HORIZON-EUROHPC-JU-2021-COE-01

EXCELLERAT P2

The European Centre of Excellence for Engineering Applications

Project Number: 101092621

D2.1 Use-Case Execution Roadmap

The EXCELLERAT P2 project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101092621. The JU receives support from the European Union's Horizon Europe research and innovation programme and Germany, Italy, Slovenia, Spain, Sweden and France.

Work Package:	WP2	Use-Case Execution	
Author(s):	Thomas Hösgen, Ansgar Niemöller, Matthias Meinke		RWTH
	Jonathan Vincent		KTH
	Gabriel Staffelbach		CERFACS
	Micheal Wagner		DLR
	Herbert Owen		BSC
	Matic Brank		UL
	Francesco Salvadore		CINECA
Approved by	Executive Centre Management		30.06.2023
Reviewer	Jonathan Vincent		KTH
Reviewer	Simone Bnà		CINECA
Dissemination Level	Public		

Date	Author	Comments	Version	Status
2023-06-05	Thomas Hösgen, Ansgar Niemöller, Matthias Meinke	Initial draft	V0.1	Draft
2023-06-19	Thomas Hösgen	Draft	V0.2	Draft
2023-06-21	Thomas Hösgen	Draft	V0.3	Draft

List of abbreviations

ACARE	Advisory Council for Aeronautical Research in Europe
ACF	Autocorrelation functions
AMR	Adaptive mesh refinement
ASMR	Automatic static mesh refinement
CAA	Computational aeroacoustic
CCC	Center coordination committee
CFD	Computational fluid dynamics
DLR	German Aerospace Center
DNS	Direct numerical simulation
DRL	Deep reinforcement learning
DMD	Dynamic mode decomposition
DOF	Degree of freedom
FA	Further application
FSDM	FlowSimulator DataManager
FWH	Ffowcs Williams-Hawkings
HPC	High-performance computing
LES	Large-eddy simulation
MDA	Multi-disciplinary analysis
MOS	Method of snapshots
POD	Proper orthogonal decomposition
RANS	Reynolds-averaged Navier-Stokes
SAneg	Spalart-Allmaras one-equation turbulence model in its negative form
SEM	Spectral element method
SVD	Singular value decomposition
SPL	Sound pressure level
UAV	Unmanned aerial vehicle
UC	Use case
UQ	Uncertainty quantification
VR	Virtual reality
VTK	Visualization toolkit

Executive Summary

This document summarises the roadmap of the workflow development connected to all use-cases defined in EXCELLERAT P2. In total, six use cases and one further application were defined in the proposal, which cover the three application profiles relevant in HPC, i.e., hero runs, smaller scale, strong scaling production runs and large scale, strong scaling applications.

This document will serve as a reference to monitor the progress of the work connected to the workflow development. This allows the introduction of appropriate measures if unforeseen delays occur. The defined success criteria will be used to assess the achievements towards the end of the project.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

The document outlines the roadmap for the workflow development for all use cases (UCs) in the EXCELLERAT P2 project. For each UC, firstly a brief introduction of the UC is given with a summary of the main objectives. Subsequently, the workflow and dependencies are described. Additionally, the timeline and milestones are defined, which are illustrated with a Gantt chart. Finally, success criteria are specified.

	Topic	Partner	Code
UC-1	External aircraft aerodynamics	DLR	CODA
UC-2	Hydrogen combustion for propulsion	CERFACS	AVBP
UC-3	Mitigation of aeroacoustic noise	RWTH	m-AIA
UC-4	Fully integrated aircraft simulations with emission models	BSC	Alya
UC-5	High-fidelity simulations of rotating parts	KTH	Neko
UC-6	Active control for drag reduction of transonic airfoils	CINECA	FLEW
FA-1	Engineering design and digital twin of the first wall of a tokamak fusion reactor	UL	OpenFoam, ElmerFEM, Raysect, Mitsuba 2

2 UC-1 External Aircraft Aerodynamics

Partner: DLR, Code: CODA

2.1 Use Case Introduction

Computational fluid dynamics (CFD) simulations are an increasingly important part of aircraft design. They allow in-depth insight into the aerodynamic behaviour of components and help reducing cost and time in development.

This use case simulates steady airflow at subsonic speed and computes typical characteristics like air velocity and direction, pressure and turbulence (Fig.1). For the use case, CODA solves the Reynolds-averaged Navier-Stokes equations (RANS) with a Spalart-Allmaras one-equation turbulence model in its negative form (SAneg). It uses a second-order finite-volume spatial discretisation with an implicit Euler time integration.

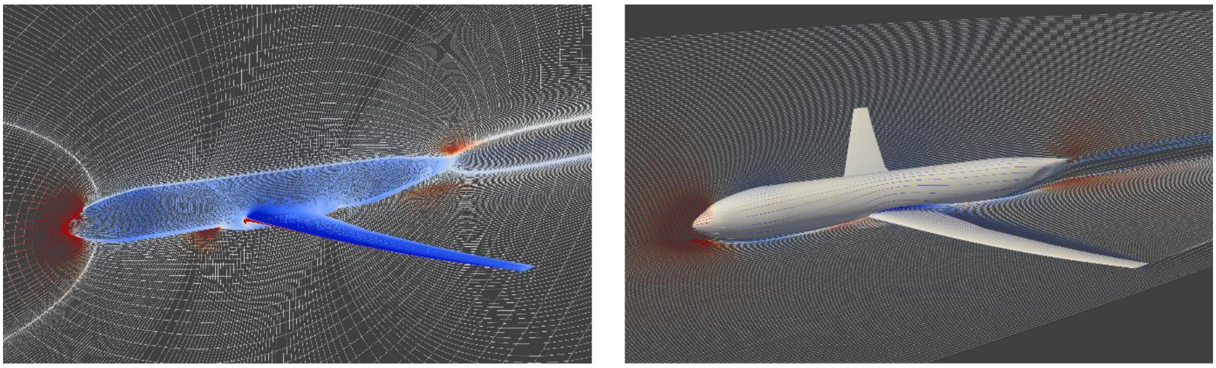

Figure 1: Example for a use case simulation: aircraft configuration with mesh (left) and airflow around wing and fuselage (right); both with air pressure as colour gradient

The major objective of the use case is to analyse the CODA reference application with respect to node and system level performance. EXCELLERAT P2 will provide the opportunity for early access to pre- or near-exascale architectures and will allow verification of extrapolations from previous strong scaling tests, which indicated pre-exascale readiness for CODA. Actual near-exascale testing will show if further performance optimisation is necessary to reach this goal.

Planned external aircraft aerodynamics simulations will require large scale runs in the range of pre-exascale systems for research and production as well as large sets of medium-scale runs for aerodynamics data production (ensemble runs).

The inputs of the use case are aircraft geometries in the form of unstructured meshes with varying sizes ranging from tens of millions to several billions of elements, typically with six degrees of freedom per element (ranging up to several hundreds). The aircraft geometries will include public models such as the NASA CRM Common Research Model as well as internal models from DLR and Airbus.

2.2 Workflow

The CODA CFD software is part of the MDA framework FlowSimulator, which provides plug-ins for all steps of a full aircraft simulation as well as a seamless integration into multi-disciplinary simulation.

In particular, CODA uses FlowSimulator's core component, the FSDM for I/O, where various I/O libraries are supported. FSDM is an open-source software hosted by DLR. FSDM provides the FSMesh class, which is the preferred container for the exchange of data among FlowSimulator plugins. FSDM is MPI parallelised and an FSMesh instance is a distributed representation of the data, usually containing information on the geometry, the (computational) mesh, flow fields/solutions, as well as coupling strategies.

Furthermore, CODA uses Spliss for solving linear equation systems as part of implicit CFD methods. The sparse linear system solver Spliss aims to provide a linear solver library that, on the one hand, is tailored to requirements of CFD applications but, on the other hand, independent of the particular CFD solver. Focusing on the specific task of solving linear systems allows for integrating more advanced, but also more complex, hardware-specific optimisations, while at the same time hiding this complexity from the CFD solver. For instance, Spliss enables the execution of the computationally intensive linear solver on GPUs without the necessity of any code adaption.

2.3 Timeline

The technical developments will be performed in the following steps:

- Definition and requirements for the use case and workflow (M6).
- Analysis of CODA baseline efficiency (M12).
- Analysis of CODA improved efficiency (M36).
- CODA efficiency demonstrator (M48).
- Continuous improvement of CODA efficiency (M48).

2.4 Dependencies

CODA is the CFD software being developed as part of a collaboration between the French Aerospace Lab ONERA, the German Aerospace Center (DLR), Airbus, and their European research partners. CODA is jointly owned by ONERA, DLR and Airbus.

2.5 Gantt Chart

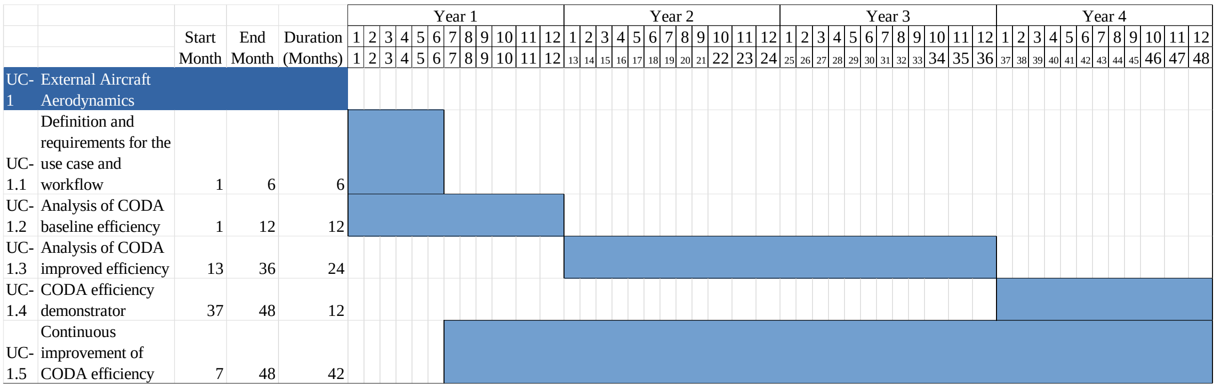

Figure 2: UC-1 Gantt Chart

2.6 Success criteria

For this use case the following criteria will be used to verify the success:

- Improved efficiency on available HPC systems.
- Improved scalability towards exa-scale systems.

3 UC-2 Hydrogen combustion

Partner: CERFACS, Code: AVBP

3.1 Use Case Introduction

In this use case, the simulation of hydrogen combustion in propulsion conditions is performed, which is representative of problems encountered in aerospace engine design.

The motivation for the use case is part of the ambitious climate neutrality or net zero greenhouse emissions of the EU for 2030. One of the pillars to achieve this is the transition to renewable energy. Hydrogen is considered a promising clean energy carrier that can help to decarbonize various sectors, including industry, transport and heating.

The simulation of hydrogen combustion requires high-fidelity high-performance models to solve the Navier Stokes compressible equations with accurate chemistry modelling and the full range of scales introduced by the turbulence in a complex geometrical design.

3.2 Objectives

When performing computational fluid dynamics, meshing is done using a trial-and-error approach: a mesh is generated, and then compared to best practices. If adequate, draft CFD runs are performed. If found lacking, the mesh goes back to the generation stage. The real simulation starts only after this preparatory cycle. When dealing with exascale simulation this heuristic approach is bound to be prohibitive. Instead, we propose to automate the trial-and-error cycle in a workflow generating goal-informed meshes within pre-requisite resources.

The use case targets the simulation of a hydrogen combustion chamber using an ASMR workflow using automatic feedback from the CFD software. Such an endeavour requires the efficient use of exascale hardware as well as efficient mesh adaptation techniques to fully exploit the available resources.
In this use case, all the necessary components of the workflow such as parallel mesh adaptation, automatic quality verification, and efficient parallel solver execution for the combustion prediction on exascale hardware will be performed. The ASMR should be capable of starting from a low-resolution coarse case setup and converging towards a billion-element mesh ensuring throughout its run, the conservation of the input quality and cost criteria.

3.3 Execution profile

The accurate prediction of combustion on a complex geometry is challenging and requires the accurate discretisation of the features of interest. The localisation and scale of the discretisation are complicated to predict a priori. For these use cases, an iterative ASMR workflow allows the global quality control of the simulation and ensures the best simulation result within a user-defined budget and target case size.

Figure 3 explains schematically the dependencies of the execution profile.
An initial setup of the use case is generated by the user and used to perform a first guess of the simulation using the state-of-the-art solver optimised for exascale hardware (CPU or GPU). Then quality and convergence criteria verification occur, triggering the improvement of the mesh or the continuation of the simulation.

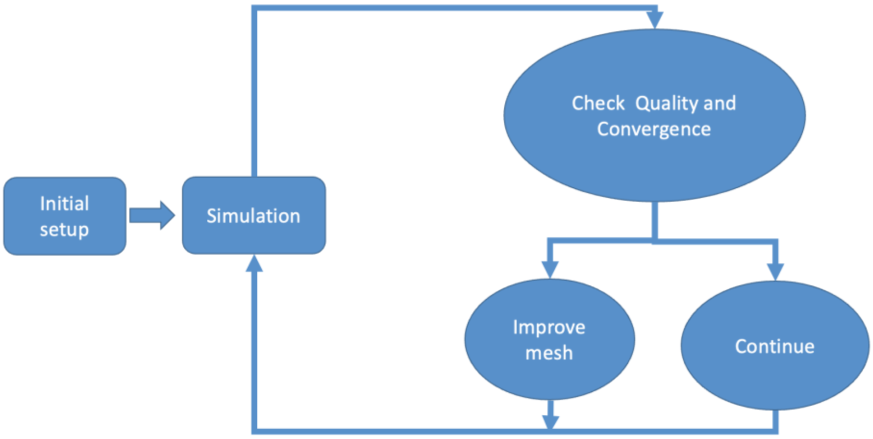

Figure 3. UC-2 Execution profile

3.4 Workflow

A typical workflow is depicted in Figure 4. The detailed components are:

- Mesh adaptation framework.
    - Open source static and parallel mesh adaptation frameworks HIP and Treeadapt. Serial are limited to < 200M elements. Treeadapt scalability demonstrated up to 4096 cores (AMD Epyc2) generating 1.2B elements in 10min.
- LES/CFD simulation.
    - State of the art AVBP solver scaling up to 100k cores and 1024 NVIDIA GPUs.
- Quality and convergence checking methods.
    - Python tools (Figure 5 shows the convergence of y+ on a test case).

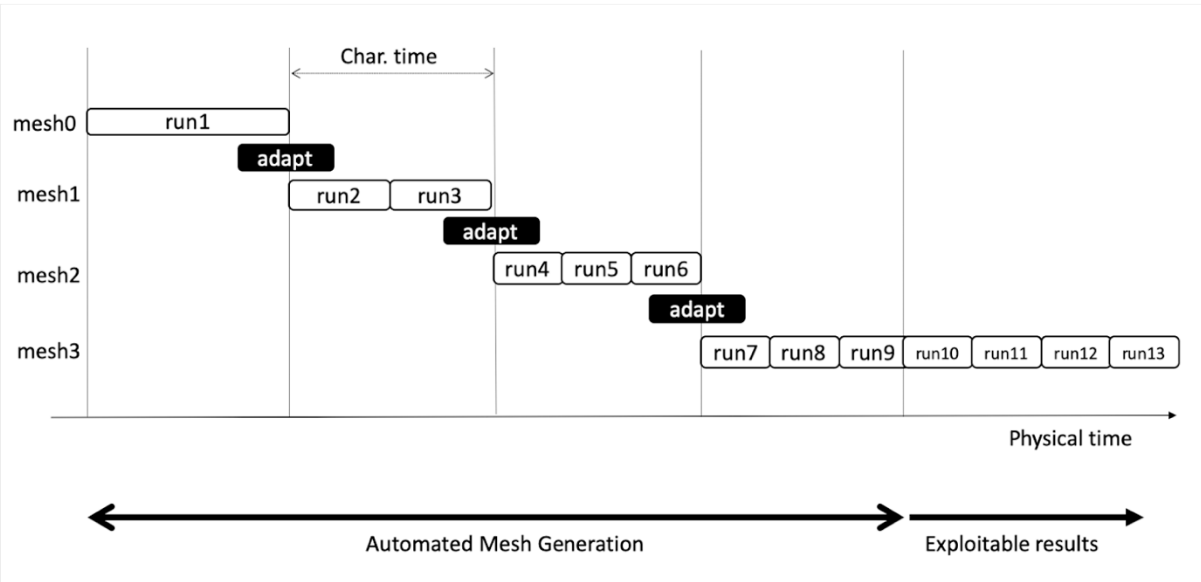

Figure 4: UC-2 Typical workflow

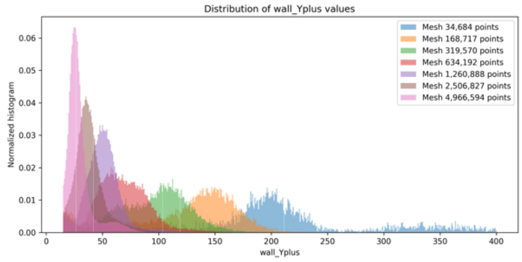

Figure 5: Convergence of y+ on a test case

The exascale execution profile requires a robust highly efficient solver and mesh adaptation frameworks capable of running many dependent simulations with increasing mesh size and complexity. A typical ASMR workflow should consist of 20 steps with 5 mesh improvement steps and 15 convergence steps. Initial runs should start at O(10) compute nodes and gradually increase to O(1000) nodes to reach exascale.

3.5 Timeline

The different components of the workflow can be developed/tested in parallel. The workflow itself handling the resource allocation and the dependencies will be added to the open-source lemming software developed at CERFACS.

- Definition and requirements for the use case and workflow (M6).
- Define a setup for a simplified problem for small-scale runs (M6).
- Establish necessary workflow building blocks (M12).
- (Semi-)Automation of the workflow (M18).
- Automation of the workflow on EuroHPC hardware (M24).
- Define a test setup for exascale demonstration (M24).
- Perform automatic static mesh refinement of the simulation (M36).

3.6 Dependencies

The execution of the workflow on NVIDIA GPUs will require the acceleration of the localisation algorithms defined in WP3 and/or their acceleration to other GPUs like AMD.

3.7 Gantt Chart

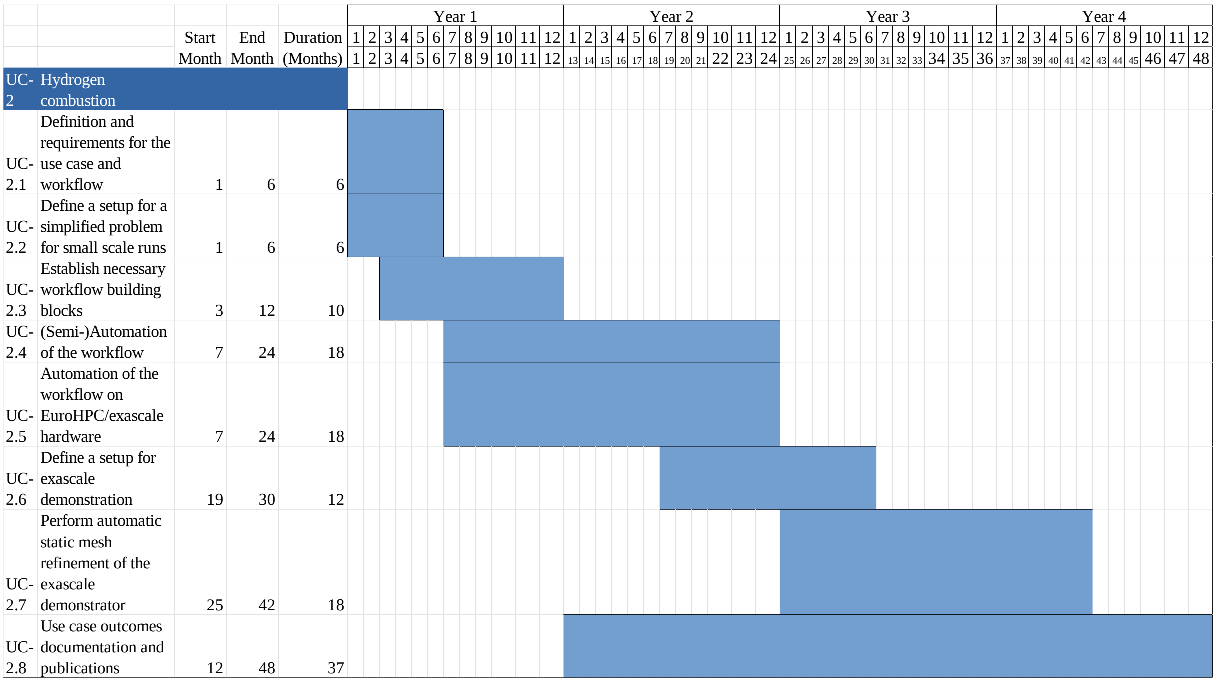

Figure 6: UC-2 Gantt Chart

3.8 Success criteria

For this use case, the following criteria will be used to verify the success:

- Development of a workflow, which can automatically perform mesh quality and convergence for the combustion application.
- Scalability: The workflow should demonstrate scalability and efficient utilisation of resources as the mesh size and complexity increase.
- Robustness: The workflow should be robust and capable of handling various scenarios and inputs without errors or failures. It should be able to handle different geometries, boundary conditions, and initial conditions.
- User-friendliness: The workflow should be user-friendly and easy to set up, configure, and execute.
- Portability: The workflow should be portable and compatible with different high-performance computing (HPC) architectures and environments. It should be able to run on different systems, including CPU and GPU-based architectures, without significant modifications.
- Extensibility: The workflow should be designed in a modular and extensible manner, allowing for future enhancements and integration of additional features or components. It should facilitate the incorporation of new algorithms, solvers, or tools to further improve the simulation capabilities.
- Documentation and Support: The workflow should be accompanied by comprehensive documentation, including user guides and tutorials.

4 UC-3 Mitigation of Aeroacoustic Noise

Partner: RWTH, Code: m-AIA

4.1 Use Case Introduction

In this use case, a shape optimisation of chevron nozzles is performed, which is representative for problems with expensive objective function evaluations in the important technical field of noise reduction. It includes a constraint of minimum jet thrust loss, which has to be weighted with the goal of noise reduction. The accurate prediction of the emitted sound is based on a turbulence scale resolving CFD method, directly coupled with a computational aeroacoustics solver, which is a typical example of a multiphysics application. The necessary large number of objective function evaluations can only be performed on exascale HPC systems. An AI-based optimiser, developed by the project partner FhG, will be used to efficiently identify the optimal solution. The motivation for the use case is given e.g., by ACARE, who established the Flightpath 2050, a new goal for more rigorous noise reduction by 65% relative to the capabilities of typical new aircraft in 2000. Despite the progressive introduction of high-bypass-ratio aircraft engines and chevron nozzles, which possess a sawtooth-like shape at the engine's trailing edge, jet noise still is a significant source of aircraft engine noise.

Unlike the optimisation of the aerodynamic performance or structural weight, noise reduction is still to a large extent an unsolved problem. One of the challenges connected to noise reduction is a reliable and accurate prediction of the sound pressure level in the far field, which is often generated by intricate flow phenomena. Therefore, turbulence scale resolving simulations have to be performed in many cases to obtain the correct sound pressure level. For example, the tip gap vortex in an axial fan can generate noise, which cannot be predicted with methods based on Reynolds averaged solutions. At present, shape optimisations are not possible for such cases due to the computational expensive evaluation of the objective function.

A typical chevron nozzle shape is shown in Figure 7 and the setup for the flow and the acoustic field prediction is depicted in Figure 8.

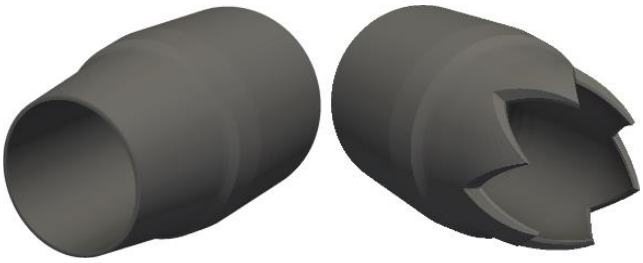

Figure 7: Baseline (left) and typical chevron nozzle (right)

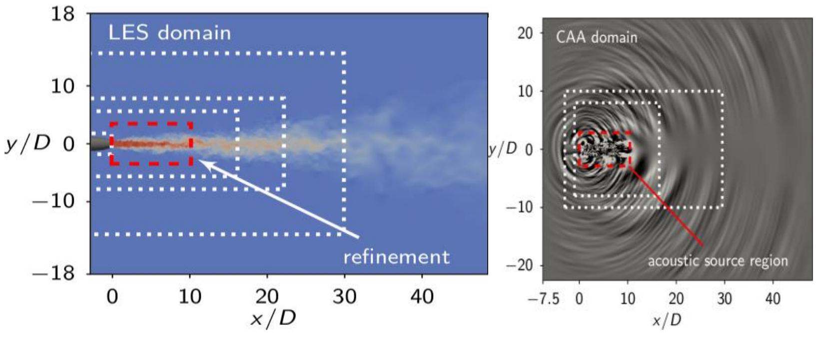

Figure 8: Computational setup with the LES or CFD, and the CAA domain for the jet noise prediction

4.2 Objectives

It is the objective of this use case to perform a shape optimisation for the shape of the chevrons at the nozzle trailing edge to minimize the jet noise. During aircraft take-off and cruise flight, however, the use of chevron nozzles can lead to a substantial and unwanted loss of thrust. Therefore, the optimisation has to be performed under the constraint that the thrust is minimally affected. Since the objective function evaluation is computationally extremely expensive, HPC clusters of exascale have to be used efficiently. Therefore, an advanced workflow has to be designed, which can fully exploit the available computing resources. The goal is to implement all necessary workflow components such as the automatic generation of chevron shapes, the execution of the simulation for the flow field prediction and the determination of the complex objective function. Additionally, highly resolved simulations should be performed, which enable the identification of the essential noise source locations and to determine the effect of the various chevron parameters on the generation of the acoustic waves. For the execution of the use case, the workflow couples a high-fidelity 3-D aeroacoustics solver with state-of-the-art AI optimisation algorithms provided by the project partners FhG and data analytic tools to systematically perform thrust-constrained noise minimisation of chevron nozzle shapes.

4.3 Execution profile

The accurate prediction of the overall sound pressure level in the far field requires several numerical methods which must be combined to obtain the final result for the objective function. The CFD solver for the prediction of the turbulent flow field is directly coupled to the CAA solver for the determination of sound pressure levels in the acoustic near field, which then delivers unsteady data for a FWH method, which can compute the overall sound pressure level in the far field. The coupling between the CFD and CAA solver is schematically explained in Figure 9.

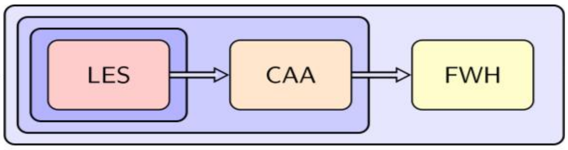

Figure 9: Coupling of the LES or CFD with the CAA solver and the FWH method

A single simulation is performed in various stages to minimize the required computational effort. First, the CFD solver is executed alone, until a fully developed flow field and a sufficiently converged time averaged flow field is obtained. Subsequently, the CFD and CAA solver is executed in a fully coupled manner until the unsteady acoustic pressure signal is obtained in the near sound field. Finally, the history of the unsteady pressure signal on a closed control surface is used to predict the noise in the acoustic far field. The overall sound pressure level is determined at reference locations for the evaluation of the objective function. In addition, the flow field is postprocessed to compute the nozzle thrust.

The simultaneous execution of the individual runs poses a significant challenge for the coordination of the computational resources (various hardware platforms, queueing systems). Since the runs are expected to last for different lengths of time an intelligent, optimised coordination of the simulations is necessary. Since the AI-based optimisation algorithm is directly fed with data from the simulation (in situ) the simultaneous provision of large CPU/GPU resources is challenging. To enable a subsequent analysis of the simulation results, it is necessary to efficiently store the data on long-term storage systems which is challenging due to the large amount of data.

4.4 Workflow

The workflow is schematically depicted in Figure 10:

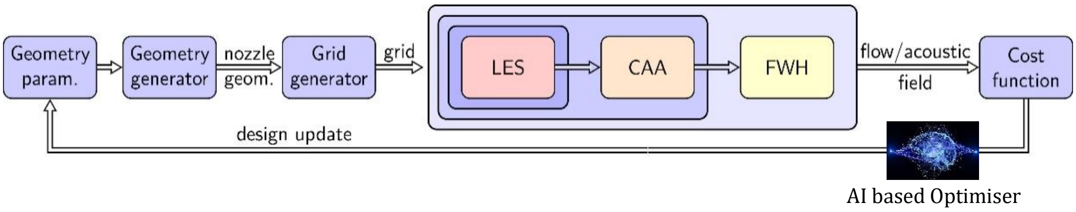

Figure 10: Workflow components for the constrained shape optimisation of chevron nozzles

In more detail, the workflow components for the jet noise prediction are:

- AI-based optimiser provides a chevron nozzle design, i.e., new chevron geometry parameters.
- Geometry generator generates STL surfaces for the mesh generation for the specific chevron nozzle shape.
- The grid generation generates refined meshes for the LES/CFD and CAA simulation.
- The LES/CFD simulation is performed to determine the time averaged mean flow field.
- The coupled CFD/CAA simulation is performed to predict acoustic pressure in the near field.
- FWH method to predict the acoustic far-field.
- Postprocessing of the FWH and CFD results for the cost or objective function evaluation.

Various aspects of the individual components are summarised in the following:

AI-based optimiser

- provides a shape update by respecting design space constraints.
- will be executed on a subset of the computational resources.
- communicates via MPI with the simulation workflow components.

Geometry generator

- Uses a parametrisation of the shape defined by analytical functions.
- Generate new geometry in form of STL data for the given set of parameters (Matlab).
  - o STL geometry files as input for grid-generator/simulation.
  - o Check validity (e.g., waterproof surface, no surface element overlaps).

Grid Generation

- Grid generator for hierarchical Cartesian grids integrated in m-AIA code.
- Generates both the CFD and CAA meshes.
- Serial execution: O(100,000,000) cells, high memory usage, O(10min).
- Parallel execution for larger grids (small scale run).

LES/CFD Simulation

- Interpolate fully developed flow field to new grid; requires intermediate grid for mapping; full scale job, O(1min) to obtain the initial condition.
- Ensure high parallel efficiency by pre-balancing the grid(s) using static weights.
- Use dynamic load balancing at beginning of a simulation to reduce imbalances.
- O(250,000,000) cells for small sized grid.
- Perform simulation until the flow field is fully developed, use an automatic convergence criterion.
- Compute mean flow field by time averaging.
- Small sized job: O(10,000) cores, O(24 hours).
- Mean quantities for CAA; sample data for postprocessing.

Coupled CFD/CAA simulation

- Run LES and CAA concurrently with source term exchange.
- O(500,000,000) degrees of freedom in CAA.
- LES: start from fully developed solution of the turbulent flow field.
- CAA: initial phase and propagation through the computational domain.
- Sampling phase: acoustic pressure at observer locations; surface data for FWH.
- Long enough sampling time for accurate SPL→ near far-field SPL.
- Single job size: O(10,000) cores, O(24 hours).
- Combine with in situ postprocessing/visualisation.

Ffowcs Williams-Hawkings (FWH)

- Time-resolved surface data → far-field extrapolation.
- Potentially large data volume to be read.
- Comp. Effort: O(#observer * #time steps * #surface elements).
- Smaller scale run possible: O(1000) cores, O(1h); very good scalability.

To summarise, multiple simulations with different phases/configurations; various output files, large data volume have to be performed. This requires a reliable and fault tolerant automation and high efficiency without the necessity of user interaction or supervision.

The exascale execution profile is defined by many small scale runs O(1000) for the shape optimisation. In addition, a few large scale runs O(10) will be performed for advanced data analytics to identify physical mechanisms. Pre-exascale hero runs O(1) are planned for higher Reynolds number, coaxial jet configurations and very high-fidelity results to demonstrate exascale readiness.

4.5 Timeline

The development of the AI-based optimisation workflow will be first performed with a simplified shape optimisation problem, which is currently being defined together with the partner FhG.

The technical developments will be performed in the following steps:

- Definition and requirements for the use case and workflow (M6).
- Define a setup for a simplified shape optimisation problem for small scale runs (M6).
- Establish necessary workflow building blocks (M12).
- (Semi-)Automation of aeroacoustic prediction workflow (M24).
- Define a test setup for AI-based optimisation for FhG (M24).
- Perform shape optimisation for chevron nozzles for a reduced set of shape parameters without constraints (M30).
- Perform shape optimisation for a larger set of shape parameters including constraints (M36).
- Perform large scale runs for selected nozzle parameters with in situ data analytics (M30).

4.6 Gantt Chart

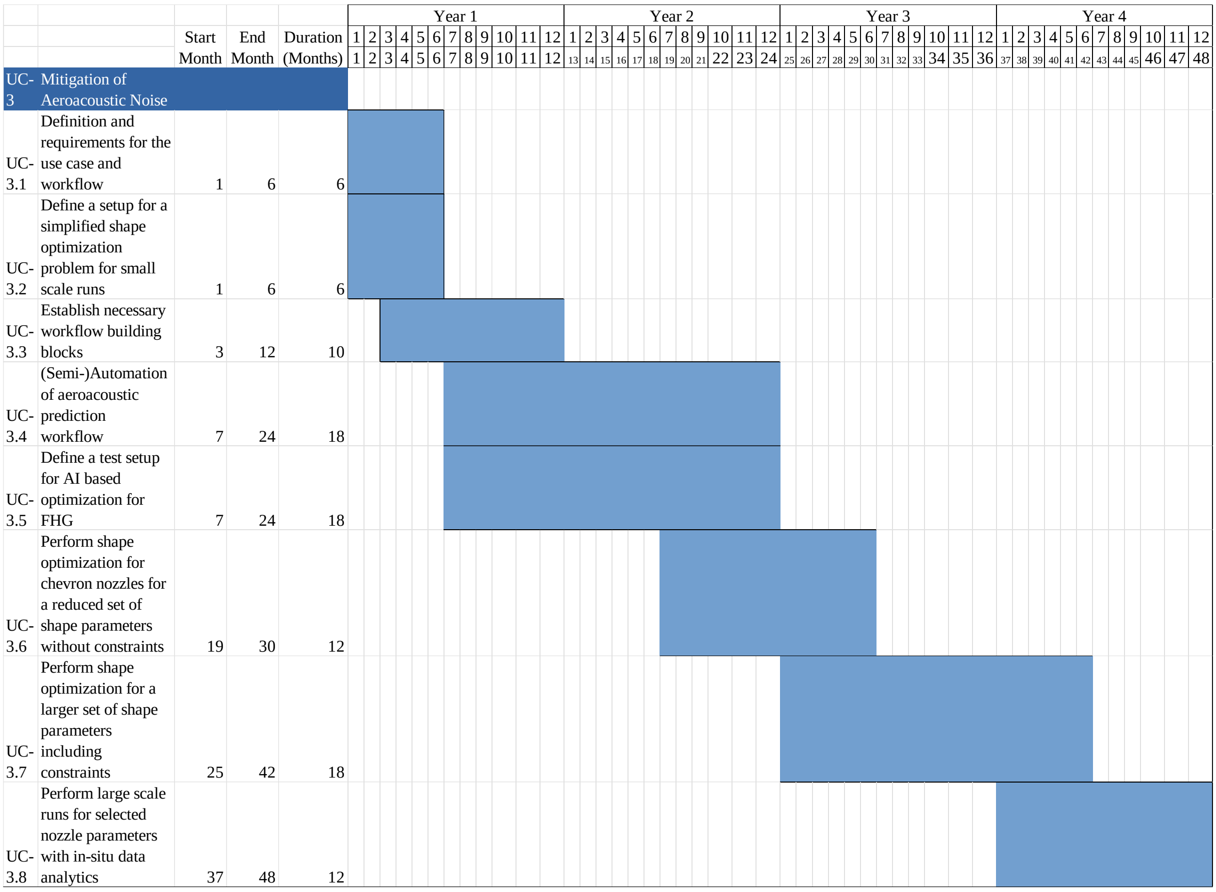

Figure 11: UC-3 Gantt Chart

4.7 Dependencies

The workflow has to be developed by the cooperation of the partners RWTH and FhG. Additionally, partner HLRS contributes in terms of in situ visualisation, which will be used for the flow field analysis.

4.8 Success criteria

For this use case the following criteria will be used to verify the success:

- Development of a workflow, which can automatically perform shape optimisations.

- Efficient usage of HPC hardware during the workflow execution.
- Identification of chevron shapes with minimized noise emission.
- Successful analysis of noise generation mechanisms based on large scale simulation runs on exascale hardware.
- Analysis of noise source mechanisms based on large scale simulations.

5 UC-4 Fully integrated aircraft simulations with emissions models

Partner: BSC, Code: Alya

5.1 Use Case Introduction

This use case is aligned with the strategies in the aeronautical industry to achieve the EU decarbonisation objectives by 2050 in air transportation, which includes substantial reductions in CO2, NOx emissions, and noise as defined in the Flightpath 2050. Generating digital tools that can be used to obtain detailed information on the relevant parameters from the aircraft and propulsive systems is an essential step toward designing low-emissions ultra-silent aircraft. An efficient strategy to develop these digital tools is using data-driven methods based on deep learning, where a neural network is fed with data from experiments and/or high-fidelity simulations. These tools can be integrated into design and optimisation workflows to reduce the computational cost, turnaround time, and energy consumption during the design phases.

This use case focuses on generating a multiphysics platform that integrates advanced simulation techniques from CFD codes and massively parallel workflows using leading-edge HPC architectures. Fully integrated simulations of the aircraft and propulsive system will be conducted with advanced HPC algorithms to obtain detailed predictions of aerodynamic parameters and emissions characteristics, including non-CO_2 effects like NOx or soot.

Figure 12 shows the flow field around the Common Research Model, a simplified aircraft geometry developed to test the accuracy of numerical simulations. Alya has already participated in previous versions of the High Lift Prediction Workshop. It is currently active in the latest version (https://hiliftpw.larc.nasa.gov), where we expect to showcase the improvements developed within EXCELLERAT P2.

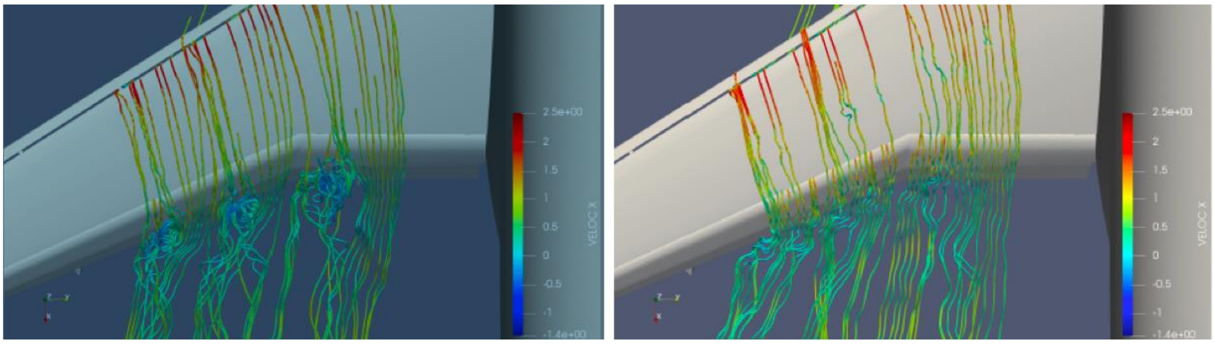

Figure 12: Instantaneous vortical structures calculated with Alya

5.2 Objectives

The accomplishment of such simulations involves different levels of hierarchies and parallelism in the algorithms responsible for controlling the different tasks. This requires the revision of the load balancing strategies for inter- and intra-nodes in the CFD codes, but also communication with the other tasks during post-processing and data analytics. A multicode strategy will be used to perform coupled multiphysics simulations, which brings additional challenges to ensure high computational performance, flexibility, and portability. As Alya contains a fully parallel workflow, including the reading and partitioning, a fully parallel workflow can be achieved after the mesh generation step. Such a parallel workflow is required in a pre-processing step. The I/O will be adapted to facilitate the reading/writing of files during runtime, and parallel algorithms will be used to communicate the data.

5.3 Execution profile

To enable digital twins of fully integrated aircraft simulations, we plan to integrate coupled external aerodynamics with engine emissions simulations over scale resolving simulation methods, including:

1. flow control devices (AFC, active surfaces, etc.).
2. high-fidelity combustion simulations with emissions.
3. Deep reinforced learning for flow control of high lift wings.

In EXCELLERAT P1, Alya's standard low-order finite element version was used. However, for EXCELLERAT P2, we are working on a new high-order version. It is based on a spectral element method of arbitrary order. An entropy viscosity stabilisation is used for shock capturing. Implicit and explicit time discretisation's are available. The use of OpenACC provides a code that is valid for both CPU & GPU.

5.4 Workflow

The workflow to be executed with Alya and its auxiliary tools is schematically depicted in Figure 13:

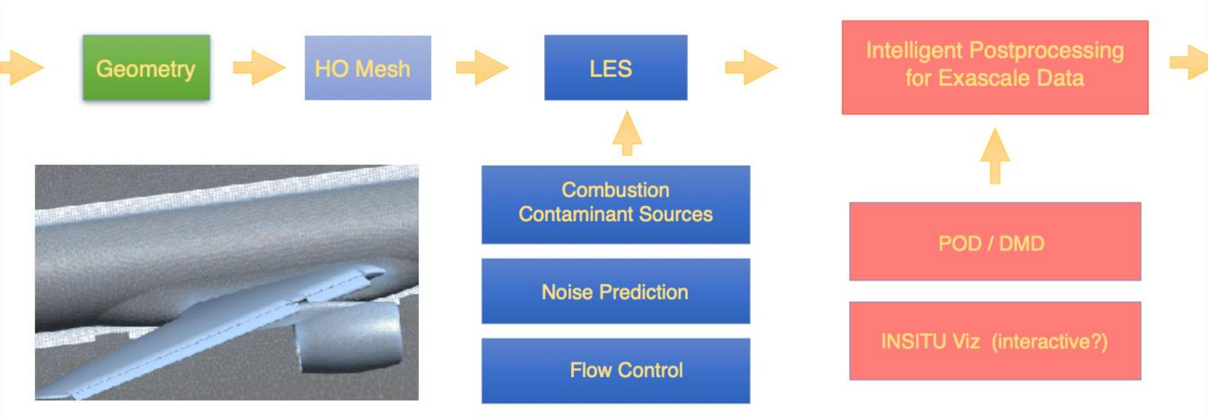

Figure 13: Workflow for fully integrated aircraft simulations with emissions models

The workflow starts with a complex geometry provided in some CAD format. In the case of the High Lift Prediction Workshop, the geometry is provided by the Workshop organiser. Before performing the simulation, a mesh must be obtained. We typically rely on commercial tools for complex geometries, ANSA (https://www.beta-cae.com/ansa.htm) being our default option. However, we have also been experimenting with the open-source tool GMSH (https://gmsh.info) for high-order meshes. For the High Lift Prediction Workshop, the organisers also provide suggested grids to promote a better comparison between the different codes.

Wall-modelled Large Eddy Simulation lies at the heart of our workflow. Accuracy is still a challenge for high Reynolds number turbulent flows over complex geometries. The substantial computational cost means high-fidelity numerical methods must be used optimally in the most energy-efficient architectures. We are opting for a spectral high-order discretisation implemented optimally on GPUs. As the workflow indicates, the flow simulation is coupled to contaminant sources from a Combustion model also available in Alya.

The Alya team has experience in active methods to control turbulence and separation, a crucial aspect of achieving high aerodynamic efficiency. Moreover, data-driven methods and their application to flow control will be explored with a focus on deep reinforcement learning (DRL).

The ever-growing computational power of modern HPC systems enables us to perform detailed simulations of the flow, which give us access to massive information on the flow, however, typically only a tiny portion of the information is used. In an extreme case, even only the drag and lift coefficients are evaluated. The amount of data that is extracted depends heavily on the experience of the CFD user. Thus, for exascale simulations, intelligent postprocessing tools help to analyse and understand the flow making the workflow more efficient. DMD and POD tools can help reduce the complexity of the data and identify the underlying physical processes. DMD decomposes the data into a set of modes that capture the dominant dynamics of the system. It helps to identify the frequencies and amplitudes of the dominant oscillations in the system and provides insight into its stability and behaviour. POD decomposes the data into a set of orthogonal modes that capture the dominant spatial structures in the data. It can be used to identify the most important patterns of variability in the system and help identify regions of high turbulence and flow separation. Using DMD and POD together can provide a complete understanding of the system by identifying the dominant modes of variability and the dominant spatial structures. BSC develops its own tools to use DMD/POD in HPC systems.

Additionally, we will use in situ visualisation and virtual reality (VR) to further enhance the analysis of exascale CFD data. Alya will be interfaced with the Distributed Data-parallel Scientific Visualization tool VISTLE (https://vistle.io). In situ visualisation helps analyse and visualise the data while it is being generated, rather than waiting until the simulation is complete, making our workflow more efficient. Additionally, it reduces the amount of data that needs to be stored and transferred, which is critical at Exascale. On the other hand, VR capabilities provided by VISTLE allow a more immersive and intuitive way to visualise and explore the data. This can be particularly useful for visualising complex, multi-dimensional datasets like the ones we generate. A combination of DMD, POD, in situ visualisation, and VR will help us transform how we analyse and visualise the data we generate.

We resume the key feature of each of the components of our workflow:

Mesh Generation

- Unstructured high-order tetrahedral grids generated with commercial (ANSA) or free (GMSH) tools.
- Partitioning to hexahedral third-order meshes in GMSH and improvement with HOPR (https://www.hopr-project.org).
- In-house mesh partitioner that uses the parallel mesh partitioning library GeMPa based on Space-Filling Curves.
- Using high memory nodes, we expect to generate O(10^9) nodes.

LES/CFD Simulation

- Fully parallel workflow starting from an already partitioned grid using external tools.
- O(10^9) degrees of freedom. Higher number of DOFs either using other external mesh generation tools to be investigated, automatic mesh subdivision, or higher order elements.
- Perform simulation until the time averaged values of interest are sufficiently converged.
- Medium sized job*: O(100) GPUs, O(96 hours).
- Instantaneous and averaged quantities passed to intelligent post processing tools using parallel output (HDF5) or in situ.
- For contaminants dispersion coupling with combustion module as a source.

Reactive flow

- Extension of the compressible Navier-Stokes solver based on spectral elements to turbulent reactive flows.
- Study of thermoacoustic instabilities of the PRECCINSTA burner from the high-pressure combustion test rig HIPOT at DLR Stuttgart.
- Expected runs with highly resolved LES with O(10^9) degrees of freedom.
- Job size: O(100) GPUs, O(48 hours).
- Combine with parallel post-processing/analysis using POD/DMD.

Flow control

- Passive methods based on modifying the airflow by means of geometrical additions on the shape.
- Active methods based on directly modifying the flow (e.g., by blowing-sucking means, etc.).
- Used for drag reduction/lift enhancement (aerodynamic efficiency) in aerospace applications.
- Classic methods (based on linear control theory) vs new generation methods based on AI (deep reinforcement learning, etc.).
- Several possible actuators are available.
- How to sense the flow for the actuation is still an open issue.
- Job size: O(10) GPUs, O(48 hours).

Intelligent postprocessing

- Parallel POD/DMD tools.
- In situ visualisation with Vistle with the possibility of Virtual Reality.

To summarise, an efficient workflow for scale-resolving simulation of aircraft simulations with emissions models is proposed. Apart from the mesh generation, the workflow is fully parallel. A high-order discretisation is used for the LES simulation. It is implemented to work optimally on GPUs leading to a significant reduction in energy consumption.

According to their sizes, the runs that have been planned for EXCELLERAT P2 can be classified as follows:

- Hero run: coupled external aerodynamics with engine emissions simulations.
- Large-scale: External aerodynamics with scale-resolving simulation methods, including flow control devices (AFC, active surfaces, etc.), high-fidelity combustion simulations to predict pollutant formation NOx and soot.
- Small scale/ensemble: Deep Reinforce Learning for flow control of high lift wings.

5.5 Timeline

Developing fully integrated aircraft simulations with emissions models relies on a high-order version of Alya implemented optimally on GPUs. It is matched with fully parallel tools for mesh partitioning and intelligent post-processing.

For the technical developments, we propose the following milestones:

- Definition and requirements for the use case and workflow (M6).
- Perform LES simulations on simplified geometry (M12).
- Coupling with combustion model (M24).
- LES simulations with more complex geometry (M30).
- Flow control testing of classical and AI based approaches (M30).
- Intelligent postprocessing including DMD and POD (M36).
- Perform hero run including pollutant dispersion and in situ postprocessing (M48).

5.6 Gantt Chart

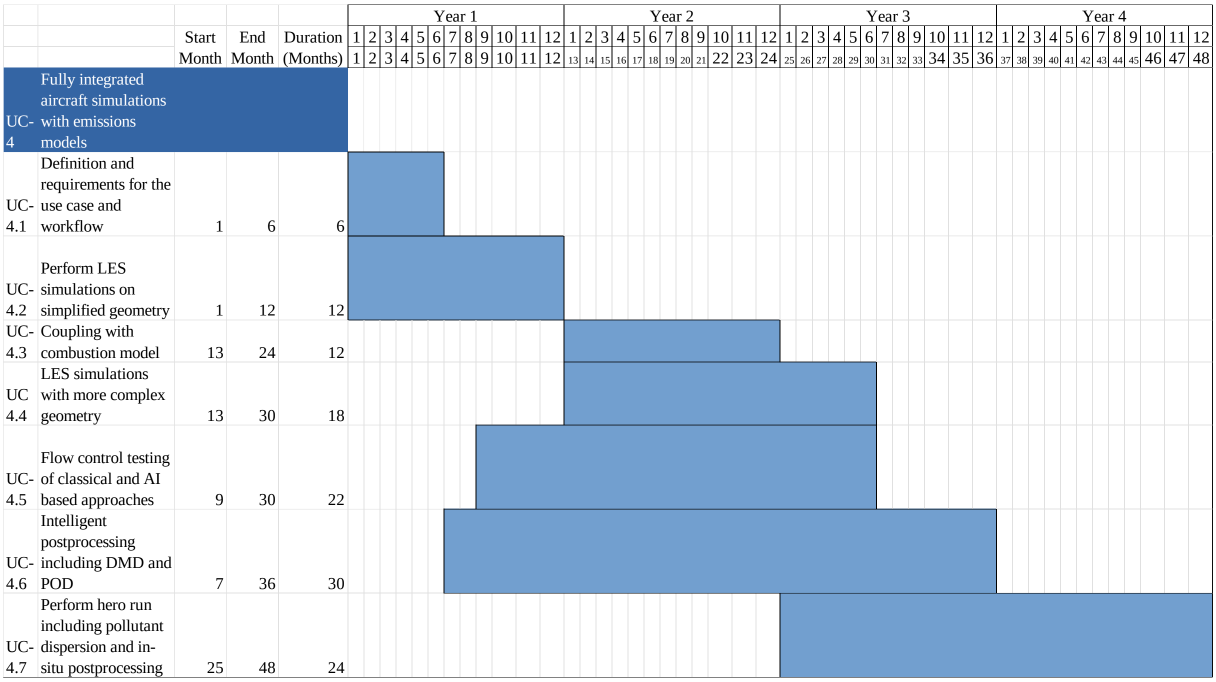

Figure 14: UC-4 Gantt Chart

5.7 Dependencies

The workflow relies mainly on tools developed by BSC. For in situ visualisation and virtual reality we will use the tool VISTLE (https://vistle.io) developed by HLRS.

5.8 Success criteria

For this use case, the following criteria will be used to verify the success:

- Development of a workflow for integrated aircraft simulations with emissions models.
- Optimal implementation of the key kernels on GPUs.
- Creation of intelligent postprocessing procedures merging DMD, POD, in situ visualisation, and VR.
- Successful Alya results in the 5th AIAA CFD High Lift Prediction Workshop (HLPW-5).
- Progress in flow control for aircraft using high-order scale resolving simulations.

6 UC-5 High-fidelity simulation of rotating parts

Partner: KTH, Code: Neko

6.1 Use Case Introduction

This use case is a continuation of C1U3 use case introduced in EXCELLERAT phase 1 by CINECA. We have decided to continue with this use case as, despite of the significant development done during EXCELLERAT P1, it is still of scientific interest, challenging, and requires further code improvements. In addition, this class of flow cases is very industrially relevant, as rotors are used in many mechanical devices, where they allow the devices to exchange kinetic energy between the fluid and the device. It is very important to better understand the dynamics of the flow around rotors, as by improving the energy efficiency, we can for example decrease the energy consumption of vehicles or increase the energy production of wind turbines. In EXCELLERAT P2 we will study rotors in context of aero mobility, looking closer at a drone rotor in hover configuration. This choice is motivated by the rapid expansion of the small UAV market, that is expected to reach USD 58.4 billion by 2026 [1]. Diverse use of drones, covering e.g., package delivery or law enforcement, makes them commonly found in daily life. A side effect of the frequent use of UAVs is noise pollution. This motivates detailed studies of the properties of the flow around the drone blade, both experimental and numerical.

We are going to investigate a two-bladed, twisted rotor studied by Ning [2]. This configuration was later investigated in several works including numerical ones e.g., Delorme et al. [3] or recent Nathanael et al. [4], but there is lack of results of high-fidelity simulations that do not rely on turbulence modelling. This flow case also constitutes an excellent platform to exploit high-order numerical methods using advanced meshing techniques (e.g., adaptive mesh refinement) in an industrially relevant case. The geometry of this case is relatively complex, but still affordable computationally for hex-based meshing. In addition, the complex flow dynamics are rich in a variety of flow features (e.g., tip vortices). This allows the testing of the whole simulation workflow starting with mesh generation, going through mesh adaptation strategies, in situ data analysis, and ending with post-processing.

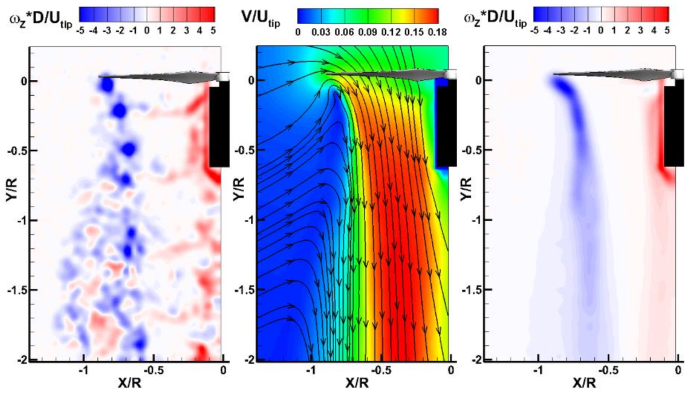

Figure 15: "Free-run" PIV measurement of the propeller in hover configuration. Source [2]

6.2 Objectives

The main goal of UC-5 is to perform high-fidelity simulation of an "Iowa" rotor in hover configuration focusing on the rotor thrust, the wake characteristics and noise generation. We will perform the runs using the high-order CFD solver Neko using GPU accelerators. Neko is a SEM solver sharing multiple features with Nek5000 (which could be considered a predecessor of Neko). During EXCELLERAT P1 we worked successfully on the robust, fully three-dimensional h-type AMR framework for Nek5000, that was applied to two different use cases.

The main limitation of this approach came from the problems inherent in Fortran77 when using GPUs. These problems are solved in Neko using modern object-orientated Fortran 2008. Neko supports various hardware-backends by defining a device abstraction layer and then writing hardware specific kernels in CUDA, HIP or OpenCL. Although Neko provides very good support for GPUs, it lacks some of the features added to Nek5000 during EXCELLERAT P1 e.g., AMR or UQ. That is why our first aim is porting our previous AMR and UQ implementations to Neko. We will then further develop these methods. In case of AMR, we will focus on the performance of pressure preconditioner, work balance, P_N-P_N implementation and efficient GPU porting. For UQ further improvement of assessing the uncertainly of time-averaged quantities will be considered.

Moreover, we are going to implement in Neko acoustic solvers based on integral formulations as well as FEM/SEM solvers for the acoustic wave equation with different source term formulations [5]. The source term will be directly computed from the incompressible flow field solutions. The last aspect we are going to consider is an in-situ data analysis. A good example is that POD and DMD analysis are widely used in CFD community, but not affordable for exascale simulations due to large amount of required data. We are going to implement parallel streaming algorithms [6] for both POD and DMD in Neko and perform the whole analysis in situ. We also plan to implement in Neko the data compression for SEM solvers developed as a part of EU project ADMIRE.

6.3 Execution profile

We are going to perform multiple varied simulations of the rotor while developing this use case. They will include small test runs for different code components (e.g., acoustics or streaming algorithms) as well as a final "hero" run performed on a large number of GPUs. Right now, it is difficult to estimate the size and the number of these runs, so we present a rough estimate of a cumulative time needed each year (see Table 1). Specified time is consistent with requirements reported in the special access scheme document.

	Benchmarking	Development	Regular	Extreme
Year 1	50'000 Node hours	100'000 Node hours	1M CPU core hours; 0.1M GCD hours	2M CPU core hours; 0.2M GCD hours
Year 2	50'000 Node hours	100'000 Node hours	2M CPU core hours; 0.2M GCD hours	4M CPU core hours; 0.4M GCD hours
Year 3	50'000 Node hours	100'000 Node hours	5M CPU core hours; 0.5M GCD hours	10M CPU core hours; 1M GCD hours
Year 4	50'000 Node hours	100'000 Node hours	5M CPU core hours; 0.5M GCD hours	10M CPU core hours; 1M GCD hours

Table 1: Rough estimate of a cumulative time needed each year

6.4 Workflow

During EXCELLERAT P2 we are going to work on the whole simulation workflow covering pre-processing, running and post-processing stages. At the pre-processing step the main challenge is the generation of a proper hex-based, high-order, conforming mesh. This mesh is later evolved during the simulation through element splitting/coarsening according to the evaluated error indicator/estimator. This process is performed dynamically in a recurrent way, until a statistically converged mesh/solution is reached. Note, we assume here a statistically converged solution can be reached and we do not intend to track the time dependent features of the flow. After a statistically converged mesh/solution is reached, the refinement/coarsening process is stopped, the resulting nonconforming mesh gets frozen and is used for collecting and processing in situ data. At this stage the incompressible fluid solver will be coupled with an acoustic solver, the streaming POD/DMD, the tool collecting statistics and UQ tool. At the post-processing step the statistics collected over several runs will be further analysed and the results of UQ will be used to quantify the error of the time-averaged quantities. In more detail, the workflow components are:

High-order hex-based meshing: Neko similarly to Nek5000 relies on a hex-based, high-order, conforming meshes. The generation of such meshes for complex geometries is a challenging problem that has not yet been fully solved. AMR and the resulting non-conforming meshes increase the meshing flexibility significantly, but even in those cases the starting point for the simulations are coarse conforming meshes. Most of the open source and proprietary mesh generators are very efficient in creating meshes with mixed-type elements, but building a hex-only mesh usually is a tedious and time-consuming exercise. The rotor geometry studied in UC-5 is relatively complex, but it is still possible to mesh it with most of the available software. As Neko is intended for both academia and industrial users, we are going to investigate both proprietary (e.g., pointwise) and the open source (gmsh) mesh generators. During EXCELLERAT P1 we created a simplified "toy" rotor mesh using gmsh and its scripting language and concluded that this scripting language is not flexible enough for more complex projects. That is why in EXCELLERAT P2 we test the gmsh modern Fortran interface, where we benefit from the advantages of a more mature programming language.

H-type AMR: An important component of our simulation is AMR, which allows an optimal mesh to be built dynamically for a given flow case setup according to the estimated computational error. This feature becomes critical for large-scale simulation with limited a priori knowledge of flow dynamics. During EXCELLERAT P1 we developed a h-type AMR workflow in Nek5000 and now we will port it to Neko. In this approach we take advantage of a standard SEM mesh decomposition into a set of non-overlapping subdomains (elements) and perform recurrent split of existing subdomains into the smaller ones. As both codes are SEM solvers, most of the workflow will remain unchanged and we can directly use our experience from EXCELLERAT P1. The most significant difference is a velocity-pressure coupling, as in Nek5000 we used P_N-P_{N-2} formulation, and in Neko only supports P_N-P_N formulation. In addition, Neko would require a few additional features like variable time step, or surface projection tool. The starting point will be a CPU implementation, but the next important task will be an efficient GPU implementation. Finally, we will assess the influence of the P_N-P_N AMR implementation on collected statistics.

Aeroacoustics: Aeroacoustic noise is an important factor that needs to be considered when designing a rotor for any application as it can affect its useability. For drones that operate in urban environments, it is essential to study their noise footprint so they are less disruptive in its flying environment. Hence for the rotor that we simulate, we will investigate its noise emissions by using methods that can be coupled to our incompressible flow solver, Neko. As for most applications, the far-field acoustic pressure is of interest, we initially implement and utilize the integral method (FWH [7]) in situ. This method has a low computation cost while proven to be effective for rotating blade noise prediction [8]. However, integral methods cannot account for the reflection and refraction of acoustic waves. Therefore, the acoustic wave propagation needs to be simulated using an acoustic wave equation. For this reason, we will solve the acoustic wave equation using FEM/SEM by considering different source term analogies. The source term will be computed from the incompressible flow field results in Neko. The acoustic source term and the predicted acoustic pressure will be analysed in combination with the analysis of the flow statistics in order to identify the noise sources.

Streaming algorithms: POD and DMD are among a widely used class of methods that aim to obtain a dimensionality reduction of the time dependent data sets. They are used with the aim to perform data compression, reduced order modelling and analysis of data by means of extracting meaningful flow structures from the field. An efficient way to perform POD is to use the so-called MOS [9], but for our implementation, we focus of methods based on SVD, a mathematical factorisation technique. We focus on combining certain aspects of parallel algorithms for the computation of SVD [10] and streaming algorithms [6] along with several optimisations to ensure that we can perform parallel and streaming SVD on many nodes, as is required by the large amount of data produced by large scale simulations. To be able to execute the in-situ task asynchronously while the main simulation runs, we implement the in-situ engines that ADIOS2 provide and that were initially implemented to allow data compression in Neko predecessor Nek5000.

Uncertainty quantification: Due to finite time averaging, computed statistics of turbulent flow simulations are always to some extent uncertain. The prediction of these error bounds is not a trivial task, in particular if the corresponding analysis should be done during the runtime, i.e. without writing out long time series. We will continue our work on a low-storage updating (streaming) algorithm for the online prediction during the run-time of a simulation. The proposed framework, partially implemented in Nek5000 and now being transferred to Neko and corresponding GPU runtime, is based on modelled ACF, through a VTK-Catalyst interface. Special focus will be placed on an extension of the previous framework, developed in collaboration with Fraunhofer SCAI and the University of Manchester, to higher-order general turbulence statistics such as the turbulent kinetic energy.

In situ visualisation: In situ visualisation is important because it allows the analysis of simulation results that would otherwise be impossible due to the size of the data sets that would be needed and the required post-processing execution time. During EXCELLERAT P1 in situ visualisation was implemented for Nek5000 and will be it implemented for Neko in EXCELLERAT P2. The SENSEI open-source interface library will be used in the implementation. This interface allows the simulation to be coupled with a wide range of visualisation tools e.g., VISTLE, VisIt and ParaView.

6.5 Timeline

The development of Neko will concentrate on three aspects: AMR, in situ data analytics and noise prediction. Initially each of these features will be developed independently and then combined in single hero run.

The technical developments will be performed in the following steps:

- Mesh generation.
    - Generation of a zero-level rotor mesh with gmsh interface (M6).
- AMR development.
    - Implementation of a variable time step, a surface projection, an error indicator/estimator and the “refinement machinery” (M12).
    - CPU implementation of h-type refinement for P_N-P_N approach (M24).
    - Nonconforming mesh solver using GPU (M30).
- Acoustics development.

  - Implement and validate the FW-H solver (M12).
  - Incorporating the acoustic wave equation solver (M30).
  - Generation of the acoustic mesh and source term interpolation (M32).
  - Acoustic simulations with velocity/pressure-based source terms (M36).
- Streaming data analysis development.
  - Implement ADIOS2 streaming functions (M12).
  - Implement parallel/streaming SVD algorithms (M18).
- Uncertainty Quantification.
  - Implement Catalyst-UQ framework to Neko (M12).
  - Theoretical development of ACF framework for higher-order statistics (M24).
  - Finalisation of final UQ statistics framework (M30).
- Coupling AMR with in situ data analytics and acoustics and performing large-scale runs (M48).

6.6 *Gantt Chart*

		Start Month	End Month	Duration (Months)
UC-5	High-fidelity simulation of rotating parts			
UC-5.1	Generation of a zero-level rotor mesh with gmsh interface	1	6	6
UC-5.2	Implementation of a variable time step, a surface projection, an error indicator/estimator and the "refinement machinery"	1	12	12
UC-5.3	CPU implementation of h-type refinement for PN-PN approach	12	24	13
UC-5.4	Nonconforming mesh solver using GPU	24	30	7
UC-5.5	Implement and validate the FW-H solver	1	12	12
UC-5.6	Incorporating the acoustic wave equation solver	12	30	19
UC-5.7	Generation of the acoustic mesh and source term interpolation	24	32	9
UC-5.8	Acoustic simulations with velocity/pressure-based source terms	32	36	5
UC-5.9	Implement ADIOS2 streaming functions	1	12	12
UC-5.10	Implement parallel/streaming SVD algorithms	7	18	12
UC-5.11	Implement Catalyst-UQ framework to Neko	4	12	9
UC-5.12	Theoretical development of ACF framework for higher-order statistics	4	24	21
UC-5.13	Finalisation of final UQ statistics framework	24	30	7
UC-5.14	Coupling AMR with in-situ data analytics and acoustics and performing large-scale runs	30	48	19

Figure 16: UC-5 Gantt Chart

6.7 *Dependencies*

The workflow will be developed in collaboration with partners within EXCELLERAT P2 project, mainly RWTH (acoustics) and FhG (UQ and in situ data analytics). Moreover, we are going to interact with other EU projects (ADMIRE and CEEC) and national centres (SeRC). Additional academic collaboration with colleagues from Friedrich-Alexander-Universität Erlangen-Nürnberg and the University of Manchester.

6.8 *Success criteria*

For this use case the following criteria will be used to verify the success:

- High quality, high-order mesh for an "Iowa" rotor.
- 3D AMR demonstration performed on CPU.

- 3D AMR demonstration performed on GPU.
- Coupled 3D AMR with in situ streaming POD/DMD.
- Coupled 3D AMR with acoustic solver.

7 UC-6 Active control for drag reduction of transonic airfoils

Partner: CINECA, Codes: FLEW

7.1 Use Case Introduction

In this use case we consider the aerodynamic performance in terms of lift and drag of a typical supercritical airfoil in the presence of wall actuation, in the form of spanwise traveling waves, following the scheme originally introduced by [11]. As shown in that publication through direct numerical simulation (DNS), use of spanwise oscillations modulated in space and time can yield up to 30% net frictional drag reduction when applied to a flat plate. However, several issues need to be resolved, including scaling of the actuation performance with the Reynolds number, as well as applicability to non-trivial flow geometries, such as airfoils, also in the presence of significant compressibility effects occurring in practical applications. The answer to the first question came from a recent study [12], showing that net drag reduction can also be achieved at Reynolds numbers of relevance for aeronautics. In another paper the authors also devised a practical method to implement spanwise traveling waves through in-plane transverse momentum injection to disrupt the large-scale eddies in a turbulent boundary layer [13]. Hence, besides substantial theoretical interest, wall actuation through spanwise traveling waves bears promise of practical implementation. In a recent paper [14] the authors have tested the use of spanwise traveling waves on a limited portion of the suction side of a supercritical airfoil, considering a moderate Reynolds number (Re=300000, based on the chord), and a single flight condition ($M\infty = 0.7$, $\alpha = 4°$), with moderate (C1) and strong (C2) wall actuation. As expected, reduction of frictional drag was achieved, as in the case of flow over a flat plate. Quite unexpectedly, a favourable effect on lift was also found as a by-product, resulting from downstream shift of the lambda shock, as depicted in figure 1, which is equivalent to an increase of the Mach number on the suction side of the wing, and significant improvement of the lift/drag ratio. For a constant angle of attack, the efficiency was observed to increase by 13.5 % (with drag decreased by 0.8 % only). Higher aerodynamic efficiency allows the required lift to be achieved at a lower angle of attack, yielding significant reduction of the total drag, which was quantified via DNS to be about 15 %. It was also estimated that this may lead to overall drag reduction of about 9 % for a full aircraft in cruise flight, and that the energy cost for the active control is about 1 % of the total power expenditure.

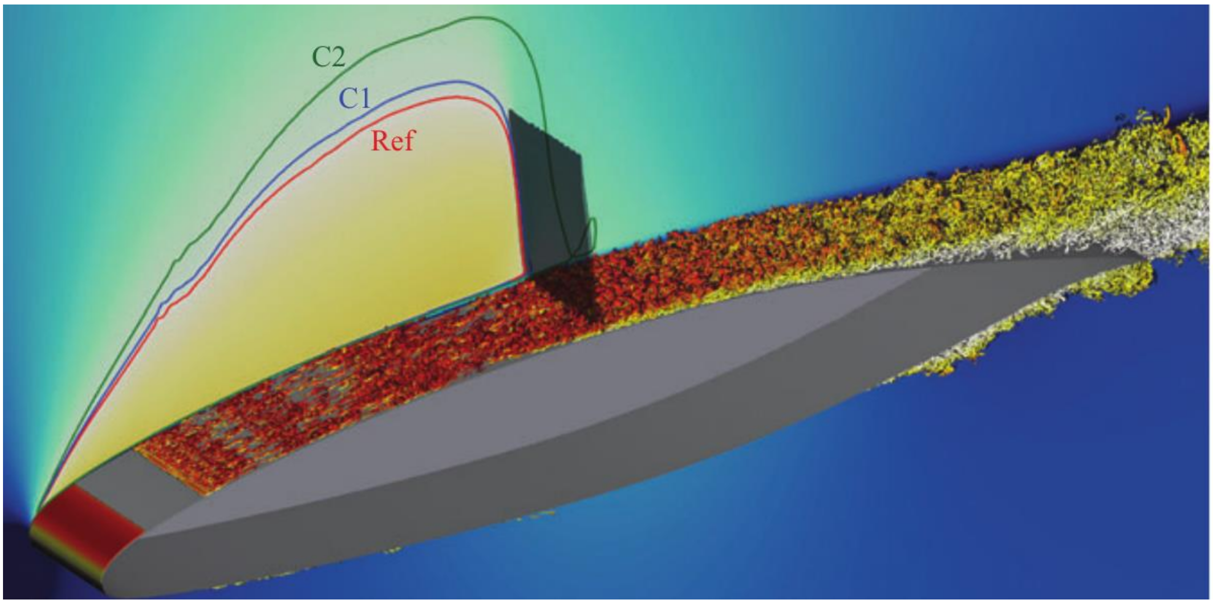

Figure 17: Flow over supercritical airfoil. Lines denote the position of the mean sonic line, with no actuation (Ref), and with moderate (C1), and strong (C2) wall actuation

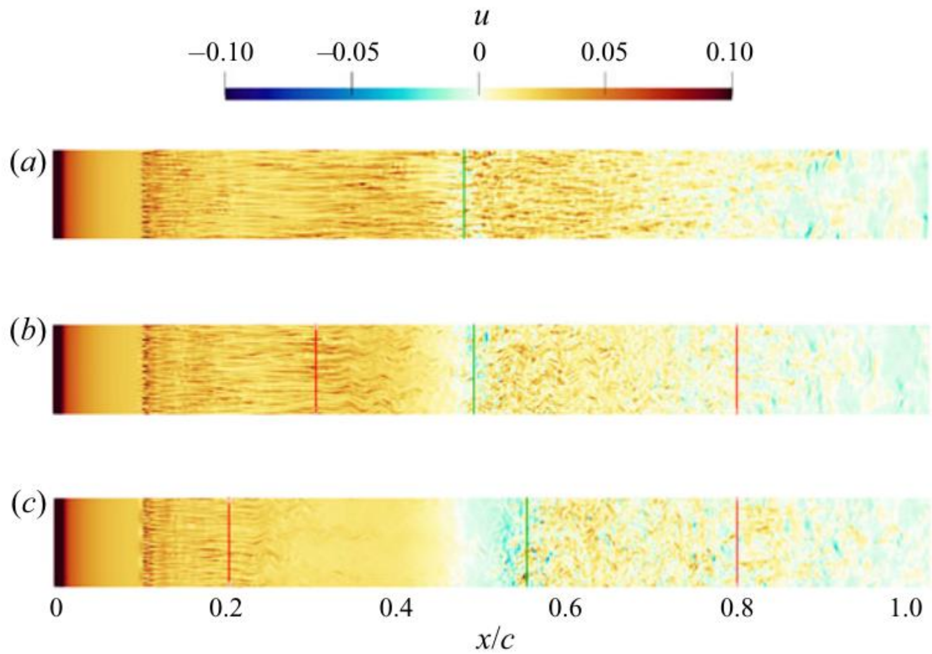

Figure 18: near-wall streamwise velocity contours with no actuation (a), C1 actuation (b), and C2 actuation (c)

7.2 Objectives

This background provides the motivation for the present use case, which is meant to dive deeper into the physics of drag reduction in realistic flow conditions, and to determine the conditions for optimum performance. Starting from our previous study conducted by Quadrio et al. [14], we aim at conducting a systematic campaign, whereby: 1) the effects of varying wall actuation parameters (frequency and wavenumber) are studied, to determine optimal performance for a given flow condition; 2) the flight Mach number and incidence angle are varied, to explore aircraft performance in a large flight envelope, in the presence of wall actuation. In order to make this parametric study meaningful, Reynolds numbers should be achieved which, although possibly scaled with respect to the real case, make the flow devoid of major transitional effects. Hence, we target Reynolds numbers of about one million for this parametric study. We estimate that about 100 of such DNS should be carried out to span the parameter space, and each DNS would include about 10 billion grid points. Last, at the end of the exploratory part, we aim at carrying out a limited set of DNS at higher Reynolds number (Re=3 million) corresponding to the most promising conditions in terms of wall actuation, and of course for the uncontrolled case. The meshes for these large-scale DNS would include from 200 to 300 billion grid points.

7.3 Execution profile

In order to achieve the goals set for this project, a high-fidelity solver is needed, which should be devoid of numerical dissipation. In our previous study, an in-house second-order, finite-volume solver was used which was able to guarantee discrete conservation of kinetic energy in the absence of viscosity, hence with nominally zero numerical diffusion. However, in order to further increase accuracy and numerical efficiency, in this project we use a novel finite-difference solver for generalized curvilinear coordinates, with arbitrary order of accuracy, and which is also capable of guaranteeing discrete conservation of the total kinetic energy, according to the formulation developed by Pirozzoli [15]. A series of code validation activities has been carried out, which has shown full reliability and accuracy for complex compressible flows. The current implementation of the FLEW solver is currently performing on classical CPU architectures with standard synchronous MPI message passing.

Successful completion of the DNS campaign set out above should involve an efficient implementation of the solver to exploit all available heterogeneous CPU/GPU resources, which requires intensive code porting activities, to be carried out by the CINECA specialists. Porting will be however facilitated as the solver shares the same numerical backbone of the STREAmS solver, which can be regarded as its Cartesian sub-case. An important issue to be dealt with during the execution of the project is also efficient storage of data to guarantee subsequent analysis of the simulation results.

7.4 Workflow

The workflow is sketched in Figure 19 below:

- Input selection: parameters are set in a text file.
- Grid generation: single block structured mesh for DNS.
- DNS: simulation is performed to predict flow statistics.
- Post-processing: benefit evaluation in terms of drag reduction for a given lift.

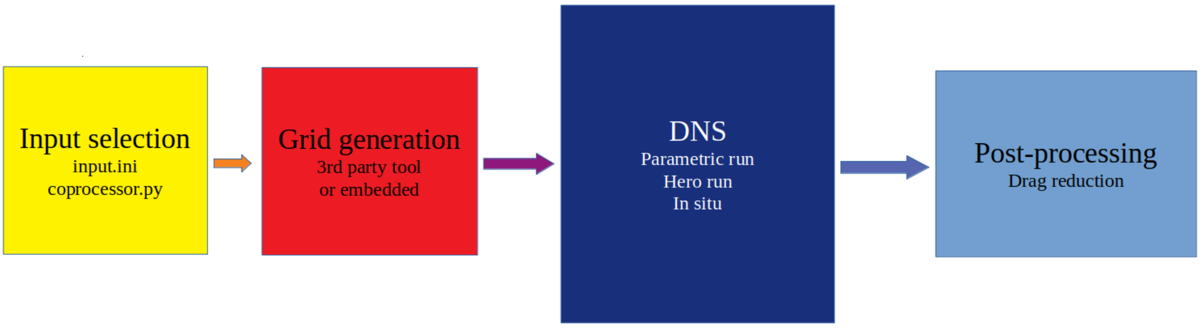

Figure 19: UC-6 Workflow

More details about the individual components in what follows:

Input selection

- Only textual inputs for the parameter selection.
- Input sections for: grid related inputs, boundary conditions, MPI partitioning, time integration, spatial numerical scheme, flow parameters, fluid type, I/O control, in situ.
- Based on our previous experience, in situ control will be split in two different files.
    - The first is the one summarised above, the second is a python script which handles the ParaView pipeline. The first file cannot be modified during runtime, and specifies parameters which is easier and most effective to control in FLEW rather than with ParaView's APIs, such as: exported quantities, freezing time ranges (e.g., for having different views of the same instantaneous field in a video), image extraction frequency.
    - The second file, the python script, is mainly used to control the ParaView visualisations (camera movements, variables to be shown, filters, colour ranges).

Grid generation

- The grid consists of a two-dimensional, single block, structured mesh, which is then extruded in the third dimension within FLEW.
- Grid is fixed for a given Reynolds number.
- Grid checking will be attempted within FLEW.

- Currently, grid generation is performed by third party commercial software, but we will try to integrate grid generation process within FLEW.

DNS

- The flow is initialized with an impulsive start.
- Domain decomposition ensures optimum load balancing thanks to grid simplicity.
- Target number of nodes: for production runs we expect 10 billion points, for hero runs we expect 50 billion points.
- Control optimisation: large number of flow conditions should be explored. The sketched workflow is only devoted to the single simulation, the control optimisation is considered an external task.
- In situ statistics, computed as the simulation progresses.
- Each simulation is performed up to statistics convergence, which ideally will be checked automatically by an ad-hoc routine, e.g., by sending a signal to a python daemon, allowing to perform a clean stop of the simulation in case of statistics convergence or when close to scheduled time-limit.

Post-processing

- Evaluation of the drag reduction effectiveness for a given lift.
- Merging of the frames obtained from in situ to produce a movie.
- Peer-reviewed publication of the results.

7.5 Timeline

- Baseline FLEW solver has most of the functionalities strictly required to perform the simulations envisaged in the context of the use case: airfoil boundary conditions will be implemented by M8.
- Validation of baseline FLEW code running on traditional CPUs will be completed by M9.
- FLEW will be completely restructured to follow object-oriented design by M15 so that the code can be suitable for development of multiple computing backends.
- FLEW porting to multi-GPU architectures – both NVIDIA and AMD GPUs – will be carried out by M27 (NVIDIA porting will be completed earlier) to make the code suitable for most EuroHPC architectures.
- Complete validation and benchmarking of the refactored/ported code will be performed by M32 deploying the code on different EuroHPC machines.
- The management of FLEW workflow will be improved and partially automated to improve usability and overall utilisation efficiency (M36).
- First set of production runs will be performed from M18 to M40 to simulate wide ranges of physical parameters involved in the use case.
- The largest production runs will be performed from M34 (or earlier) to M44 addressing unprecedented Reynolds number of this type of controlled flows.
- Post-processing and statistics analyses will be completed by M46.

- Finally, scientific studies will be disseminated through conference and peer-reviewed papers hopefully submitted from M44 (or possibly earlier) to the end of the project (M48).

7.6 Dependencies

The workflow has to be developed by the cooperation of the partners CINECA and URMLS. The use case completion has several interdependencies with WP3 and WP4 tasks. Code restructuring and GPU porting is mandatory to fully exploit the computational potential of EuroHPC machines (WP3). On the other hand, workflow improvement and in situ visualisation are important topics which improve the usability and the quality of result usage (WP4).

Another dependency is given by the availability of computing time that will be provided within the project. The availability of a sufficient computing budget is also required to achieve the results planned in this use case roadmap.

7.7 Gantt Chart

		Start Month	End Month	Duration (Months)	Year 1	Year 2	Year 3	Year 4
UC-6	Active control for drag reduction of transonic airfoils							
UC-6.1	Implementation of airfoil boundary condition	1	8	8				
UC-6.2	Validation of baseline CPU code FLEW	6	9	4				
UC-6.3	Code restructuring following object oriented design	9	15	7				
UC-6.4	Code porting to multi-GPU architectures oriented to EuroHPC machines	14	27	14				
UC-6.5	Validation and benchmarking of restructured/ported code	18	32	15				
UC-6.6	Workflow automation improvement	6	36	31				
UC-6.7	Parameter optimization runs spanning a wide range of parameters	18	40	23				
UC-6.8	Large scale run for optimized parameters at unprecedented Reynolds number	34	44	11				
UC-6.9	Post-processing and statistics analyses	36	46	11				
UC-6.10	Dissemination of results in conference and peer-reviewed journals	44	48	5				

Figure 20: UC-6 Gantt Chart

7.8 Success criteria

For this use case the following criteria will be used to verify the success:

- Efficient usage of HPC hardware during the workflow execution.
- Identification of optimal performance parameters for wall actuation through traveling waves.
- Identification of airfoil efficiency increase corresponding to optimal wall actuation.
- Identification of Reynolds number effects on computed results.

- Successful execution of DNS of transonic flow at unprecedented Reynolds number.

8 FA-1 Engineering design and digital twin of the first wall of a tokamak fusion reactor

Partner: UL, Codes: OpenFOAM, ElmerFEM, Raysect, Mitsuba 2

8.1 Use Case Introduction

In this use case, the setup of a digital twin of the first wall of a tokamak fusion reactor is conducted. During the fusion reaction in a tokamak, determining the real temperature at the surface of the first wall of the tokamak from infra-red (IR) cameras is a crucial task for appropriate engineering design of the first wall. Furthermore, while the fusion reaction is running, it is important to detect in real time overheating parts to be able to react before these parts of the reactor are damaged. To this end, the performance of the engineering parts of the tokamak (dependent on geometry and temperatures in the system) are modelled with Finite Element Methods, i.e., via the Elmer FEM code and with Finite Volume Methods, i.e., via OpenFOAM. The results will be validated using experimental data that is available from ITER first plasma experiments (using a temporary limiter) and synthetic diagnostic that combines IR and visible light camera images.

The optical performance of visible light and IR cameras is simulated, based on Monte Carlo methods using forward ray-tracing algorithms to predict the optical flow. Optical parameters such as surface reflectivity are modelled with Cook-Torrance models, which are assessed with ray-tracing codes (Raysect and Mitsuba2). An example of predicted camera output is shown below:

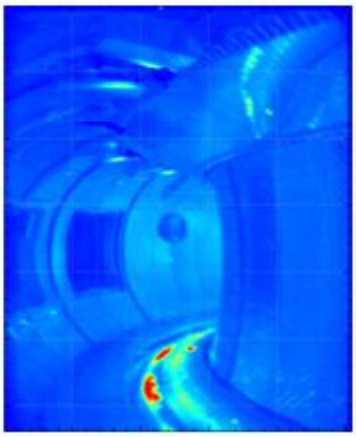

Figure 21: Synthetic camera image of WEST tokamak during operation

8.2 Objectives

The goal of the simulations is to accurately predict stray light, hot spots and uniformity to ensure performance and evaluate illuminance and luminous intensity in the visible and IR light range. The coupled thermal-optics simulations under high reflectance are used to correctly describe the visible light in cameras that see not only surface temperature but also reflected light from other surfaces. When reflections are subtracted, correct surface temperatures are determined. Therefore, only complete simulations (digital twins) can correctly predict what is seen by the cameras. The ray-tracing and thermal models of cooled surfaces are exascale relevant simulations with good scaling prediction. The inverse problem, when we have a real image from a camera from which surface temperatures should be determined, knowing that the surfaces are reflective, can only be solved by having previously completed forward simulations and taking some AI correlation algorithms to determine heating scenario. The latter case is an actual requirement for real-time control of the engineering system. For this problem FPGA and accelerators are a perfect technology for real use cases.

8.3 Execution profile

The accurate assessment of a synthetic image requires three principal numerical simulations whose execution needs to be performed in sequence. The field line tracing the solver's output is an input to the thermal model and the thermal model's output is an input to the optical simulation which in the end returns a result in the form of a synthetic measurement, i.e. an image. The simple scheme is explained in Figure 22.

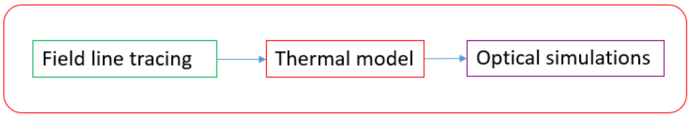

Figure 22: Sequential execution of codes for digital twin

For field line tracing and thermal modelling, the geometry components are independent of each other and they are easily parallelisable. For simpler first wall geometries (components with simple shaping with no need for much detail, components which are not actively cooled, etc.) the assessment of heat fluxes and temperatures is expected to be delivered faster than more complex shapes such as actively cooled components with complex shapes such as different casellations etc. However, for optical simulation a full temperature distribution is needed in order to give accurate results. Along with large input and output data, there is a significant challenge for appropriate coordination of the computational resources (different hardware and queue system).

For simulations and post-processing, it is planned to first test the VEGA system through development access mode. As mentioned above, the described modelling work has been partially done at the University of Ljubljana even though the executed simulations were limited and porting to HPC is still necessary. Therefore, UL will follow the classical task path: (1) Getting access to a petascale system (VEGA), (2) deploying code and data to the system, (3) preparing and running the simulations by porting the workflow and moving it to larger scales, (4) evaluating and validating the results.

After that UL will approach significant workflow improvements to reduce time to solution and work towards the solution of the real time problem.

8.4 Workflow

The workflow is depicted in Figure 23.

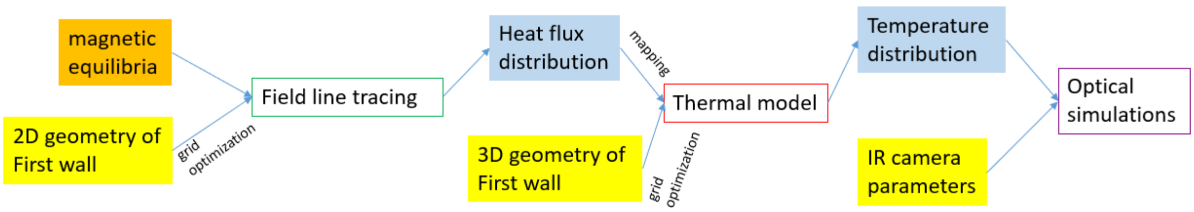

Figure 23: Workflow components for the simulation of IR camera

In more detail the workflow components are explained below:

- Magnetic equilibria will be supplied by UL partners (ITER, CEA).
- 2D geometry of the first wall is given in STEP file formats and then meshed. Grid density will be assessed after each run and compared with results to find a compromise between accuracy and execution time.

- Field line tracing is assessed taking into the account the magnetic equilibria and 2D geometry.
- Heat flux distribution on the first wall is provided and mapped on the simplified 3D geometry of the tokamak first wall.
- Thermal model assesses the temperature distribution on the first wall.
- Temperature is given as input to the optical camera simulation.
- Based on camera parameters and defined reflection and radiation on the wall (a function of temperature) optical simulation is performed which returns synthetic camera image.

Various aspects of the individual components are summarised in the following:

Field line tracing

- Magnetic equilibria, currently stored in an ASCII file standard (within the fusion community) eqdsk file format. For the need of exascale computing the data is expected to be stored instead in binary format such as HDF5 or netCDF.
- 2D geometry of the first wall is given as CAD model in STEP file format. Here the grid generation can in principle be accomplished prior to the simulation and stored as input data. Later the grid can be optimised for the new set of simulations.
- Field line tracing is a RKF4 algorithm to integrate magnetic field and assess the streamline movement of particle. The intersection between the streamline and 2D geometry (triangles) is calculated in each step (for 9000 streamlines and 450000 triangles, the execution time is 123s running on 36 cores).

Thermal model

- Heat flux is mapped from the 2D first wall to the 3D model to solve heat equation using Nearest Neighbour Interpolation.
- The structure is a 3D mesh (tetrahedrons). Serial execution of grid generation (millions of elements).
- The transient heat equation is solved to obtain the temperature distribution. Serial execution (currently 6 million elements, 3 timesteps → computational time ~2 minutes).

Optical simulations

- Camera parameters are loaded (Field of View, camera resolution and pixel size)
- Temperature distribution is passed from the thermal model → reflection coefficients are assessed from temperatures and used in the simulation.
- Backwards ray-tracing is performed. Rays are launched from each pixel randomly into space. Intersection with the first wall is calculated and energy of the ray is estimated which then contributes to the total radiance on the pixel.

To summarise, three types of simulations with different configurations and input parameters; various output files and large datasets need to be properly stored and performed. This requires automated algorithms for storage, computation and high efficiency without the need for user to interact with simulation.

8.5 Timeline

The development of the workflow will be first performed with a basic first wall component for ITER temporary limiter which is currently underway. The technical developments will be performed in the following steps:

- Definition and requirements for the use case and workflow (M6).
- Definition of a setup for small scale runs (basic first wall component) (M6).
- Establish necessary workflow building blocks (M12).
- Further optimisation of the workflow and update of simulation tools (M24).
- Definition of test setup for larger scale runs (M24).
- Perform the test using full geometry of tokamak for larger scale runs and check accuracy (M30).
- Perform large scale runs for selected tokamak machines (ITER, WEST) (M48).

8.6 Gantt Chart

		Start Month	End Month	Duration (Months)
FA-1	Engineering design and digital twin of the first wall of a tokamak fusion reactor			
FA-1.1	Definition and requirements for the use case and workflow	1	6	6
FA-1.2	Definition of a setup for small scale runs (basic first wall component)	1	6	6
FA-1.3	Establish necessary workflow building blocks	3	12	10
FA-1.4	Further optimisation of the workflow and update of simulation tools	7	24	18
FA-1.5	Definition of a test setup for large scale runs	7	24	18
FA-1.6	Perform the test using full geometry of tokamak for larger scale runs and check accuracy	25	42	18
FA-1.7	Perform large scale runs for selected tokamak machines (ITER and WEST)	37	48	12

Figure 24: FA-1 Gantt Chart

8.7 Dependencies

The workflow has to be developed at UL with the input data supplied by ITER and CEA

8.8 Success criteria

For this use case the following criteria will be used to verify the success:

- Development of a workflow which outputs synthetic image without user interaction.
- Efficient usage of HPC hardware during the workflow execution (in terms of computational resources and data storage).
- Identification of hot spots on the first wall based on large scale simulation.

- Successful analysis of synthetic images to predict thermal loading on the first wall for a given magnetic equilibria.

9 Conclusion

For each challenging use case, the workflow development has been presented together with a timeline, which allows to monitor the individual progress. At this early stage of the project, i.e., M6, all tasks connected to the UCs and FA are in accordance with the specified timelines.

10 References

[1] A. Mehra, "CISION PR Newswire," MarketsandMarkets INC., 11 June 2021. [Online]. Available: https://www.prnewswire.com/news-releases/unmanned-aerial-vehicle-uav-market-worth-58-4-billion-by-2026--exclusive-report-by-marketsandmarkets-301310782.html. [Accessed 14 June 2023].

[2] Z. Z. Ning, Experimental investigations on the aerodynamic and aeroacoustic characteristics of small UAS propellers, Ph.D. thesis, Iowa State University, 2018.

[3] Y. Delorme, R. Stanly, S. Frankel and D. Greenblatt, "Application of actuator line model for large eddy simulation of rotor noise control," *Aerosp. Sci. Technol.,* vol. 108, pp. 1-12, 2021.

[4] J. Nathanael, C.-H. Wang and K. Low, "Numerical studies on modelling the near- and far-field wake vortex of a quadrotor in forward flight," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering,* vol. 236, no. 6, p. 1166, 2022.

[5] A. Hüppe, J. Grabinger, M. Kaltenbacher, A. Reppenhagen, G. Dutzler and W. Kühnel, "A non-conforming finite element method for computational aeroacoustics in rotating systems," *20th AIAA/CEAS Aeroacoustics Conference,* 2014.

[6] F. Liang, R. Shi and Q. Mo, "A split-and-merge approach for singular value decomposition of large-scale matrices," *Statistics And Its Interface*, vol. 9, p. 453–459, 2016.

[7] J. Ffowcs Williams and D. Hawkings, "Sound generation by turbulence and surfaces in arbitrary motion," *Mathematical and Physical Sciences,* vol. 264, no. Philosophical Transactions of the Royal Society of London. Series A, pp. 321-342, 1969.

[8] F. Farassat, "Linear acoustic formulas for calculation of Rotating Blade Noise.," *AIAA Journal,* vol. 19, no. 9, p. 1122–1130, 1981.

[9] S. Lawrence, Turbulence and the dynamics of coherent structures. Parts I-III., Q Appl Math, 46(3), 561., 1987.

[10] Z. Wang, B. McBee and T. Iliescu, "Approximate partitioned method of snapshots for POD," *Journal of Computational and Applied Mathematics,* vol. 307, p. 374–384, 2016.

[11] M. Quadrio, P. Ricco and C. Viotti, "Streamwise-traveling waves of spanwise wall velocity for turbulent drag reduction," *Journal of Fluid Mechanics,* vol. 627, pp. 161-178, 2009.

[12] I. Marusic, D. Chandran and A. Rouhi, "An energy-efficient pathway to turbulent drag reduction," *Nat Commun,* vol. 12, p. 5805, 2021.

[13] Smits.USA Patent 0044837, 2023.

[14] M. Quadrio, A. Chiarini, J. Banchetti, D. Gatti, A. Memmolo and S. Pirozzoli, "Drag reduction on a transonic airfoil," *Journal of Fluid Mechanics,* vol. 942, p. R2, 2022.

[15] S. Pirozzoli, "Stabilized non-dissipative approximations of Euler equations in generalized curvlinear coordinates," *Journal of Computational Physics,* vol. 230, no. 8, pp. 2997-3014, 2011.