UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

Design of Aircraft and Flight technologies

Aircraft design tools developed at the University of Naples Federico II and their integration with DLR software

Agostino De Marco

University of Naples (UniNa) Federico II, Naples – Italy

DAF, Design of Aircraft and Flight technologies research group

www.daf.unina.it

agostino.demarco@unina.it

Workshop: Aircraft geometries with TiGL – 11-12 September 2018, DLR Cologne

Presentation Overview

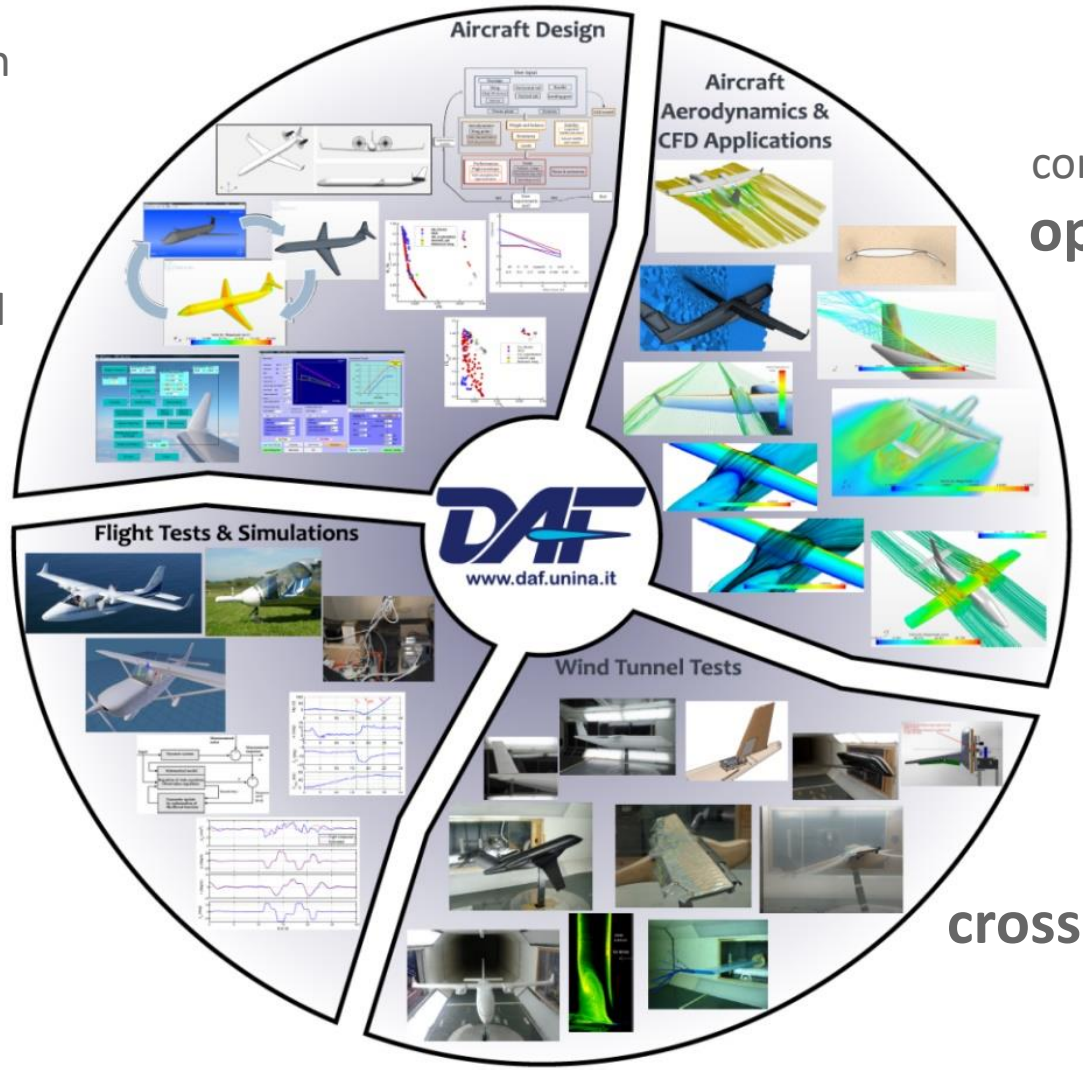
- DAF research unit – Design of Aircraft and Flight technologies @ UniNa
- JPAD – a modular Java API for aircraft designers
- JPADCAD – a CAD module for programmatic aircraft geometric modelling
- UniNa & DLR tools integration (AGILE Project, RCE s/w)

DAF research unit

Design of Aircraft and Flight technologies

design analysis and optimization of light aircraft, general aviation aircraft and commercial transport aircraft

development of software and new methodologies for aircraft design and optimization



aerodynamics analyses on complete aircraft and components

optimization with high fidelity aerodynamic tools

simulations of aircraft, flight quality and parameter estimation

flight tests and aircraft certification

wind tunnel tests on complete aircraft, visualizations and Pressure distributions

cross-validation with high fidelity aerodynamic tools

www.daf.unina.it

DAF research unit

Design of Aircraft and Flight technologies

Aircraft Design & Flight Mechanics

- Design and optimization of light aircraft, general aviation aircraft and commercial transport aircraft.
- Cooperations with:
  - *Tecnam Aircraft Industries*
  - *ATR*
  - *Leonardo Company*
- Development of new methodologies for aircraft preliminary design.
- Development of advanced software tools for aircraft design and MDO (JPAD).

DAF research unit

Design of Aircraft and Flight technologies

Aircraft Design & Flight Mechanics

- Design and optimization of light aircraft, general aviation aircraft and commercial transport aircraft.
- Cooperations with:
  - *Tecnam Aircraft Industries*
  - *ATR*
  - *Leonardo Company*
- Development of new methodologies for aircraft preliminary design.
- Development of advanced software tools for aircraft design and MDO (JPAD).

Aircraft Aerodynamics & CFD applications

- UniNa SCoPE parallel computing center
- CFD solvers: Star-CCM+, SU2, OpenFOAM
- Applications:
  - Complete aircraft aerodynamic database generation
  - High-lift devices analysis and design
  - Wing-Fuselage junctions analysis
  - Vertical tail and dorsal fin analysis and design
  - Control surface design and analysis
  - Wing loads alleviation
  - Propeller simulations
  - Winglet design and analysis

DAF research unit

Design of Aircraft and Flight technologies

Ongoing research projects

- Horizon 2020 — **ADORNO: Aicraft design and noise rating for regional aicraft**.

  Development of aircraft models for a regional aircraft engine platform to provide requirements and trade factors for specific fuel consumption, engine drag and engine weight on fuel burn for both a year 2000 reference aircraft and a CleanSky 2 target aircraft.

  In partnership with MTU Aero Engines. 2018-2022

- Horizon 2020/CleanSky JU — **IRON: Innovative turboprop configuration**.

  Green and cost efficient Conceptual Aircraft Design including Innovative Turbo-Propeller Power-plant.

  In partnership with CIRA, ONERA, Politecnico di Torino, NLR, TUDelft, General Electric Deutschland, GE Aviation System LTD. 2016-2019

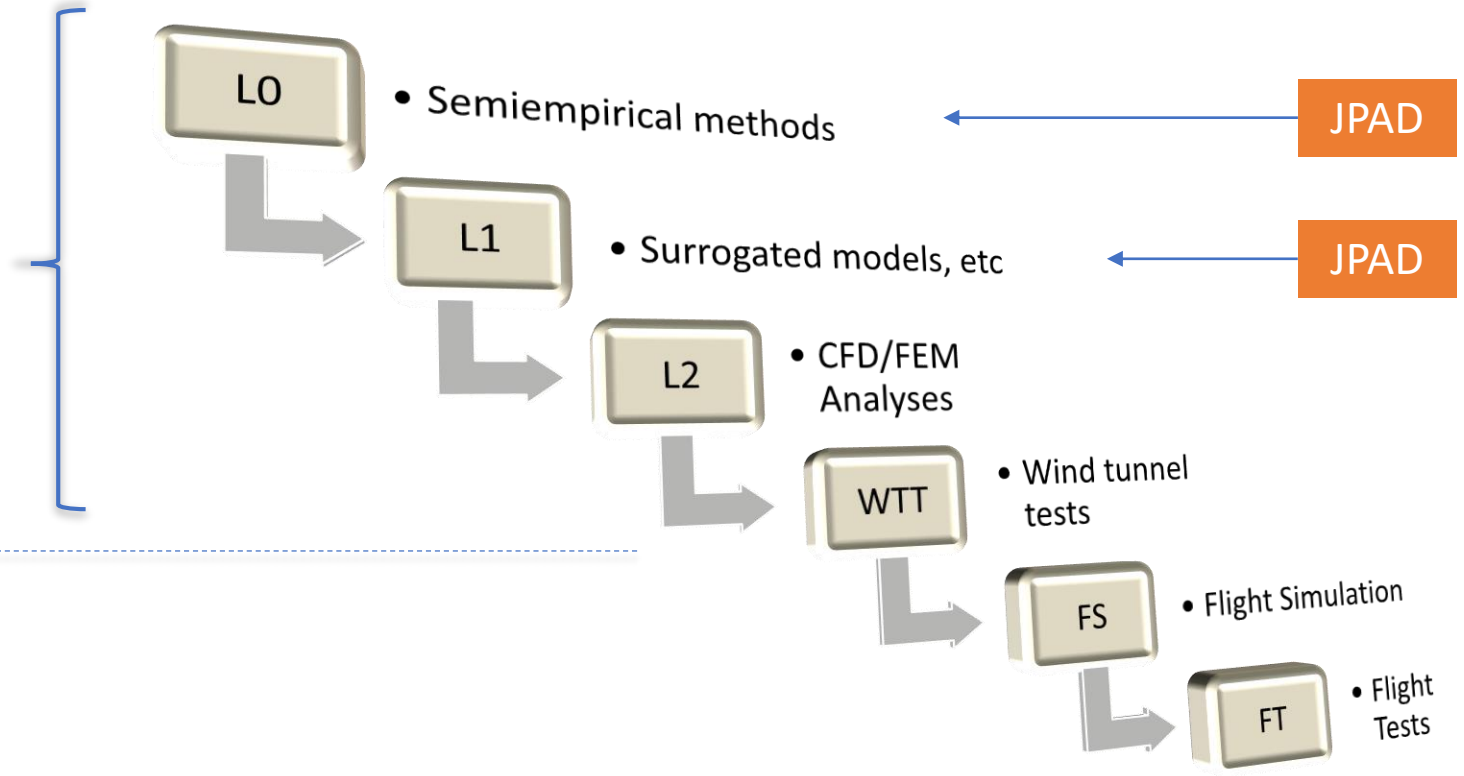
- Horizon 2020 — **AGILE: Aircraft 3rd generation MDO for innovative collaboration of heterogeneous teams of experts**.

  In partnership with DLR, Leonardo, EADS Deutschland, Fokker Aerostructures, NLR, Noesis Solutions, ONERA, Politecnico di Torino, University of Aachen, TUDelft, Bombardier, TSAGI, CIAM. 2015-2018

Aircraft Design Process

Aircraft Design Process

Typical fidelity levels L0, L1, L2
of multidisciplinary analyses
(aerodynamic, structural, etc)
in preliminary design

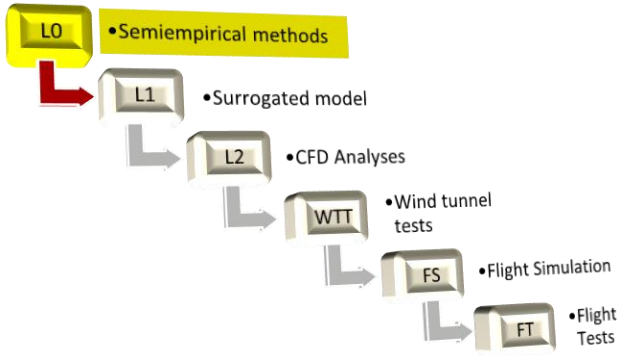


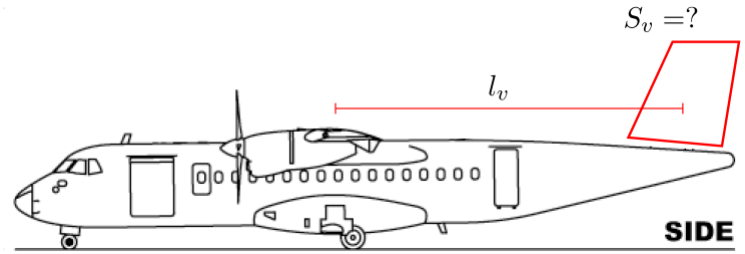
Preliminary Design

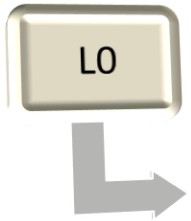
Detailed Design

Aerodynamic Design

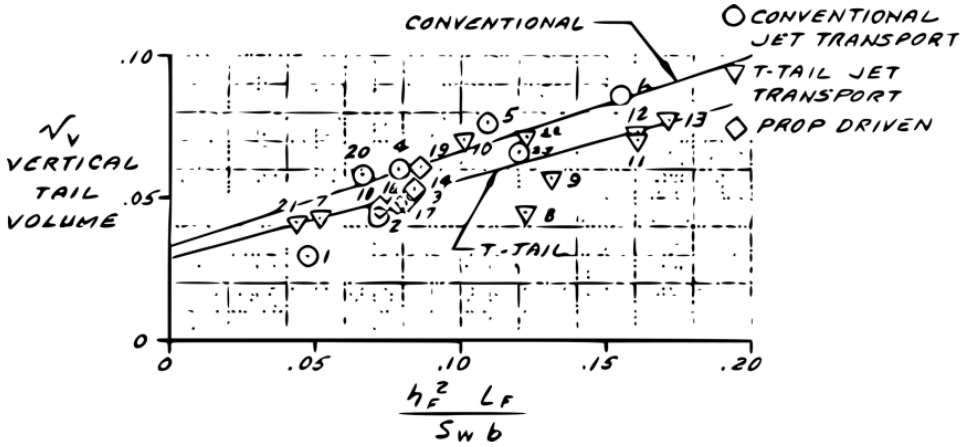
Aircraft CAD
is useful





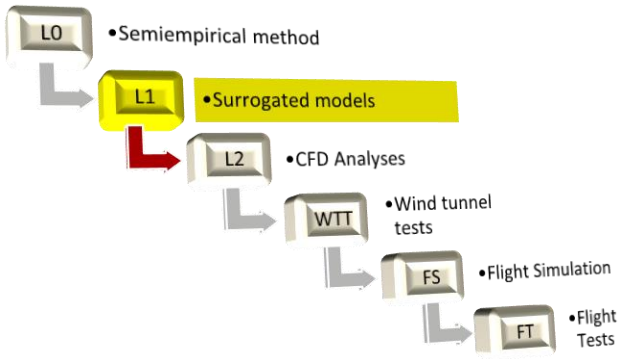


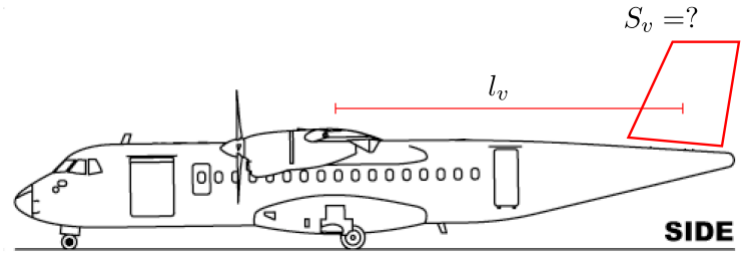
• Semiempirical methods

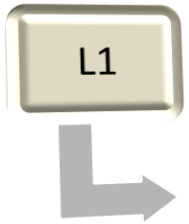


Aerodynamic Design

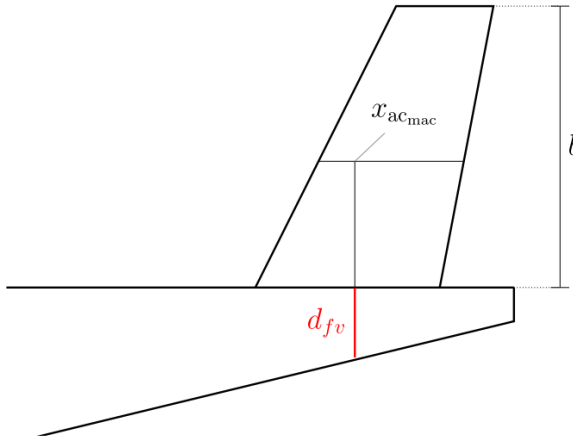
Aircraft CAD
is relevant

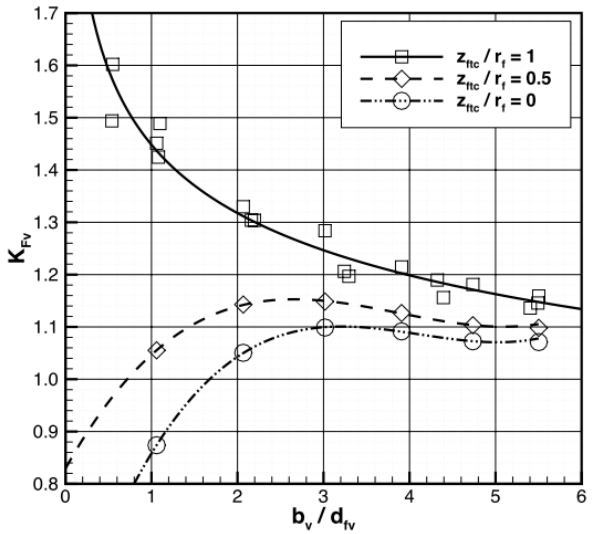






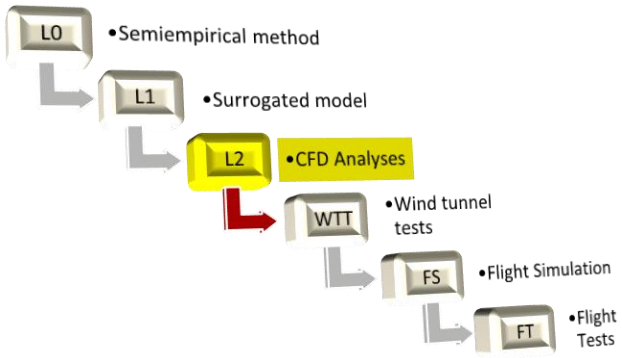
• Surrogated models

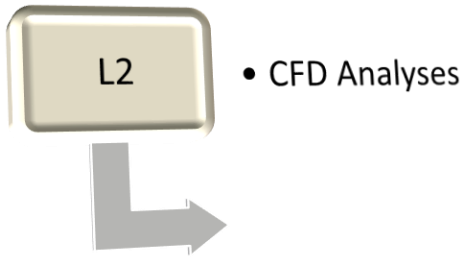


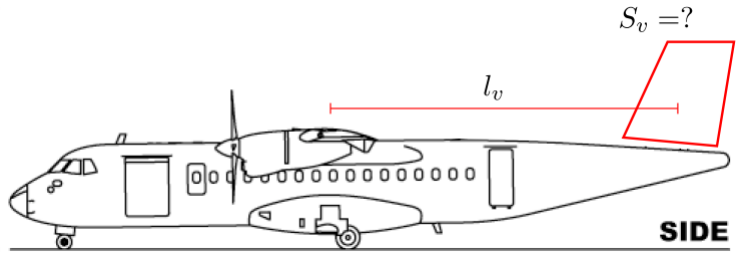


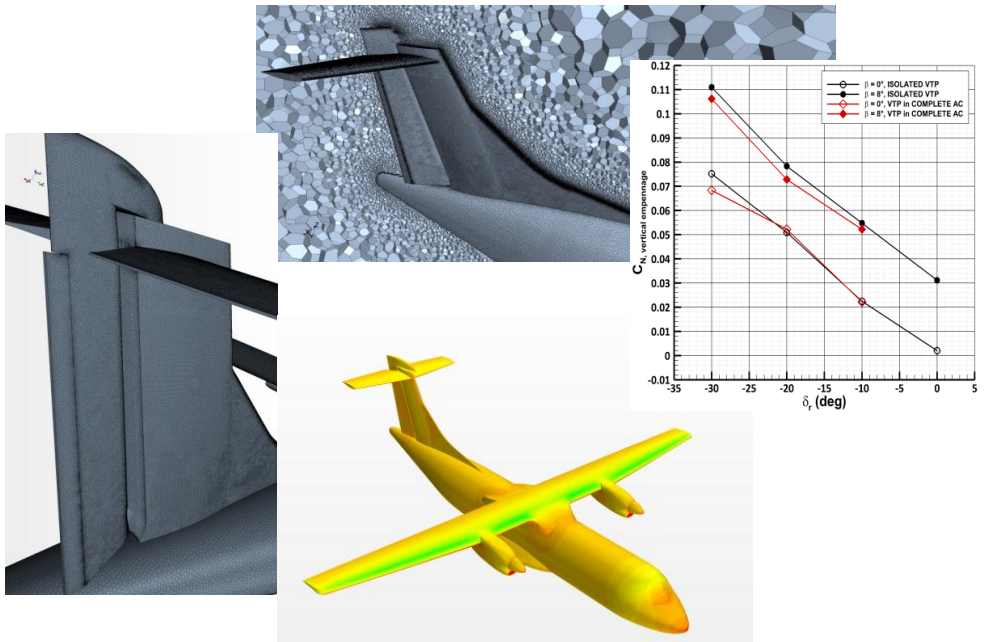
Aerodynamic Design

Aircraft CAD
is important!









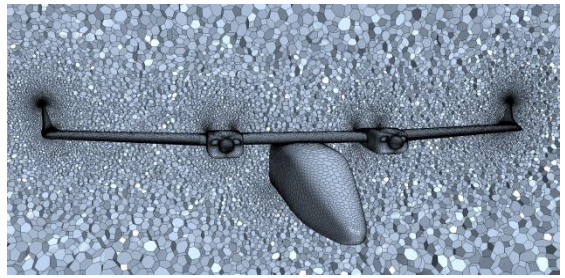
Selected Examples of L2 Aerodynamic Analyses & Designs

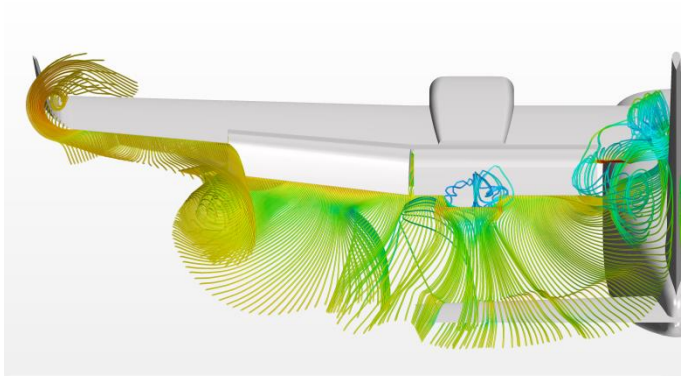
Selected Examples of L2 Aerodynamic Analyses & Designs

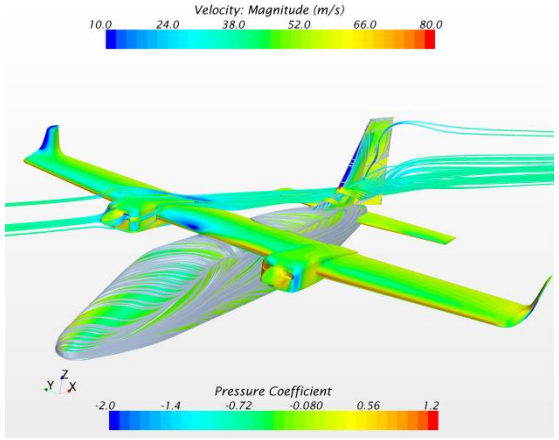
- Tecnam P2012 Traveller – Complete aerodynamic database
- High-lift system design of a next generation turboprop
- Wing-fuselage junction optimization

Aircraft Complete Aerodynamic Database

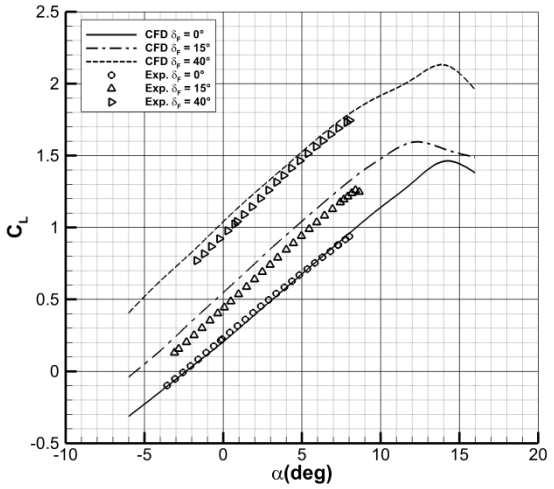
Tecnam P2012 - Traveller

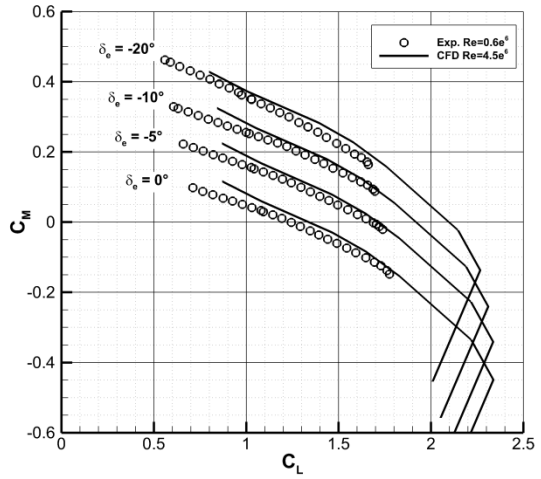




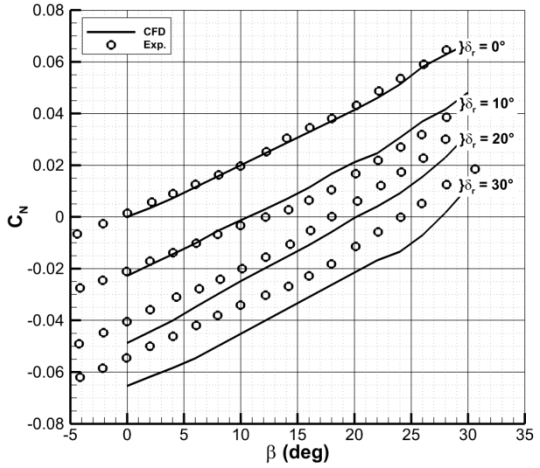


Longitudinal





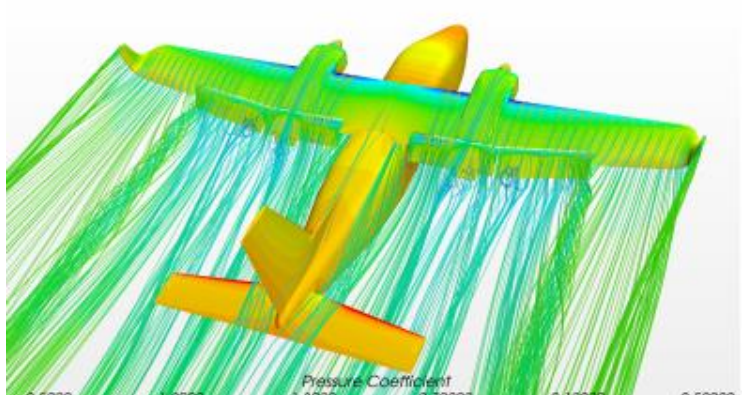
Lateral-Directional

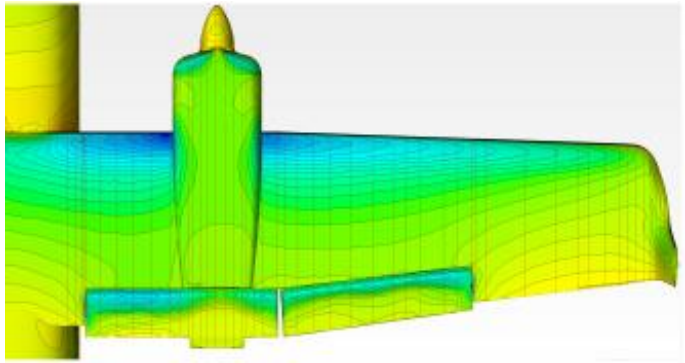


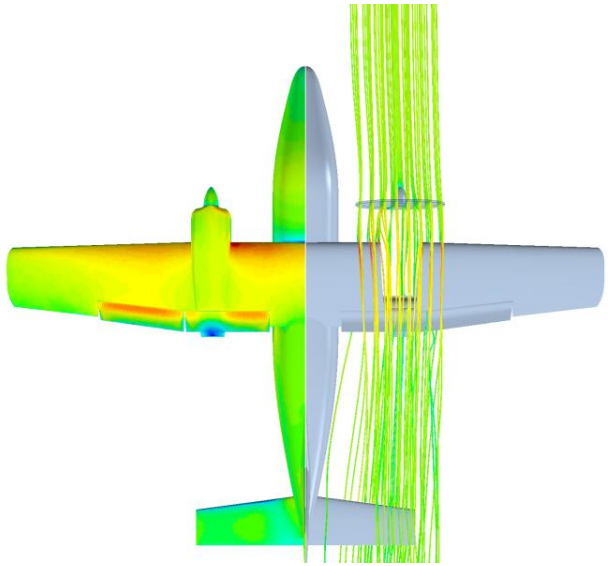
Aircraft Complete Aerodynamic Database

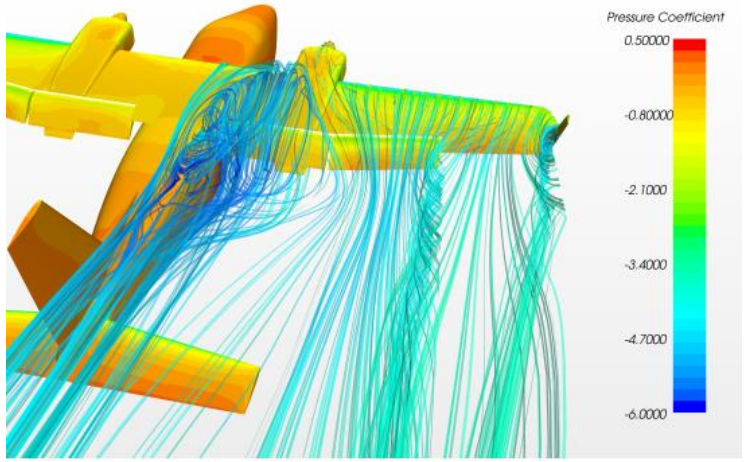
Tecnam P2012 - Traveller

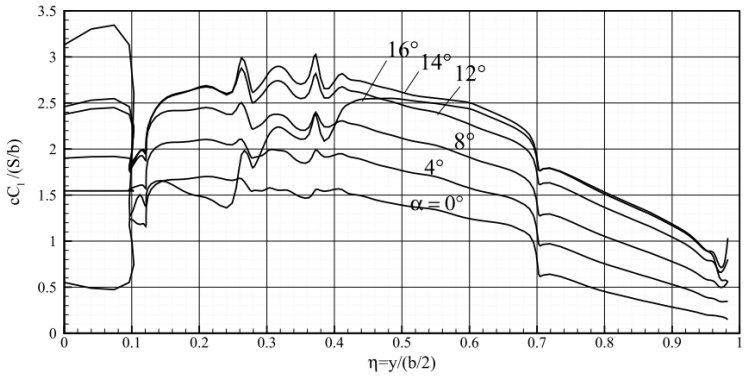
High-Lift Characterization & Propeller Effects



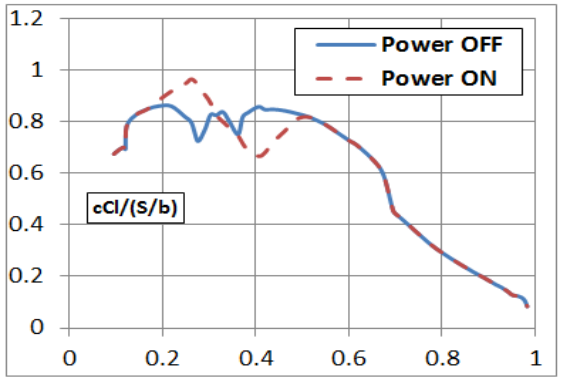




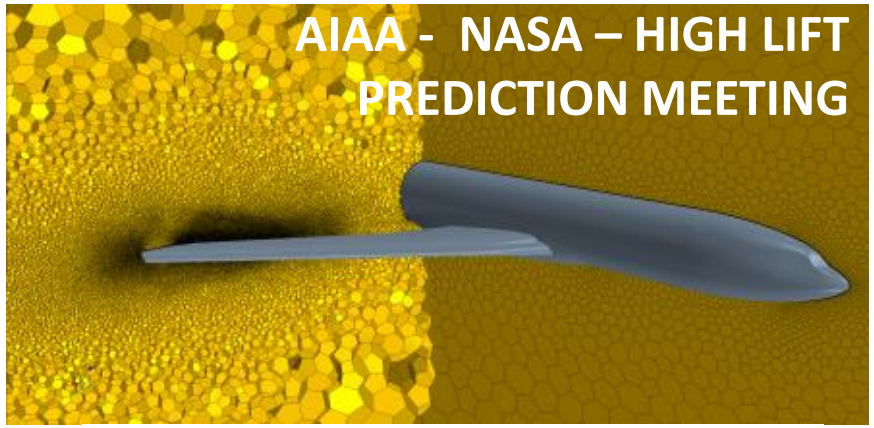


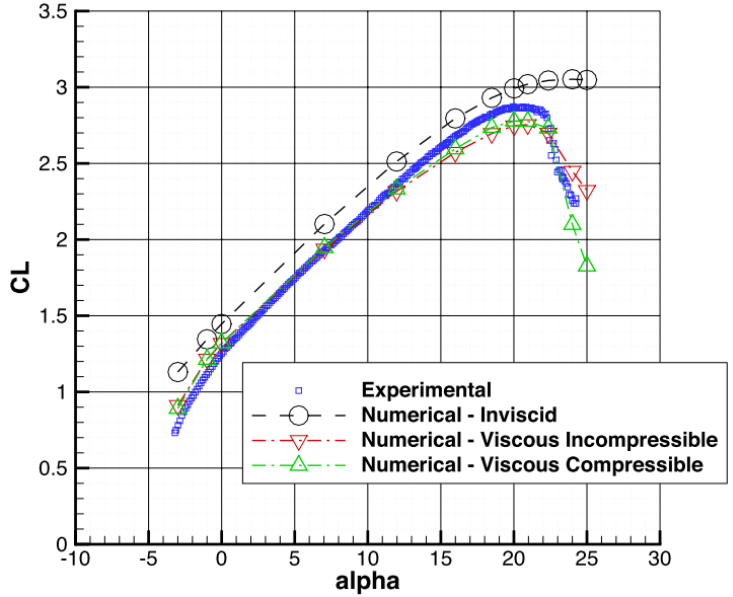


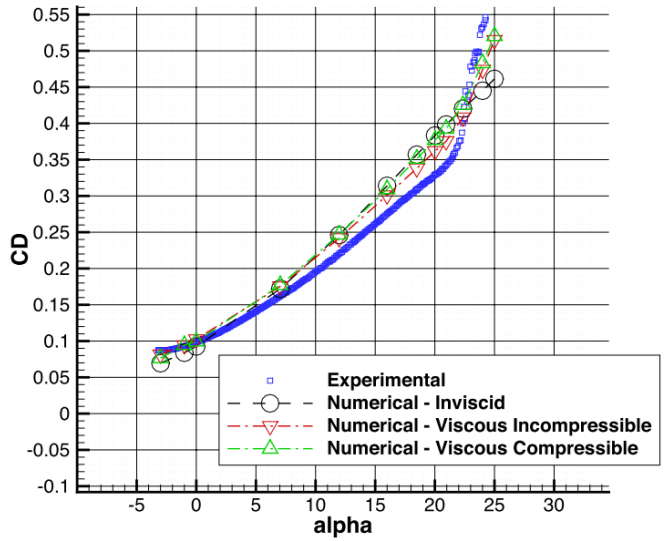
(b) Flap deflection 40°.



High-Lift System Design and Analyses

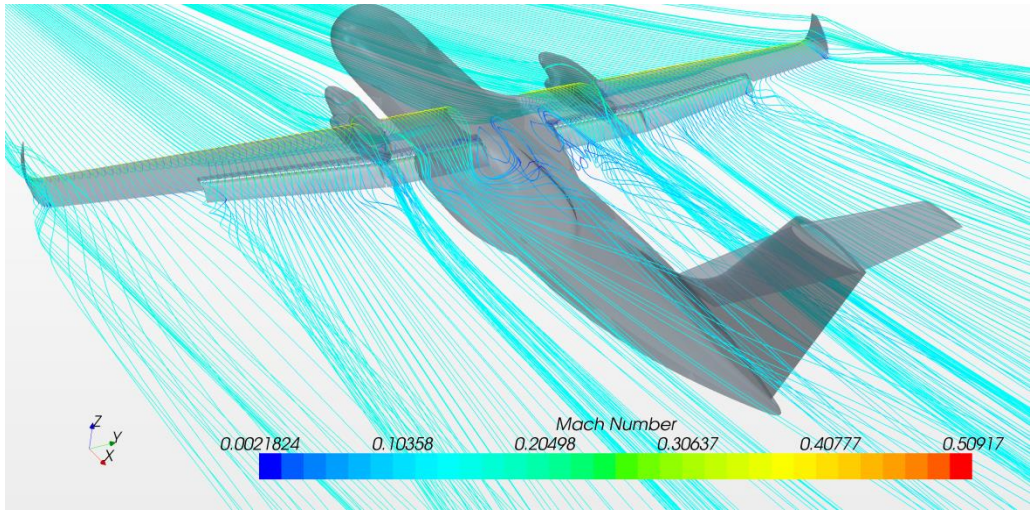


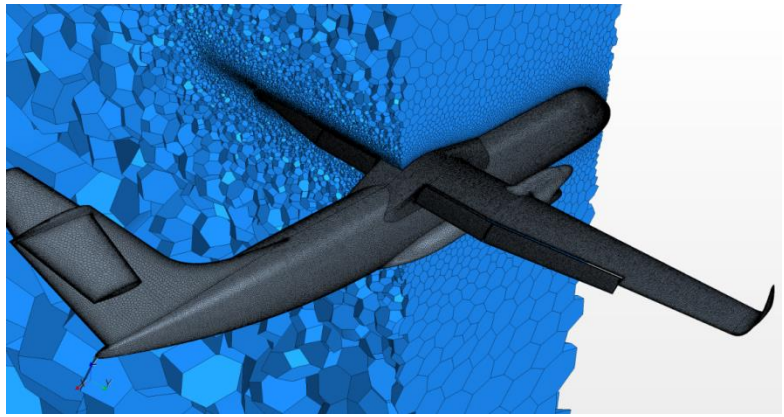


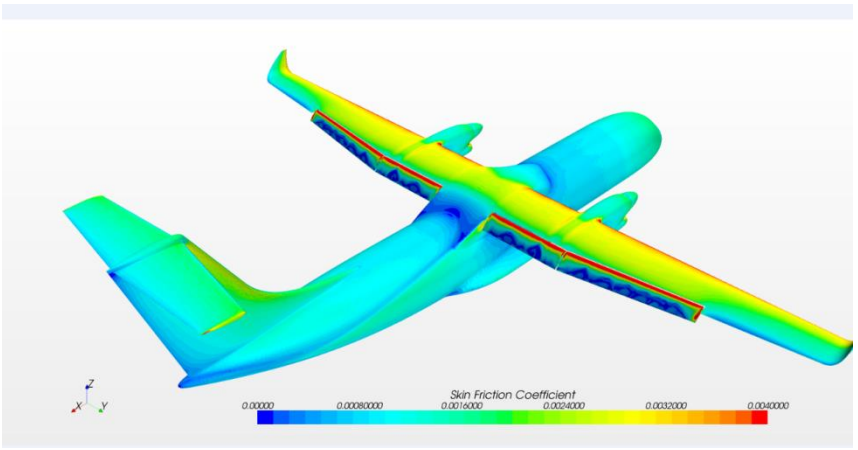


High-Lift System Design and Analyses

Turboprop design (ATR)

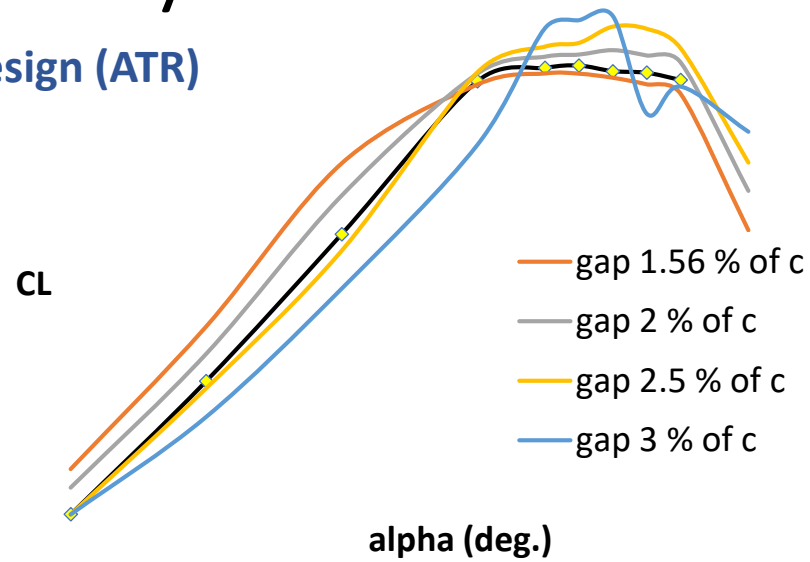


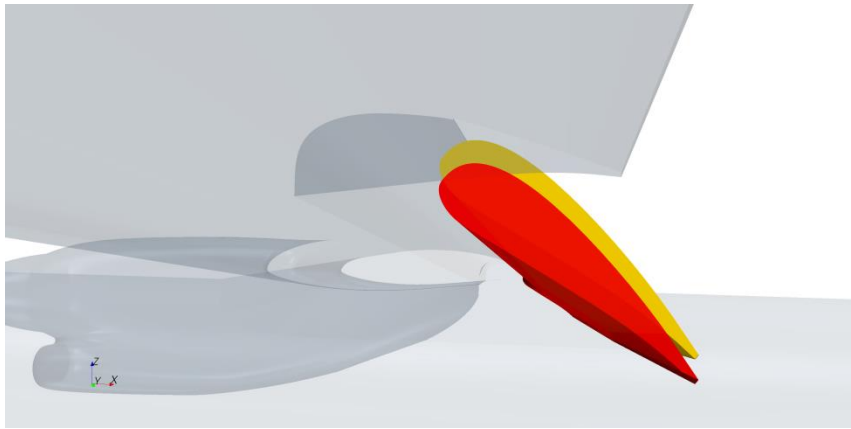


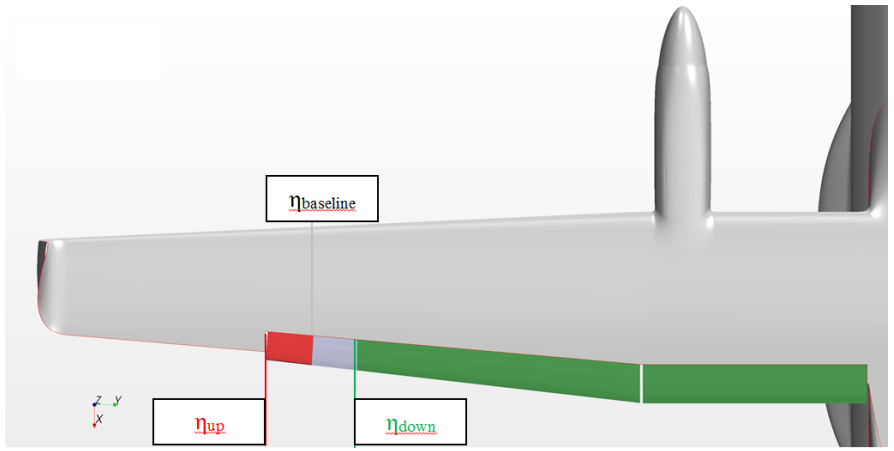


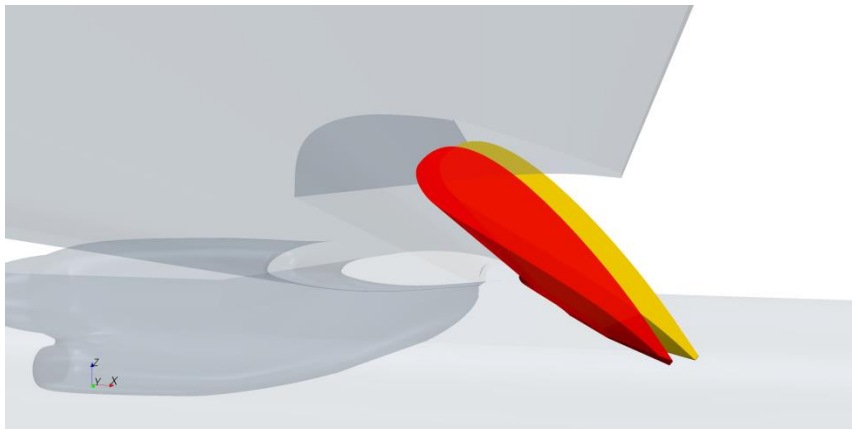
High-Lift System Design and Analyses

Turboprop design (ATR)



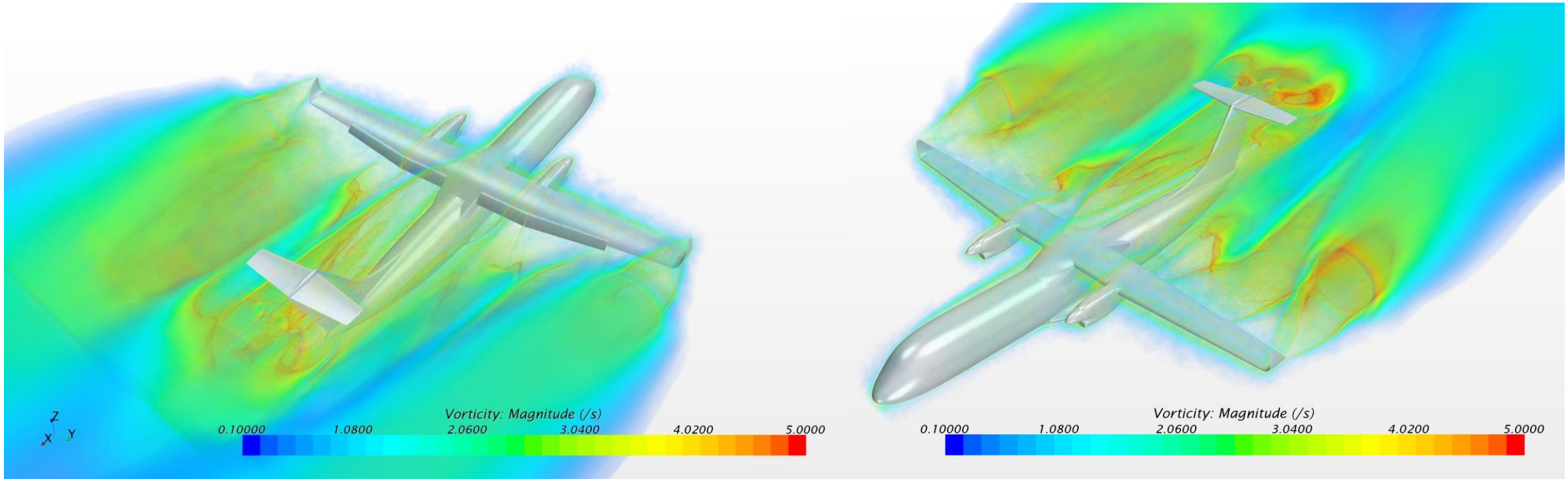






High-Lift System Design and Analyses

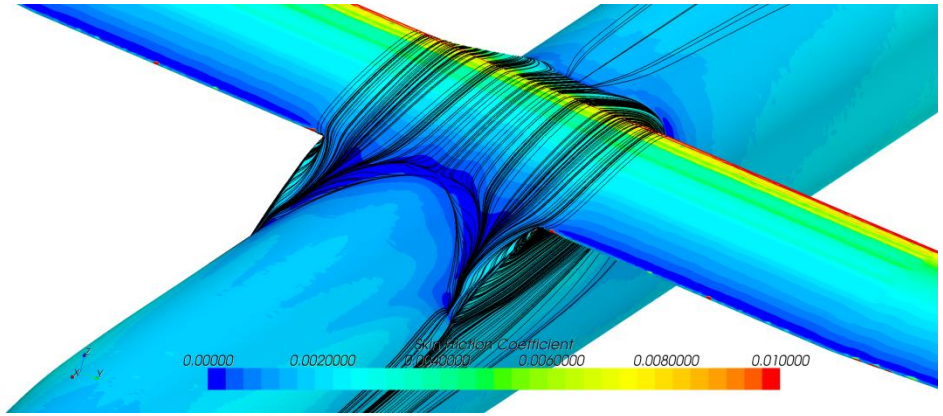
Turboprop design (ATR)

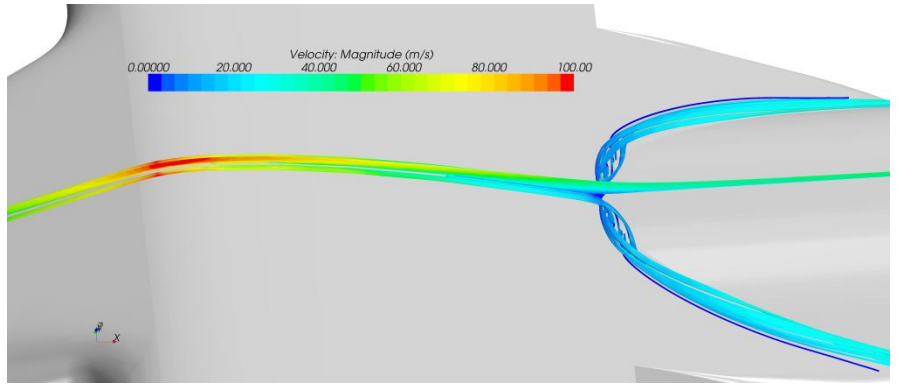


Wing-Fuselage Junction Design

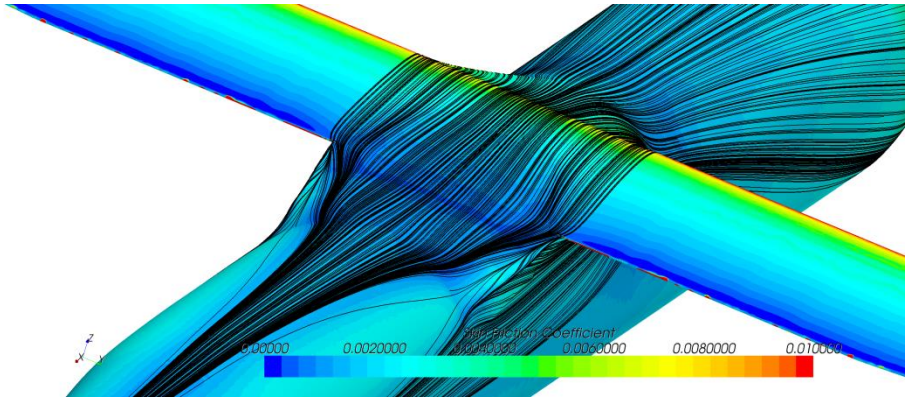
Turboprop optimization

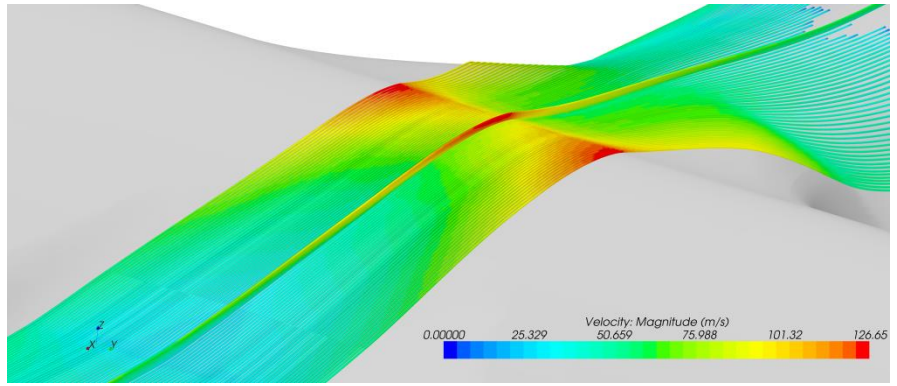
ORIGINAL





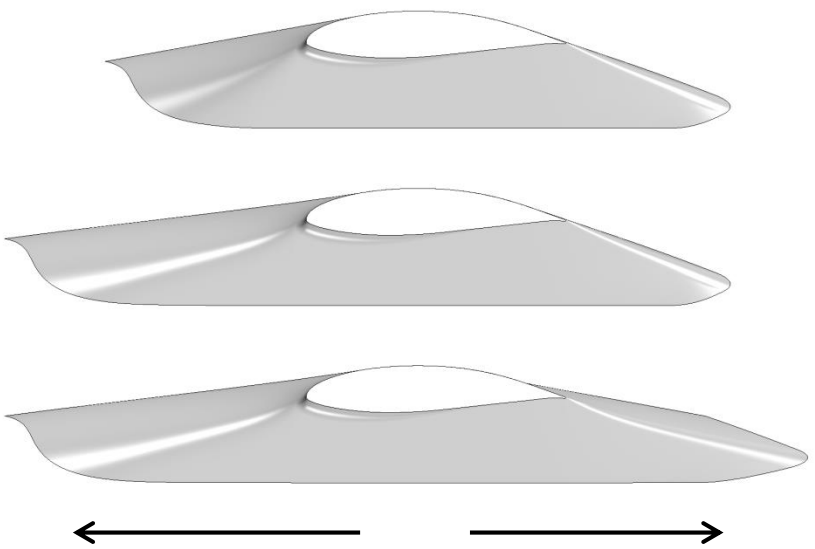
OPTIMIZED

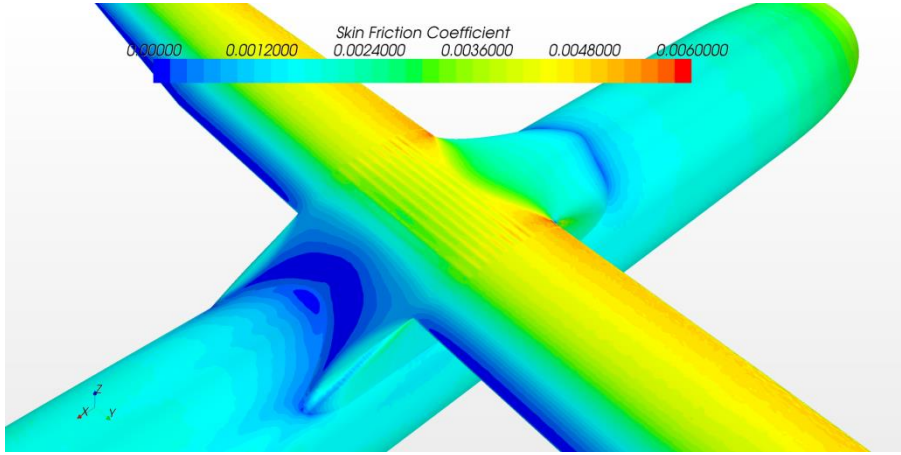


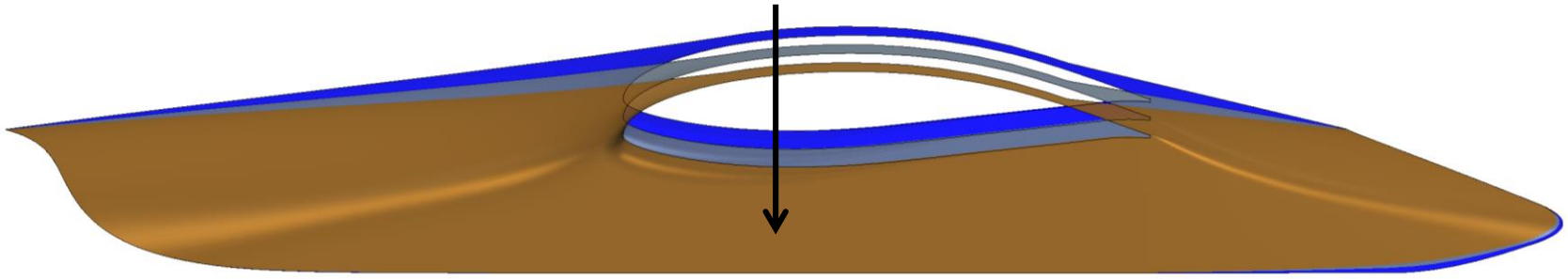


Wing-Fuselage Junction Design

Turboprop optimization

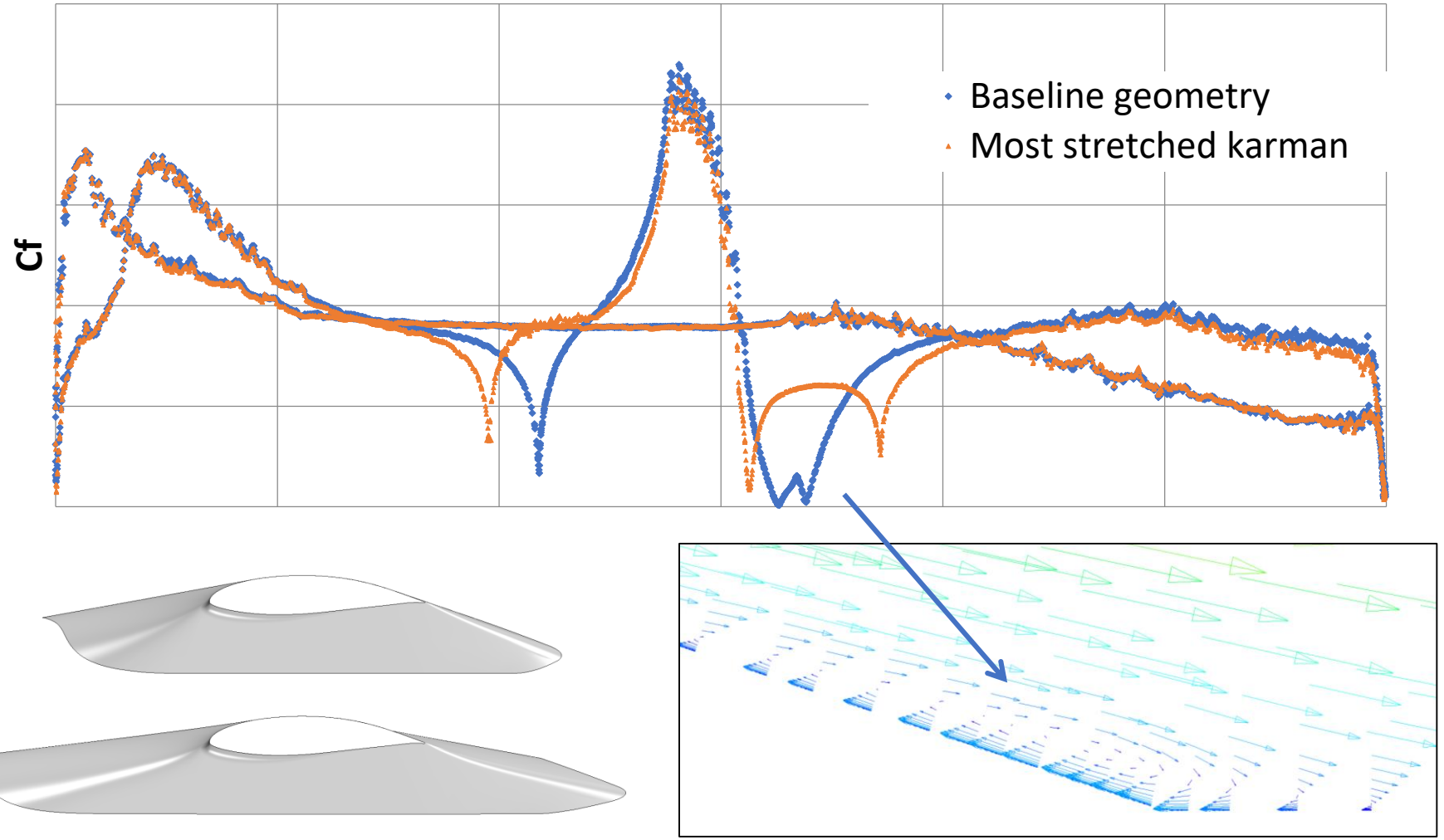






Wing-Fuselage Junction Design

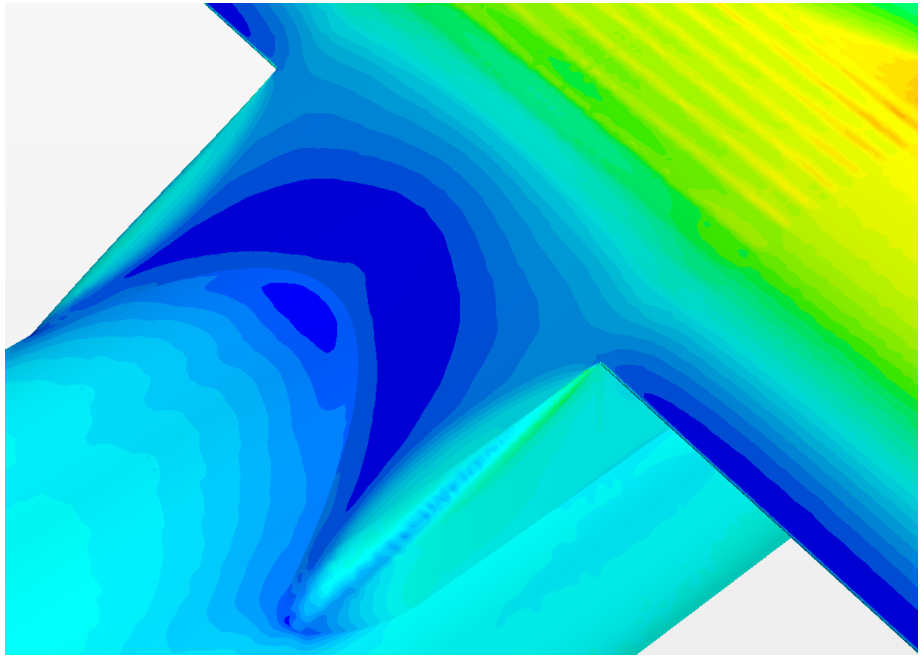
Turboprop optimization



Wing-Fuselage Junction Design

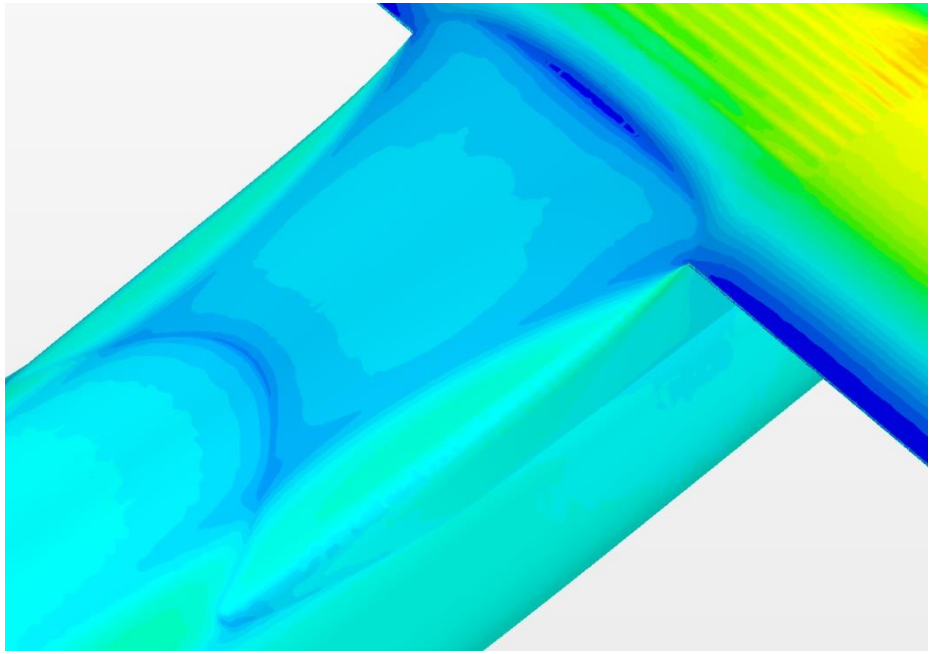
Turboprop optimization

Baseline



Karman CDf = 8 counts
Karman CDp = 25 counts

Stretched and lowered



Karman CDf = 10 counts
Karman CDp = 14 counts

JPAD

Java Programming Interface for Aircraft Design

JPAD

Java Programming Interface for Aircraft Design

- A modular, open-source system of Java API-s and applications, conceived as a fast, reliable and user-friendly computational aid for engineers in the preliminary design phase of transport aircraft.
- JPAD API-s provide a set of building blocks to develop software applications.
- JPAD-based applications are typically used to perform multi-disciplinary analyses of a parametrically-defined aircraft.
- In preliminary design, these tools are typically employed to search for the optimum configuration which cope with the objectives and constrains set by the user.

JPAD

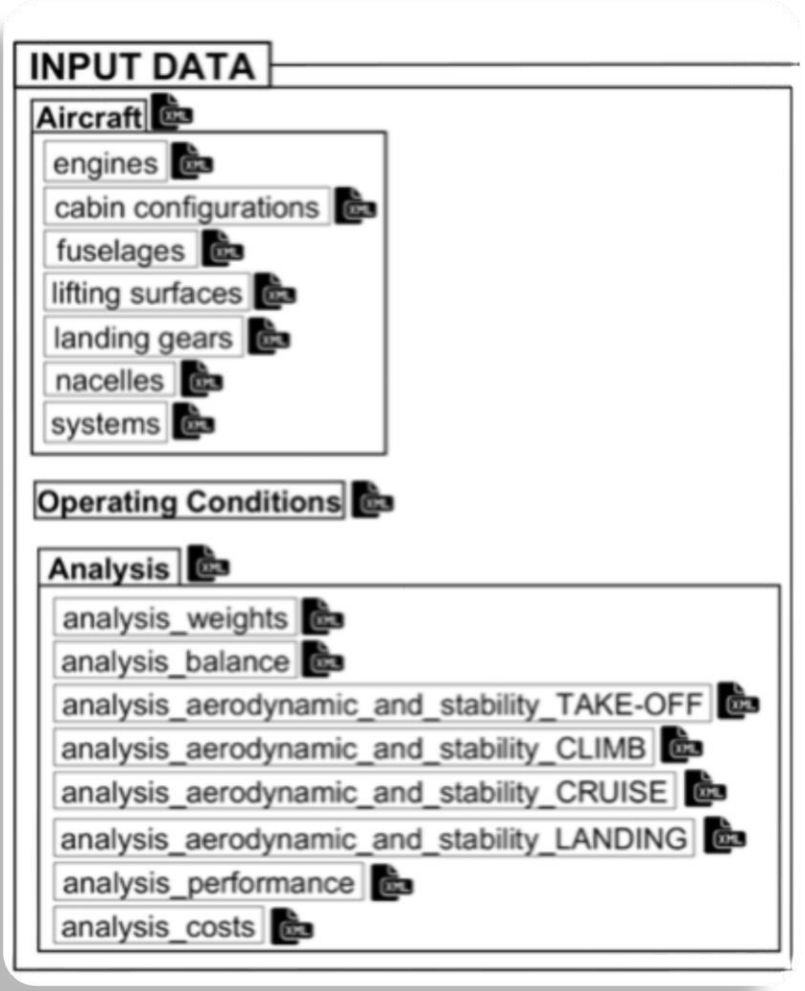
Java Programming Interface for Aircraft Design

Main Features:

- **Smart management of I/O operations.**
- **A modern, user friendly, modular framework.**
- **Support for simultaneous management/analysis of several aircraft and/or ‘varied’ configurations of the same aircraft.**
- **Conceived for collaborative design activities.**
- **Interoperability with other tools/disciplines (CAD/CFD/FEM analysis).**
- **Capability to generate and export the CAD model of the aircraft.**
- **Use of new methodologies to estimate aerodynamic characteristics (VeDSC, FusDes databases).**
- **Use of simulation-based approach for performance evaluation.**
- **Integration of the most advanced metaheuristic optimization algorithms.**

JPAD

Configuration/Input Files



- Object Oriented Programming approach is automatically reflected in the XML structure. There's not just one input file which stores all aircraft's data.
- Instead, there's an XML file for each aircraft component and each component can call other XML files. For example, the wing XML file calls for airfoil XML files.
- The setup of hierarchy among the XML files helps to reduce dramatically the dimension of each file and makes working with configuration files, especially at high level, much easier.

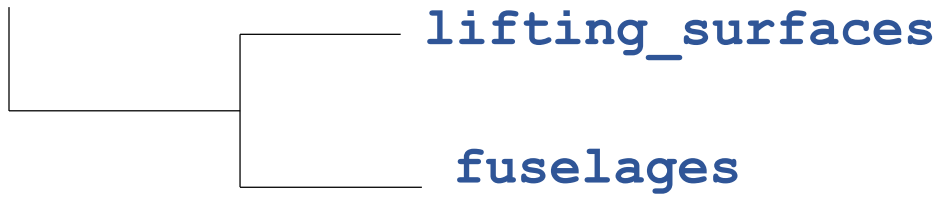
JPAD

Configuration/Input Files

```xml
<?xml version="1.0" encoding="UTF-8"?>
<jpad_config>
    <aircraft id="ATR-72" regulations="FAR_25" type="TURBOPROP">
        <global_data>
            <cabin_configuration file="cabin_configuration_ATR72.xml"/>
        </global_data>
        <lifting_surfaces>
            <wing file="wing_ATR72.xml">
                <position>
                    <x unit="m">11.0000</x>
                    <y unit="m">0.0000</y>
                    <z unit="m">1.6000</z>
                </position>
                <rigging_angle unit="deg">2.0000</rigging_angle>
            </wing>
            <vertical_tail file="vtail_ATR72.xml">
                <position>
                    <x unit="m">21.6000</x>
                    <y unit="m">0.0000</y>
                    <z unit="m">1.3000</z>
                </position>
                <rigging_angle unit="deg">0.0000</rigging_angle>
            </vertical_tail>
            <horizontal_tail file="htail_ATR72.xml">
                <position>
                    <x unit="m">25.3000</x>
                    <y unit="m">0.0000</y>
                    <z unit="m">5.7374</z>
                </position>
                <rigging_angle unit="deg">1.0000</rigging_angle>
            </horizontal_tail>
        </lifting_surfaces>
        <fuselages>
            <fuselage file="fuselage_ATR72.xml">
                <position>
                    <x unit="m">0.0000</x>
                    <y unit="m">0.0000</y>
                    <z unit="m">0.0000</z>
                </position>
            </fuselage>
        </fuselages>
        </aircraft>
</jpad_config>
```

XML structure:

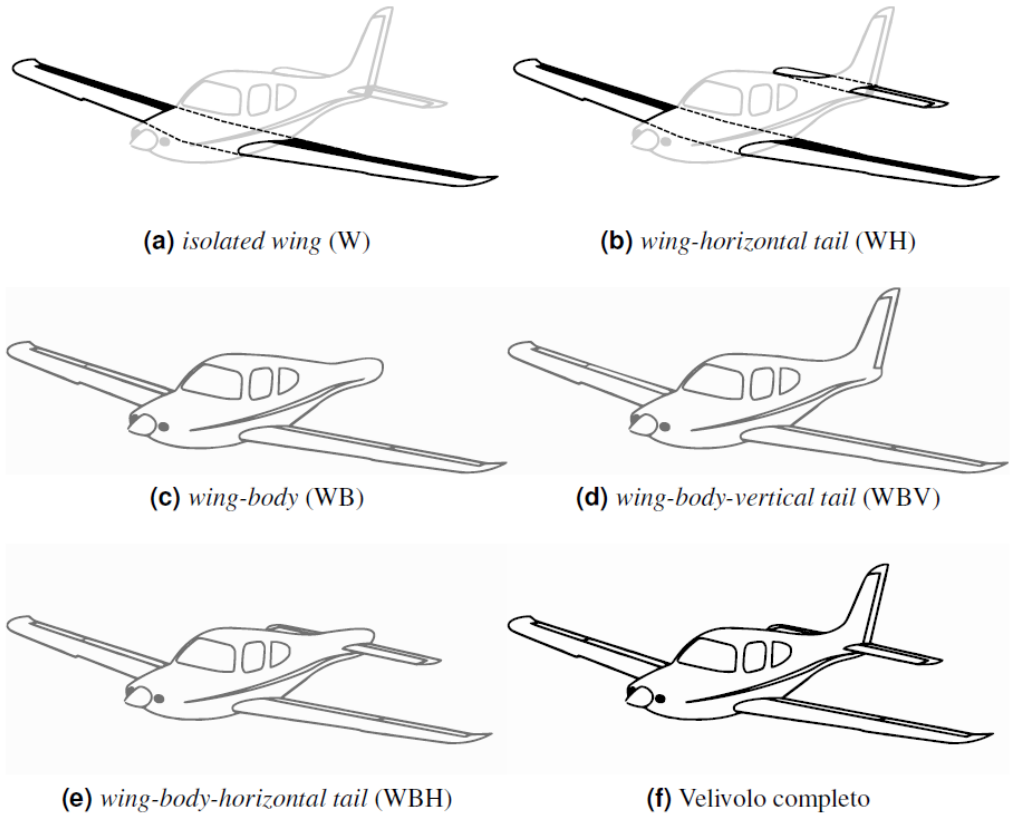
- Tags: **`aircraft`**, **`lifting_surfaces`**, **`fuselages`** …
- Attributes **`id`**, **`file`**, **`unit`**
- tree structure

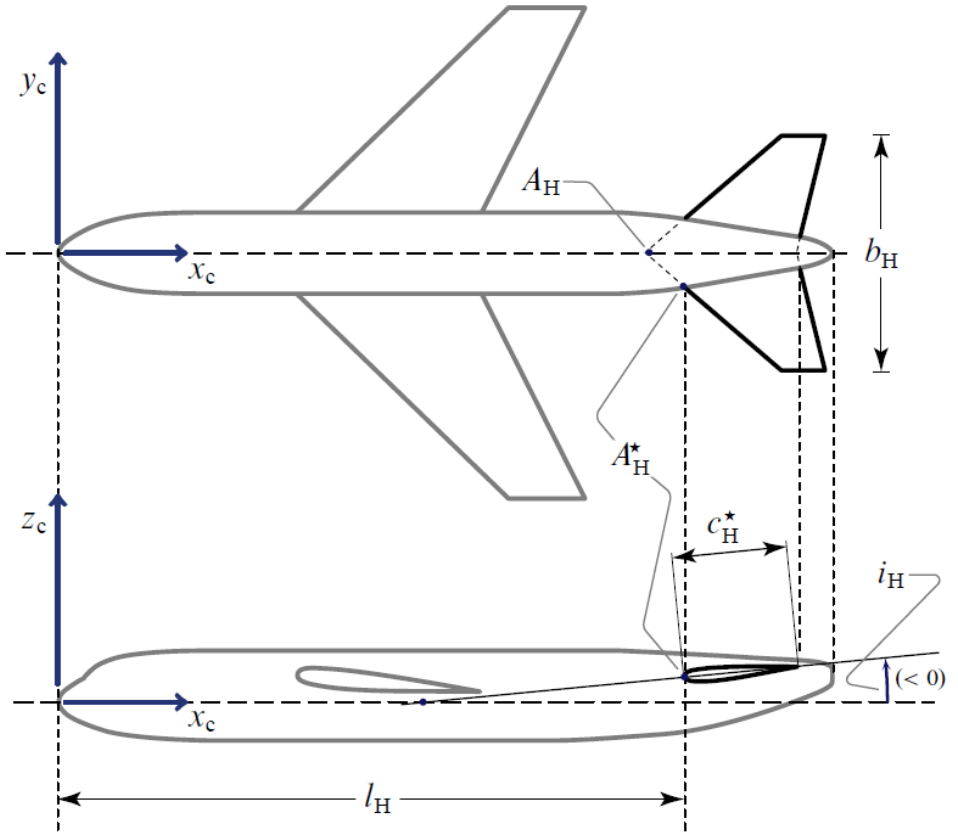


JPAD

Parametrically Defined Aircraft Components

Aircraft: JPAD Business Model

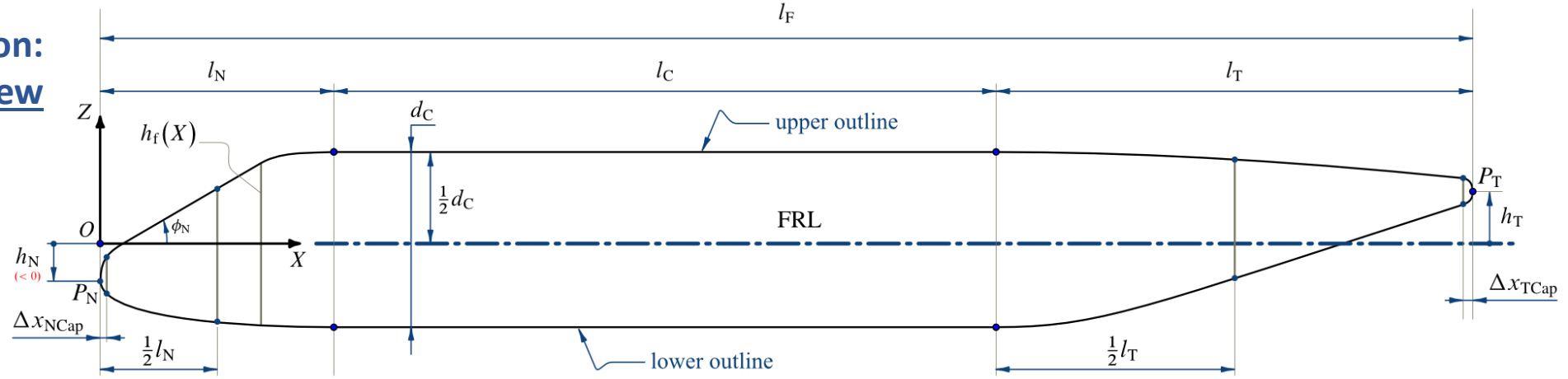




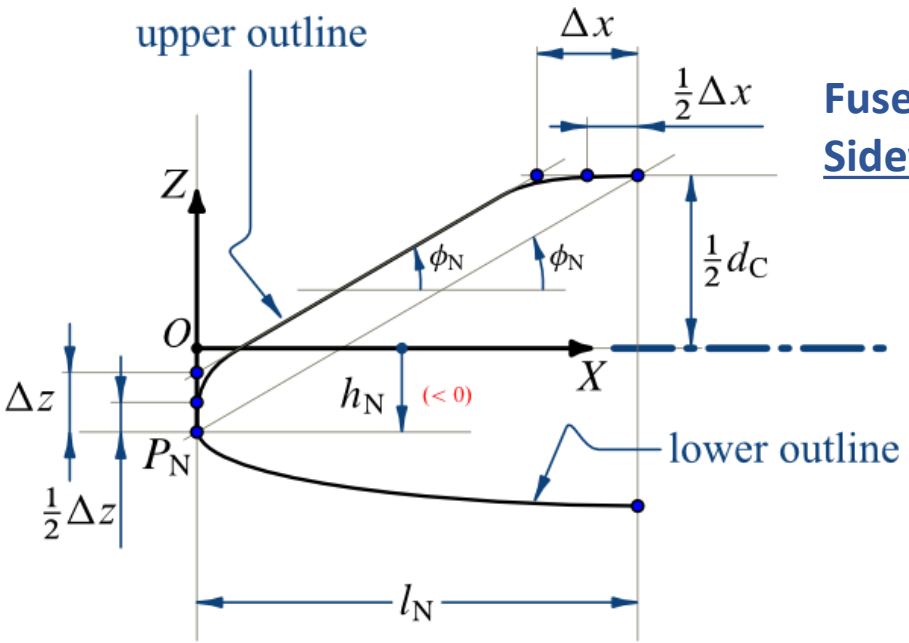
JPAD

Parametrically Defined Aircraft Components

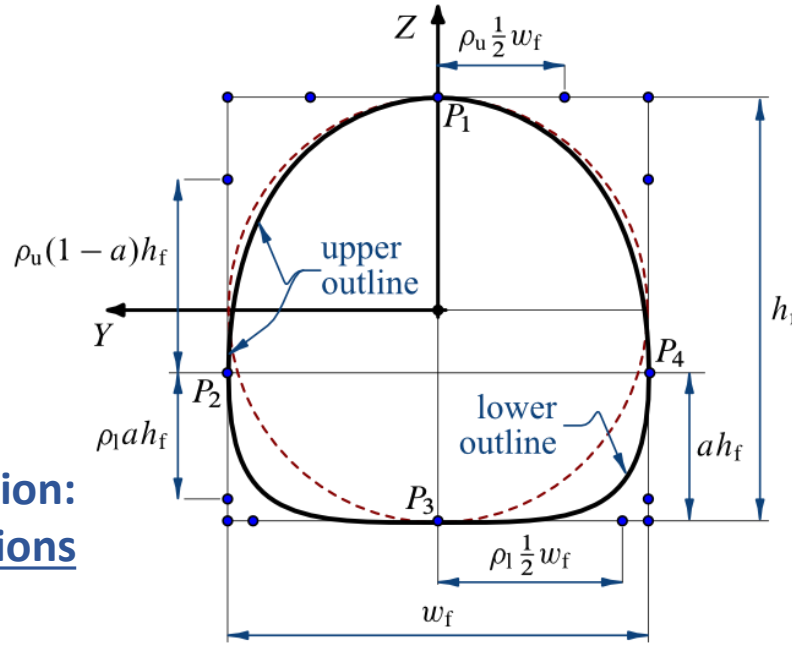
Fuselage parametric definition:
Sideview



Fuselage parametric definition:
Sideview Nose



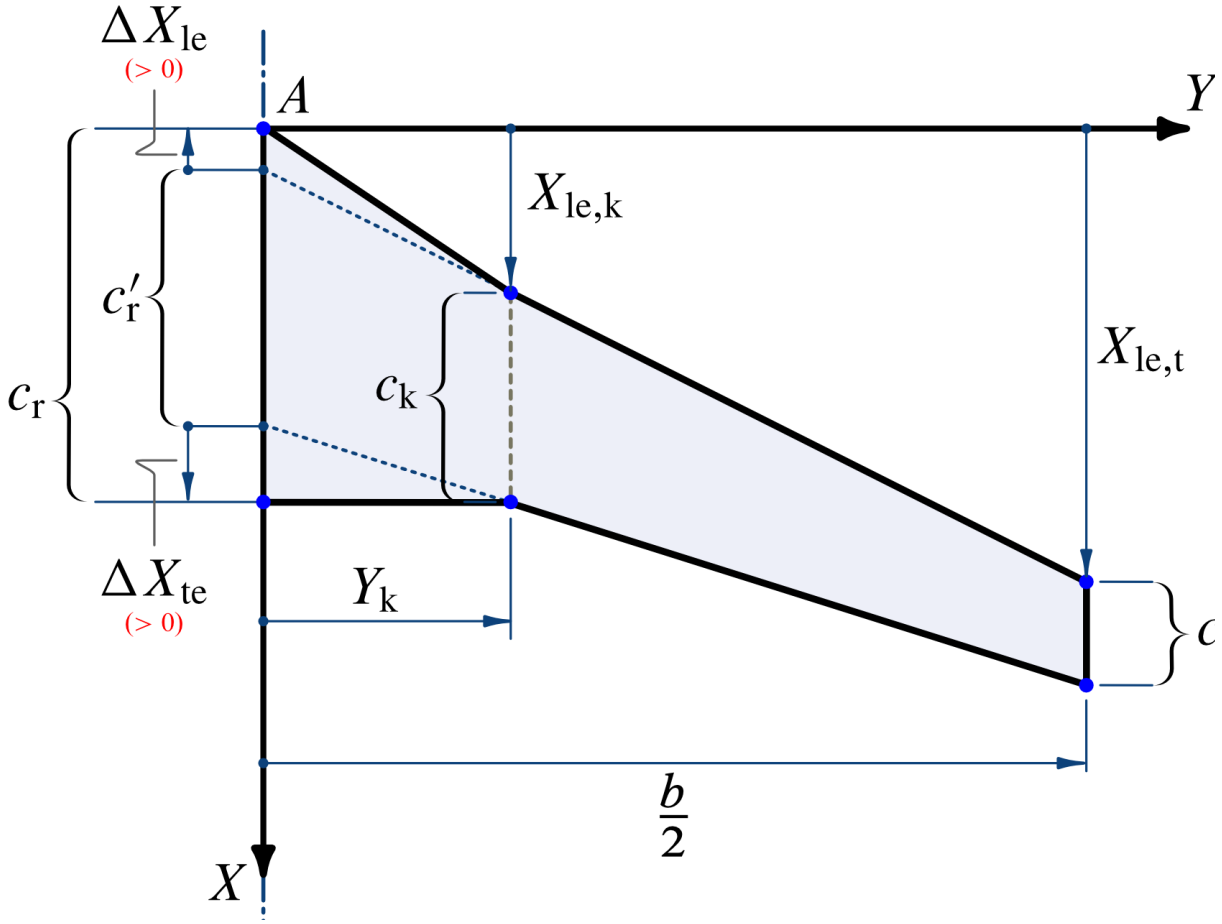
Fuselage parametric definition:
Cross sections

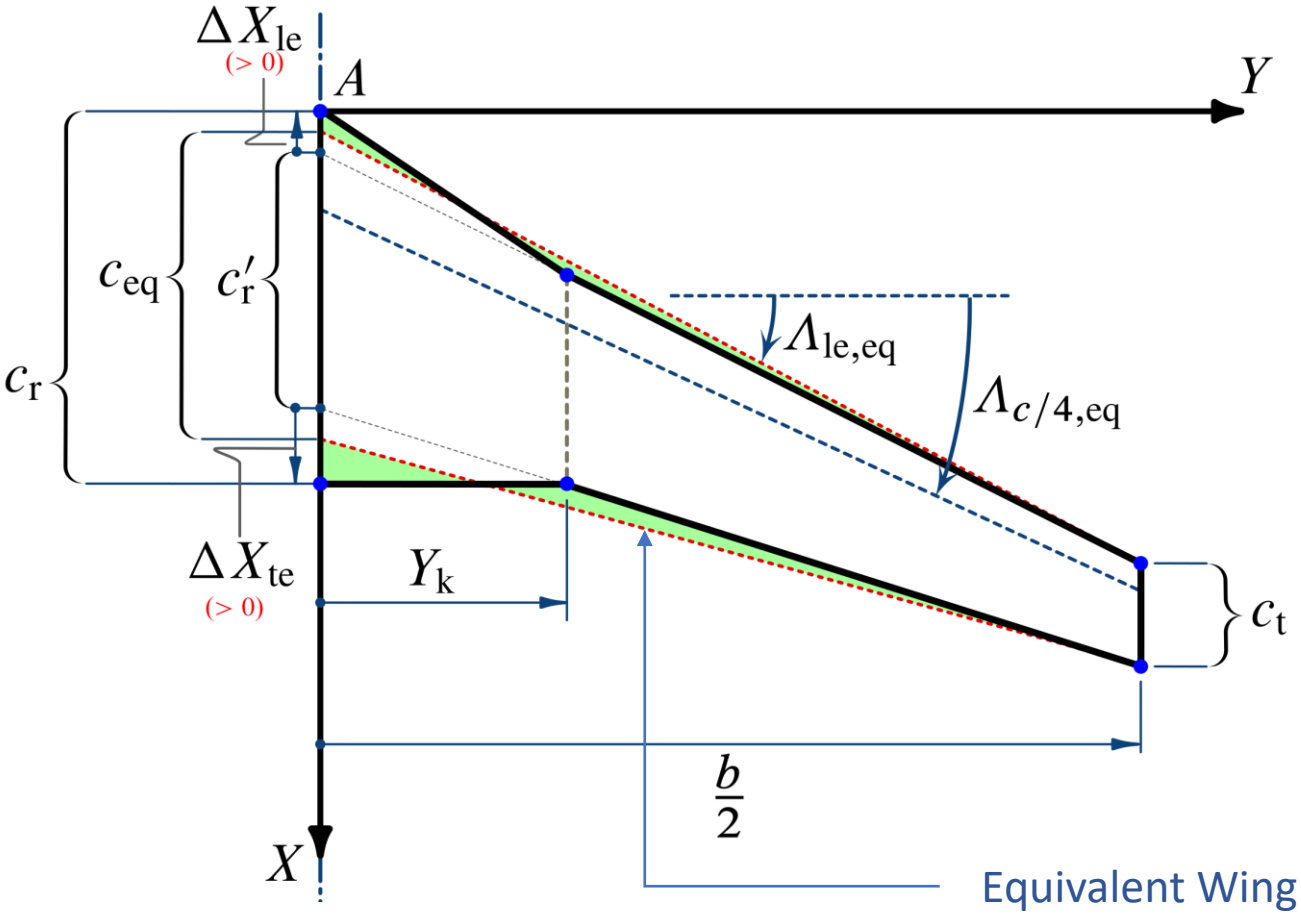


JPAD

Parametrically Defined Aircraft Components

Wing parametric definition: Topview





JPAD

From configuration files to the *Builder Pattern*

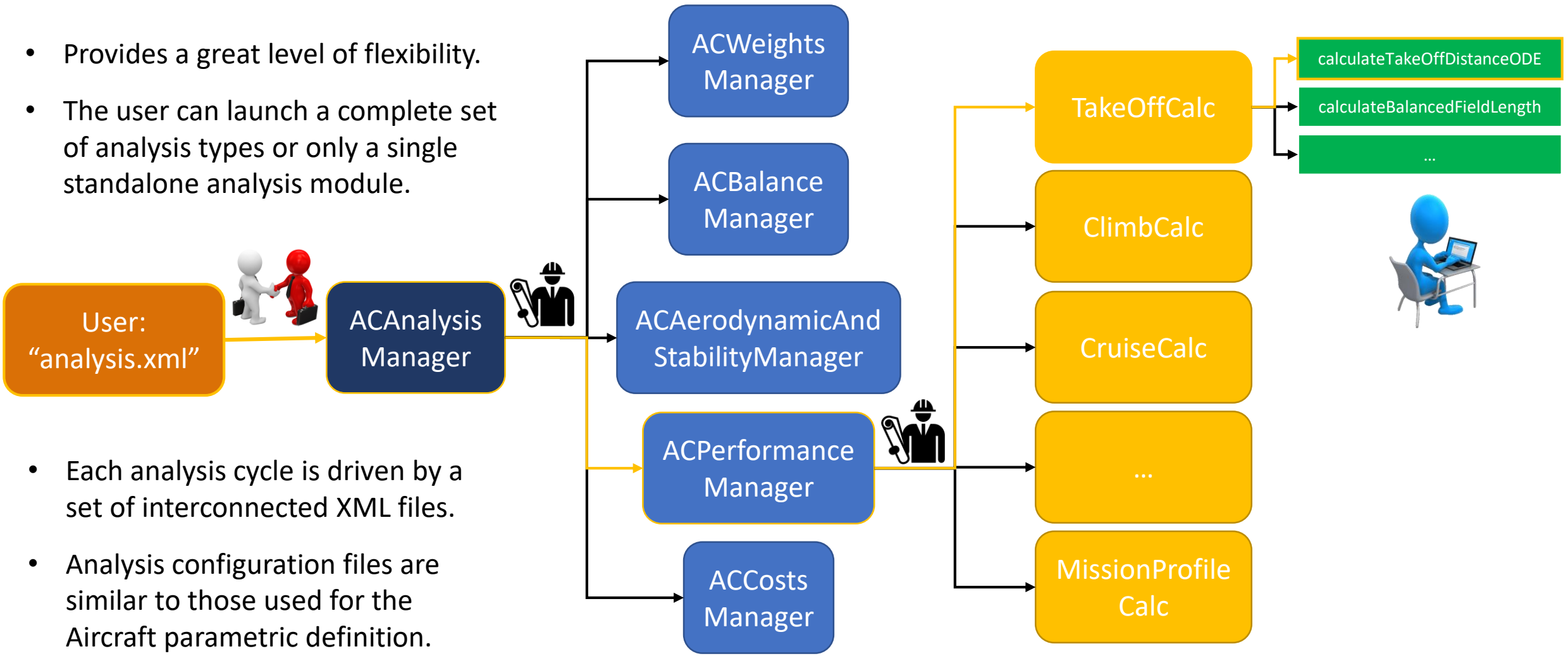
```java
@FreeBuilder
public interface IFuselage {

    String getId();
    int getDeckNumber();
    boolean isPressurized();
    Amount<Length> getFuselageLength();
    double getNoseLengthRatio();
    double getCylinderLengthRatio();
    Amount<Length> getSectionCylinderWidth();
    Amount<Length> getSectionCylinderHeight();
    Amount<Length> getHeightFromGround();
    Amount<Length> getRoughness();
    Amount<Length> getNoseTipOffset();
    Amount<Length> getTailTipOffest();
    double getNoseCapOffsetPercent();
    double getTailCapOffsetPercent();
    WindshieldTypeEnum getWindshieldType();
    Amount<Length> getWindshieldHeight();
    Amount<Length> getWindshieldWidth();

    // how lower part is different from half diameter
    double getSectionNoseMidLowerToTotalHeightRatio();
    double getSectionCylinderLowerToTotalHeightRatio();
    double getSectionTailMidLowerToTotalHeightRatio();
    // shape index, 1 --> close to a rectangle; 0 --> close to a circle
    double getSectionCylinderRhoUpper();
    double getSectionCylinderRhoLower();
    double getSectionMidNoseRhoUpper();
    double getSectionMidTailRhoUpper();
    double getSectionMidNoseRhoLower();
    double getSectionMidTailRhoLower();

    List<SpoilerCreator> getSpoilers();

    class Builder extends IFuselage_Builder  {
        public Builder() {


        }
    }
}
```

```java
// create the fuselage via its builder
Fuselage fuselage = new Fuselage(
        new IFuselage.Builder()
        .setId(id)
        .setPressurized(pressurized)
        .setFuselageLength(len)
        .setDeckNumber(deckNum)
        .setRoughness(roughness)
        .setNoseCapOffsetPercent(dxNoseCapPercent)
        .setNoseTipOffset(heightN)
        .setNoseLengthRatio(lenRatioNF)
        .setSectionNoseMidLowerToTotalHeightRatio(sectionNoseMidLowerToTotalHeightRatio)
        .setSectionMidNoseRhoUpper(sectionMidNoseRhoUpper)
        .setSectionMidNoseRhoLower(sectionMidNoseRhoLower)
        .setWindshieldType(windshieldType)
        .setWindshieldHeight(windshieldHeight)
        .setWindshieldWidth(windshieldWidth)
        .setCylinderLengthRatio(lenRatioCF)
        .setSectionCylinderHeight(sectionCylinderHeight)
        .setSectionCylinderWidth(sectionCylinderWidth)
        .setHeightFromGround(heightFromGround)
        .setSectionCylinderLowerToTotalHeightRatio(sectionCylinderLowerToTotalHeightRatio)
        .setSectionCylinderRhoUpper(sectionCylinderRhoUpper)
        .setSectionCylinderRhoLower(sectionCylinderRhoLower)
        .setTailTipOffest(heightT)
        .setTailCapOffsetPercent(dxTailCapPercent)
        .setSectionTailMidLowerToTotalHeightRatio(sectionTailMidLowerToTotalHeightRatio)
        .setSectionMidTailRhoUpper(sectionMidTailRhoUpper)
        .setSectionMidTailRhoLower(sectionMidTailRhoLower)
        .addAllSpoilers(spoilers)
        .build()
        );
```

JPAD

Analisys Manager Model

- Provides a great level of flexibility.
- The user can launch a complete set of analysis types or only a single standalone analysis module.



- Each analysis cycle is driven by a set of interconnected XML files.
- Analysis configuration files are similar to those used for the Aircraft parametric definition.

JPAD

External Analysis Tool Interface

- JPAD based applications can run external analysis tools in batch mode, and parse their results.
- Specific classes in the API address the construction of a command to launch the tools, and the assimilation of results.
- At this time, JPAD is provided with launchers for various external software, including:
  - AVL – a program by Mark Drela (MIT) for aerodynamic and flight dynamics analysis,
  - USAF Digital Datcom – a static stability and dynamic-derivative calculator,
  - Star-CCM+ – a CFD analysis software (JPADCAD, .Step exporter),
  - JSBSim – a flight simulation software github.com/JSBSim-Team/jsbsim
    - Import from CPACS (including aerodynamics and propulsion databases). ◀CPACS
- Aircraft export in CPACS format is functional, but still under development. ◀CPACS

JPADCommander

A GUI for aircraft authoring and analysis

GUI (Graphic User Interface)
Work in progress (will be completed by the end of the year)

JPADCommander incorporates all the functionalities provided by JPAD API.

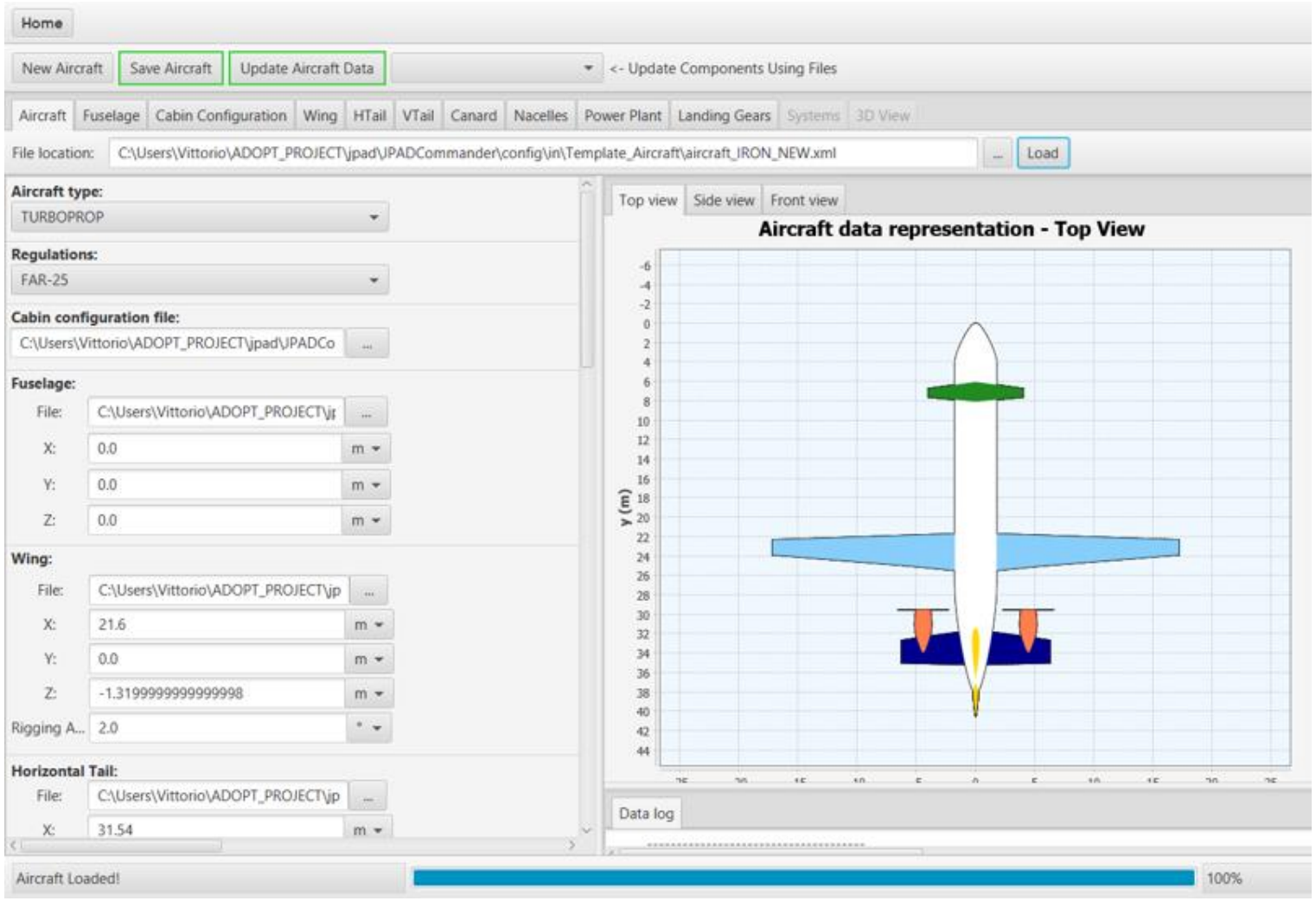
Users are guided throughout all the steps required to:

- Set up the aircraft components
- Prepare and run a set of analyses
- Visualize results

JPADCommander

A GUI for aircraft authoring and analysis

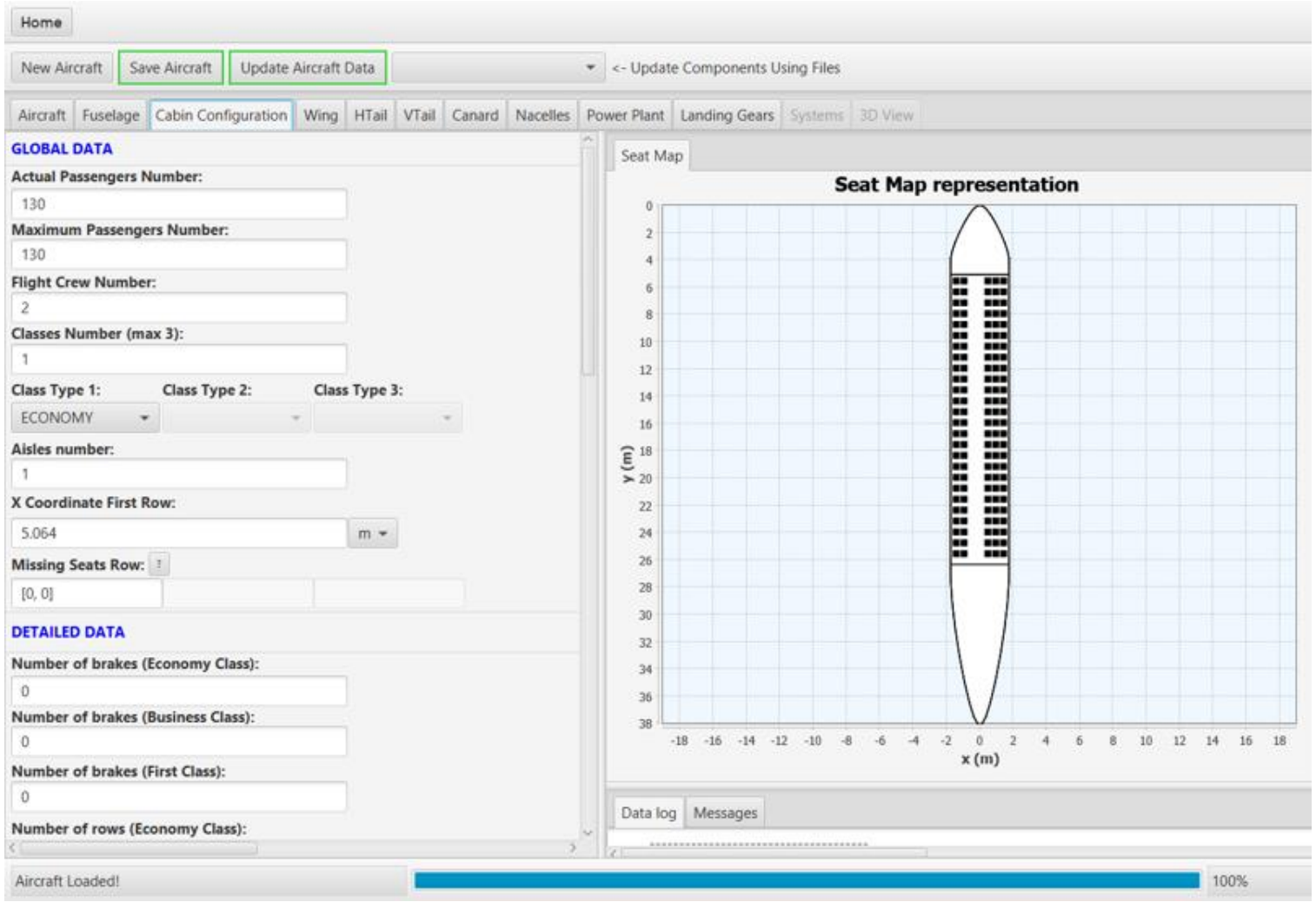
Input Manager:
Main Window



JPADCommander

A GUI for aircraft authoring and analysis

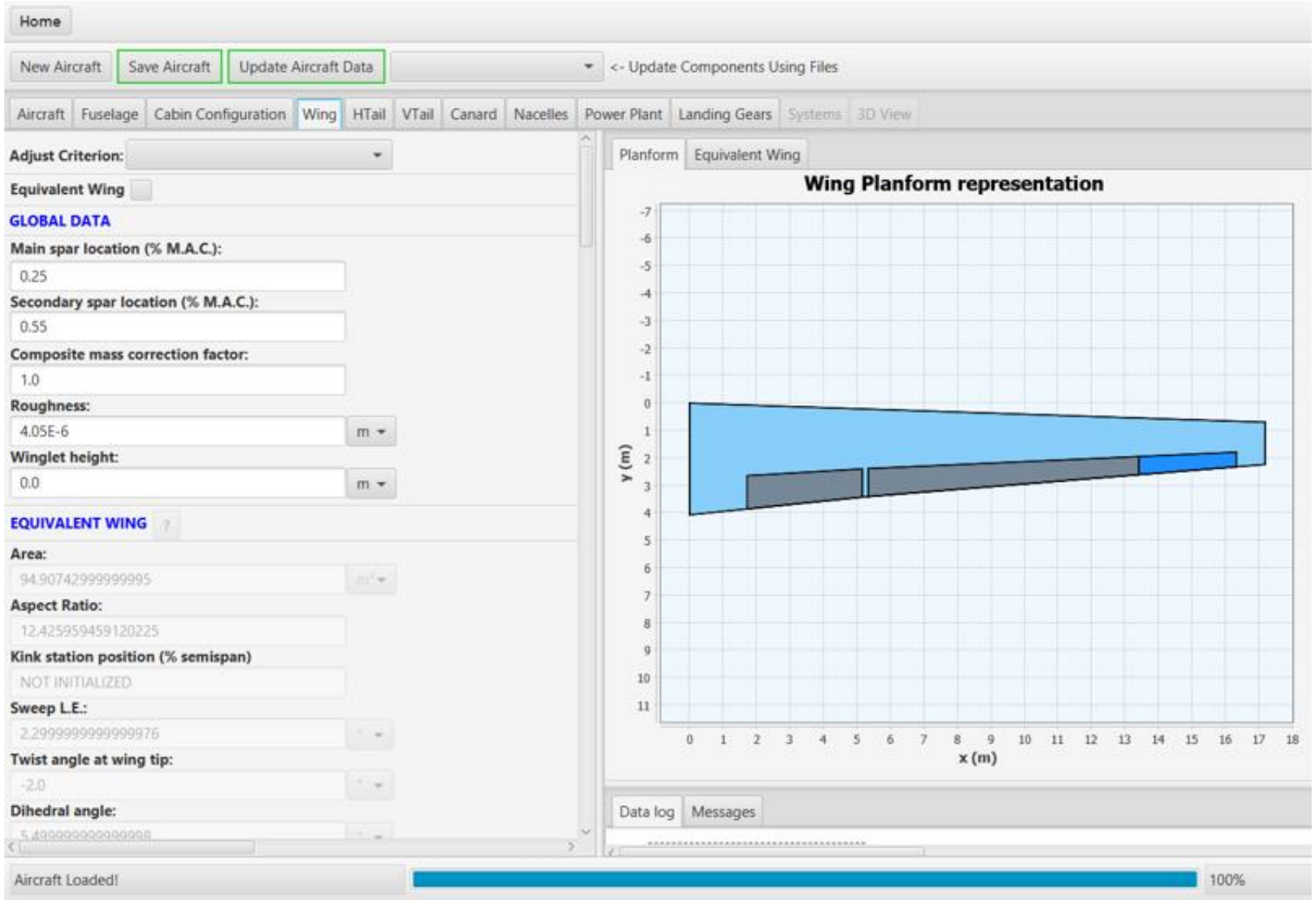
Input Manager:
Cabin Configuration Window



JPADCommander

A GUI for aircraft authoring and analysis

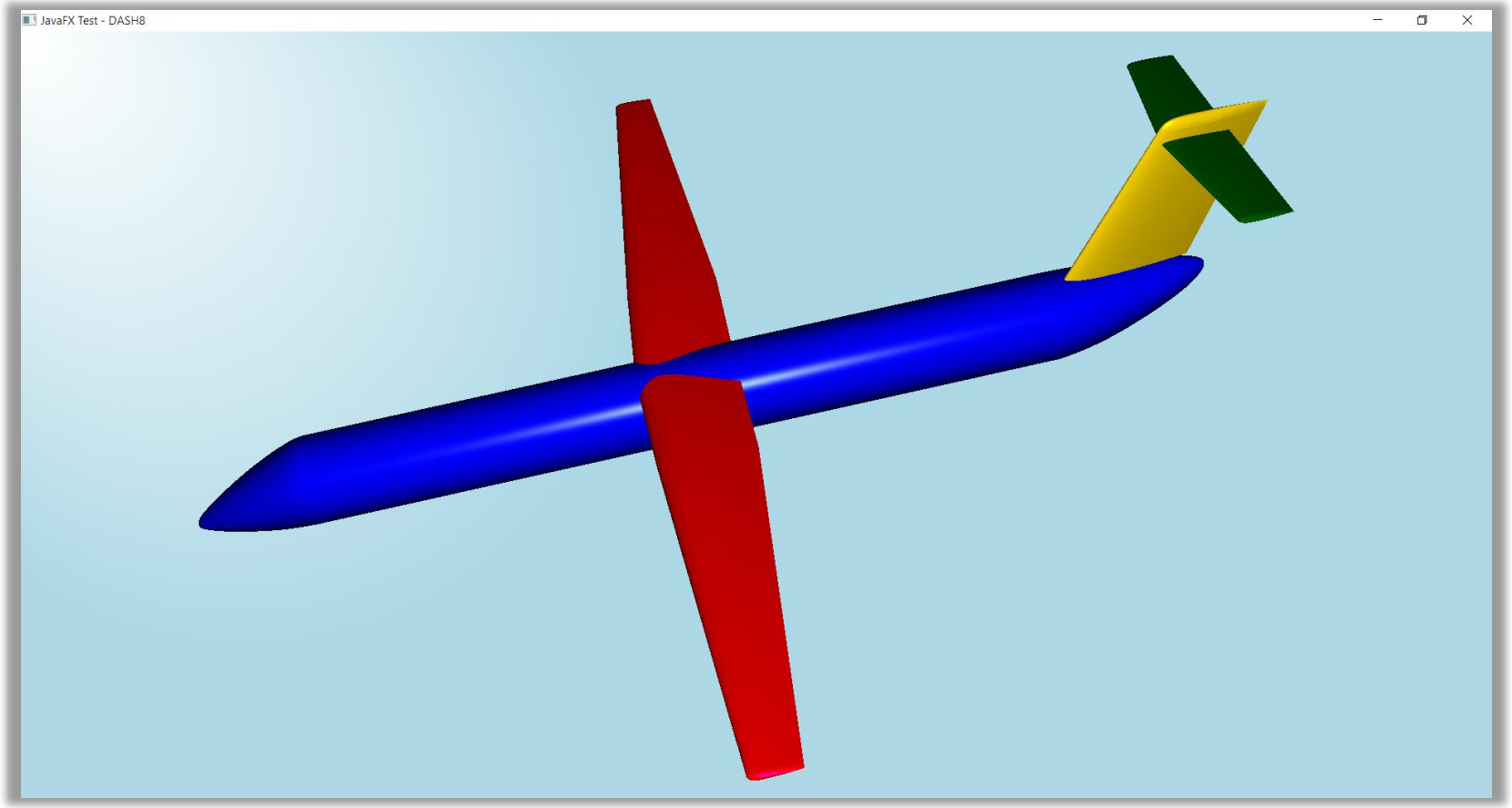
Input Manager:
Wing Configuration Window



JPADCommander

A GUI for aircraft authoring and analysis

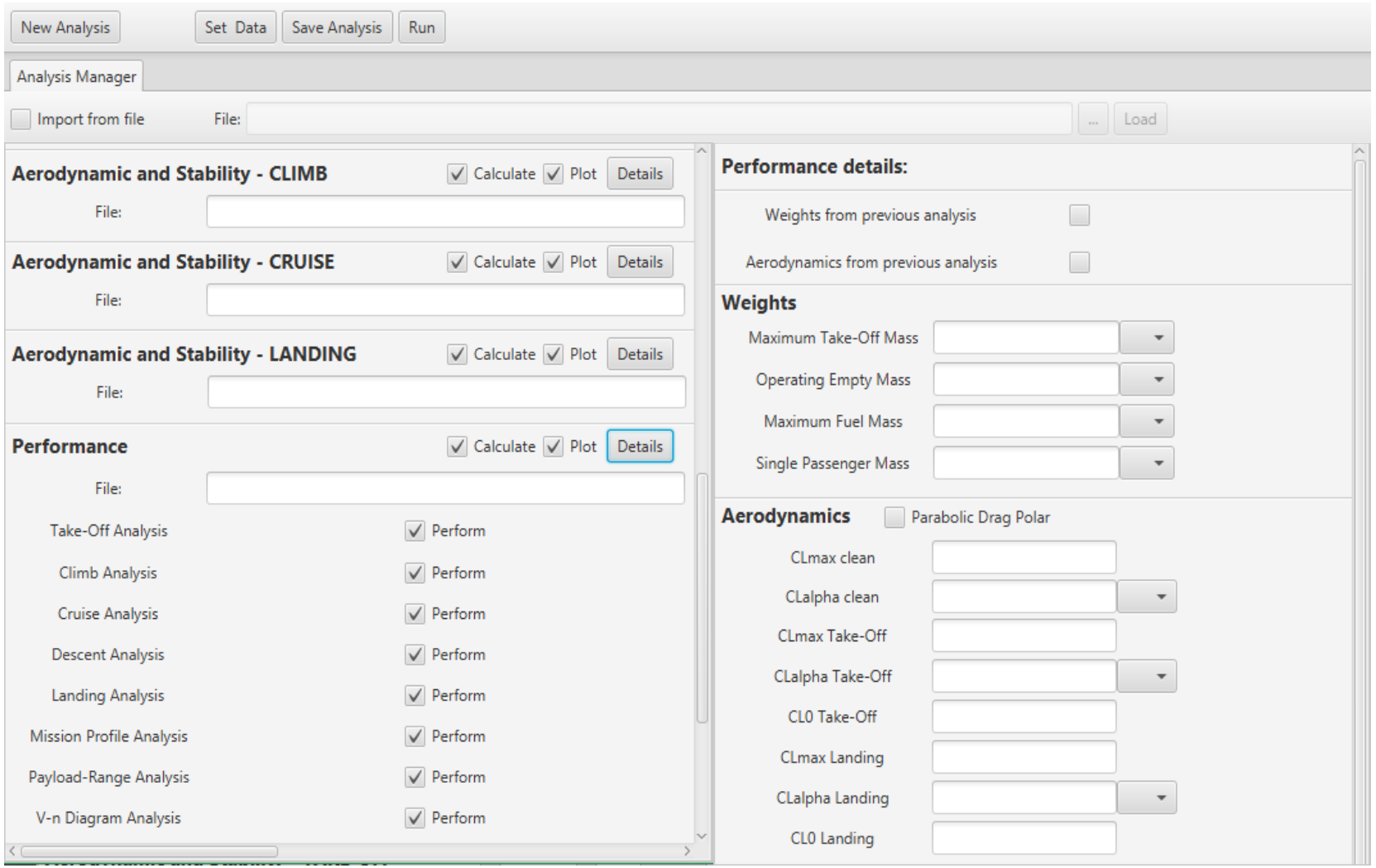
3D View as a JavaFX MeshView



JPADCommander

A GUI for aircraft authoring and analysis

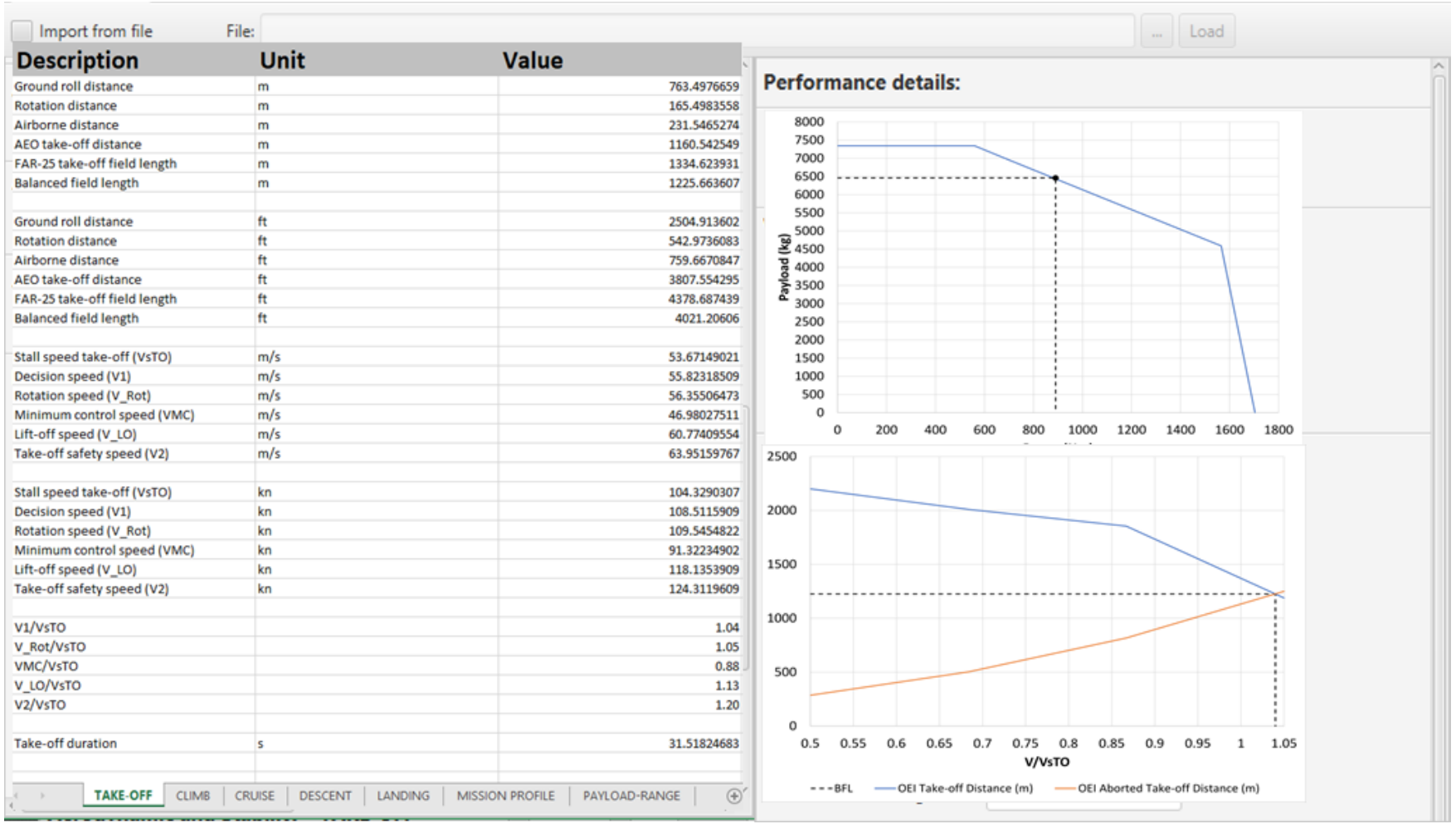
Analysis Manager



JPADCommander

A GUI for aircraft authoring and analysis

Results



Description	Unit	Value
Ground roll distance	m	763.4976659
Rotation distance	m	165.4983558
Airborne distance	m	231.5465274
AEO take-off distance	m	1160.542549
FAR-25 take-off field length	m	1334.623931
Balanced field length	m	1225.663607
Ground roll distance	ft	2504.913602
Rotation distance	ft	542.9736083
Airborne distance	ft	759.6670847
AEO take-off distance	ft	3807.554295
FAR-25 take-off field length	ft	4378.687439
Balanced field length	ft	4021.20606
Stall speed take-off (VsTO)	m/s	53.67149021
Decision speed (V1)	m/s	55.82318509
Rotation speed (V_Rot)	m/s	56.35506473
Minimum control speed (VMC)	m/s	46.98027511
Lift-off speed (V_LO)	m/s	60.77409554
Take-off safety speed (V2)	m/s	63.95159767
Stall speed take-off (VsTO)	kn	104.3290307
Decision speed (V1)	kn	108.5115909
Rotation speed (V_Rot)	kn	109.5454822
Minimum control speed (VMC)	kn	91.32234902
Lift-off speed (V_LO)	kn	118.1353909
Take-off safety speed (V2)	kn	124.3119609
V1/VsTO		1.04
V_Rot/VsTO		1.05
VMC/VsTO		0.88
V_LO/VsTO		1.13
V2/VsTO		1.20
Take-off duration	s	31.51824683

JPADCommander

A GUI for aircraft authoring and analysis

Video

JPADCAD

An API for aircraft automatic CAD modelling

JPADCAD

An API for aircraft automatic CAD modelling

- JPAD native aircraft models (external shapes) are exported as CAD models through the use of the Open CASCADE Technology (OCCT) library.
- OCCT is an open source software development kit, written in C++ and released by Open Cascade SAS.
- An advanced Java Wrapper Tool (Commercial, based on SWIG, by Open Cascade SAS) has been used to wrap OCCT C++ classes into Java classes, to allow their use from within Java applications.
- JPADCAD is a hierarchy of Java classes that incorporates high-level CAD functionalities.

JPADCAD

An API for aircraft automatic CAD modelling

- JPADCAD relies on the OCCT Java Wrapper, and contains all the classes and utilities for CAD implementation.
- CAD entities, such as edges and faces, etc, are modelled following an Object-Oriented Paradigm.
- OCCShape implements the Abstract Factory Pattern.
- Non-abstract geometric entities, such as curves, surface etc, have been implemented as OCCXXX classes.
- Example: An interface for geometric 3D curves (CADGeomCurve3D) has been created, alongside its implementation (OCCGeomCurve3D).
- OCCUtils is the utility class used to accomplish common CAD operations.

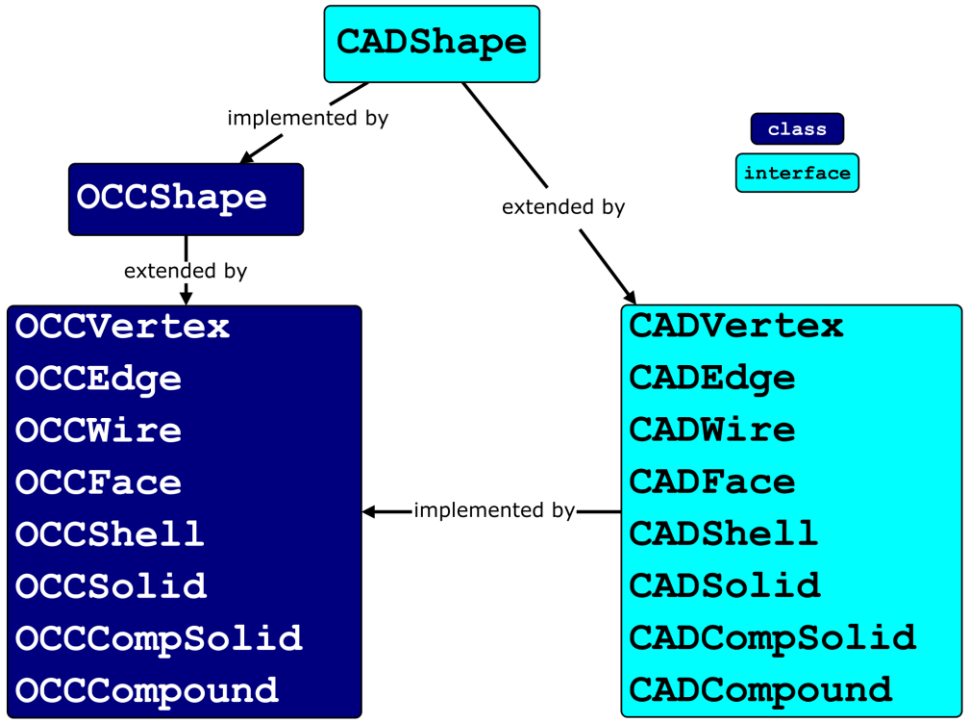


Figure 2.4 Relationship diagram for JPADCAD topological classes and interfaces

JPADCAD

An API for aircraft automatic CAD modelling

Open CASCADE	JPAD	
	Abstract Topology	Concrete Topology
TopoDS_Shape	CADShape	OCCShape
TopoDS_Vertex	CADVertex	OCCVertex
TopoDS_Edge	CADEdge	OCCEdge
TopoDS_Wire	CADWire	OCCWire
TopoDS_Face	CADFace	OCCFace
TopoDS_Shell	CADShell	OCCShell
TopoDS_Solid	CADSolid	OCCSolid
TopoDS_CompSolid	CADCompSolid	OCCCompSolid
TopoDS_Compound	CADCompound	OCCCompound

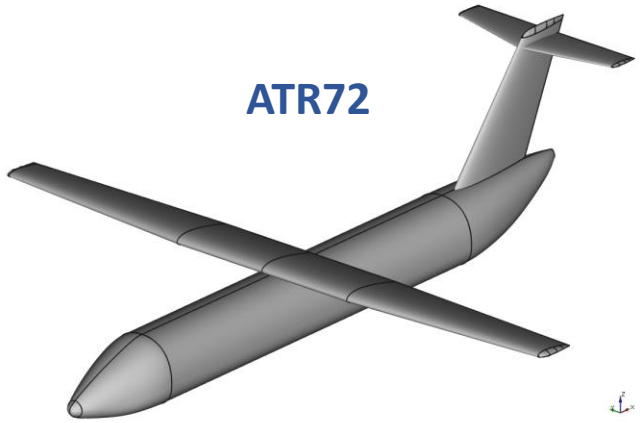
Table 2.2 JPADCAD topology classes overview

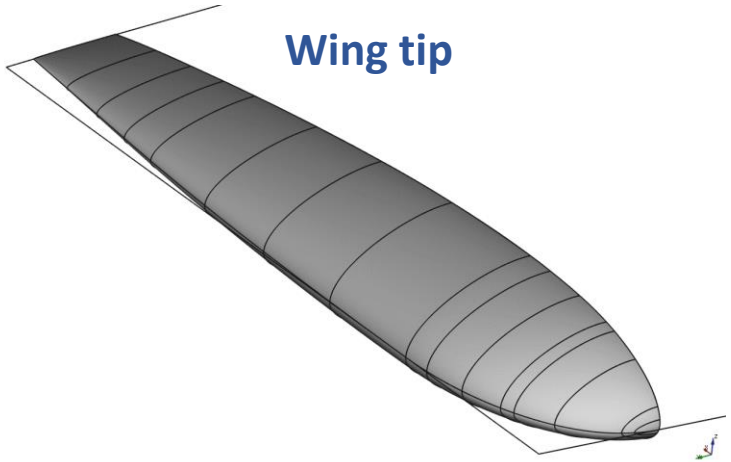
Open CASCADE	JPAD	
	Abstract Geometry	Concrete Geometry
Geom2dAdaptor_Curve	CADGeomCurve2D	OCCGeomCurve2D
GeomAdaptor_Curve	CADGeomCurve3D	OCCGeomCurve3D
Geom_Surface	CADGeomSurface	OCCGeomSurface

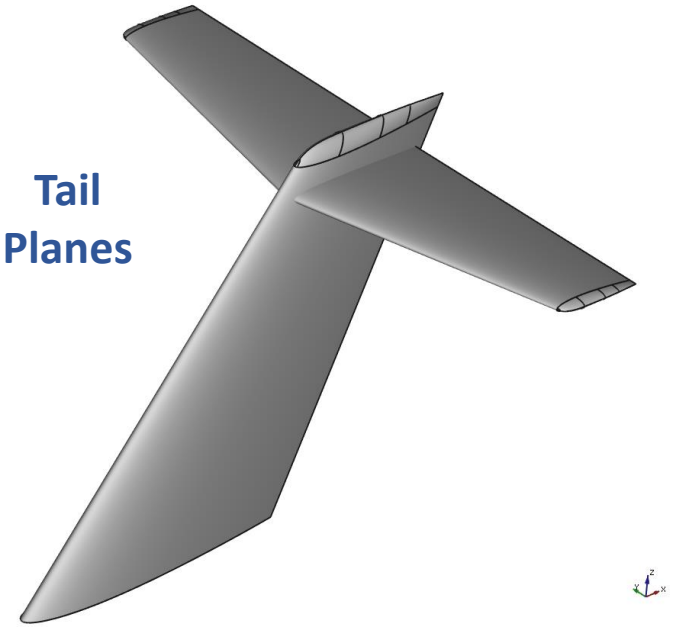
Table 2.4 JPADCAD geometry classes overview

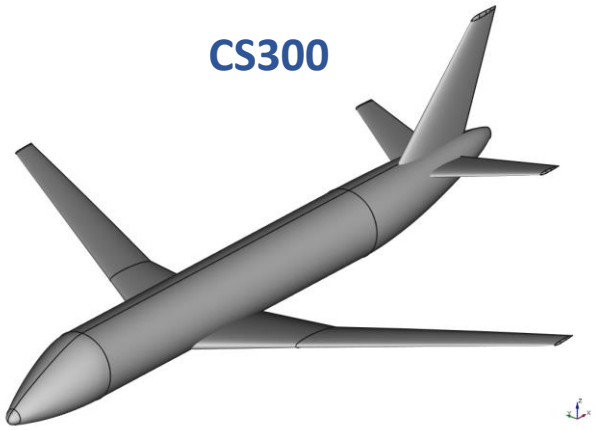
JPADCAD

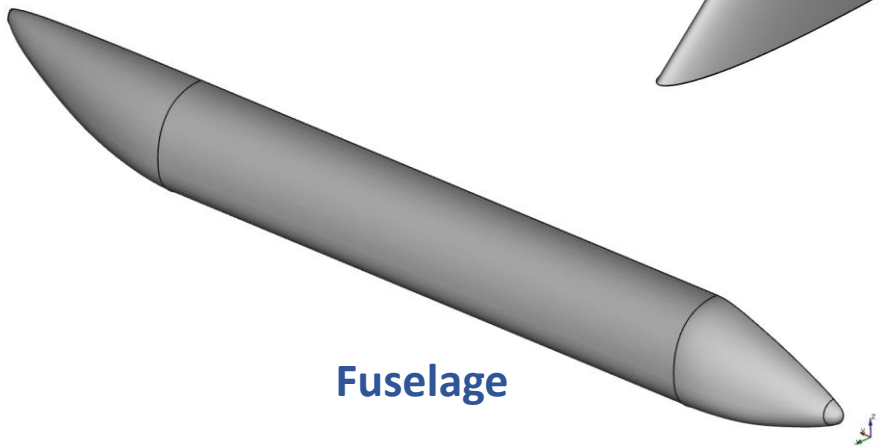
An API for aircraft automatic CAD modelling







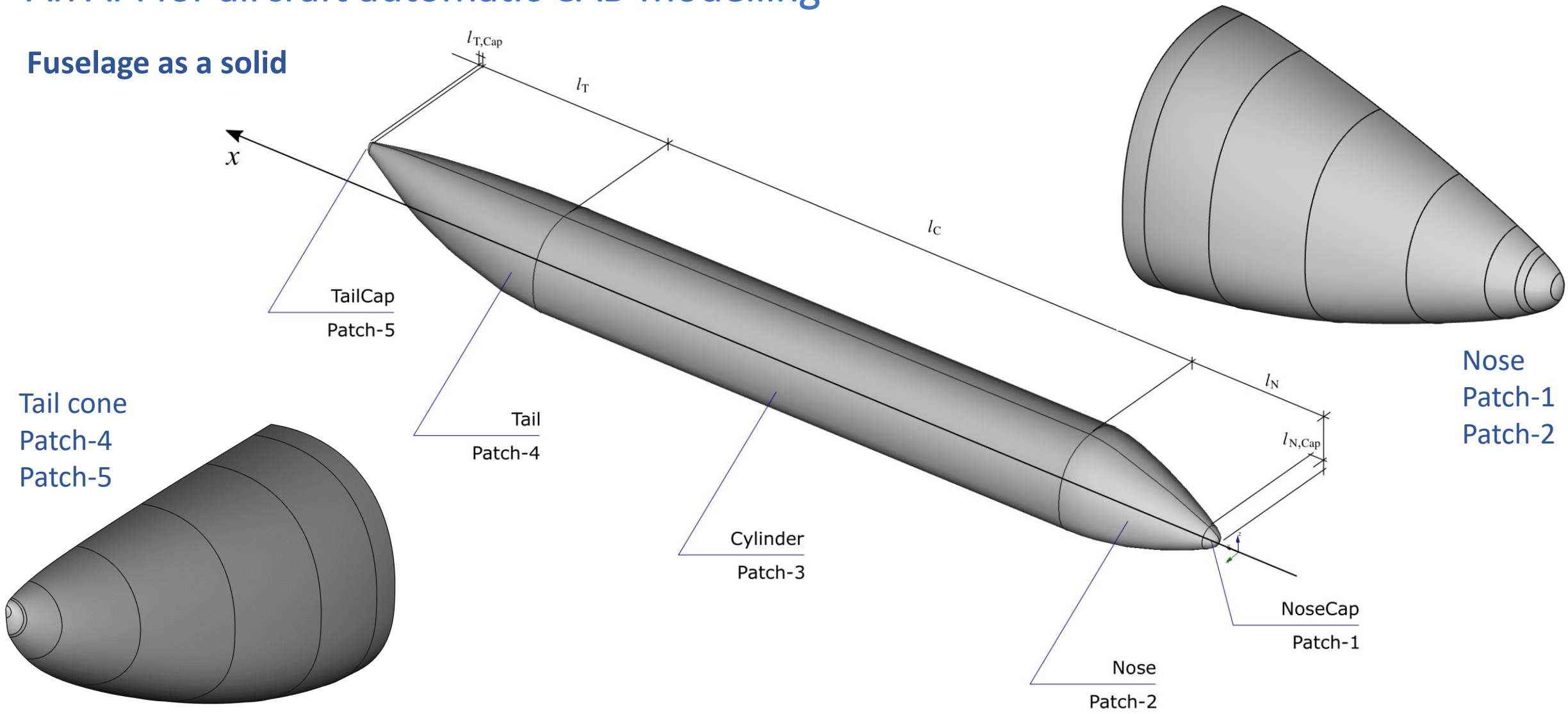




JPADCAD

An API for aircraft automatic CAD modelling

Fuselage as a solid



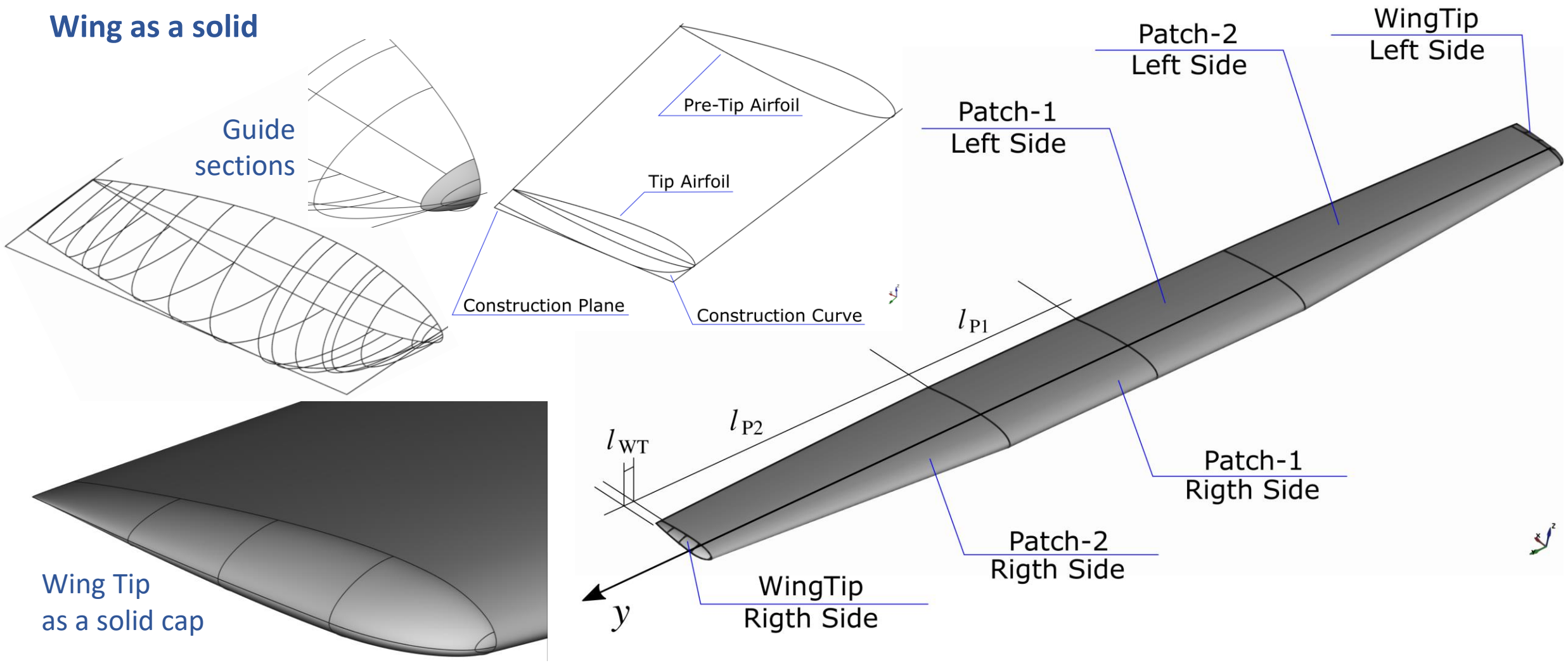
Tail cone
Patch-4
Patch-5

Nose
Patch-1
Patch-2

JPADCAD

An API for aircraft automatic CAD modelling

Wing as a solid

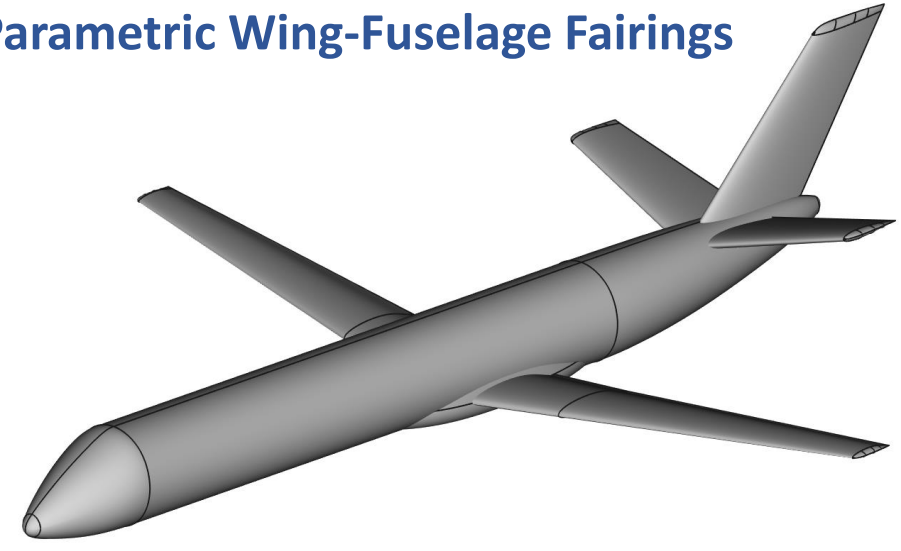


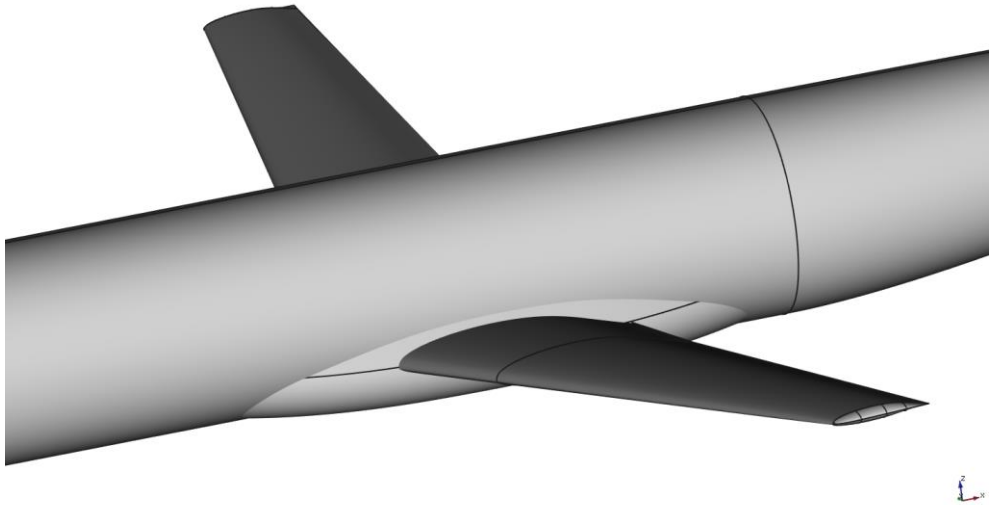
Wing Tip as a solid cap

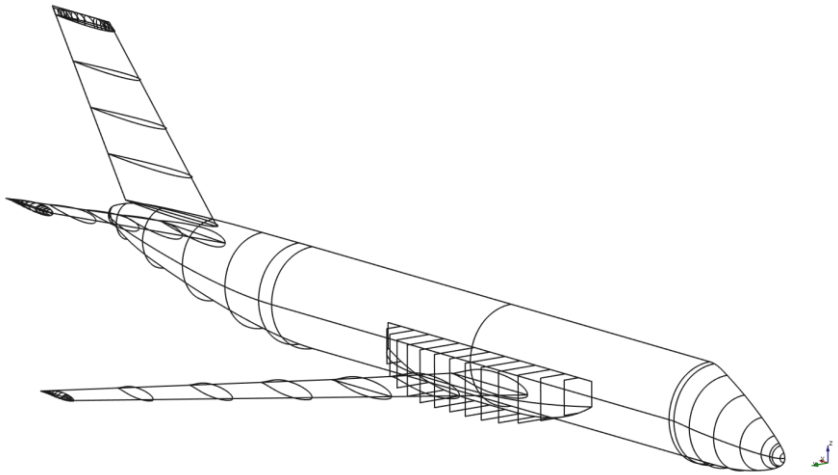
JPADCAD

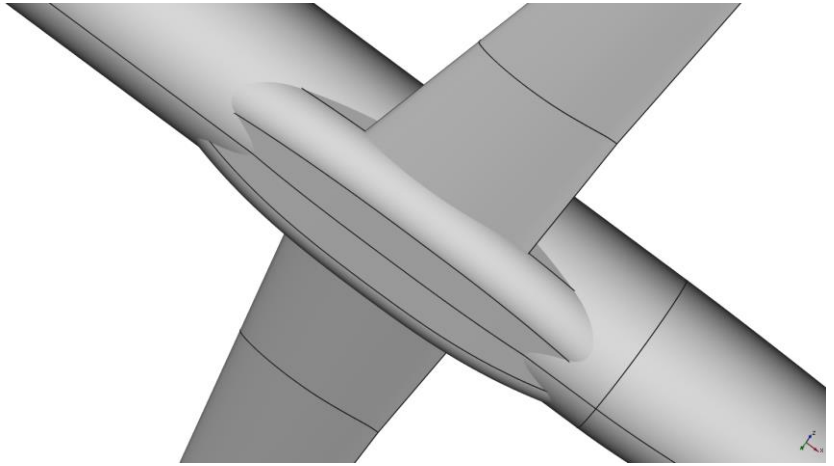
An API for aircraft automatic CAD modelling

Parametric Wing-Fuselage Fairings





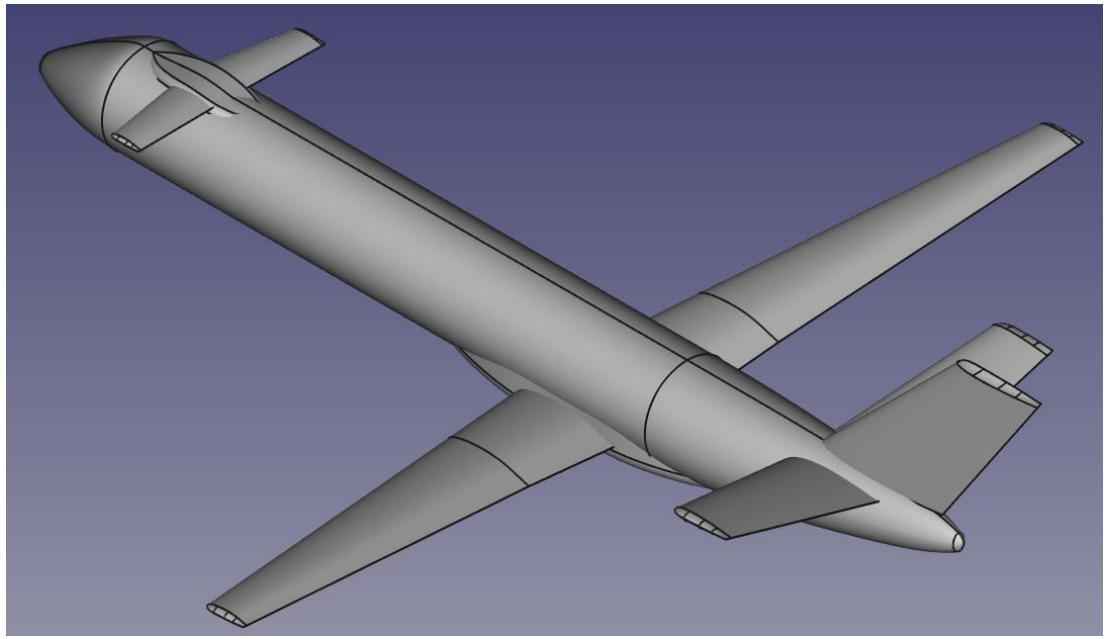


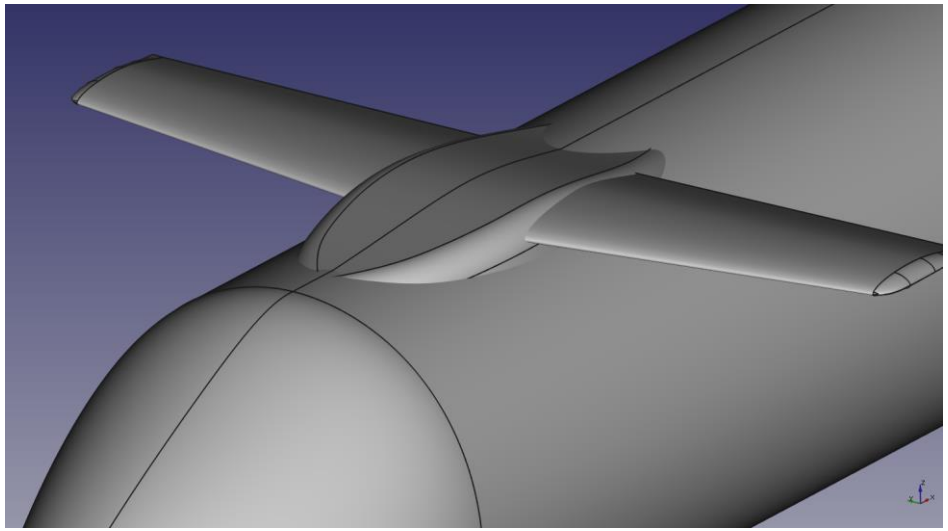


JPADCAD

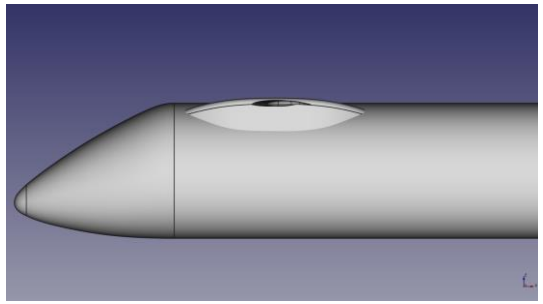
An API for aircraft automatic CAD modelling

Parametric Wing-Fuselage Fairings

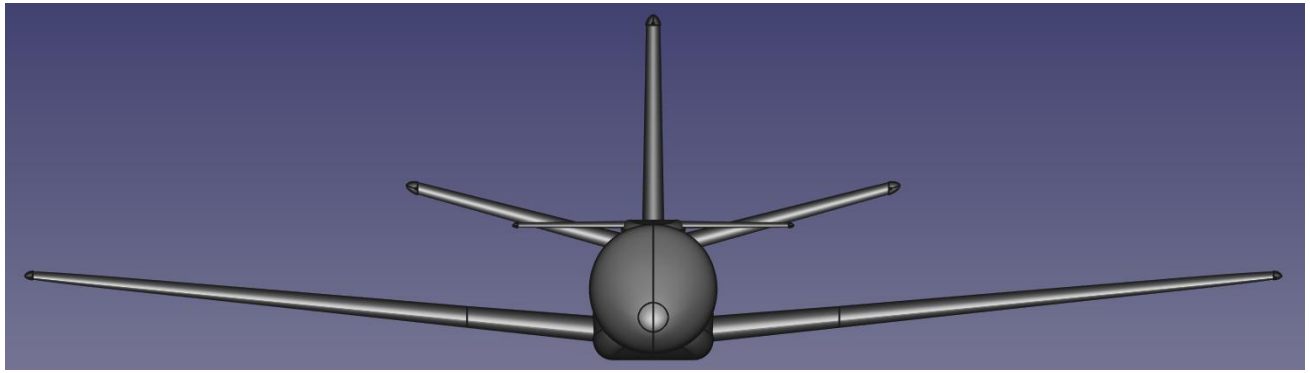




EU H2020
Project IRON

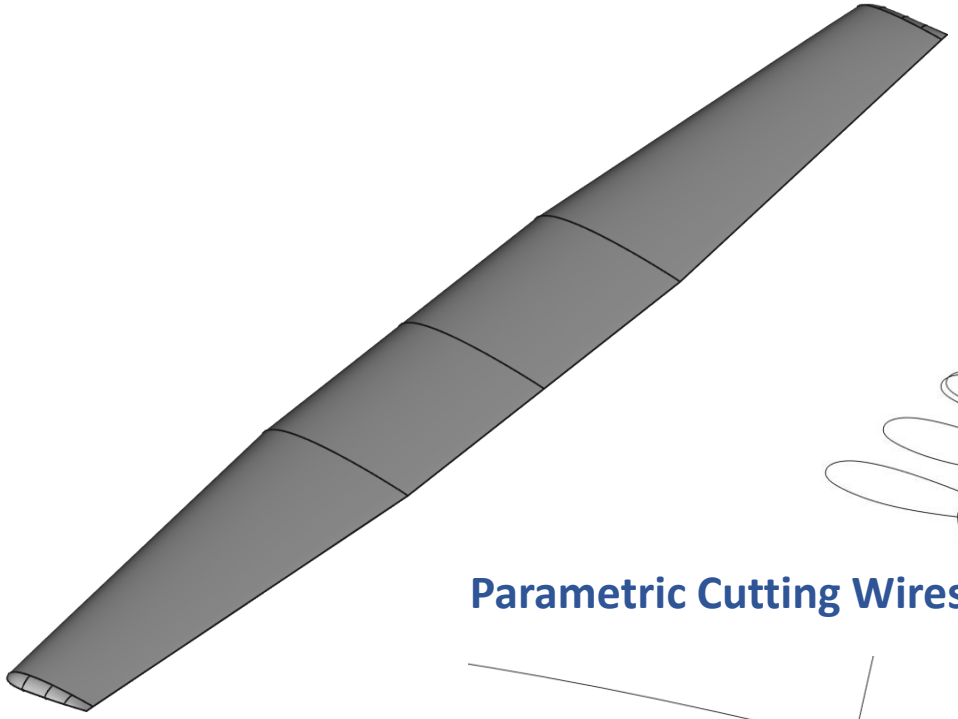


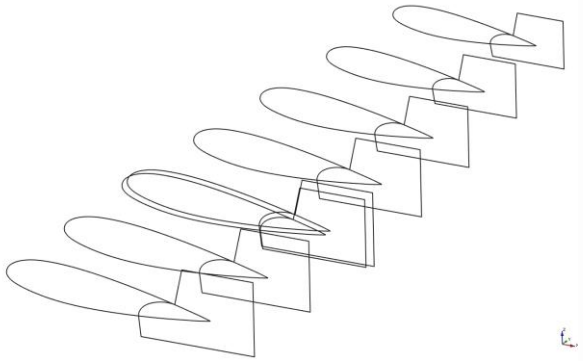
Horizontal-Tail
mounted engines
(not shown)



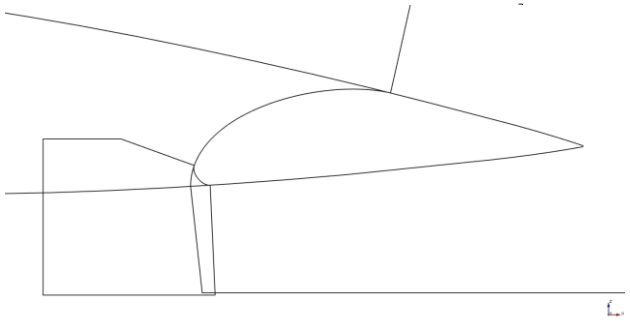
JPADCAD

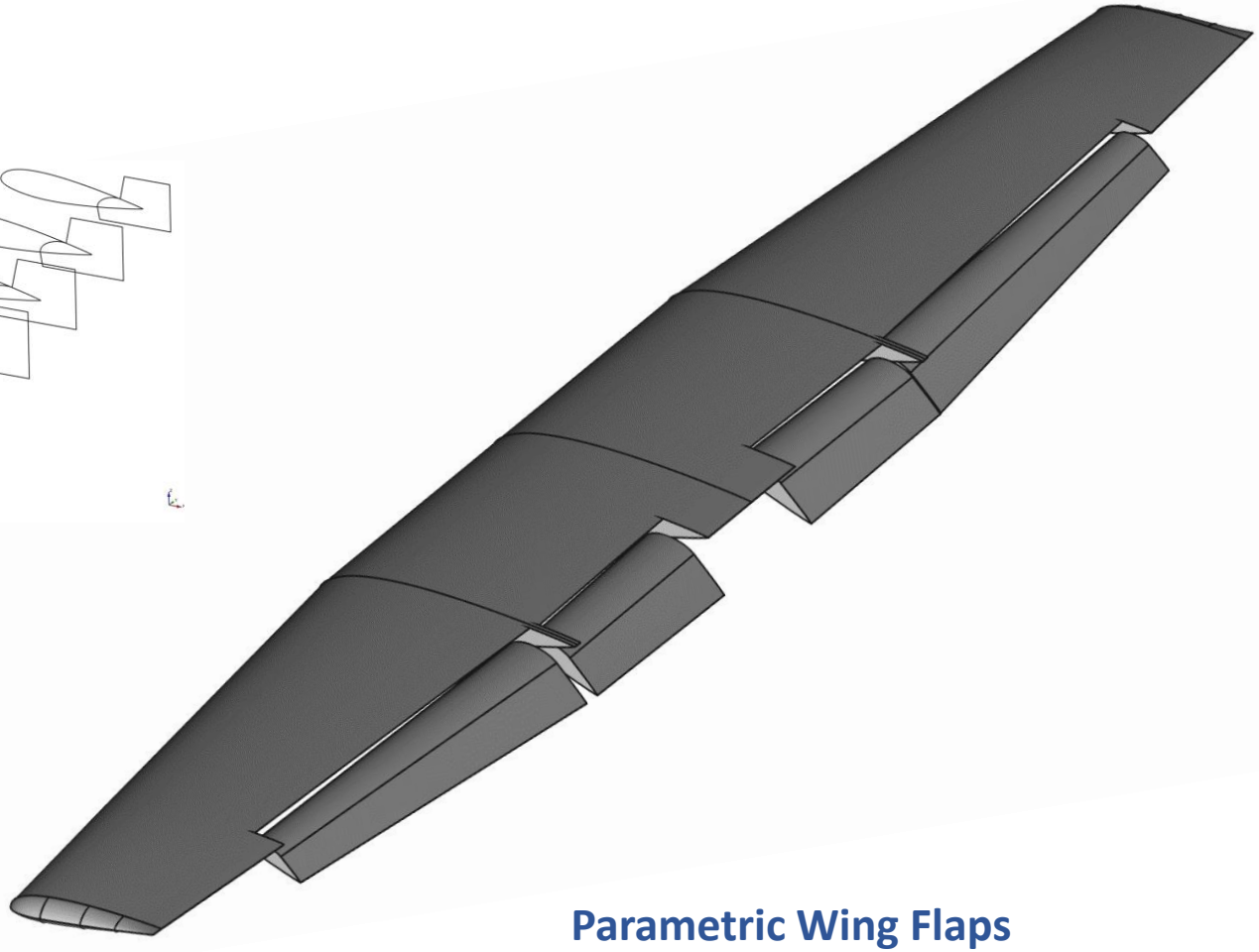
An API for aircraft automatic CAD modelling





Parametric Cutting Wires

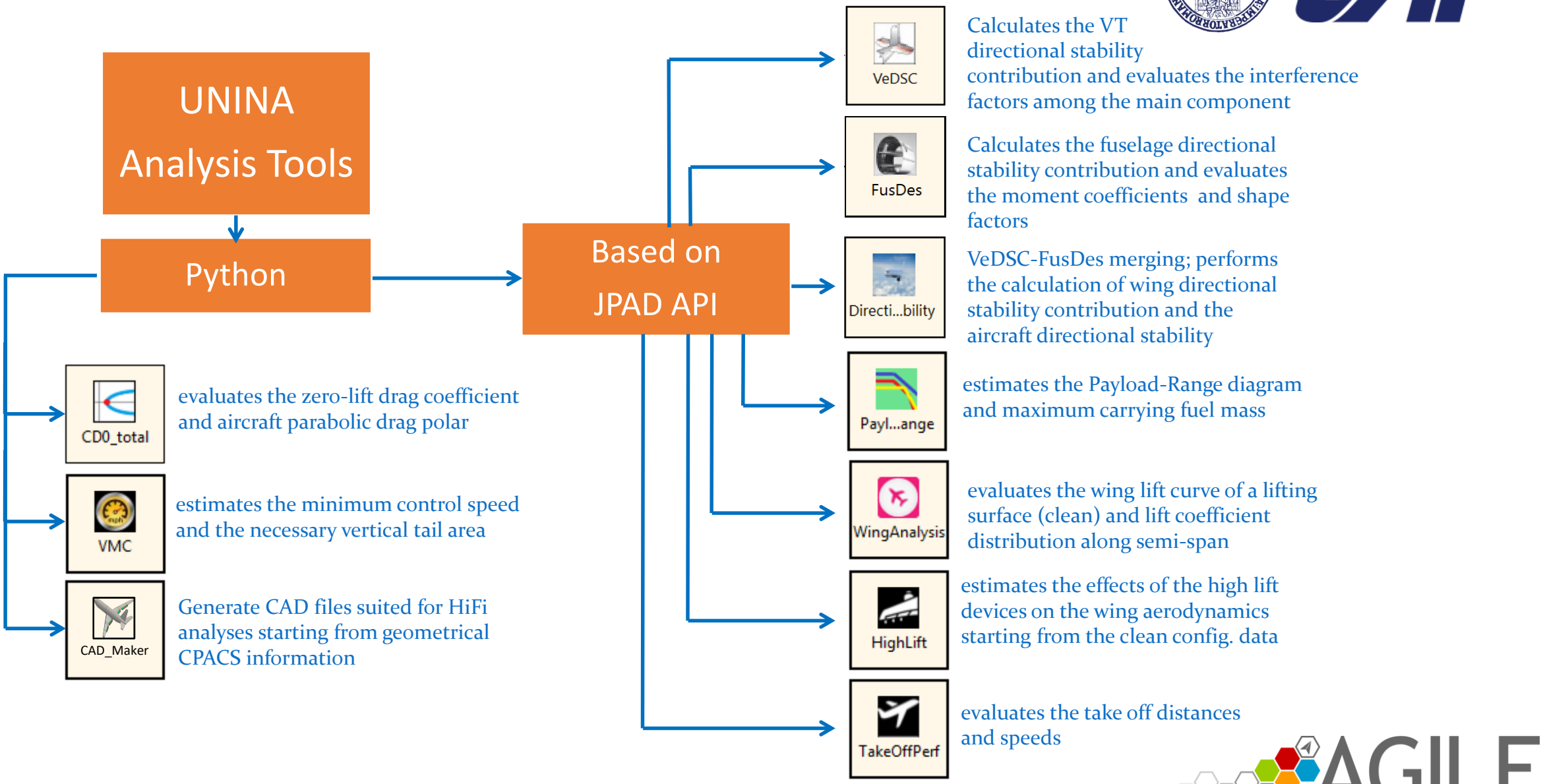




Parametric Wing Flaps

UniNa & DLR Tools

UniNa & DLR Tools – Project AGILE



UniNa & DLR Tools – Project AGILE

cpacs
A Common Language for Aircraft Design

Stored aircraft data

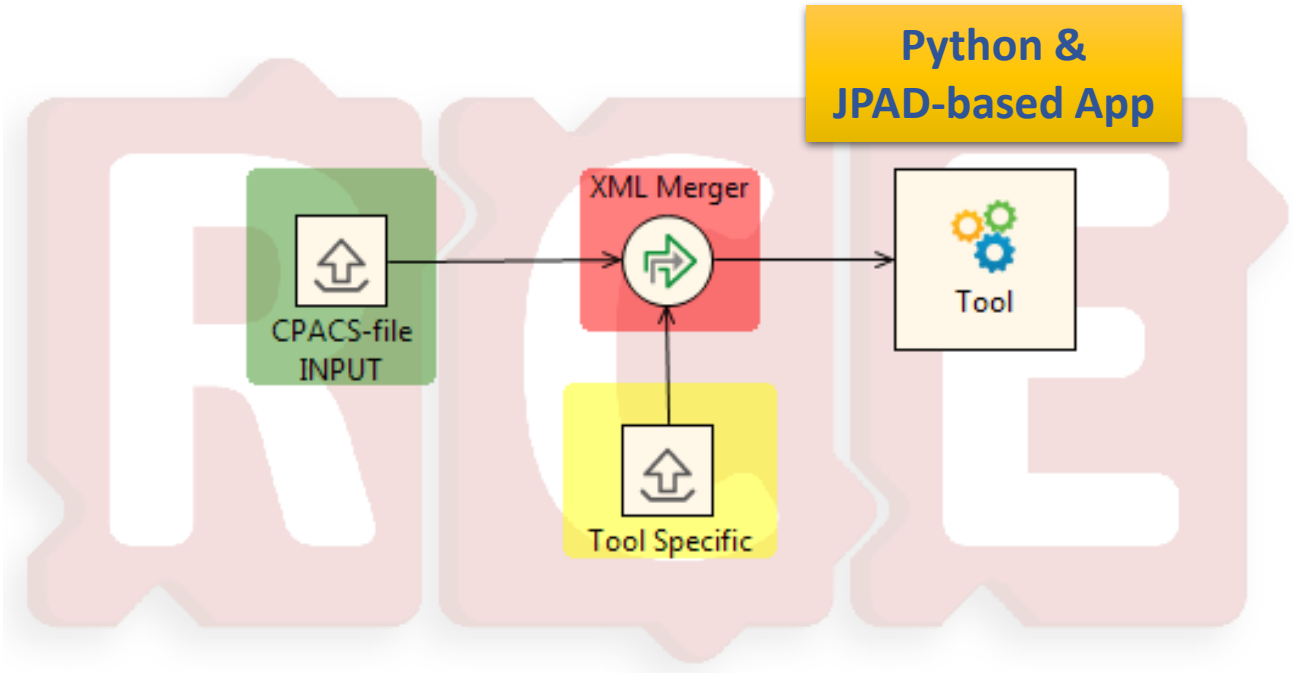
python

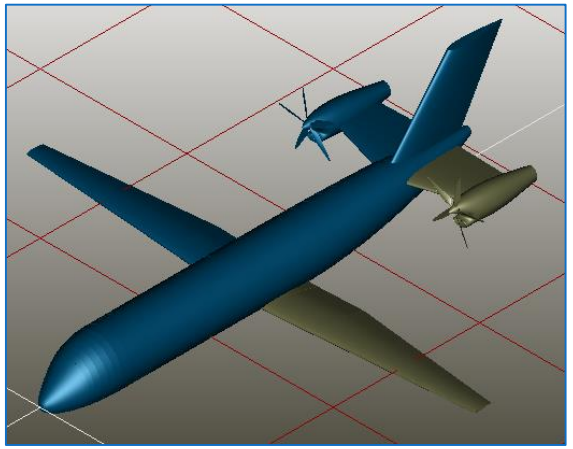
Python algorithm

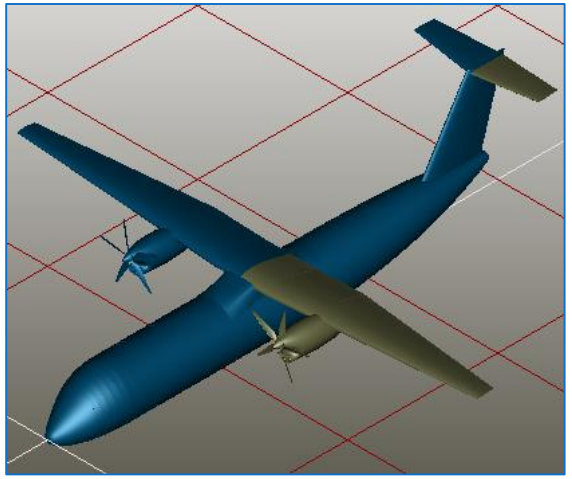
cpacs
A Common Language for Aircraft Design

Aircraft data and module results

RCE a distributed, workflow-driven integration environment by DLR



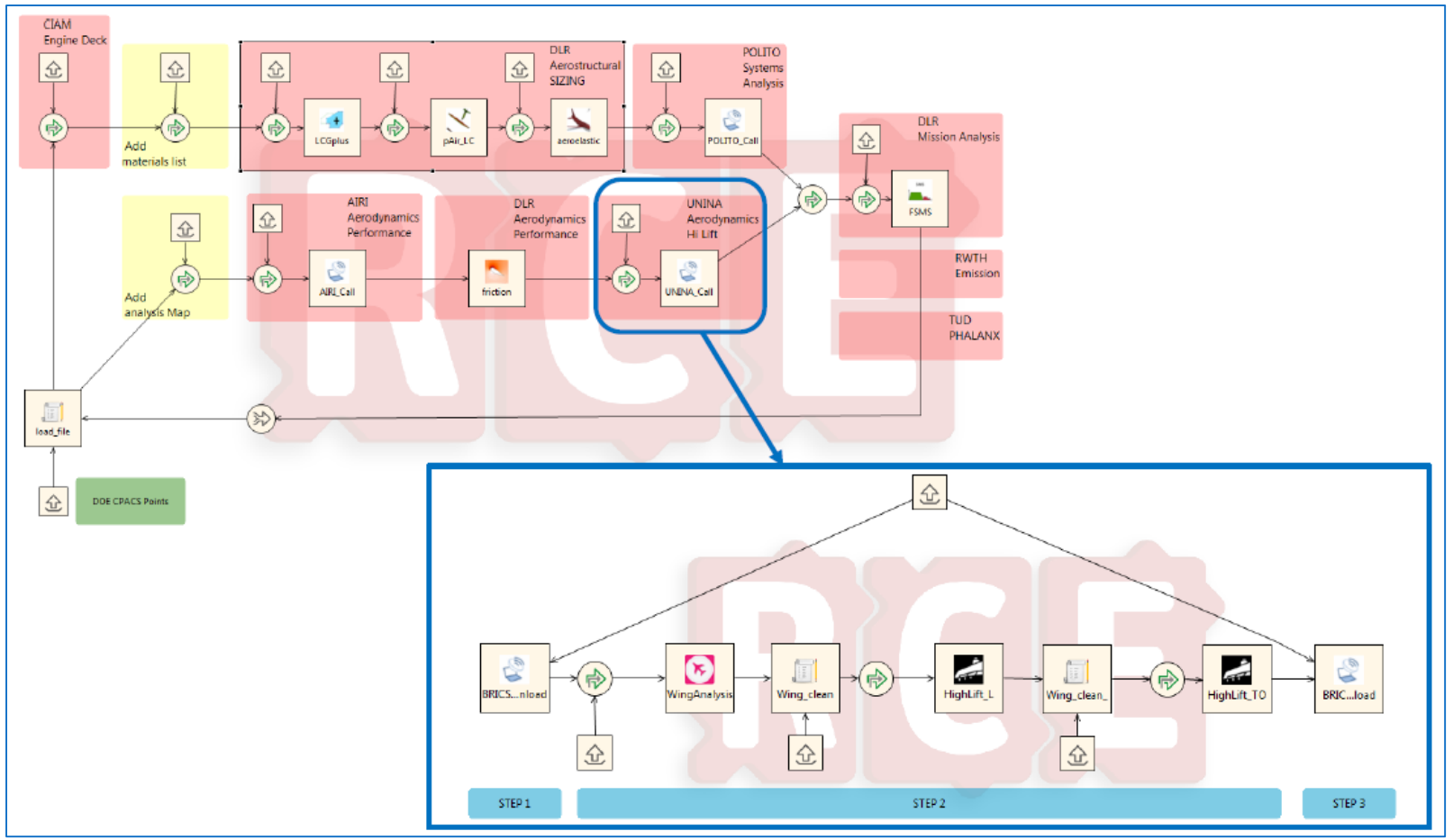




AGILE

MDAO – Project AGILE

UNINA and partners' tools are used to setup an entire Multidisciplinary Design Analysis and Optimization (MDAO) workflow.



Tools can be used in local or through remote connection.

It is possible to assemble a workflow composed by parteners' tools worldwide distributed.

Conclusions

- JPAD is a mature software library, approaching release as v1.0
  - The Analysis Manager and its submodules … need more testing
  - The Optimization module … needs to be completed
- JPADCAD is well integrated in a workflow with StarCCM+
  - Automatic flapped wing generation … needs more testing
  - Automatic wing-fuselage fairing generation … needs more testing
  - Investigate low-level integration with TiGL … a collaboration with DLR is welcome, mainly to enhance/refine TiGL bindings (Java, Python, etc)
- JPAD-based aerodynamic & performance analysis tools are integrated effectively into RCE workflows.