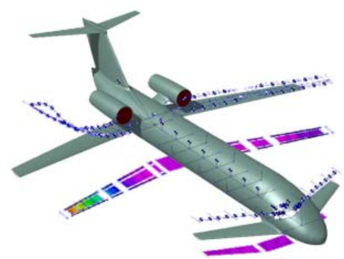
___ Flutter_param.inc structure ___



SMATCAD : FLUTTER problem

V-g plot resulting from flutter analysis. In this case, no flutter is detected.

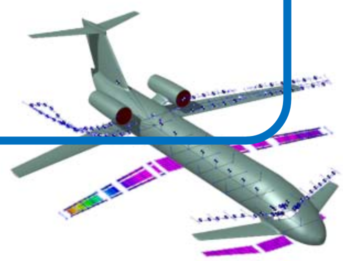




PLOTs review

In this section the main predefined plot available in NeoCASS are highlighted.
In particular:

- Placard diagram
- Velocity margins vs altitude
- section-wise sizing shear force in fuselage, wing, vertical and horizontal tail
- section-wise sizing bending moment in fuselage, wing, vertical and horizontal tail
- section-wise moment of inertia of fuselage
- section-wise sizing torque of wing, vertical tail and horizontal tail
- Fuselage thickness
- *s*pan-wise skin thickness of wing, vertical tail and horizontal tail
- *s*pan-wise web thickness of wing, vertical tail and horizontal tail
- Out-of-plane moment of inertia (I1 [m4]) wing, vertical tail and horizontal tail
- In-plane moment of inertia (I2 [m4]) wing, vertical tail and horizontal tail
- torsional constant (J [m4]) wing, vertical tail and horizontal tail
- bearing structure mass distribution along fuselage, wing, vertical tail and horizontal tail
- etc ……



PLOTs review

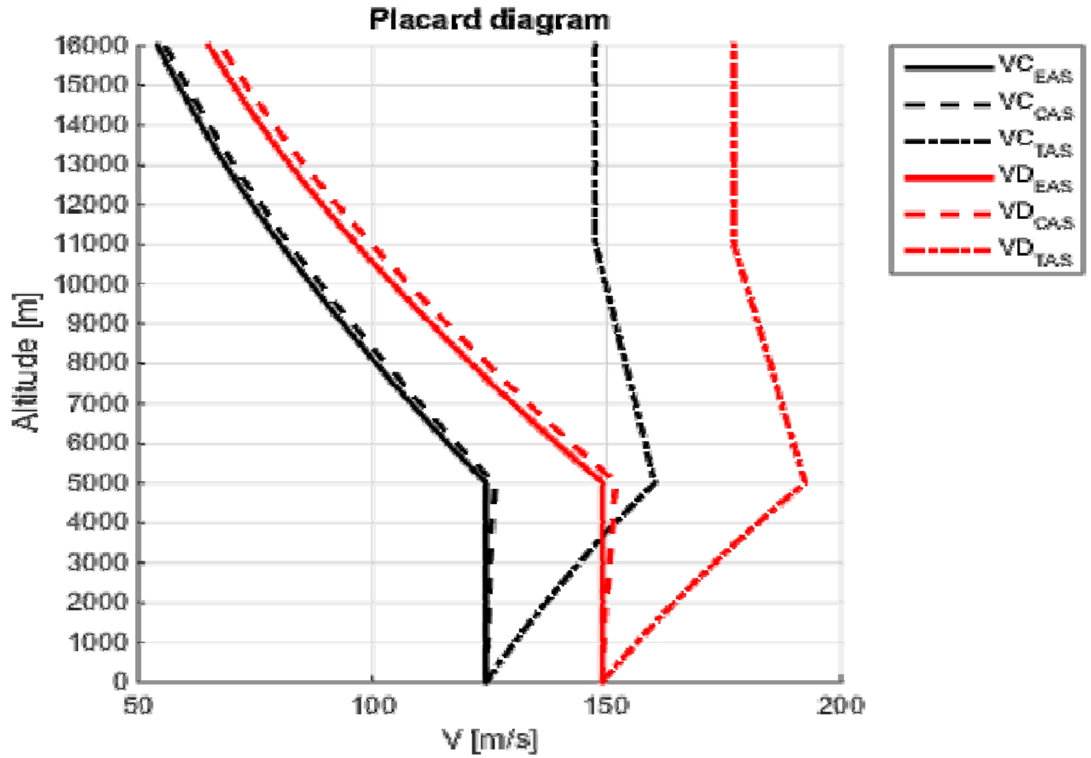
In order to plot the Placard diagram use:

placard*(nfig, HC, MC, HD, MD)*

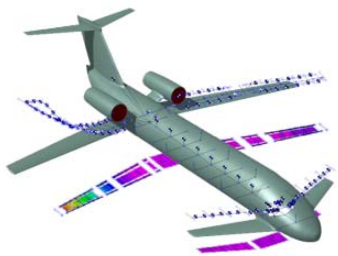
nfig : figure number

MC, MD : Mach n. Cruise and Dive

HC, HD : altitude at which the previous velocities are defined [m]



For example:
placard(1, 5000, 0.5, 5000, 0.6)

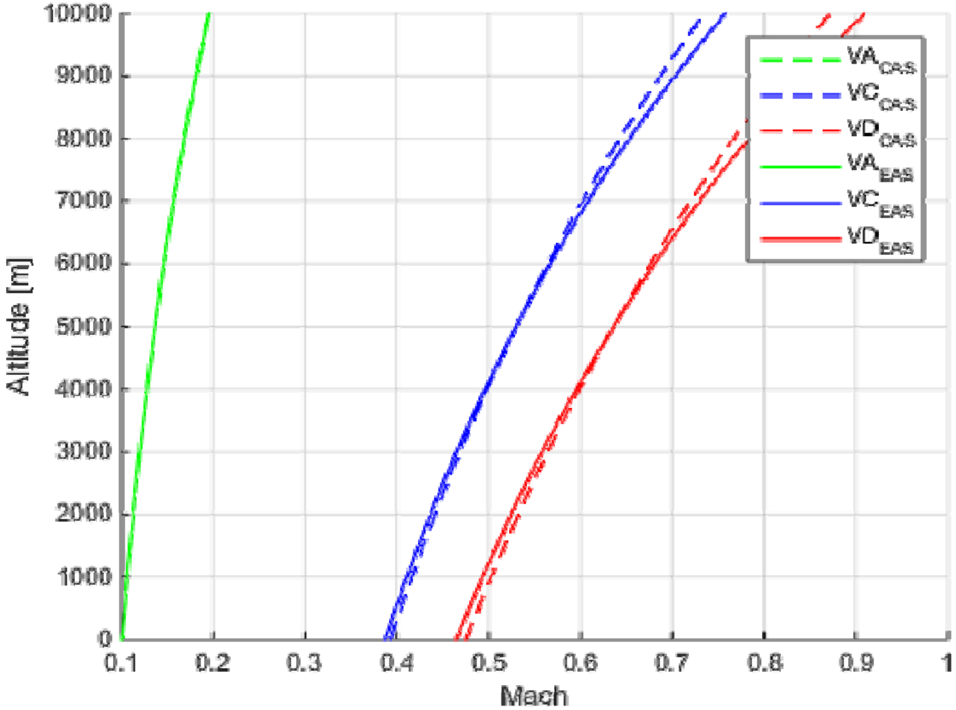


PLOTs review

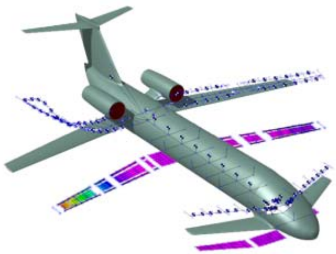
In order to plot velocity margins as function of altitude use:

plot_zM*(nfig, VA, HA, VC, HC, VD, HD)*

nfig : figure number *VA, VC, VD* : characteristic velocities [m/s]
HA, HC, HD : altitude at which the previous velocities are defined [m]



For example:
plot_zM(2,35,500,170,5000,204,5000)

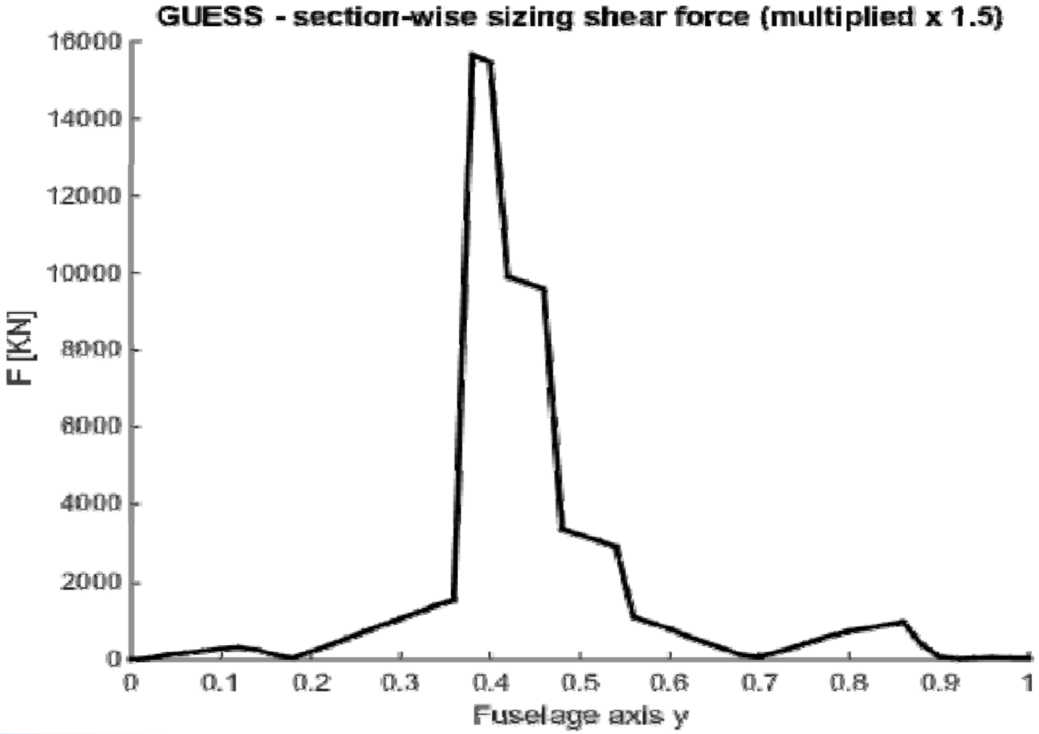


PLOTs review

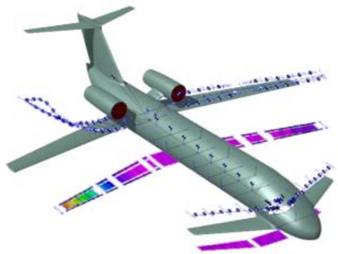
Loading the guess module's results (***load guess_model***) one could plot many interesting variables trends along the main aircraft structures. Use:

plot_guess_model*(set, fignr, guess_model)*

fignr: figure number   *guess_model*: name of the file where guess results are stored
set = 1,2,3,4,6,7,8,11,12 or13 corresponds to predefined plot

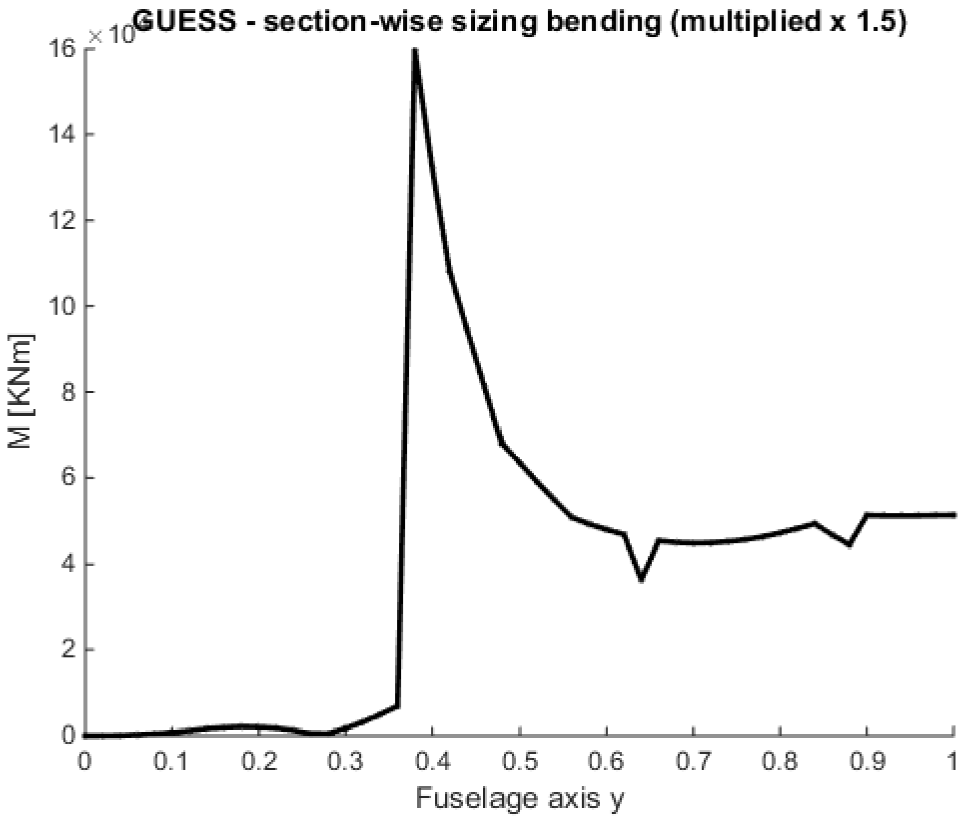


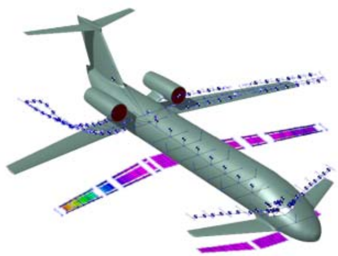
Set=1 section-wise sizing shear force in the fuselage



PLOTs review

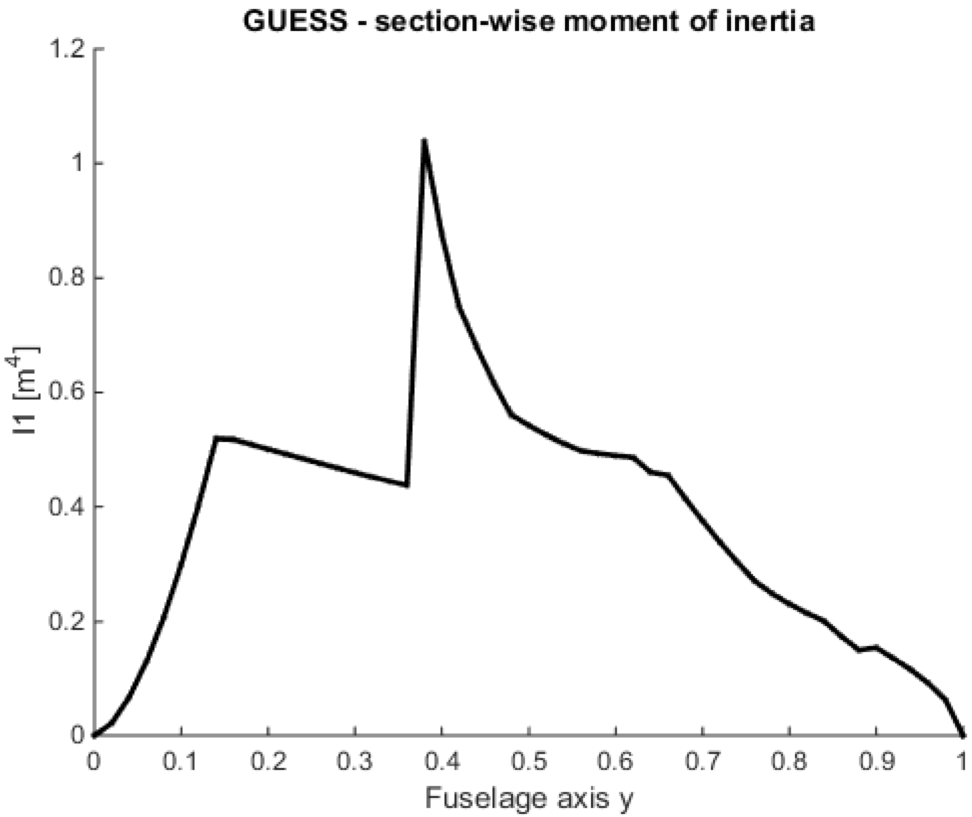
The same will be for the next figures.
Set=2: section-wise sizing bending (x1,5) in the fuselage

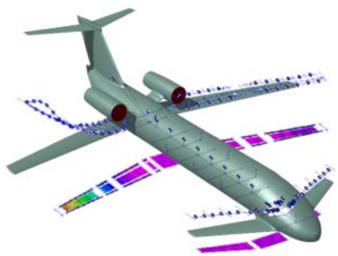




PLOTs review

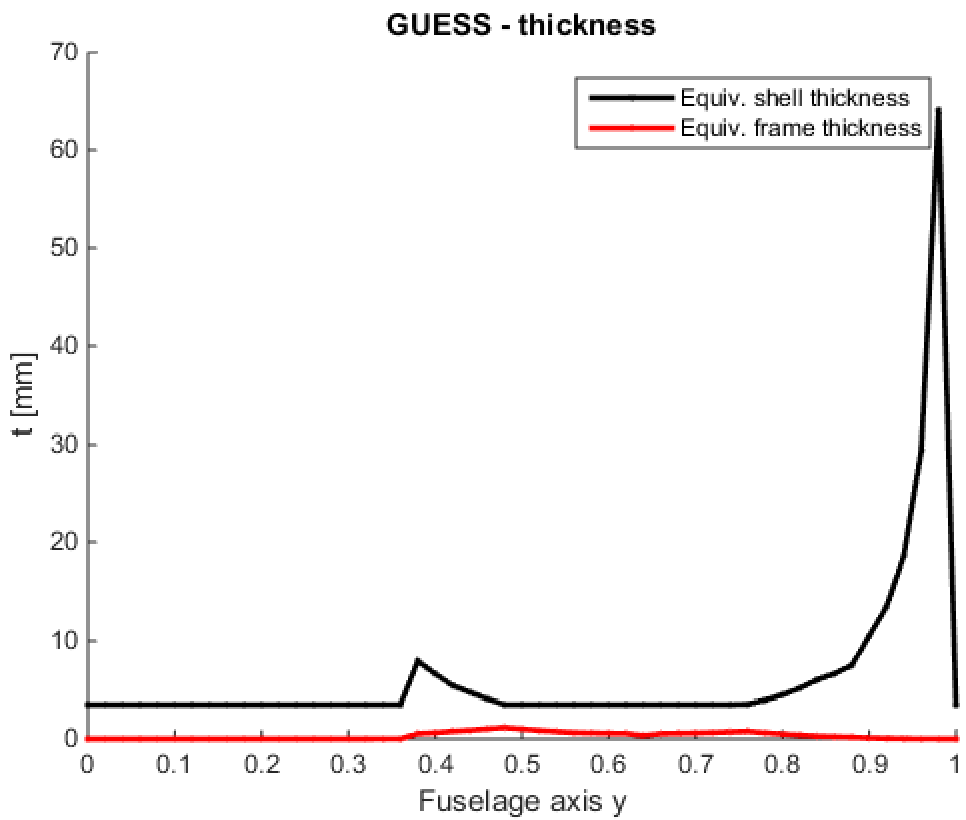
Set=3: section-wise moment of inertia of fuselage

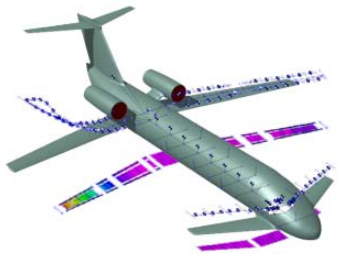




PLOTs review

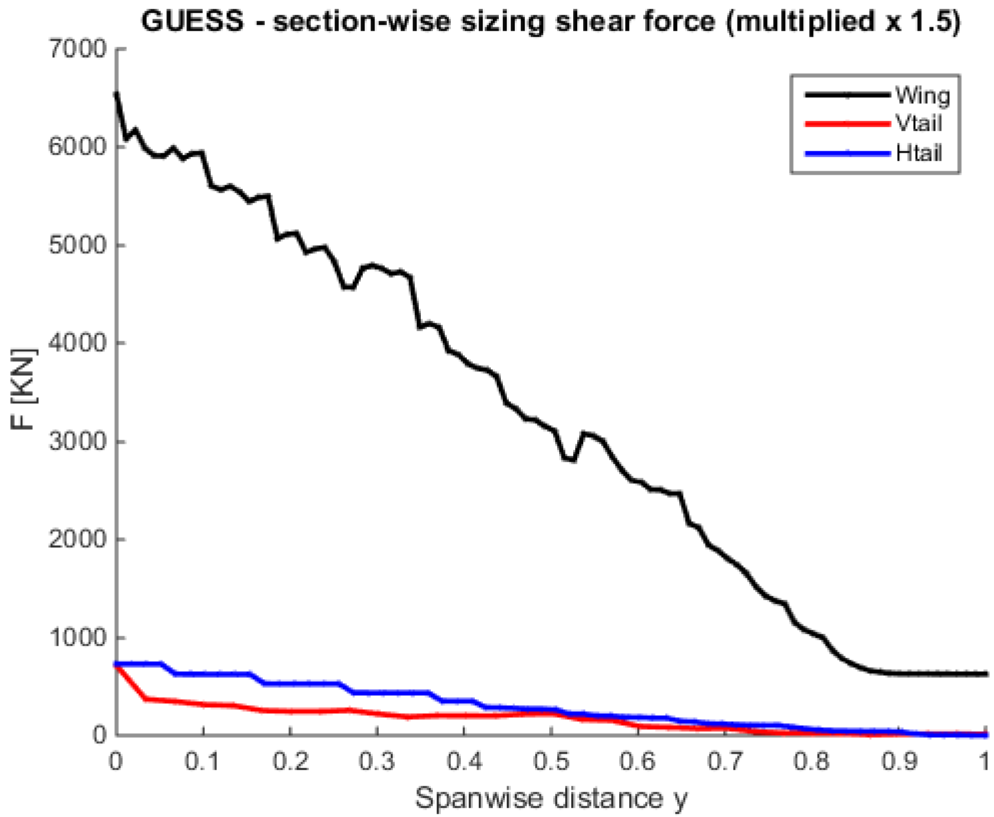
Set=4: Fuselage thickness

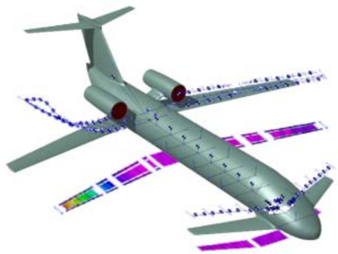




PLOTs review

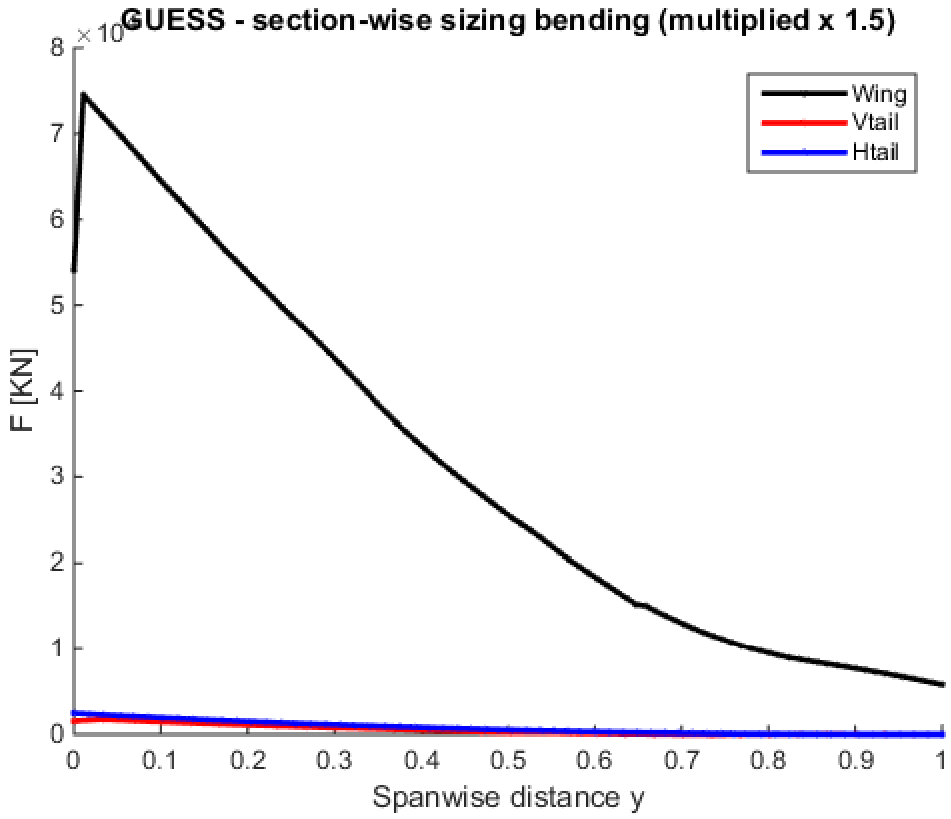
Set=6: section-wise sizing shear force (x1,5) wing, vertical tail and horizontal tail

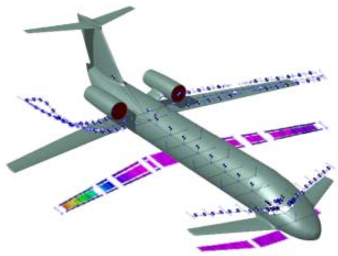




PLOTs review

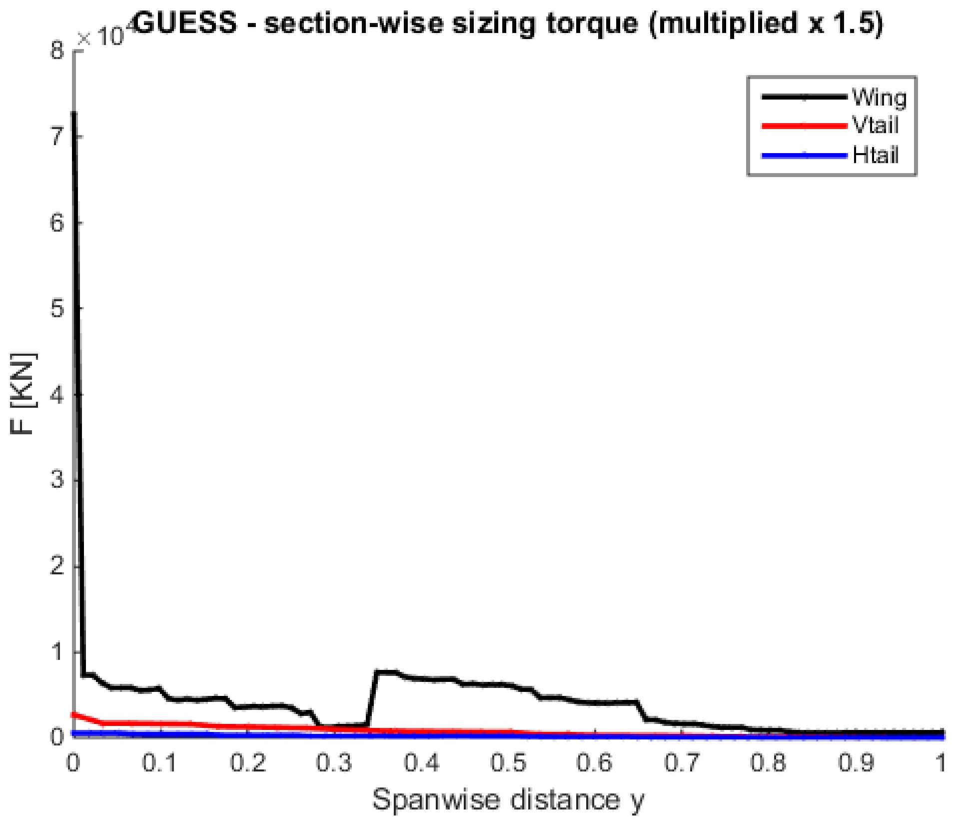
Set=7: section-wise sizing bending (x1,5) wing, vertical tail and horizontal tail

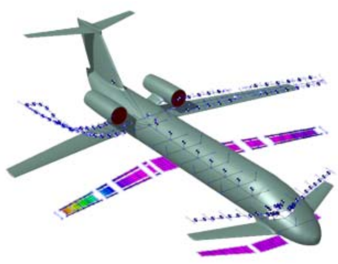




PLOTs review

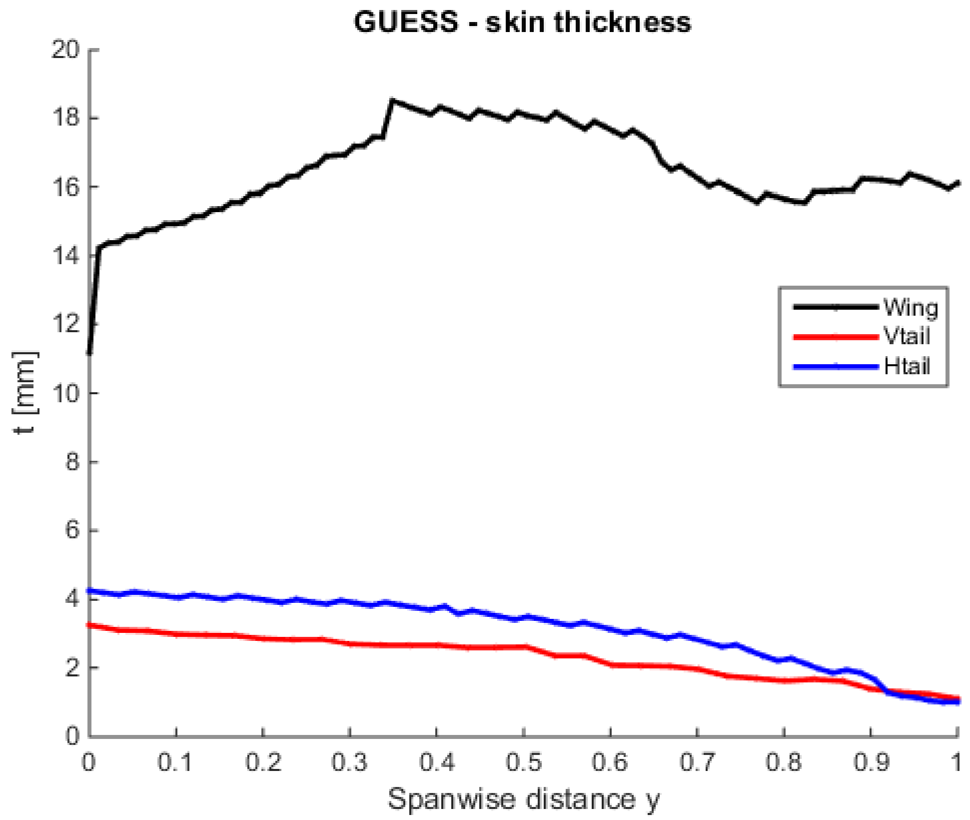
Set=8: section-wise sizing torque (x1,5) wing, vertical tail and horizontal tail

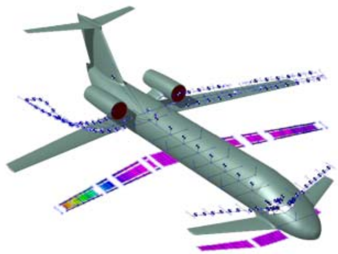




PLOTs review

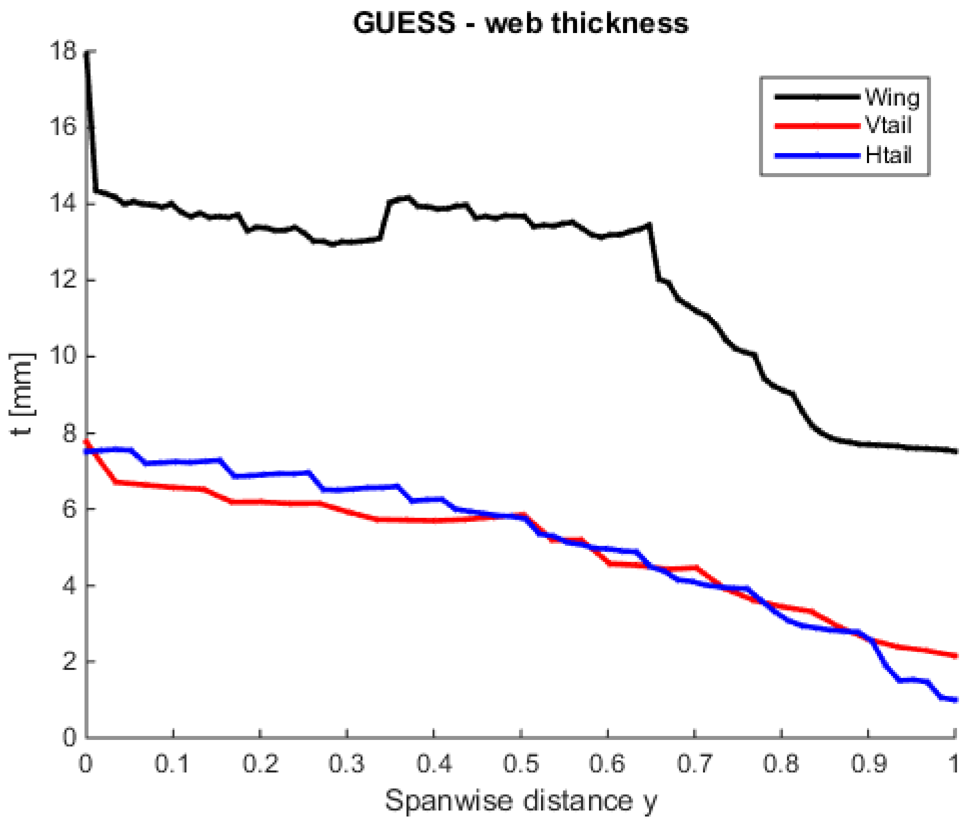
Set=9: span-wise skin thickness of wing, vertical tail and horizontal tail

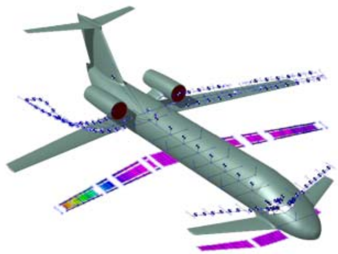




PLOTs review

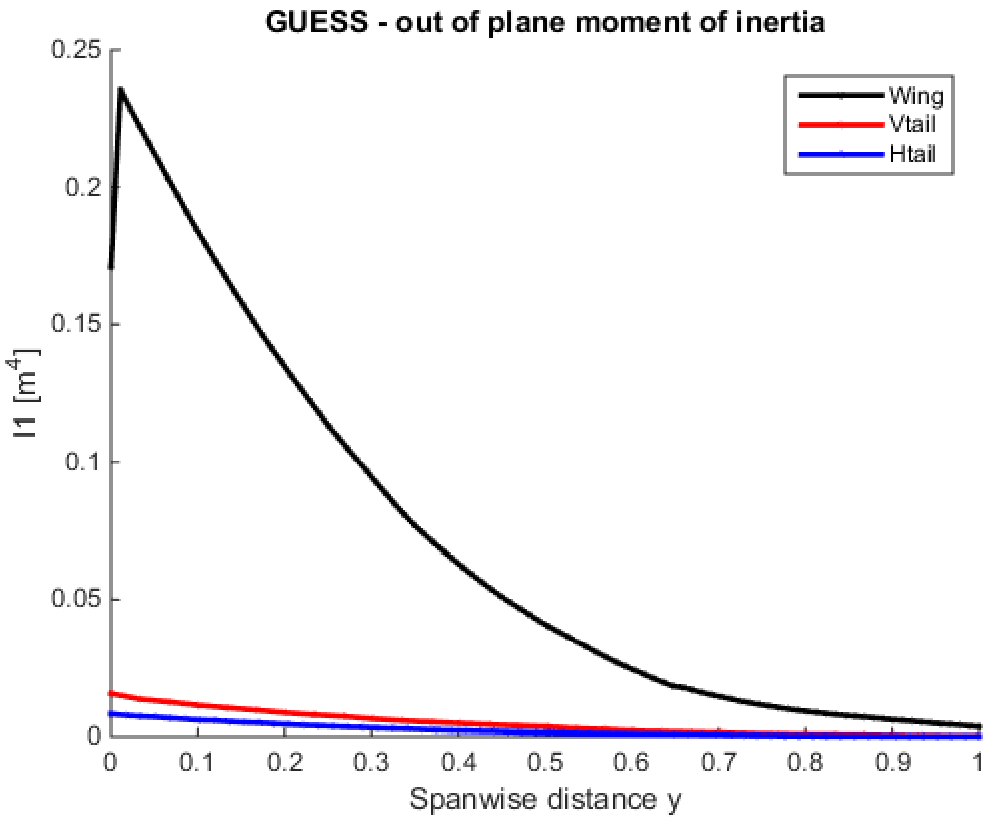
Set=10: span-wise web thickness of wing, vertical tail and horizontal tail

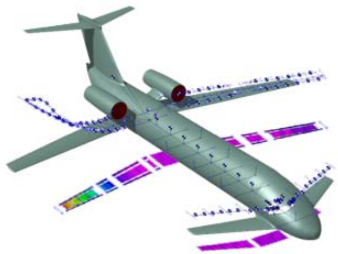




PLOTs review

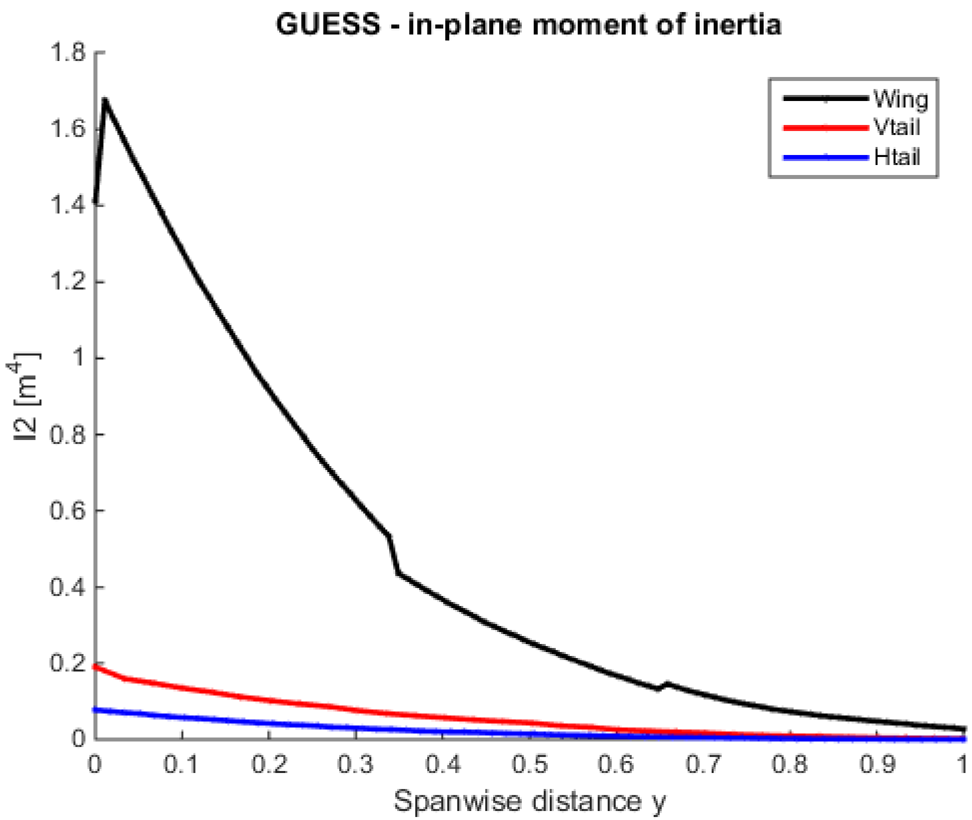
Set=11: Out-of-plane moment of inertia (I1 [m4]) wing, vertical tail and horizontal tail

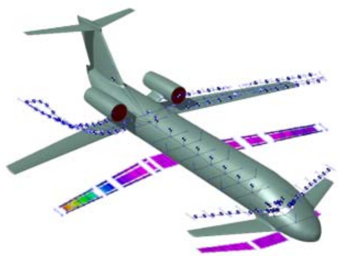




PLOTs review

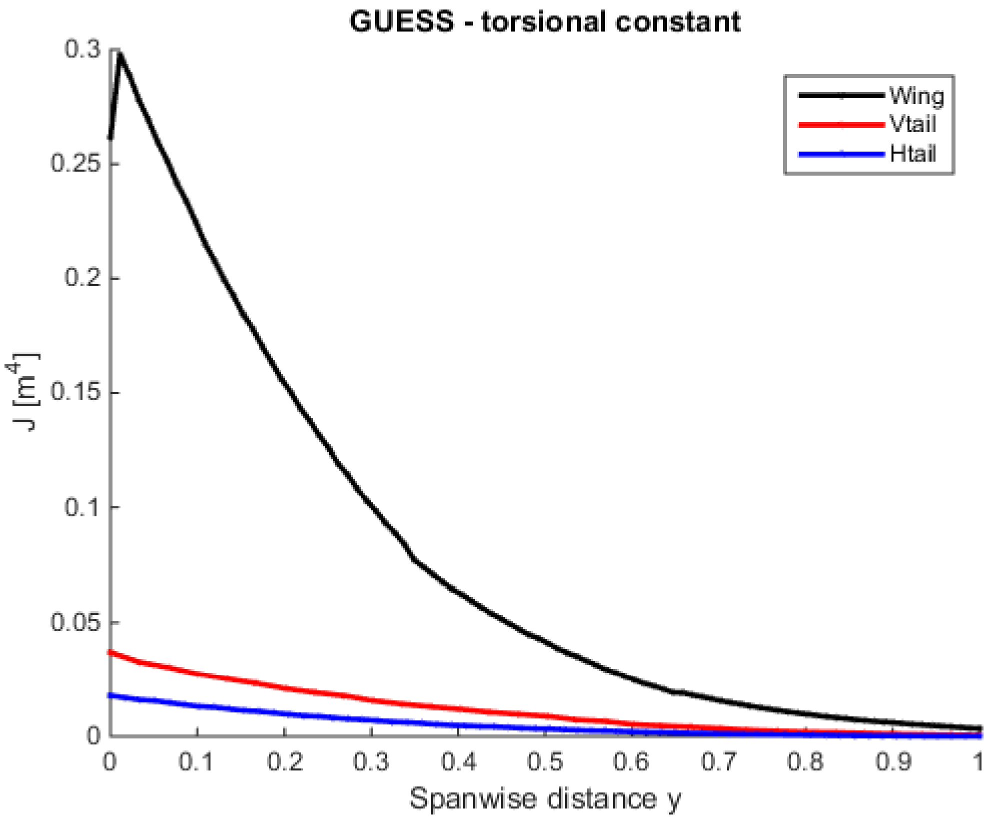
Set 12: in-plane moment of inertia (I2 [m4]) wing, vertical tail and horizontal tail

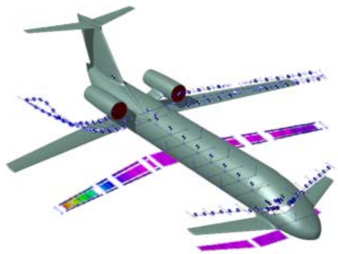




PLOTs review

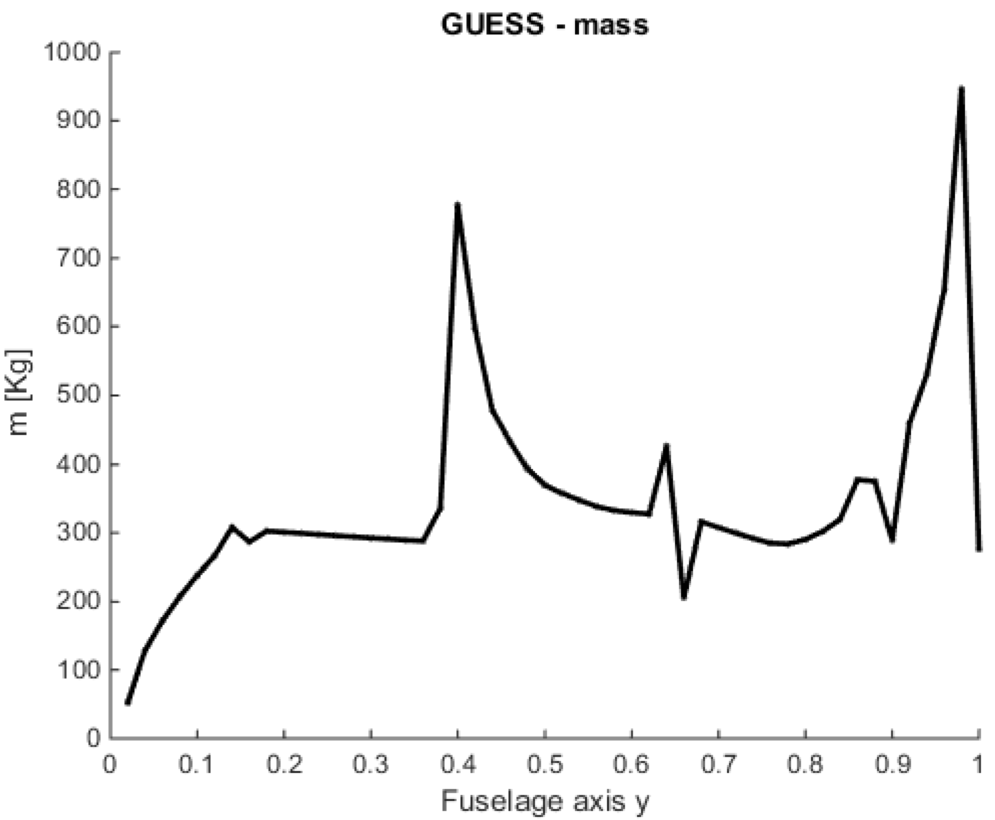
Set 13: torsional constant (J [m4]) wing, vertical tail and horizontal tail

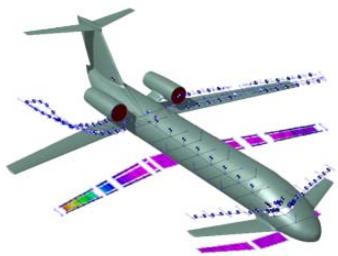




PLOTs review

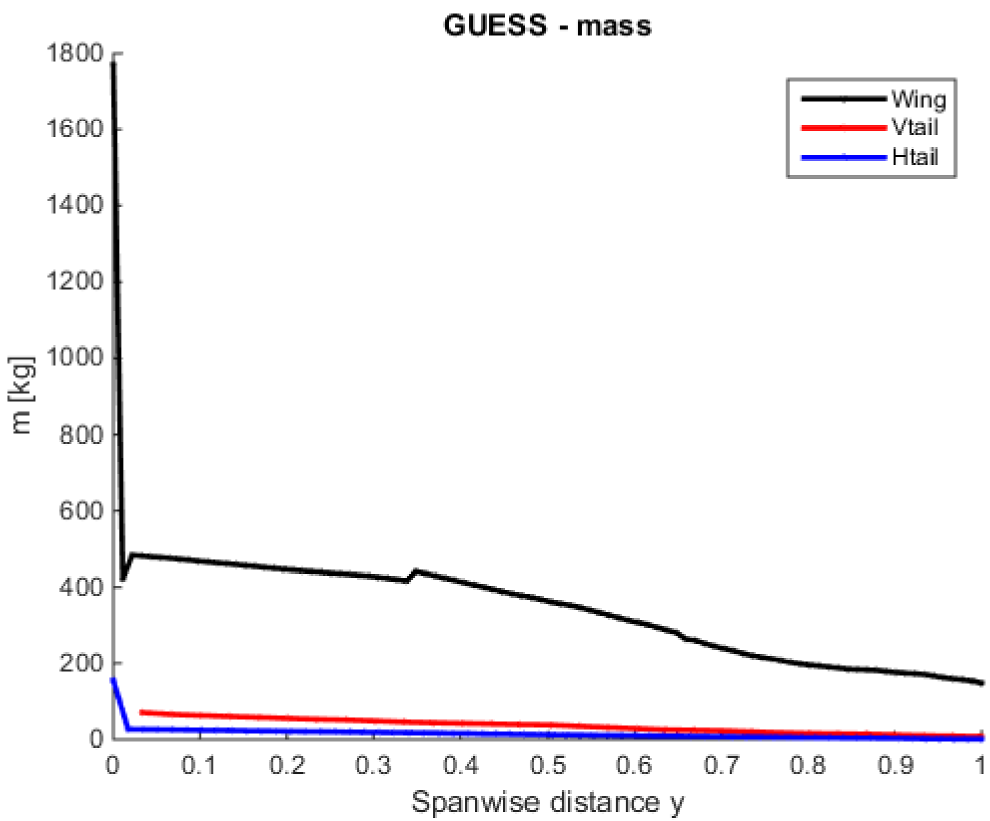
Set 14: bearing structure mass distribution along the fuselage

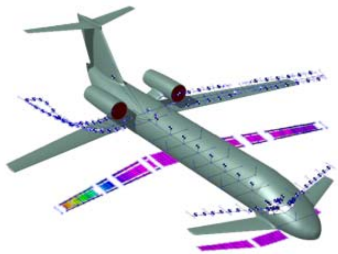




PLOTs review

Set 15: bearing structure mass distribution along wing, vertical tail and horizontal tail





PLOTs review

Internal Forces

Use the following command after performing at least a trim analysis:

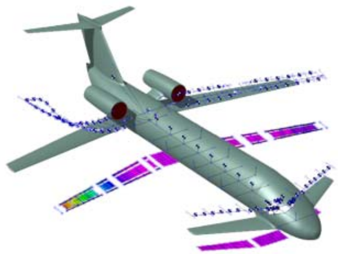
global *beam_model*
AI_plot*(ifig, IND, NLEVEL)*

where:

ifig is the index of figure

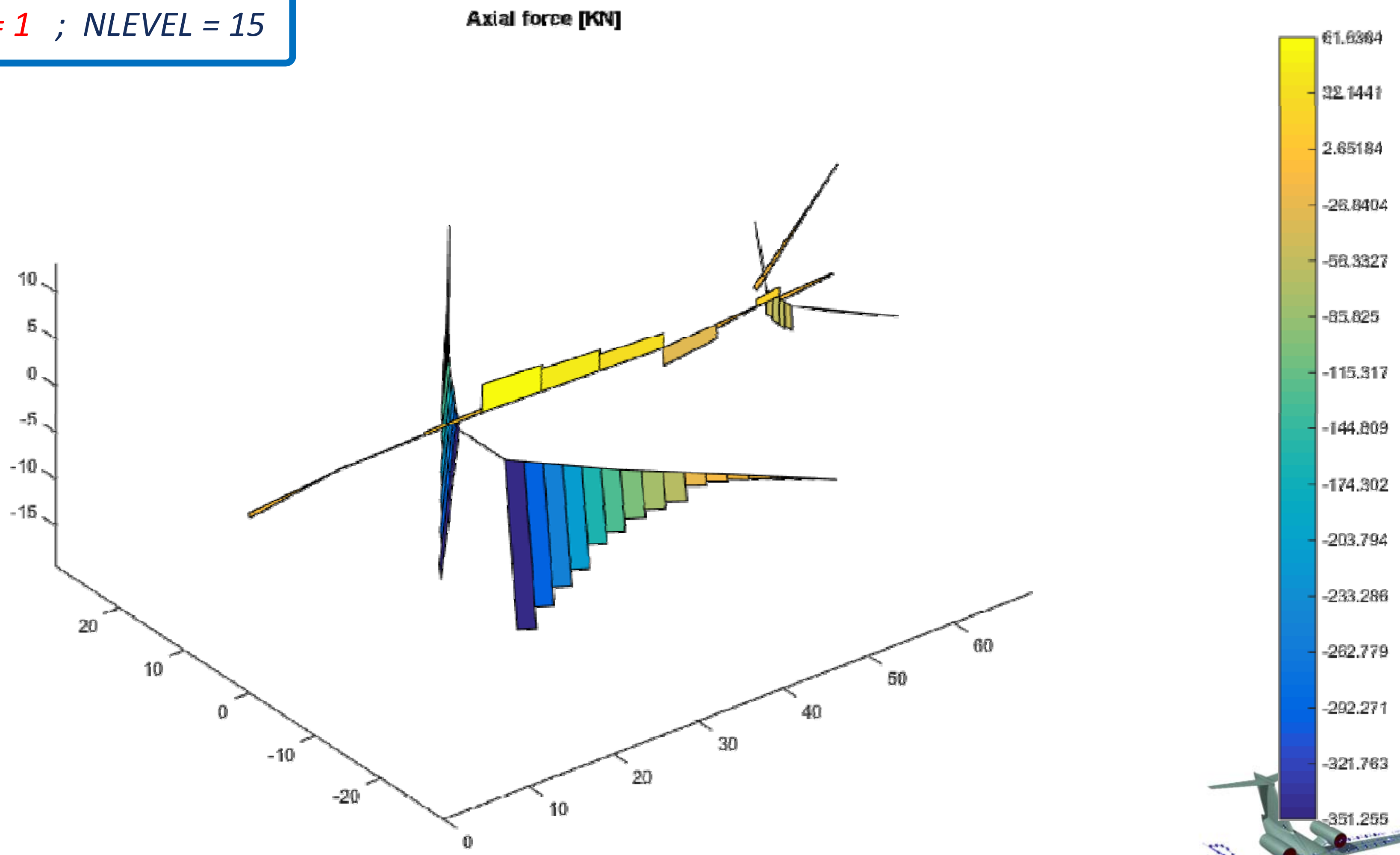
IND is the index of internal load that is needed
IND = 1 : local **Tx**; (axial)
IND = 2 : local **Ty**; (vertical shear)
IND = 3 : local **Tz**; (horiz shear)
IND = 4 : local **Mx**; (torsion)
IND = 5 : local **My**; (out of plane bending)
IND = 6 : local **Mz**; (int of plane bending)

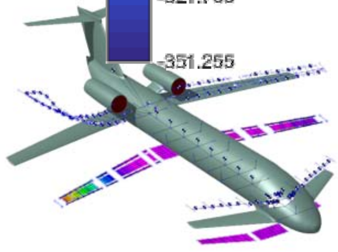
NLEVEL number of levels in contour



PLOTs review

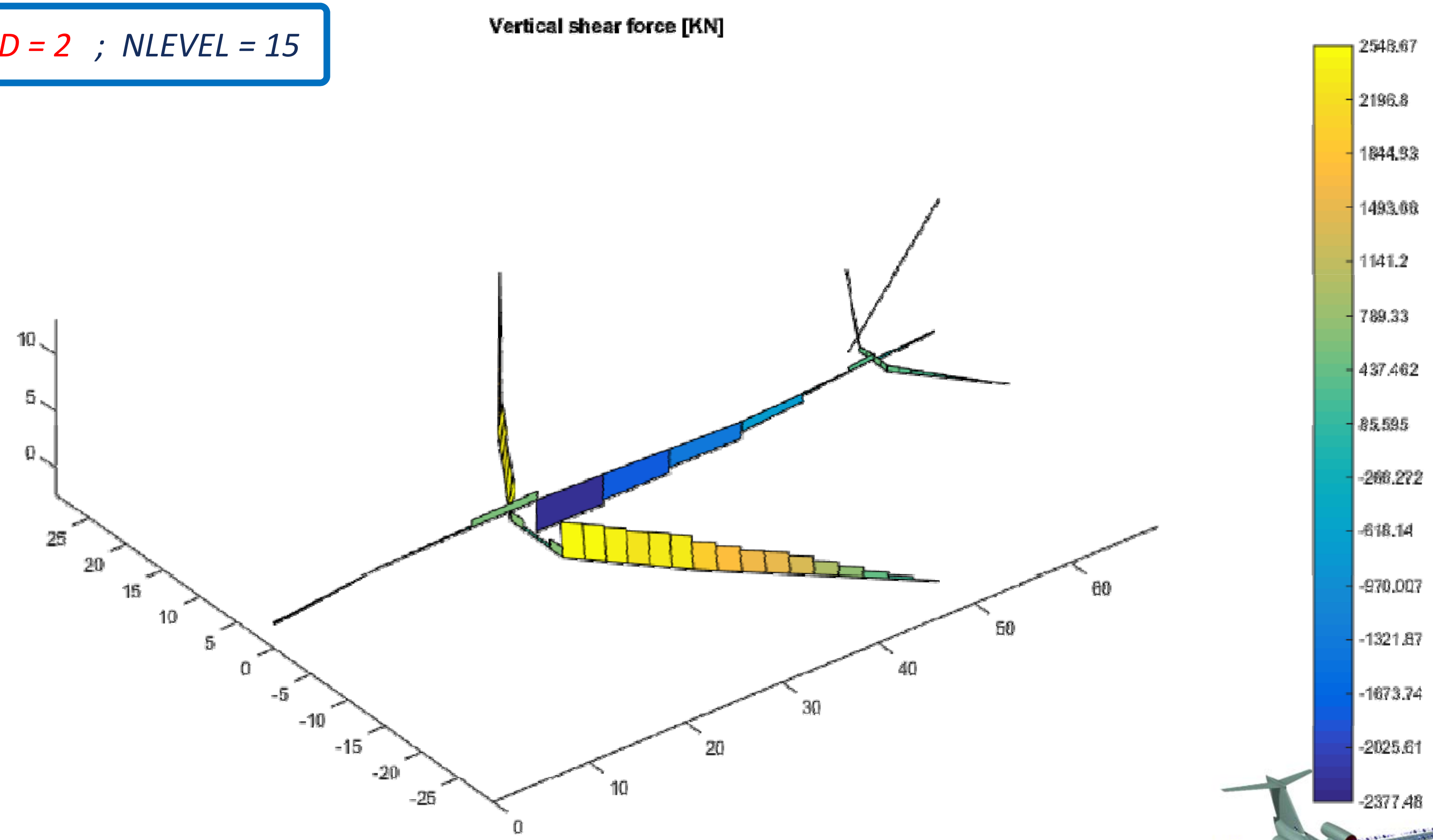
ID = 1 ; NLEVEL = 15

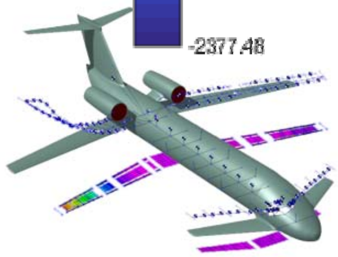




PLOTs review

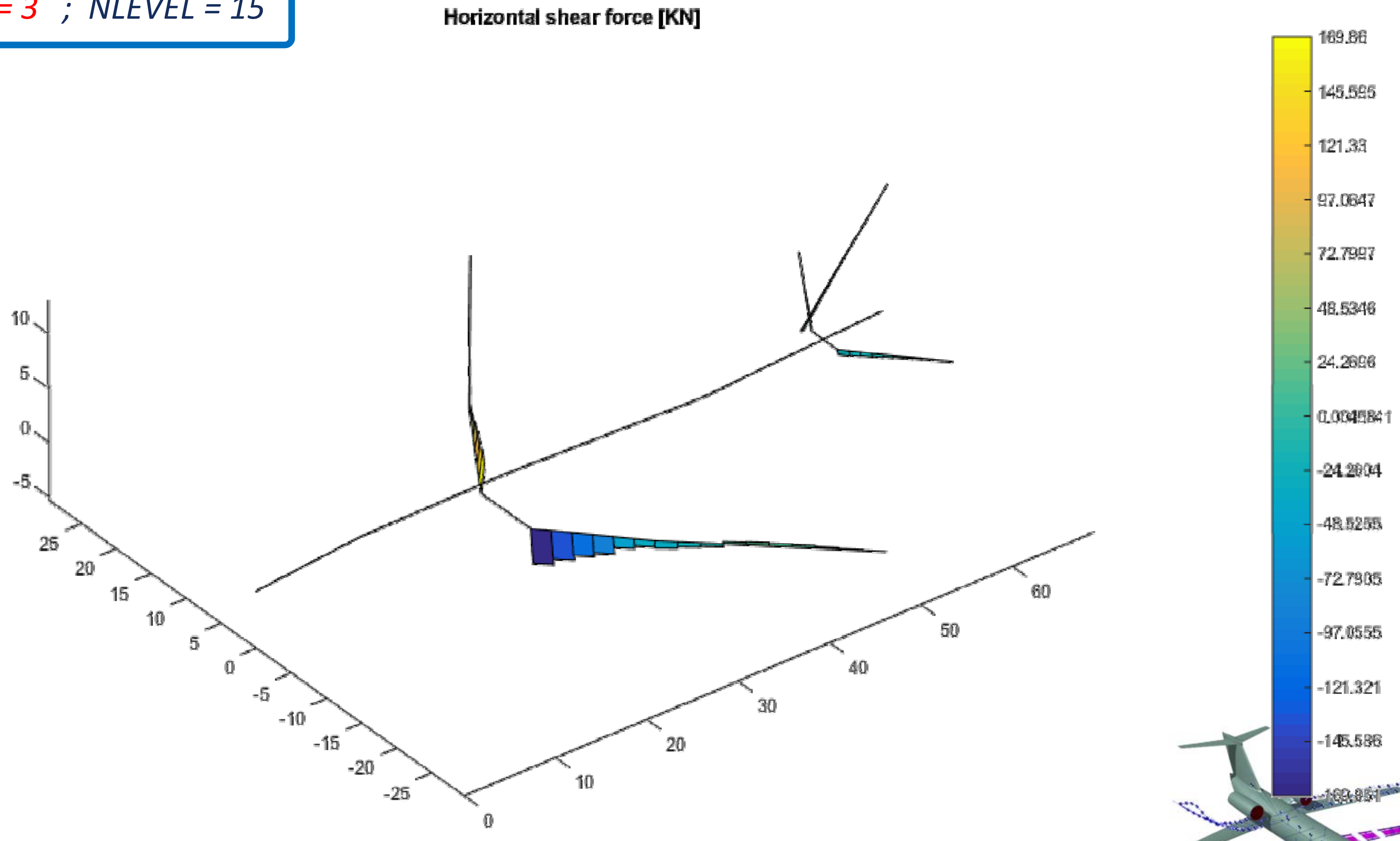
ID = 2 ; NLEVEL = 15





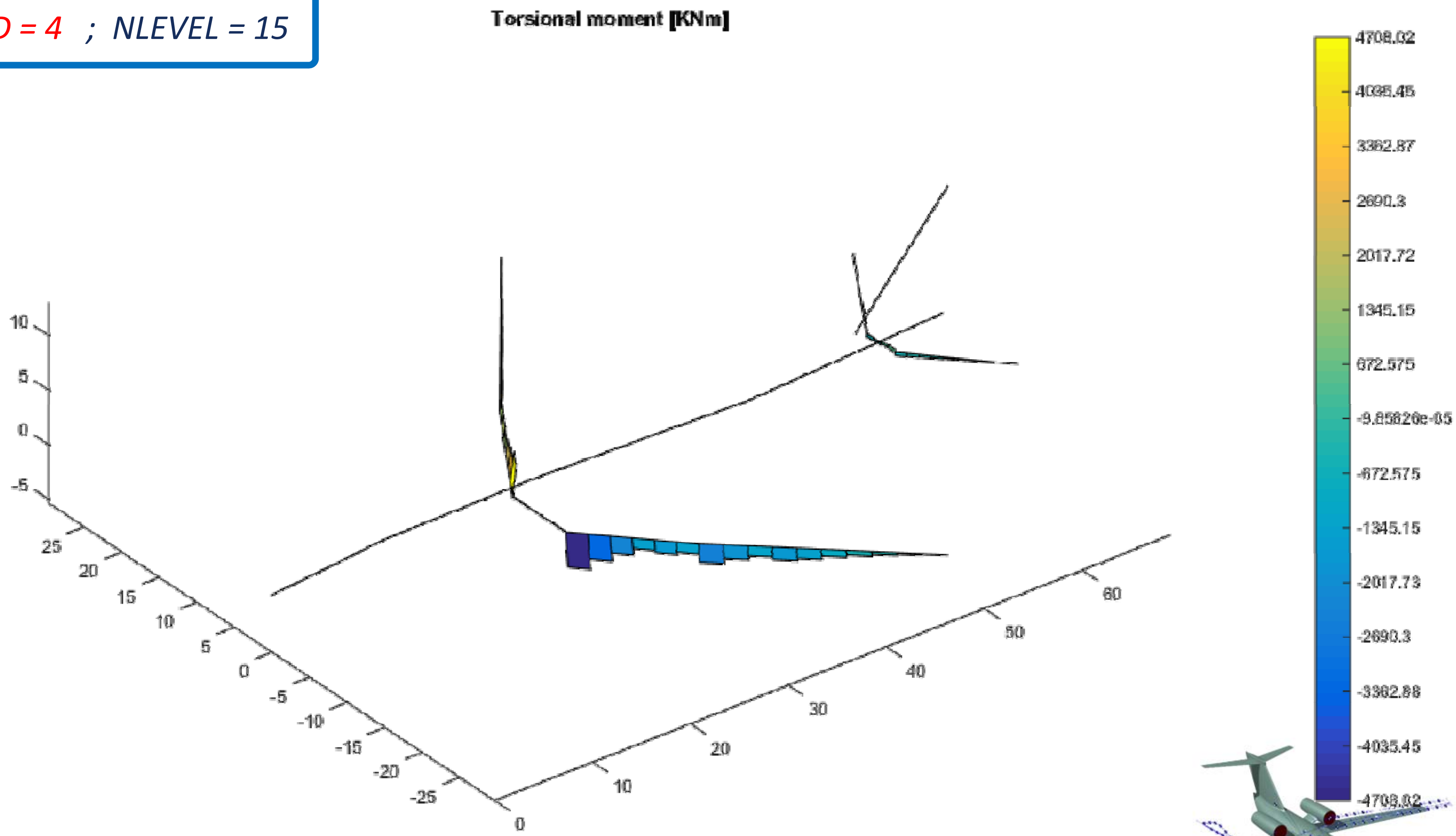
PLOTs review

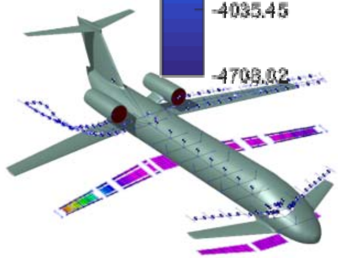
ID = 3 ; NLEVEL = 15



PLOTs review

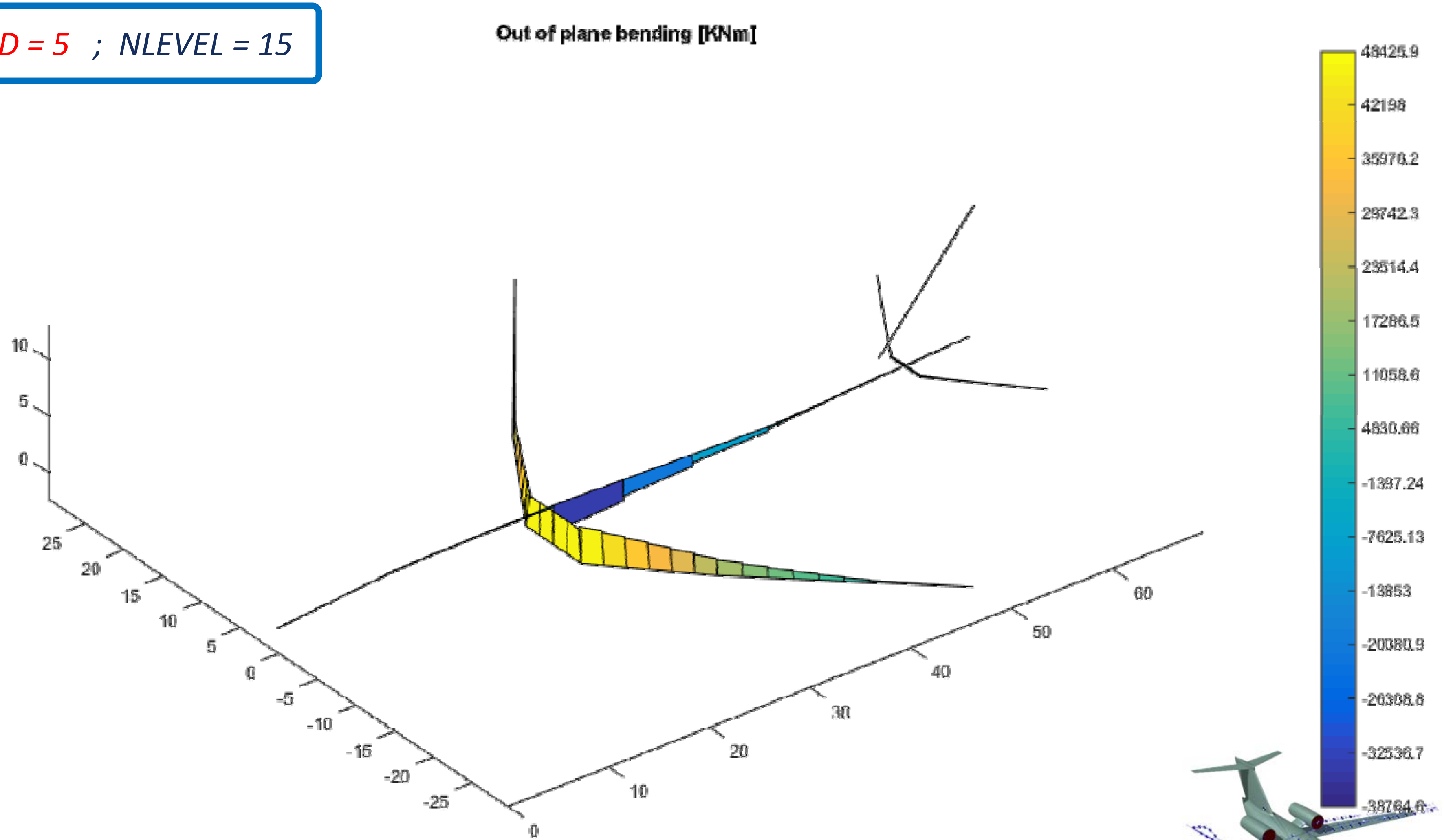
ID = 4 ; NLEVEL = 15

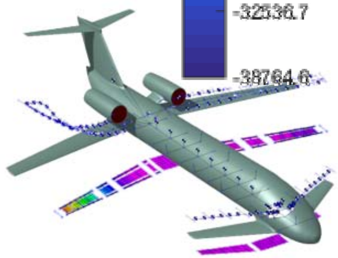




PLOTs review

ID = 5 ; NLEVEL = 15





PLOTs review

ID = 6 ; NLEVEL = 15