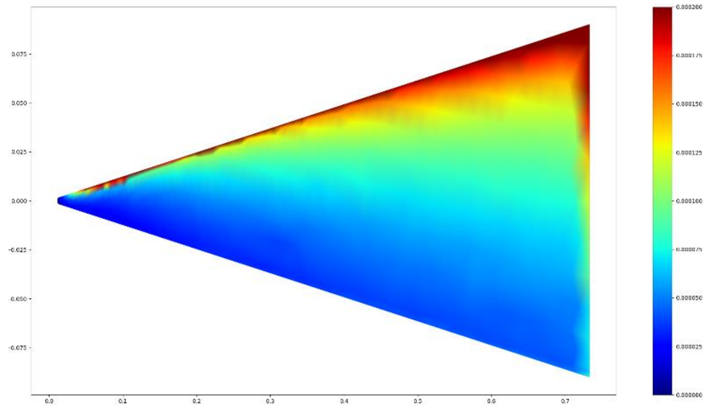
PFAT
EXECUTIVE SUMMARY

Post-Flight Analysis Toolkit



Code:	PFAT-DMS-TEC-EXS
Issue:	1.0
Approval Date:	16/12/2021
Confidentiality Level:	Unclassified

Approval Signature:

Prepared by:	PFAT Development Team
Reviewed by:	F. Letterio/ Project Manager
Approved by:	F. Letterio/ Project Manager

This page intentionally left blank

Document Status Log

Issue	Section	Change Description	Date
1.0	All	First release of the document	16/12/2021

Table of Contents

1. INTRODUCTION 7
- **1.1. Background** 7
- **1.2. Purpose** 7
- **1.3. Scope** 7
- **1.4. Acronyms and Abbreviations** 7
- **1.5. Definitions** 8

2. RELATED DOCUMENTS 10
- **2.1. Applicable Documents** 10
- **2.2. Reference Documents** 10
- **2.3. Standards** 11

3. PROJECT HISTORY 12
- **3.1. Project Objectives** 12
- **3.2. Schedule** 13
- **3.3. Project Team** 14
- **3.4. Work Breakdown Structure** 14
- **3.5. Study Logic** 15

4. POST-FLIGTH ANALYSIS TOOLKIT 17
- **4.1. SW Development** 17
- **4.2. Tool Capabilities** 17
  - 4.2.1. Computational Core 18
  - 4.2.2. Common Data Structure 19
  - 4.2.3. Executable Modules 19
    - 4.2.3.1. Aero-thermodynamics 20
    - 4.2.3.2. Propulsion 21
    - 4.2.3.3. Structural 21
    - 4.2.3.4. Trajectories 22
  - 4.2.4. Graphical User Interface 22
- **4.3. Validation and Verification Approach** 23
- **4.4. Post-Flight Analysis Campaign** 23
  - 4.4.1. Trajectory Domain 23
  - 4.4.2. Aerothermodynamic Domain 24
  - 4.4.3. Propulsion Domain 26
  - 4.4.4. Structural Domain 28

5. CONCLUSIONS AND FUTURE PESPECTIVES 29
- **5.1. Achievements of PFAT Project** 29
- **5.2. Future Work** 29

List of Figures

Figure 1: PFAT Project Schedule 13
Figure 2: PFAT Industrial Consortium 14
Figure 3: PFAT Work Breakdown Structure........ 14
Figure 4: PFAT Study Logic 15
Figure 5: PFAT Software Development Life Cycle Process Diagram 17
Figure 6: PFAT System Decomposition........ 18
Figure 7: Lift and drag coefficients processing chain 20
Figure 8: Heat flux probability density function processing chain........ 20
Figure 9: Pressure power spectral density processing chain........ 20
Figure 10: Boundary integral processing chain 20
Figure 11: Intermittency calculation processing chain........ 21
Figure 12: Propulsion analysis processing chain 21
Figure 13: Structural analysis processing chain 21
Figure 14: Separation analysis processing chain 21
Figure 15: Acoustic analysis processing chain........ 21
Figure 16: Trajectory reconstruction processing chain 22
Figure 17: Forward (left) and Smoothed (Right) Position Uncertainty Evolution (Blue X, Red Y and Green Z Coordinates) and Altitude Evolution 24
Figure 18: PFAT vs reference tool results of PSD pressure processing chain 25
Figure 19: PFAT vs reference tool results of heat flux probability density function processing chain........ 25
Figure 20: PFAT vs reference tool results of boundary integral processing chain 25
Figure 21: PFAT vs reference tool results of lift and drag coefficients processing chain 26
Figure 22: PFAT vs reference tool results of intermittency processing chain........ 26
Figure 23: Combustor model reconstructed thrust and specific impulse (blue) vs simulated (red) performances........ 27
Figure 24: Expander model reconstructed thrust and specific impulse (blue) vs simulated (red) performances........ 27
Figure 25: Pressure fed model reconstructed thrust and specific impulse (blue) vs simulated (red) performances with default fitting expressions........ 27
Figure 26: Pressure fed model reconstructed thrust and specific impulse (blue) vs simulated (red) performances with user defined fitting expressions 28
Figure 27: Separation model – time history of the nodal displacements difference between baseline and comparison model, for the nodes showing the greatest discrepancy 28

List of Tables

Table 1: Applicable documents 10
Table 2: Reference documents 10
Table 3: Standards 11

1. INTRODUCTION

1.1. Background

Throughout the life time of ESA missions, from conceptual design (Phase 0) until disposal (Phase F), an enormous amount of data is generated in each of the phases for the different engineering disciplines, e.g. aero(thermo)dynamic, structural and thermal analysis, models of the propulsion subsystem, design trajectories.

A big part of these data is primarily produced during the design phases of a mission, but more data is produced again during the operations of it. ESA has a need during the post-operational mission phases to evaluate the flight performance and conduct the so-called post-flight analysis. A large amount of data therefore needs to be analysed methodically to assess the predicted performances throughout the mission and understand the potential uncertainties. Such an analysis would increase the fidelity and future design iterations of ESA missions, and could be used to assess the discrepancies between predicted and observed flight behaviour.

These data originate from theoretical analysis, computational design, numerical simulations from engineering and high-fidelity codes, ground and flight experiments, and would thus have different dataset formats. The manual extraction, manipulation and aggregation of such large datasets is tedious, time consuming, and prone to errors. Each of those disciplines use different kind of software design and simulations tools across the phases of a given project. A deep post flight data analysis would involve the running of those different tools per discipline, and the exchange of the data between disciplines.

PFAT is an activity aimed at developing a software tool that should specifically act as glue between different discipline tools with different datasets, allowing the extraction of figures-of-merit and uncertainties, to derive engineering criteria used for further flight vehicle and mission re-design.

1.2. Purpose

The objective of this document is to provide the Executive Summary for the Post-Flight Analysis Tool (PFAT) activities, with a summary of the background, design and development, functionalities and outcome of the PFAT tool.

1.3. Scope

This document has been produced in the frame of the WP 6000 (Project Management) of the contract 4000108365/13/NL/CT between ESA-ESTEC and DEIMOS Space (DMS) for the study on.

1.4. Acronyms and Abbreviations

The acronyms and abbreviations used in this document are the following ones:

AEDB	Aerodynamic Database
AoA	Angle of Attack
AoS	Angle of Sideslip
API	Application Programming Interface
CDS	Common Data Structure
CFD	Computer Fluid Dynamics
CFI	Customer Furnished Item
CI	Continuous Integration

COTS	Commercial Off-The-Shelf
DMS	Deimos Space
EAI	Empresarios Agrupados Internacional
EDL	Entry Descent and Landing
EIP	Entry Interface Point
EKF	Extended Kalman Filter
EO	Earth Observation
ESPSS	European Space Propulsion System Simulation
FDIR	Failure Detection Isolation and Recovery
FEM	Finite Element Model
FES	Functional Engineering Simulator
FF	Formation Flying
GNC	Guidance Navigation and Control
HDF	Hierarchical Data Format
HFM	Hot Film Air Mass
IC	Initial Conditions
IDS	Internal Data Structure
INS	Inertial Navigation System
KPI	Key Performance Indicator
NEO	Near-Earth Object
OBT	On-Board Telemetry
OS	Operating System
P/T	Pressure and temperature
PFA	Post-Flight Analysis
PFAT	Post-Flight Analysis Tool
PLF	Payload Fairing
QA	Quality Assurance
RCS	Reaction Control System
RVD	Redezvous and Docking
SoW	Statement of Work
SPR	Software Problem Report
SUM	Software User Manual
SVT	Software Validation and Testing
SW	Software
TDMS	Technical Data Management Streaming
TPS	Thermal Protection System
WP	Work Package

1.5. Definitions

- ☐ The definitions of the specific terms used in this document are the following ones:
  - ☐ **Acceptance Testing**: Formal testing conducted to determine whether or not a system satisfies the acceptance criteria, previously defined by the customer.

- ☐ **Graphical User Interface**: interface that allows users to interact with the system through graphical icons and visual indicators.
- ☐ **Integration Testing**: An orderly progression of testing in which software elements, hardware elements, or both are combined and tested until the entire system has been integrated.
- ☐ **System Testing**: The process of testing an integrated hardware and software system to verify that the system meets its software requirements.
- ☐ **Test Case**: Documentation specifying inputs, predicted results, and a set of execution conditions for a test item.
- ☐ **Test Plan**: Documentation specifying the scope, approach, resources, and schedule of intended testing activities.
- ☐ **Test Procedure**: Documentation specifying a sequence of actions for the execution of a test.
- ☐ **Use Case**: a list of action or event steps, typically defining the interactions between a role (known in the UML as an actor) and a system, to achieve a goal. The actor can be a human, an external system, or time.
- ☐ **Verification:** Confirmation, through the provision of objective evidence, that specified requirements have been fulfilled [ISO 9000:2005]

  NOTE: verification process (for software) is the process to confirm that adequate specifications and inputs exist for any activity, and that the outputs of the activities are correct and consistent with the specifications and input.
- ☐ **Validation:** Confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled [ISO 9000:2005]

  NOTE: The validation process (for software) is the process to confirm that the requirements baseline functions and performances are correctly and completely implemented in the final product.

2. RELATED DOCUMENTS

2.1. Applicable Documents

The following table specifies the applicable documents that shall be complied with during project development.

Table 1: Applicable documents

Ref.	Code	Title	Issue
[AD 1]	AO/1-9765/19/NL/KML	Invitation to Tender AO/1-9765/19/NL/KML, “Toolset for Post Flight Analysis of ESA Missions”	27/03/19
[AD 2]	ESA-TECMPA-SOW-013145	“Toolset for Post Flight Analysis of ESA Missions” SoW	12/03/19
[AD 3]	PFAT-DMS-COM-PRS01-E	“Post Flight Analysis Toolset for ESA Missions” Proposal	29/05/19
[AD 4]	PFAT-DMS-TEC-TNO1.1-11	PFAT Use Case Analysis and Functional Requirements Identification	15/11/21
[AD 5]	PFAT-DMS-TEC-TNO1.2-12	PFAT Data Format and Methodology	15/11/21
[AD 6]	PFAT-DMS-TEC-TNO2-12	PFAT Architecture Design	15/11/21
[AD 7]	PFAT-DMS-TEC-TNO4-10	PFAT Validation and Testing Document	15/11/21
[AD 8]	PFAT-DMS-TEC-TNO5-10	PFAT Software Manual & Tutorials	16/12/21
[AD 9]	PFAT-DMS-TEC-FIR-10	PFAT Final Report	16/12/21
[AD 10]	PFAT-DMS-TEC-TNO6-10	PFAT Future Developments Roadmap	16/12/21

2.2. Reference Documents

The following table specifies the reference documents that shall be taken into account during project development.

Table 2: Reference documents

Ref.	Code	Title	Issue
[RD 1]	ISBN 0-201-57168-4	“The Unified Modelling Language User Guide”, G. Booch, J. Rumbaugh, I. Jacobson	2
[RD 2]	OPENSF-DMS-TEC-SUM01	openSF System User Manual	4.0
[RD 3]	OPENSF-DMS-OSFI-DM	OSFI Developer's Manual	1.19
[RD 4]	PE-ID-ESA-GS-464	ESA Generic E2E Simulator Interface Control Document	1.2.5

Ref.	Code	Title	Issue
[RD 5]		Karlgaard, Christopher D., et al. "Mars science laboratory entry atmospheric data system trajectory and atmosphere reconstruction." Journal of Spacecraft and Rockets 51.4 (2014): 1029-1047.	
[RD 6]		Karlgaard, Christopher D., et al. "Mars InSight entry, descent, and landing trajectory and atmosphere reconstruction." Journal of Spacecraft and Rockets 58.3 (2021): 865-878.	

2.3. Standards

The following table specifies the standards that shall be complied with during project development.

Table 3: Standards

Ref.	Code	Title	Issue
[SD 1]	ECSS-E-00A	ECSS Space Engineering – Policy and Principles	
[SD 2]	ECSS-M-00-02A	ECSS Space Project Management. Tailoring of Space Standards	
[SD 3]	ECSS-E-10	ECSS Space Engineering: Engineering – Part 1: Requirements and Process	
[SD 4]	ECSS-E-40B, draft	ESA software requirement standard ECSS-E40 tailored to small software projects	
[SD 5]	ECSS-E-ST-40C	Space Engineering Software	

3. PROJECT HISTORY

This chapter presents the main organizational issues of the project, mainly, the schedule, the work breakdown structure and the description of the main work packages.

3.1. Project Objectives

PFAT proposal was prepared answering the ESA ITT AO/1-9765/19/NL/KML, "PFAT: Post Flight Analysis Toolset for ESA Missions".

The joint submission of this proposal from **DEIMOS Space S.L.U. (Spain)**, as prime contractor, and the member of its consortium, **Deutsches Zentrum für Luft und Raumfahrt – DLR (Germany)**, **RUAG Space (Switzerland)** and **Empresarios Agrupados Internacional S.A. (Spain)** came from the common desire to provide ESA with the most competitive solution to the stated problem and SW product demand, as well as from the perspective of creating an asset that shall enhance our involvement in the future post-flight analysis activities.

According to Article II, Purpose, Convention of establishment of a European Space Agency, SP-1271(E), 2003 "*ESA's purpose shall be to provide for, and to promote, for exclusively peaceful purposes, cooperation among European States in space research and technology and their space applications, with a view to their being used for scientific purposes and for operational space applications systems.*"

Given the above ESA mandate, most ESA's missions are conceived to learn and increase our knowledge of space and to develop related technologies. This means that each ESA mission is designed to achieve a set of mission goals, in most of the cases defined to push forward the boundaries of the European technology capabilities. Learning from real space missions and experiments is in the essence of the ESA mandate and this objective is very relevant in particular to the domains of the ESA Flight Vehicles Engineering and Aerothermodynamics section.

In this respect, the present activity represents a valuable step ahead to pursue such objective. The specific objectives of the activity are described in [AD 2]:

- ☐ Development of the Post-Flight Analysis Tool (PFAT) software for use in post-flight analysis of ESA missions and interoperable with other engineering software tools commonly used by ESA
- ☐ Implement post-flight algorithms and analysis tools in PFAT to support the following engineering domains: propulsion, aerodynamics, thermodynamics, structures, materials and trajectories.
- ☐ Make use of standard exchange formats with PFAT
- ☐ Automatic generation of post-flight analysis reports with PFAT

Therefore, the main goal of this activity has been to develop an open-source SW tool to allow ESA performing the post-flight analysis of space missions with special emphasis on the following domains: (1) propulsion, (2) aerodynamics, thermodynamics, (3) trajectories and (4) structures and materials. To address this challenging goal, the resulting SW framework provides a generalised interface with the most common external sources in the different technical domains both at design and experimental/flight levels together with a set of capabilities to manage, process, analyse and report the required information by the user.

3.2. Schedule

Figure 1 shows the schedule followed for the PFAT project activities.

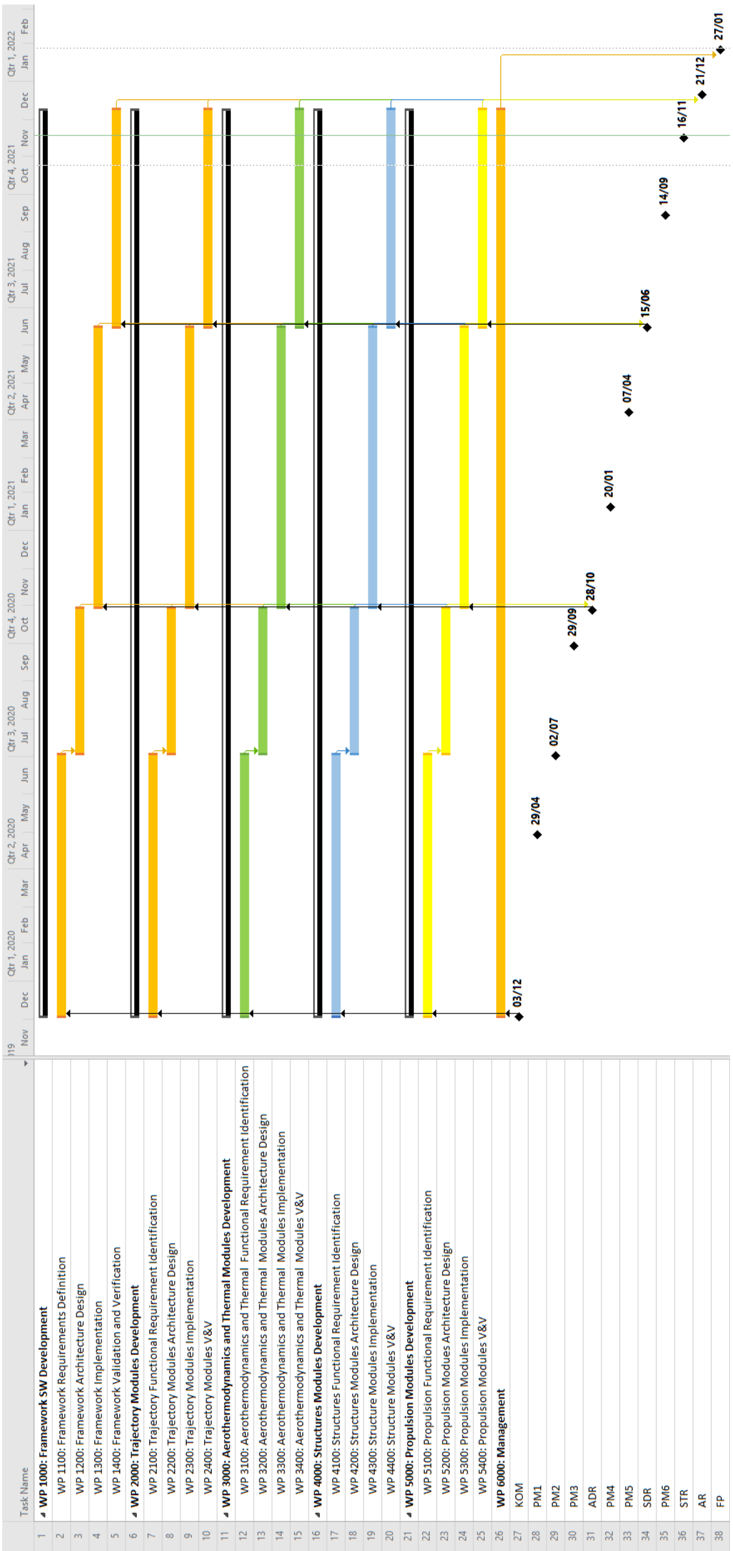


Figure 1: PFAT Project Schedule

3.3. Project Team

The following figures show the project industrial consortium and the project team, with the relative responsibilities.

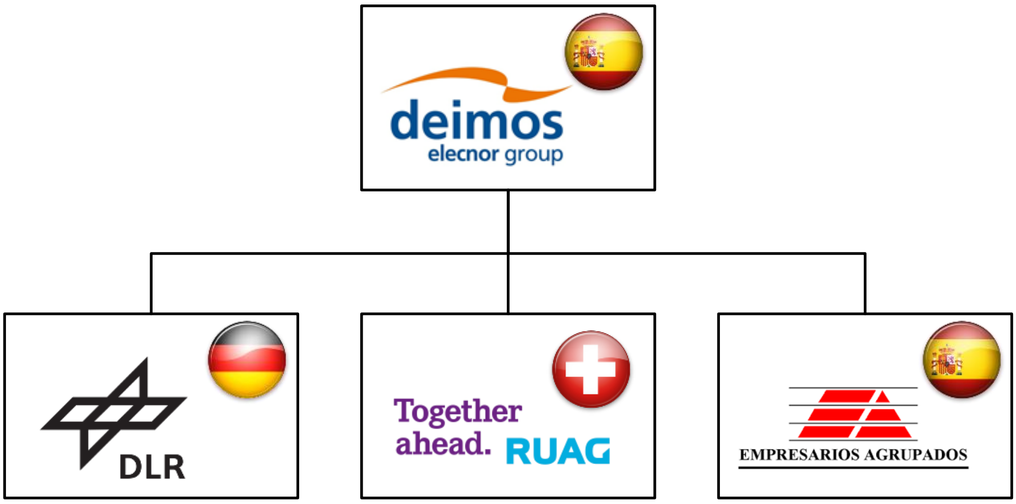


Figure 2: PFAT Industrial Consortium

3.4. Work Breakdown Structure

The Work Breakdown Structure was conceived to achieve all the objectives of the study in a timely and effective manner. The followed Work Breakdown Structure is shown in Figure 3.

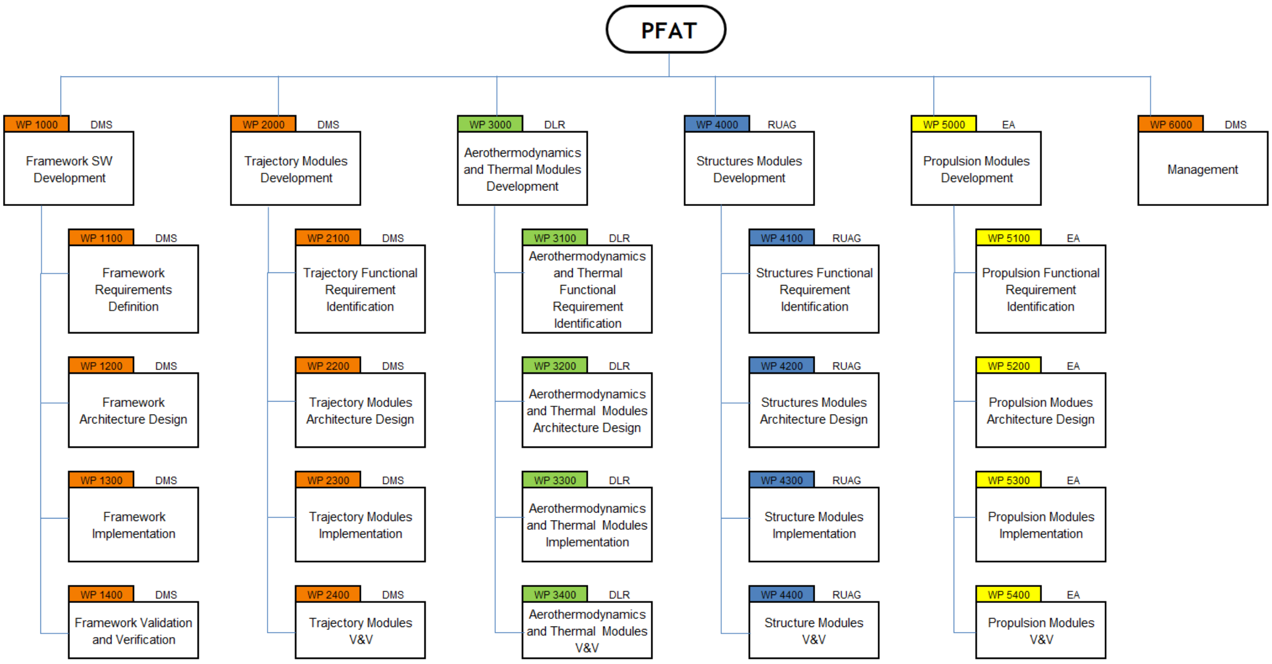


Figure 3: PFAT Work Breakdown Structure

3.5. Study Logic

The project study logic plan, which is illustrated in Figure 4, has been posed following the premises of the SOW [AD 2], showing the work logic and the tasks of the project with the logical connections among them, the project reviews and the deliverables.

The activity breakdown follows the standard SW development process as per ECSS E-ST-40-C [SD 5].

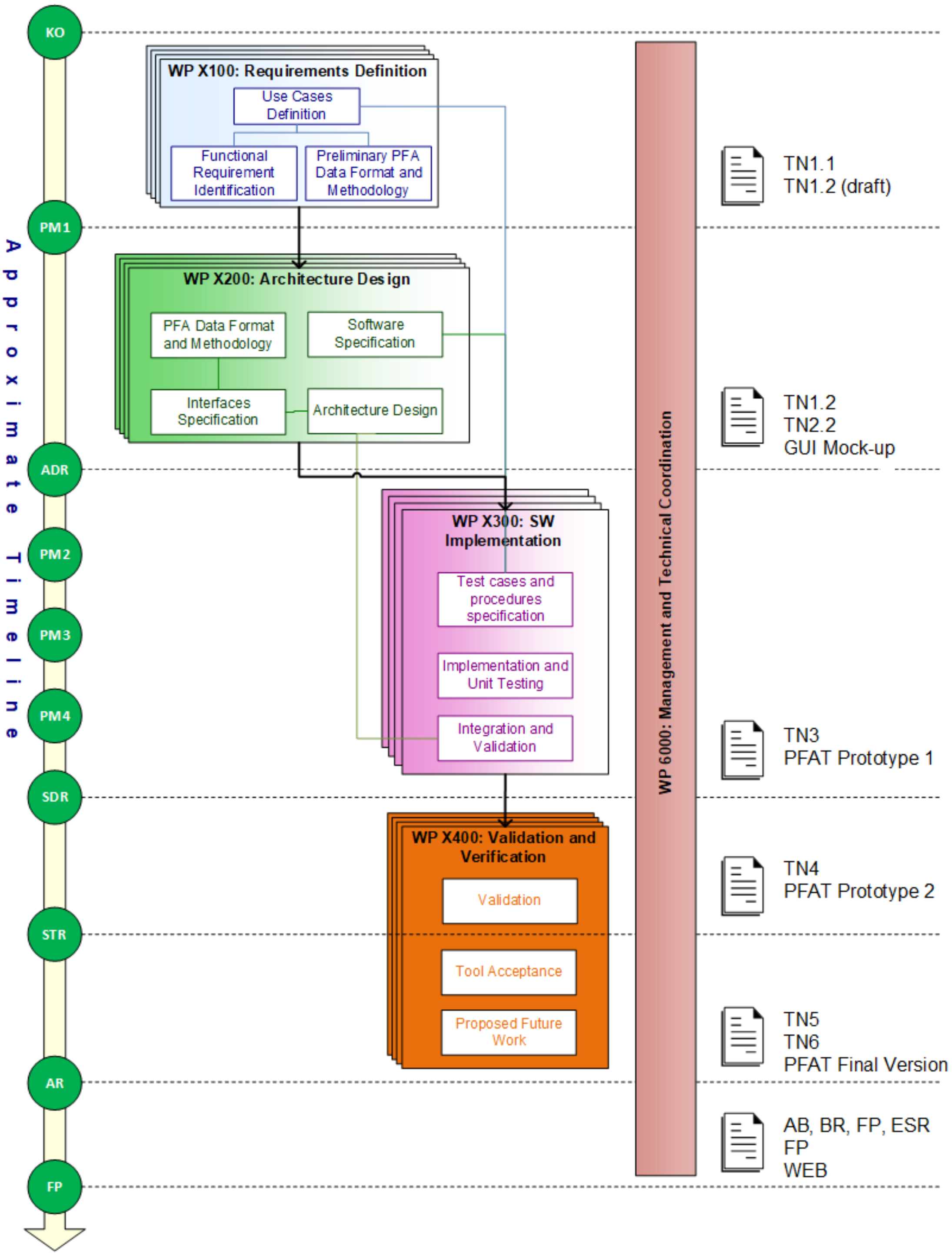


Figure 4: PFAT Study Logic

The activities of the project can be seen in two different orthogonal views: grouped according to the standard phases of the SW development process or grouped according to their technical domains. In the WBS (see Sec. 3.4) the latter has been used, however in the work logic the first approach is preferred, since it better describes the temporal succession of tasks.

According to the WP numbering, the thousands digit is associated to the technical domain (WP 1000 Framework, WP 2000 Trajectories, WP3000 Aerothermal dynamics and Thermal, WP 4000 Structures and WP 5000 Propulsion), while the hundreds digit is associated to the SW development process, as follows:

- ☐ WP X100 – Requirement Definition: in these work packages use cases for PFAT have been defined and specified and from them functional requirements and preliminary PFA data formats and methodologies derived.
- ☐ WP X200 – Architecture Design: these work packages have been meant for deriving software specifications from the functional requirements specified in WP X100, specifying PFAT interfaces, consolidating PFA data formats and methodologies, and designing the tool architecture.
- ☐ WP X300 – Software Implementation: during the SW implementation, detailed designed has been performed, including analysis and calculation algorithms specification. Test cases and procedures have been specified and the implementation performed, with the support of unit and integration testing.
- ☐ WP X400 – Validation and Verification: within this WP verification (testing that the tools calculations are numerically correct) and validation (testing that the tool implements the functionalities it has been designed for, performing PFA against the test cases scenarios derived from the use cases) have been undertaken. The activity ended with the recompilation of recommendations for future works on PFAT.

Additionally, WP6000 run in parallel to the whole project lifetime to carry out project management activities and thus to ensure a proper and adequate level of technical and programmatic progress of the tasks. This also covered the prime contractor's monitoring and control of activities performed by the subcontractor.

At the end of each WP a dedicated meeting has been held with the Agency to review the work done and to plan the following activity's steps.

4. POST-FLIGTH ANALYSIS TOOLKIT

4.1. SW Development

From SoW [AD 2], the main objective of this activity is declared *to develop an open-source software tool that uses a set of inter-related software design and analysis tools to allow ESA in the post flight analysis of space missions focusing on the following key engineering disciplines: propulsion, aerodynamics, thermodynamics, trajectories, thermal, structures and materials, and structures*.
The tasks specified in the SoW follow the software development waterfall approach, where the corresponding engineering disciplines are arranged sequentially in different phases to facilitate its implementation and control. This is schematically presented in Figure 5, where the titles of the boxes represent the process names as defined in ECSS E-ST-40-C, whereas the contents represent the activities as defined in the SoW.

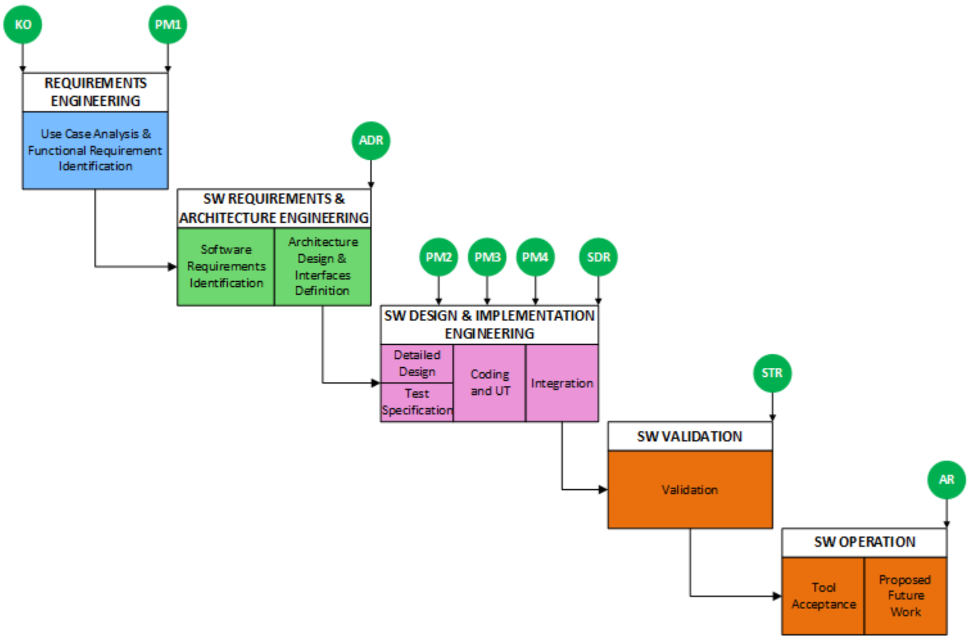


Figure 5: PFAT Software Development Life Cycle Process Diagram

4.2. Tool Capabilities

The PFAT system has being designed to be a multi-operating system desktop software application with four different functional components:

- The **computational core**, which is composed of six Python modules, related with the four engineering disciplines (aerothermodynamics, propulsion, structural and trajectories), data processing and an auxiliary module with common functionalities.
- The **executable modules**, which are a series of executable scripts that can be interfaced directly with the Graphical User Interface (GUI) and that expose the main functionalities of the six above-mentioned Python modules.
- The **Graphical User Interface**, which provides access to the executable modules and allows the user to build processing chains and manage the configuration and execution of the PFAT analyses. PFAT relies on the ESA openSF integration framework, which has been extended and upgraded whenever necessary to meet the PFAT needs.
- The **Common Data Structure** (CDS), which can be understood as the glue between the different modules, since it is a container able to store the input and output data of the modules.

From this high-level architectural decomposition, it can be grasped that the tool has been designed for a dual use, either by means of a Command Line Interface (CLI) as a Python package that provides a set of stand-alone functions, or by means of a GUI.

The resulting high-level system decomposition is the one depicted in Figure 6, where a module (or a processing chain) is orchestrated by an integration framework that will also provide access to post-processing capabilities to perform automatic report generation accessing the outputs produced by the PFAT modules.

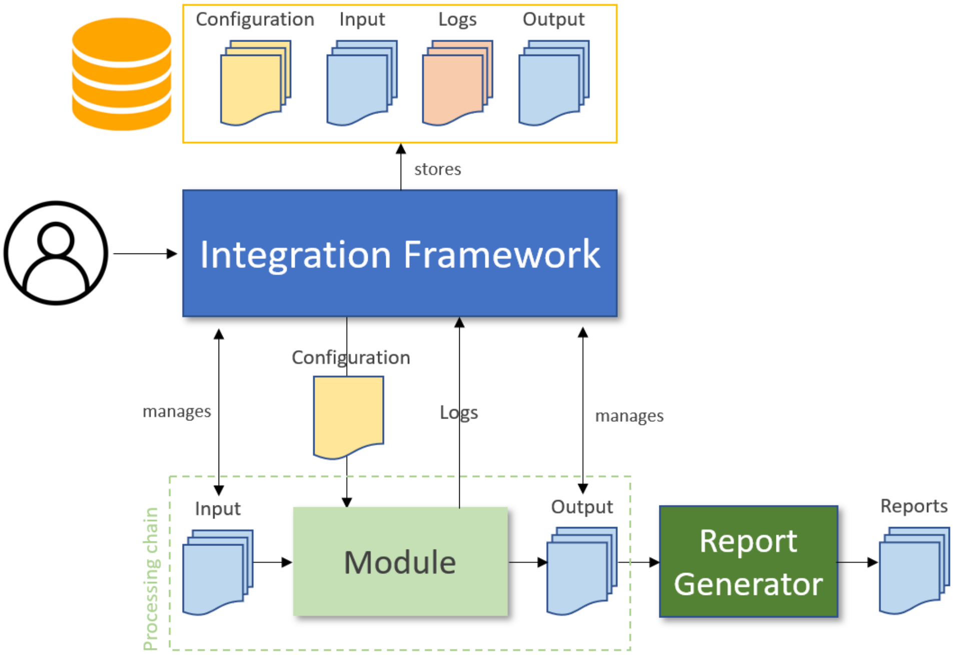


Figure 6: PFAT System Decomposition

4.2.1. Computational Core

The PFAT computational core is developed in Python and consists of six different modules:

- The aerothermodynamics, propulsion, structural and trajectories modules contain the functionalities specific of each engineering discipline.
- The data processing module contains the data processing functionalities, which can be used in any of the different engineering disciplines.
- The common module contains the functionalities that are shared among different modules, mainly for post-processing, such as comparison or plot functionalities.

The usage of PFAT as a Python package allows the user to exploit all the capabilities of the tool, including the access to any function in the library and extension or development of new functionalities. It shall be highlighted that, thanks to the extremely light interface with the GUI, the developers can contribute to the PFAT source code using the custom Python syntax for displaying errors, warnings and information messages, which means that any PFAT function can be used in other contexts and any external function can be used in PFAT.

The interface with the GUI is performed by means of a Python script which source code shall be located under the scope of the openSF simulation framework context manager. The context manager is a Python artifact that allows to allocate resources within a certain scope. In this case, the openSF simulation framework context manager provides the developer with the input, configuration, output files required to be interfaced with the GUI and it formats the error, warnings and information messages into the GUI own format without any impact on the functions used within the context manager scope. It shall be noted that PFAT contains more than 30 Python scripts or executable modules callable by the GUI, so the interface has been carefully designed to minimise code repetition, to minimise the impact on the developed functions and to ease the process of creating new GUI invokable scripts.

4.2.2. Common Data Structure

PFAT contains a series of executable modules, most of which receive the input data in binary format, concretely, as the binary flush of a CDS. The main purpose of the CDS is to provide a single interface to manage any data set used in the PFAT processing chains or executable modules. One of the benefits of this approach is the agnostic data management between processing chains or executable modules of different engineering disciplines, which reduces the time that the user needs to familiarize with PFAT algorithms and maximises the reusability of the functions. In addition, the fact that data sets are managed with the same approach in the different processing chains (associated to the different engineering domains) allows to maximize the re-usability of the algorithms, which is especially relevant for the more generic algorithms such as the data processing ones.

The CDS is able to store internally data series with one independent variable and multiple dependent variables or data series, which are univocally identified with a label and a point identifier. In addition, the CDS stores internally the coordinates of the points declared in the data series identifier and the connectivity information between points (if any), storing the point identifiers associated to an element, the type of the element (the ones currently implemented are edges, triangles, quadrilaterals, tetrahedrons, pyramids, prisms and hexahedrons) and any other additional information that the user could add, such as area or volume of the element. Finally, the CDS is able to store any kind of metadata univocally identified by a string.

The CDS implements a programmatic interface (API) that can be used to fill the main four attributes in the structure (data series, point coordinates, elements information and metadata). Even though the API is considered rich enough to fulfil the expected user needs in the scope of PFAT, the four main attributes of the structure can be directly accessed in case the API does not provide enough flexibility. The attributes of the CDS in charge of storing the data series, the point coordinates and the elements information have been implemented as pandas dataframe objects (https://pandas.pydata.org/), since they are expected to store large datasets. The metadata attribute, expected to be lighter than the others, has been implemented as a Python dictionary.

In terms of communication between modules, the CDS provides methods to write into binary files and to load from binary files. In addition, the four main attributes can be written to human readable formats, concretely to comma separated values (CSV) for the data series, point coordinates and elements information attributes and to JavaScript Object Notation (JSON) for the metadata attributes. Therefore, the binary format is the selected format for communication between modules, whereas the human readable formats are used only for report generation. It shall be noted that any module that generates a binary CDS also generates its corresponding human readable files, although the number of files generated depends on the content of the CDS, since only the non-empty attributes are written.

4.2.3. Executable Modules

Due to the nature of PFAT, most of its executable modules can be assimilated to post-flight methods or algorithms, in the sense that each of the identified methods or algorithms are implemented as an executable module. Therefore, these PFAT components have been already identified and specified in a dedicated document. For further details please refer to the "PFAT Data Format and Methodology" ([AD 5]).

The computational modules are stand-alone applications that can be run integrated in the GUI either stand-alone or within a computational chain, the latter representing a complete post-flight analysis. Both the executable modules and the GUI implement an error handling system, being the latter in charge of intercepting the error and log messages, and to manage them, either presenting them to the user or taking the necessary actions. The communication between the executable modules and the integration framework will be eased by using the openSF Integration Library (OSFI), a collection of functionalities that will help the integration of the modules within openSF.

The PFAT executable modules can be categorised as follows:

- Data Processing, to perform pre- and post- process of data stored in the CDS.
- Common, I/O, comparison and plot generation functionalities that are common to all the engineering domains.
- Aero-thermodynamics, domain specific functionalities.
- Propulsion, domain specific functionalities.

- Structural, domain specific functionalities.
- Trajectories, domain specific functionalities.

The following sections provide a high-level description of the data flows identified within the system and the overall capabilities provided in each of the four engineering domains (i.e. trajectories, propulsion, structures and aero-thermodynamics).

These data flows, with algorithms and interfaces allocated to executable modules, represent the processing chains computing post-flight analyses. The diagrams in the next sections uses the following colour convention:

- Fuchsia: file in non-standard PFAT format, generated from/to an external tool or flight/experimental data
- Green: PFAT processing module
- Brown: external tool (not included in the PFAT processing chain)
- Lilac: file in standard PFAT format
- Yellow: tables and figures files

4.2.3.1. *Aero-thermodynamics*

The aero-thermodynamics processing chain foresees five different analyses. Note that the data labelled with IDS refers to data already loaded in the common data structure.

Lift and drag coefficients calculation

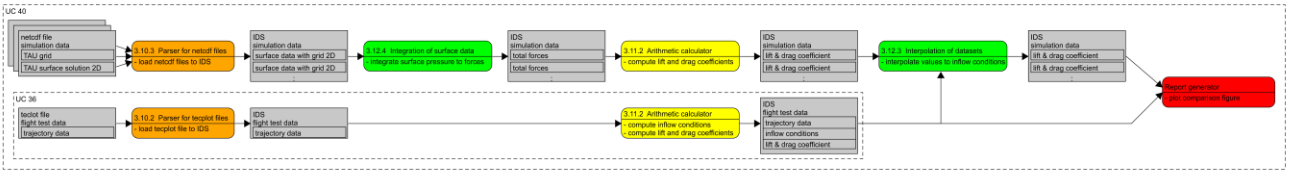


Figure 7: Lift and drag coefficients processing chain

Heat flux probability density function calculation

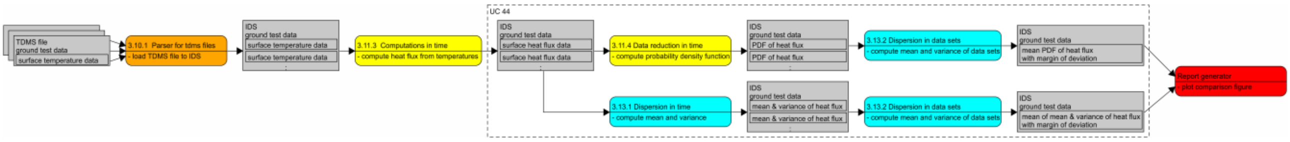


Figure 8: Heat flux probability density function processing chain

Pressure power spectral density calculation

Figure 9: Pressure power spectral density processing chain

Boundary integral calculation

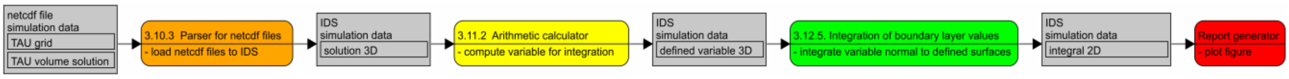


Figure 10: Boundary integral processing chain

Intermittency calculation:

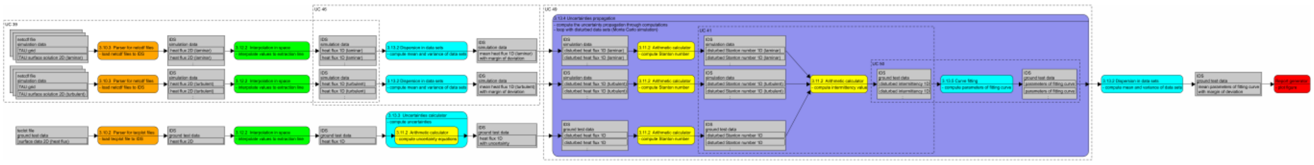


Figure 11: Intermittency calculation processing chain

4.2.3.2. Propulsion

The propulsion processing chain foresees the following steps:

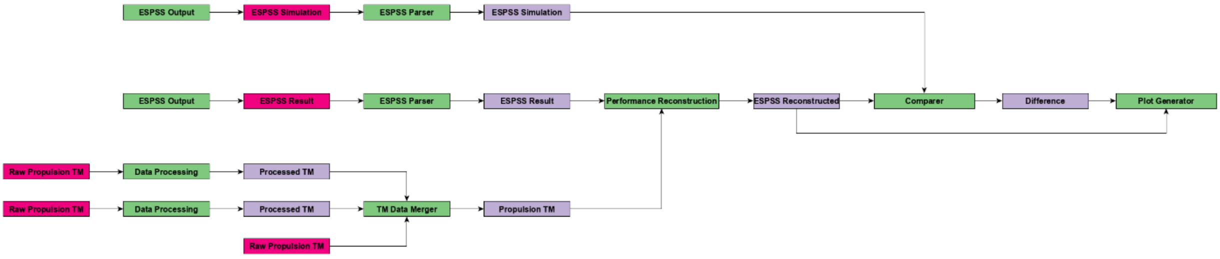


Figure 12: Propulsion analysis processing chain

4.2.3.3. Structural

The three processing chains are shown hereafter:

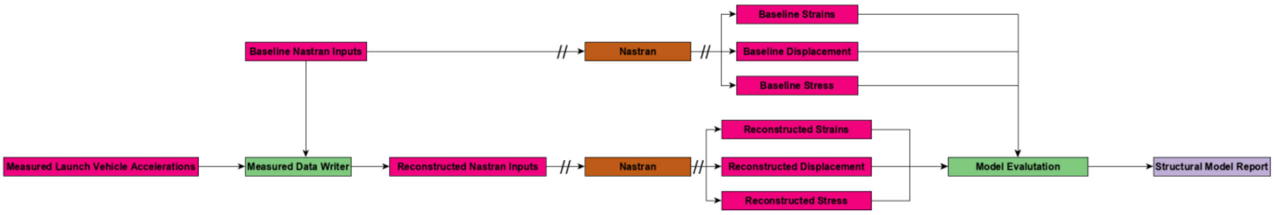


Figure 13: Structural analysis processing chain

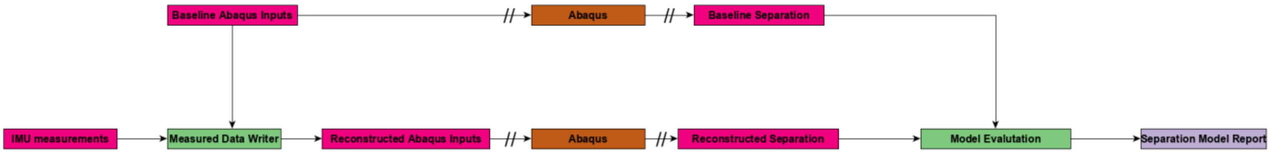


Figure 14: Separation analysis processing chain

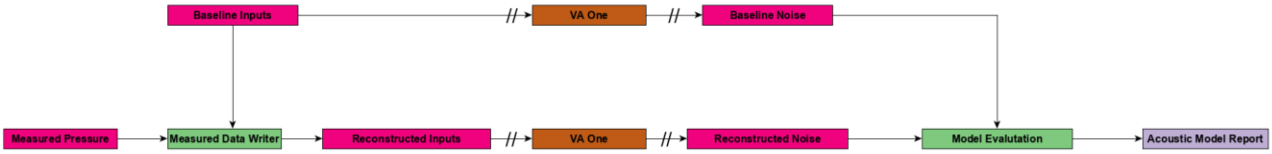


Figure 15: Acoustic analysis processing chain

4.2.3.4. Trajectories

The trajectory reconstruction processing chain foresees the following steps:

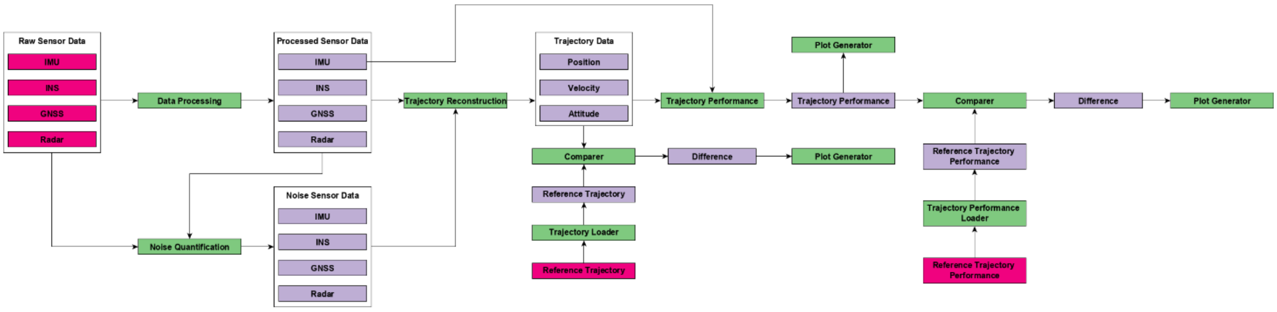


Figure 16: Trajectory reconstruction processing chain

Detailed data for the algorithms and file specifications can be found in [AD 5].

4.2.4. Graphical User Interface

PFAT uses as GUI an ESA open-source simulation framework: openSF.

openSF is a software framework aimed at supporting a standardised end-to-end simulation capability allowing the assessment of the science and engineering goals with respect to the mission requirements. Scientific models and product exploitation tools can be plugged in the system platform with ease using a well-defined integration process.

openSF has been conceived to support concept and feasibility studies for the ESA Earth Observation Programs (EOP) activities. Nevertheless, openSF has been designed and developed in a generic way, allowing its use as a simulation framework for any processing chain in domains different from EO E2E performance simulators.

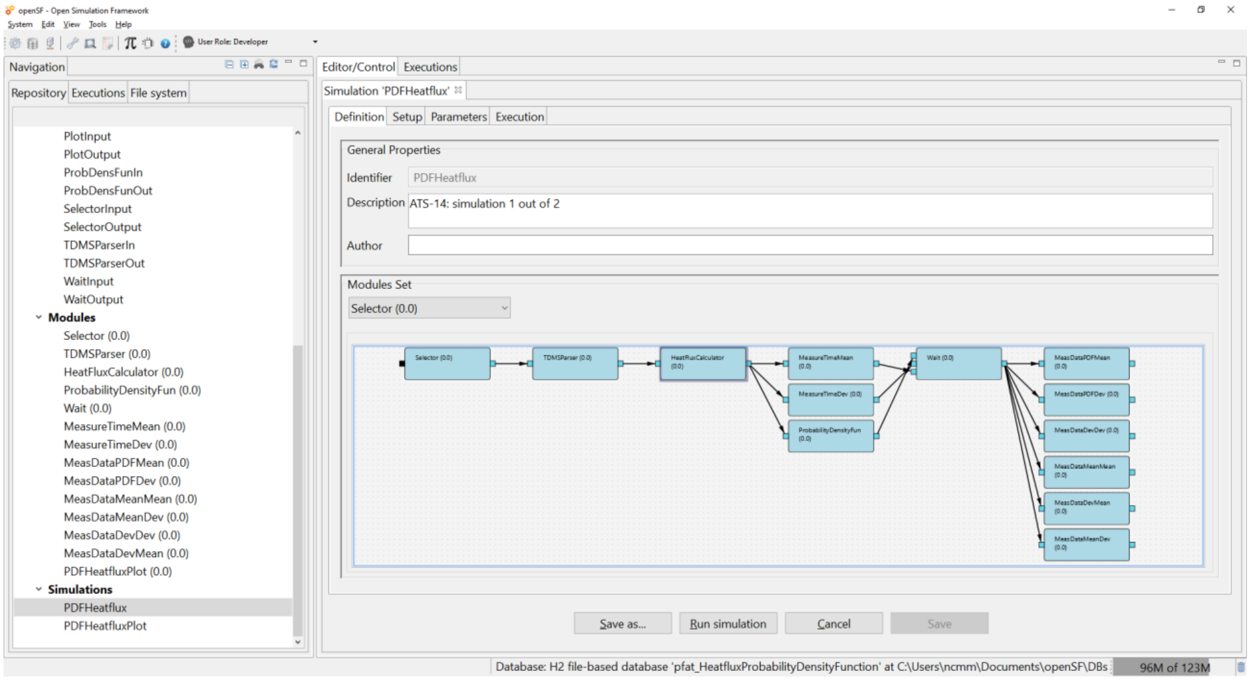
Some of the advantages provided by using openSF can be summarised as follows:

- It covers some of the requirements identified for PFAT, such as the capability to define processing chains with clear identification of computational modules and interfaces among them..
- It is distributed under the ESA Software Community License - Type 3, an open-source licensing scheme compatible with the PFAT requirement to be open source.
- It provides a reliable framework, already under development for more than 10 years and with updates and bug fixes publicly made available to the community approximately every six months.
- It is developed, maintained and distributed to the same target operating systems for which PFAT is being developed.

After having performed such assessment, the re-use of openSF and its upgrade with specific PFAT functionalities has been selected as the PFAT framework.

After a comparative analysis of the PFAT functional requirements and openSF provided functionalities, a comprehensive list of openSF features that have been re-used in PFAT were made. Moreover, some openSF features, even if not explicitly required by the current activity, have been made available to the PFAT user.

The most relevant extension has been the graphical representation of the processing chain. In order to grasp the flux of the process the PFAT framework has been upgraded with a visual viewer that shows graphically the relationships between the modules composing an analysis. In this way it will be intuitive for the user to understand how the process is defined. Moreover, a direct access to the configuration of each module will be provided by means of the visual representation of the modules. As an added feature, the graphical representation will also report the status of each module execution, either with a colour code indicating its health status (not run, in execution, failed or successfully run) and with a progress status (while being executed).



Identifier: PDFHeatflux
Description: ATS-14: simulation 1 out of 2
Author:
Modules Set: Selector (0.0)

Database: H2 file-based database 'pfat_HeatfluxProbabilityDensityFunction' at C:\Users\ncmm\Documents\openSF\DBs

Figure 27: PFAT graphical representation of the processing chain

4.3. Validation and Verification Approach

In order to guarantee a robust toolkit, complying with both software specifications and user requirements, PFAT undertook a comprehensive validation campaign.

During the implementation phase, all the functionality developed were tested with unit testing that covered 94% of the code by means of 944 unit tests.

Such a large testing guaranteed an elevated reliability of the implemented functions that minimised risks at integration level and paved the way for the validation and verification campaign.

Un order to buttress these test cases, additional tests have been specifically designed in order to reproduce in each one of the four engineering domains a real (or quasi-real) post-flight analysis campaign. 17 acceptance test cases have been thus specified in order to guarantee that the toolkit is able to perform End-to-End post-flight analysis. These acceptance tests reproduce post-flight analysis against real flight data (whenever available) or against synthetic data mimicking real flight data.

The whole validation campaign has been conducted in the three target platforms, namely: Windows 10, Linux Ubuntu 20.04 LTS, and macOS 11.0 Big Sur.

4.4. Post-Flight Analysis Campaign

4.4.1. Trajectory Domain

The validation of the PFAT modules under the trajectory's domain focused on obtaining a reconstruction of the Insight descent trajectory in towards Mars surface through the *trajectory reconstruction* module using [RD 6] as reference. The scenario analysed in [RD 6] is quite complex in terms of validation of features, including for instance data fusion of IMU, radar and landing site measurements and smoothed reconstruction (i.e., average solution of a forward and a backward reconstruction), so the validation campaign of the *trajectory reconstruction* module has been divided in two different stages: a first one where the basic capabilities of the module, such as state and uncertainty propagation, are tested against a reference tool, and a second one aiming at reproducing the results of [RD 6].

The obtained results are in great accordance with the reference ones, and the few differences detected are widely justified by differences in the approaches or data used.

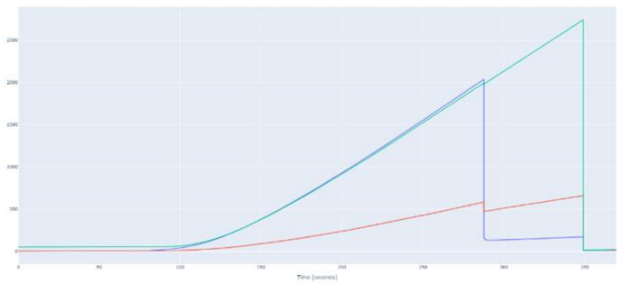
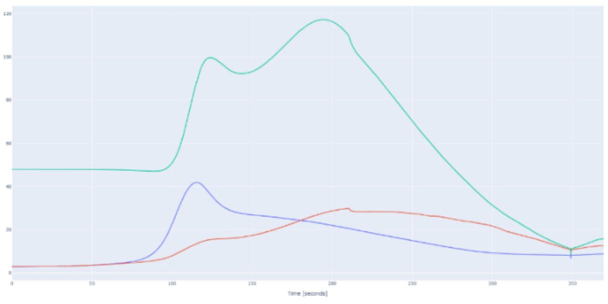
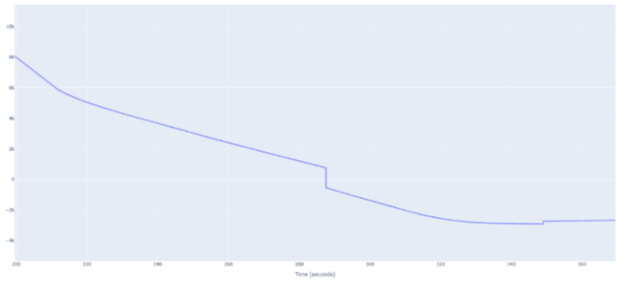
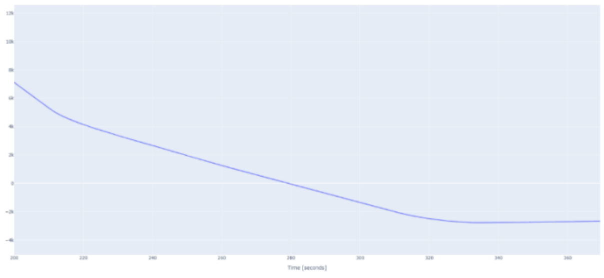
	Forward Reconstruction	**Smoothed Reconstruction**
Position Uncertainty Evolution [m]		
Altitude Evolution [m]		

Figure 17: Forward (left) and Smoothed (Right) Position Uncertainty Evolution (Blue X, Red Y and Green Z Coordinates) and Altitude Evolution

4.4.2. Aerothermodynamic Domain

The aerothermodynamic domain in PFAT includes a series of flexible modules that can be combined in different processing chains to perform a wide variety of analysis. Therefore, the validation campaign of this domain includes five different tests that show the capabilities of the PFAT aerothermodynamics modules, as described in Sec. 4.2.3.1.

Experimental or realistic synthetic datasets have been used for the campaign, and the resukts have been validated against reference data obtained with other specialised SW.

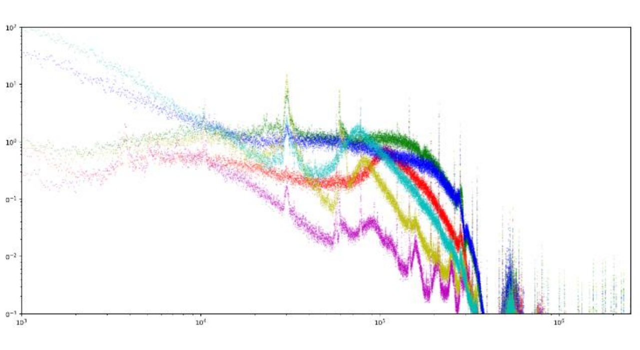
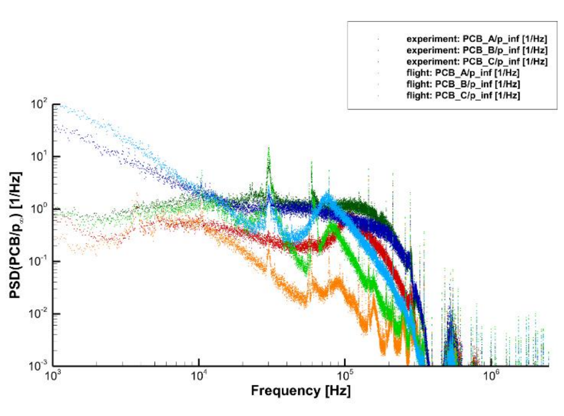
PFAT Results	Reference Tool (LabVIEW) Results
	

Figure 18: PFAT vs reference tool results of PSD pressure processing chain

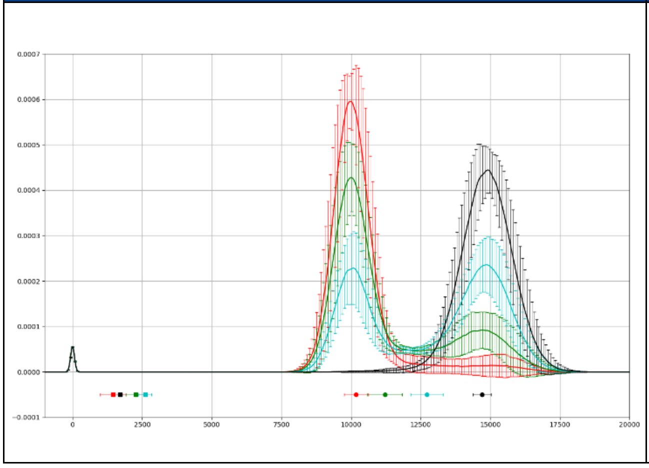
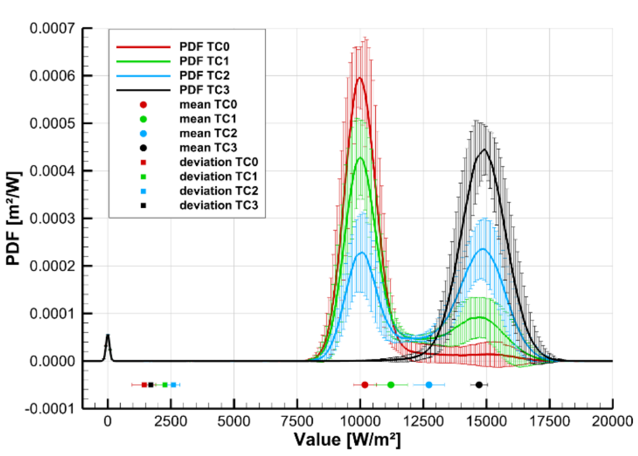
PFAT Results	Reference Tool (LabVIEW) Results
	

Figure 19: PFAT vs reference tool results of heat flux probability density function processing chain

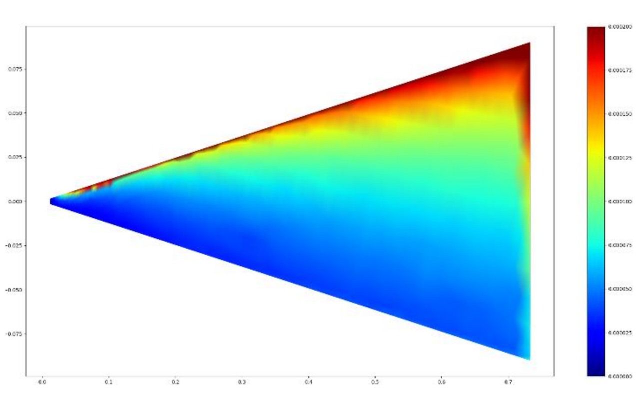
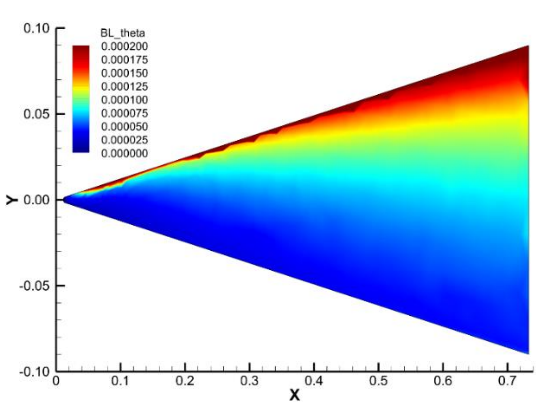
PFAT Results	Reference Tool Results
	

Figure 20: PFAT vs reference tool results of boundary integral processing chain

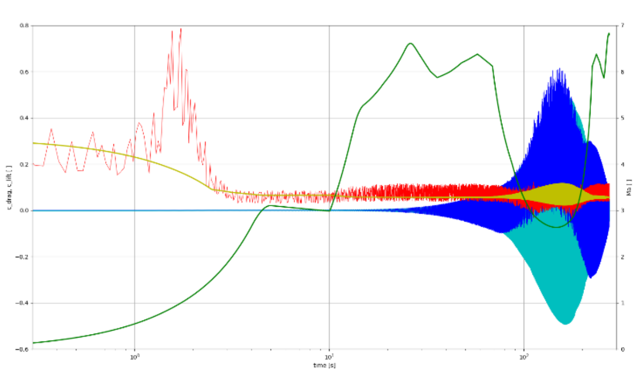
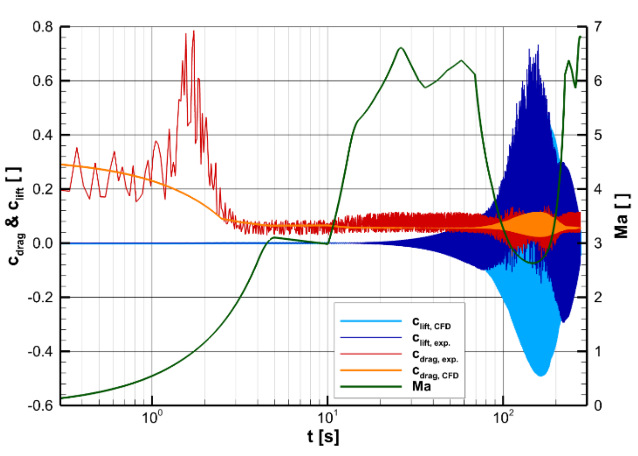
PFAT Results	Reference Tool (Tecplot and Excel) Results
	

Figure 21: PFAT vs reference tool results of lift and drag coefficients processing chain

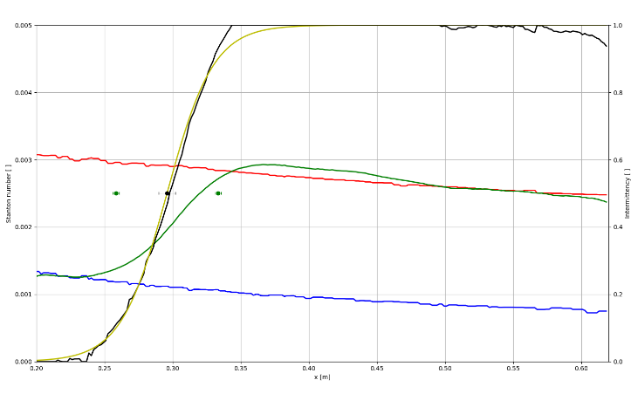
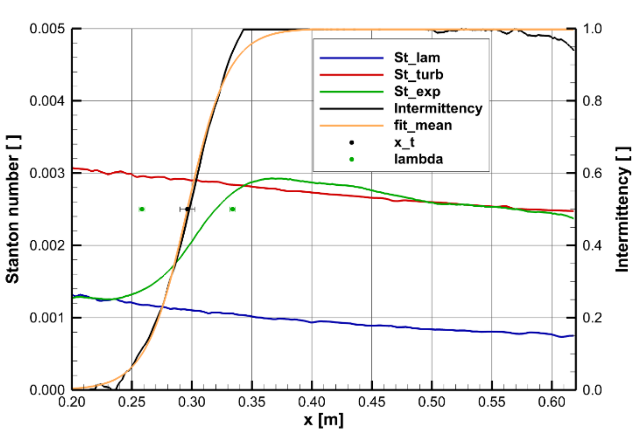
PFAT Results	Reference Tool (Tecplot and Excel) Results
	

Figure 22: PFAT vs reference tool results of intermittency processing chain

4.4.3. Propulsion Domain

The objective of the validation campaign performed in the propulsion domain has been to check that standalone propulsion ESPSS model can be loaded and executed in PFAT. At the same time, it has been checked that a proper comparison between the ESPSS simulation results and in-flight data (or reconstructed performance) takes places within a certain degree of reliability.

In order to achieve those objectives, multiple PFAT modules of the propulsion domain were executed sequentially: telemetry data loading, deck models execution, deck models results mapped to PFAT, fitting expressions computation, thrust and specific impulse reconstruction and comparison against deck model.

This structure has been used in the four different tests that have been included in the validation campaign of the propulsion domain modules.

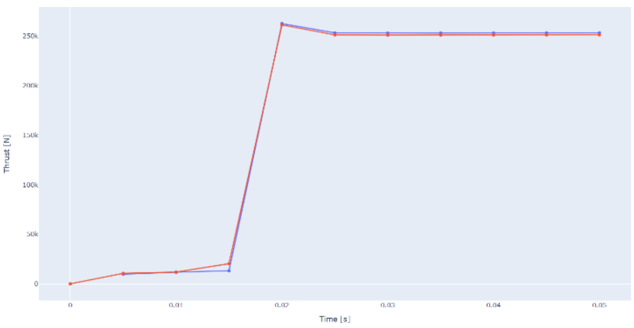
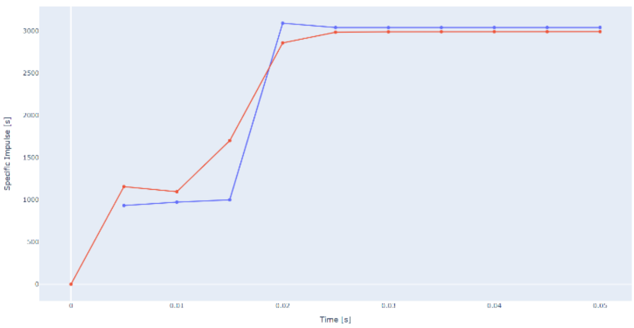
Thrust	Specific Impulse
	

Figure 23: Combustor model reconstructed thrust and specific impulse (blue) vs simulated (red) performances

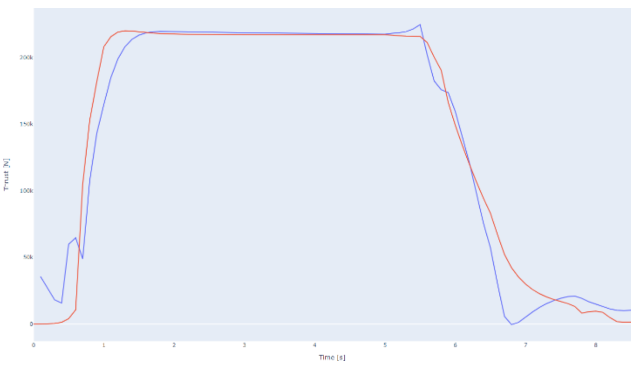
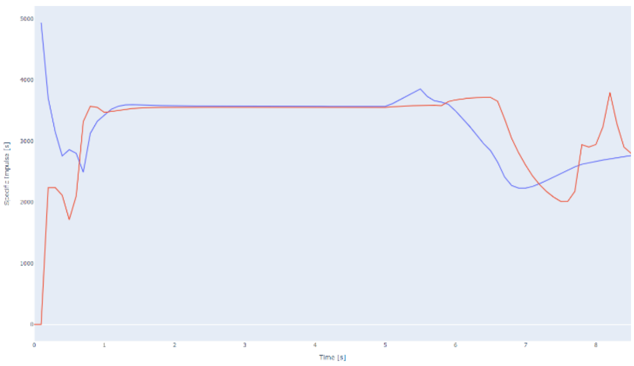
Thrust	Specific Impulse
	

Figure 24: Expander model reconstructed thrust and specific impulse (blue) vs simulated (red) performances

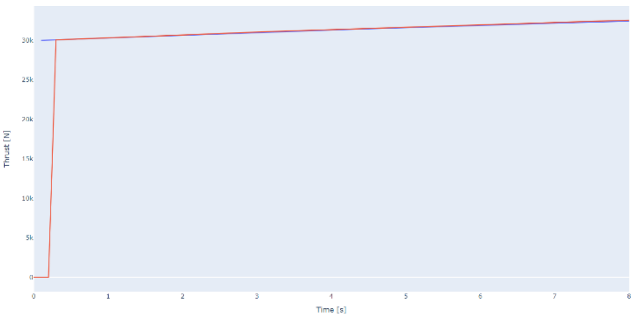
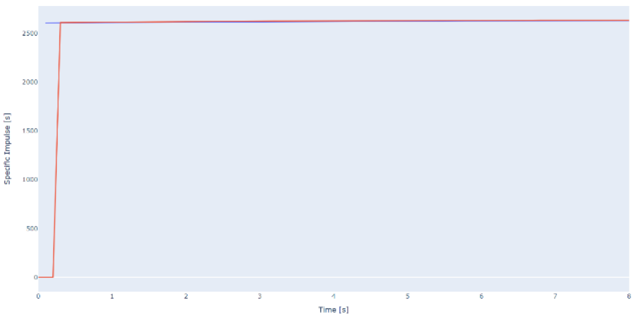
Thrust	Specific Impulse
	

Figure 25: Pressure fed model reconstructed thrust and specific impulse (blue) vs simulated (red) performances with default fitting expressions

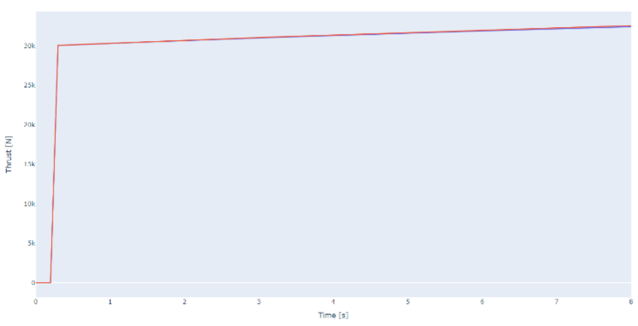
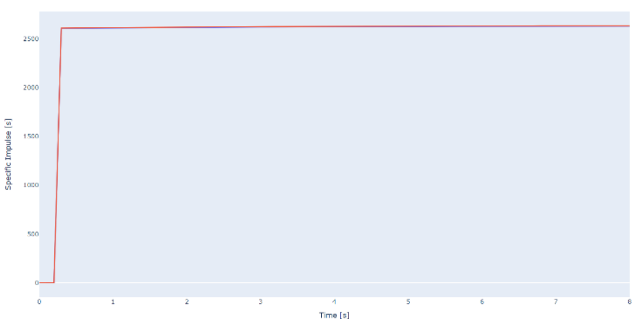
Thrust	Specific Impulse
	

Figure 26: Pressure fed model reconstructed thrust and specific impulse (blue) vs simulated (red) performances with user defined fitting expressions

4.4.4. Structural Domain

The validation campaign of the structural domain related modules can be separated in two different stages, one related with the measured data writer modules, which are in charge of reading data and inject them in the baseline input file of an external tool, resulting in the reconstructed input file; and a second one related with the model evaluation modules, which are in charge of comparing the results produced by an external tool executed with the baseline and reconstructed input files.

The tests on the three structural domains (structural, separation and acoustic) gave excellent results when tested against the same post-flight analysis performed with external tools.

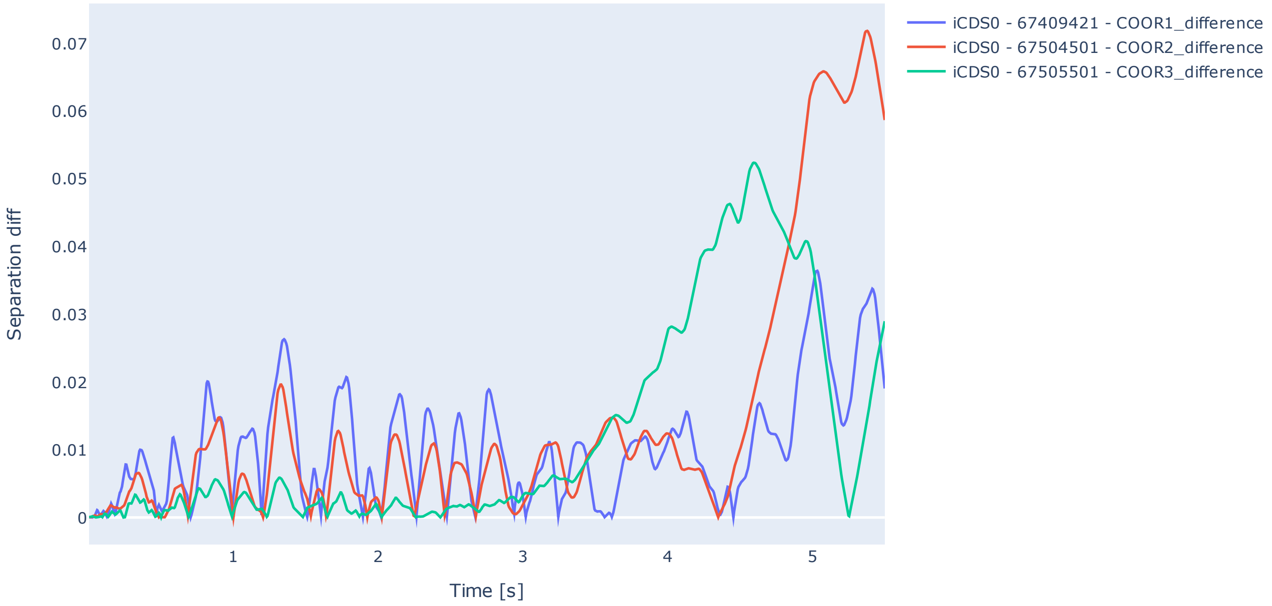


Figure 27: Separation model – time history of the nodal displacements difference between baseline and comparison model, for the nodes showing the greatest discrepancy

5. CONCLUSIONS AND FUTURE PESPECTIVES

5.1. Achievements of PFAT Project

During the PFAT activity, the following specific objectives have been fully achieved:

- ☐ A flexible and powerful Post-Flight Analysis Tool (PFAT) software has been developed, having in mind its use in post-flight analysis of ESA missions, i.e. the interoperability with other engineering software tools commonly used by ESA (e.g. TAU, NASTRAN, ASTOS);
- ☐ Robust and generic post-flight algorithms and analysis tools have been implemented, supporting the following engineering domains: propulsion, aerothermodynamics, structures and trajectories;
- ☐ A standard exchange format has been designed and implemented (the Common Data Structure), to use the toolkit interoperability;
- ☐ A powerful Graphical User Interface is provided along with a flexible Command Line Interface, providing a dual approach to PFAT use;
- ☐ Automatic generation capabilities of post-flight analysis reports have been implemented, with user friendly mechanisms to set them up.
- ☐ The toolkit capabilities have been validated against reference real post-flight data, when possible. In case of unavailability of reference flight data, realistic synthetic data have been used.

5.2. Future Work

PFAT has been an ambitious project, whose objectives have been fully reached. Notwithstanding its validation has been conducted against real datasets (or synthetic representative ones), only its use in a real context could quantify the objectives achieved and help in identifying the gaps to be filled.

However, the validation campaign already identified room for improvement, whose details have been collected and reported in the PFAT “Future Developments Roadmap” ([AD 10]). The future work should be oriented in two main directions: reinforcing the capabilities already present (e.g. adding new filters, mathematical functionalities, sophisticated methodologies) and extending them towards other engineering domains and/or data formats (e.g. providing access to further sensors/flight data).

End of Document