H2020-INFRAEDI-2018-2020

The European Centre of Excellence for Engineering Applications

Project Number: 823691

D2.4

Report on Reference Applications Outcomes

The EXCELLERAT project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 823691

Workpackage:	WP2	Application Development	
Author(s):	Mark Sawyer		UEDIN
	David Scott		UEDIN
	Niclas Jansson		KTH
	Chiara Latini		CINECA
	Adam Peplinski		KTH
	Gabriel Staffelbach		CERFACS
	Michael Wagner		DLR
	Daniel Mira Martinez		BSC
	Anne-Bernard Bedouet		Sicos
Approved by	Executive Centre Management		
Reviewer	Francesco Salvadore		CINECA
Reviewer	Christian Gscheidle		Fraunhofer
Dissemination Level	PUBLIC		

Date	Author	Comments	Version	Status
20 May 2022	Mark Sawyer (ed)	Draft following QA review	1.0	Draft
24 May 2022	Mark Sawyer (ed)	Approved version	2.0	Approved

List of abbreviations

AMR	Adaptive Mesh Refinement
ARC	Analytically Reduced Chemistry
CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
CG	Conjugate Gradient
FEM	Finite Element Method
GLL	Gauss–Lobatto–Legendre
gPCE	Generalized Polynomial Chaos Expansion
GPR	Gaussian Process Regression
GPU	Graphics Processing Unit
HPC	High Performance Computing
ISV	Independent Software Vendor
LES	Large Eddy Simulation
MPI	Message Passing Interface
PGAS	Partitioned Global Address Space
QoI	Quantity of Interest
RBF	Radial Basis Functions
RDO	Robust Optimisation Design
SEM	Spectral Element Method
UHBR	Ultra-high-bypass Ratio
UQ	Uncertainty Quantification

Executive Summary

The EXCELLERAT project was established in December 2018 with the objective of creating a European Centre of Excellence (CoE) for Engineering Applications in order to meet the challenges of the fourth wave of industrialisation and to support European engineering companies in their use of HPC and HPDA, thus increasing European industrial competitiveness.

The CoE was focused on a set of engineering software applications which were chosen to drive the technical developments necessary to progress the codes towards Exascale. A further objective of the project was to develop a service-oriented methodology to transfer knowledge and expertise to the wider engineering HPC community, while at the same time providing a potential route to sustainability for the CoE. This document reports on the work done directly on the chosen applications to develop them for Exascale.

The application codes selected were:

- **Nek5000:** Open source spectral code that solves the incompressible Navier-Stokes with a number of additional physics (heat transfer, magneto-hydrodynamics, low Mach number, electrostatics)
- **ALYA:** HPC Multiphysics code that uses Finite Element Method for large scale problems with complex geometries
- **AVBP:** Finite element/volume 3D compressible flow solver using unstructured grids and capable of Large Eddy Simulation (LES)
- **TPLS:** Open-source solver for the incompressible Navier-Stokes equations for a two fluid flow, using the level-set method to identify interfaces
- **FEniCS:** Open-source high-level problem solving environment for automated solution of PDEs by the finite element method
- **CODA:** CFD solver for Aircraft design. Specific requirements for Exascale e.g. multi-level-parallelization and asynchronous execution.

The key achievements were

- Nek5000:
    - Incorporation of adaptive mesh refinement
    - Finalised Catalyst interface for in-situ visualisation and uncertainty quantification estimation
    - Integration of Nek5000 with SENSEI and Vistle
- ALYA:
    - Developed fully parallel workflow with load balancing at node and system level
    - Ported code to GPUs
    - Incorporated adaptive mesh refinement
    - Tested the developments with industrially significant use cases
- AVBP:
    - Used automatic mesh refinement to improve computational efficiency
    - Ported code to GPU and demonstrated on 1024 GPU tier 1 system
    - Tested in use cases involving confined explosion and combustion chamber of aircraft engine
- TPLS:
    - Development of native GPU version
    - Using HDF5 to improve I/O scalability

- Integrating a new method for identifying interface between fluids
- Added temperature and proportion of vapour in the gaseous phase as degrees of freedom

- FEniCS
    - Advancements in mesh generation and refinement
    - Testing using the DrivAER standard geometry
    - Solved problem of contact between wheels and ground within the mesh
- CODA:
    - CODA and the surrounding FlowSimulator framework were ported, tested and evaluated on the AMD Naples and the Intel Cascade Lake architecture
    - Full incorporation of Sparse Linear Systems Solver (Spliss)
    - Improvements to Spliss for CPU and GPU systems carried out and evaluated
    - Extension of FlowSimulator framework

Table of Contents

Table of Figures

1 EXCELLERAT CoE Background

1.1 Motivation

The European engineering industry consists of 130 000 companies of diverse sizes. Overall, these companies employ over 10.3 million people, with high levels of qualifications and skills. Together they generate an annual output of around EUR 1,840 billion and about 1/3 of all exports from the EU. The EXCELLERAT CoE was set up to address the challenges of the fourth wave of industrialisation and to support European engineering companies in their use of HPC and HPDA, thus increasing European industrial competitiveness.

1.2 Methodology

The CoE was focussed on a set of engineering software applications which were chosen to drive the technical developments necessary to progress the codes towards Exascale. A further objective of the project was to develop a service-oriented methodology to transfer knowledge and expertise to the wider engineering HPC community, while at the same time providing a potential route to sustainability for the CoE.

The project was organised into a series of workpackages which dealt with various aspects of the progression towards Exascale:

- WP1: Management
- WP2: Application Development
- WP3: Driving Exa-HPC Methodologies and Technologies
- WP4: Enhanced Services
- WP5: Centre Implementation
- WP6: Market Context and Sustainability
- WP7: Awareness, Impact Creation and Outreach

This document focuses on WP2, the main task of which was to drive the optimization and enhancement of the chosen reference applications. These codes will act as exemplars of the assistance that EXCELLERAT can offer. The work package will track progress on the reference applications and highlight shared challenges (such as mesh generation). The outcomes of the development work will be used to provide technical material for dissemination to promote EXCELLERAT (via use cases, solution scenarios and success stories). Lastly the work package helped to define the services that EXCELLERAT can offer to the wider HPC engineering community, both from application user and application developer perspectives.

2 Summary of Reference Applications and Use Cases

2.1 Nek5000

Nek5000 is an academic high-order CFD solver based on Spectral Element Method (SEM) used for number of different flow cases including nuclear reactor thermal-hydraulics. An important improvement to the code was implementation of Adaptive Mesh Refinement (AMR) algorithm, that allows to control computational error during the simulation and to reduce it at minimal cost by proper mesh adjustment. This code is currently extended for fully three-dimensional flows making it a robust solver capable of solving industrially relevant problems. Number of different aspects of numerical modelling are considered here starting from mesh generation, solver parallel performance, through solution reliability (Uncertainty Quantification) and ending with data visualisation.

Within the framework of EXCELLERAT two use cases have been selected. Use case C1U1 involves flow around three-dimensional wing tip, which is a relatively complex test case relevant for number of industrial areas like aeronautics, automotive or bio engineering. However, despite its environmental and economical importance, there is lack of high-fidelity numerical data for high-Reynolds-number turbulent flows. This case is as well a good platform for testing all developed tools. The second use case C1U3 is a simplified rotor in a rotating reference frame which would enable research in the direction of detailed flow field and analyses of helicopters.

2.2 ALYA

Alya is a parallel multi-physics/multi-scale simulation code featuring hybrid parallelization and has been developed to run efficiently in supercomputing environments. It is part of the PRACE Benchmark Suite and has been tested showing excellent parallel performance on most European supercomputer platforms. Alya is developed by BSC and has many users worldwide (Ciemat, NCSA, University of Oxford, Surfsara and TUb Berlin among others). The main applications fields include aeronautical, aerospace, automotive, bioengineering, wind energy or power generation. Alya has also been applied in the industrial context within collaborations with companies such as EM Combustion, Seat, Repsol, Medtronic, Siemens AG and General Electric.

Alya is applied in three use-cases. The first use-case (C2U1) deals with the prediction of emissions like pollutants e.g. NOx and soot using advanced numerical simulations with detailed chemical kinetics in combustion applications, which nowadays is leading the design process of next generation of engines for road transportation and aviation. The second use-case (C2U2) targets the investigation of active flow control for full aircraft aerodynamics, which is a very relevant topic for the development of the new ultra-high bypass ratio (UHBR) engines. The third use-case (C2U3) is focused on the modelling of the mechanical structure of transportation systems with emphasis on the prediction of load and stresses as well as fatigue and fracture. The three use-cases have been selected as they correspond to fundamental challenges of modelling and simulation codes in the aeronautical and aerospace sector.

2.3 AVBP

AVBP is a compressible Navier Stokes solver for 3D unstructured meshes capable of simulating two-phase flow combustion. The prediction of these unsteady turbulent flows is based on the Large Eddy Simulation (LES) approach where small scales are modelled and large scales are resolved. Arrhenius reduced law and analytically reduced chemistry models allow investigating combustion for complex configurations. AVBP is based on a cell-vertex finite volume approximation, leveraging a Lax-Wendroff or a finite element type low-dissipation Taylor-Galerkin scheme in combination with a linear-preserving artificial viscosity models. Two-phase flows are accounted for using the Eulerian or Lagrangian approaches.

Within the framework of EXCELLERAT two use cases have been showcased: C3U1 Safety application; and C3U2 Pollutant prediction using LES for aeronautical applications;

2.4 TPLS

TPLS (Two-Phase Level Set) is a freely available finite difference CFD code which solves the Navier-Stokes equations for an incompressible two-phase flow using a level set method. The level set method enables sharp changes in interfacial topology and is therefore ideally suited to gaining an understanding of problems of interest in many chemical engineering applications (in particular in the oil and gas industries) as well as looking at the implications of the inherent instability of the interface in a two-phase system.

TPLS has been developed by Prashant Valluri of the Institute of Materials and Processes, School of Engineering at The University of Edinburgh and Lennon Ó Náraigh of the School of Mathematical Sciences, University College Dublin. TPLS uses an ultra-high 3D Direct Numerical Simulation approach combined with the Level-Set method for tracking the developing interface between phases. TPLS employs PETSc for its parallelisation, and users can choose to use PETSc or hand-written solvers at runtime.

The TPLS use-case (C4U1) is a falling liquid film with a liquid/gas interface. This is a common scenario in many chemical engineering applications, and understanding the behaviour of the interface is of great practical importance.

2.5 FEniCS

FEniCS is a high-level problem-solving environment for automated solution of partial differential equations (PDEs) by the finite element method. To manage the complexity of multiphysics problems FEniCS takes the weak form of a PDE as input in a near mathematical notation and automatically generates low-level source code, abstracting away implementation details and HPC concepts from domain scientists. FEniCS has extensively been used in academia for goal driven *a posteriori* adjoint based error estimation, which makes it possible to automate the assessment of uncertainties in computed solutions and drive an adaptive process, resulting in an optimal mesh for given outputs of interests e.g., drag, lift or acoustic noise. By utilizing error estimation and adaptivity, extended workflows iteration between complex mesh generation, simulation and evaluation of results can be fully automated. Within EXCELLERAT our vision is to complete the workflow with shape optimization for a given quantity of interested e.g. drag. Our aim is to use *a posteriori* error estimation to drive both mesh adaption and CAD morphing in an iterative process to produce an optimal design for a given output of interests. The use case C5U1 is studying the external flow for the DrivAer model

2.6 CODA

At the German Aerospace Center (DLR), CFD codes have been developed for decades, several of which are in regular industrial use. One of them is the DLR *TAU* code, which is used in the

European aircraft industry, research organizations and academia since more than 15 years, for instance, for the Airbus A380 and A350 wing design.

In 2012 DLR decided on the development of a new flexible unstructured CFD solver called *FLUCS*. This gave the opportunity to design a modern, comprehensive HPC concept from scratch as well as focusing on strong fully implicit schemes for improved algorithmic efficiency, higher-order spatial discretization, and improved integration into Python-based multi-disciplinary process chains.

Though FLUCS had been started as a DLR activity, it has become part of a larger development that is driven by Airbus, the French aerospace lab ONERA, and DLR. All three parties reached an agreement based on the industrial needs and constraints and decided to pursue a joint effort. In 2019, the common framework and architecture of the joint development of the CFD solver based on FLUCS was called *CODA* to reflect the new collaboration and the involvement of all three partners.

2.7 Summary of use cases

C1U1	Automotive	Nek5000	Automated design cycle and error control of air intake systems of engines
C1U3	Aerospace	Nek5000	High fidelity simulation of rotating parts
C2U1	Automotive/ Aerospace	Alya	Emission prediction of internal combustion (IC) and gas turbine (GT) engines
C2U2	Aerospace	Alya	Active flow control of aircraft aerodynamics including synthetic jet actuators
C2U3	Transport systems	Alya	Coupled simulation of fluid and structure mechanics for fatigue and fracture
C3U1	Aerospace/Energy	AVBP	Combustion instabilities and emission prediction
C3U2	Safety applications	AVBP	Explosion in confined spaces
C4U1	Chemical industry	TPLS	Two phase bidirectional flow
C5U1	Aerospace/ Automotive	FEniCS	Adjoint optimization in external aerodynamics shape optimization
C6U1	Aerospace	CODA	Design process and simulation of full equipped aero planes
C6U2	Aerospace	CODA	CFD coupling with computational structural mechanics including elastic effect

3 Progress and Achievements

3.1 Nek5000

3.1.1 Technical work and achievements

Nek5000 related work was focusing on an adaptive mesh refinement (AMR) framework implemented in this solver. A number of the AMR components in Nek5000 were developed within previous EU projects CRESTA and ExaFLOW. The starting point for EXCELLERAT was a fully functional workflow for two-dimensional and simple two-dimensional extruded flows, that had to be extended to the general three-dimensional flows including improvements of grid partitioning and pressure preconditioning. The goal was to make an AMR version of Nek5000 a robust solver capable of simulating industrially relevant flows at Exascale.

Adaptive Mesh Refinement (AMR)

During EXCELLERAT we worked on multiple aspects of AMR implementations for SEM covering the whole simulation workflow. We started with the pre-processing step concentrating on strategies for hex-based high order meshing and proper connectivity information for general, complex three-dimensional meshes. The next aspect was a proper representation of the surfaces and we implemented in Nek5000 a projection tool shifting the boundary points to the surfaces represented by the analytical functions or set of splines. We experimented as well with different graph partitioners (ParMETIS and PARRSB), communication kernels and possible improvements of pressure preconditioners. We integrated our solver as well with an uncertainty quantification (UQ) tool UQit and run it in a post-processing mode. Finally, we investigated in-situ visualization combined with AMR and an effect of nonconforming interfaces on the turbulent statistics. Our implementation of AMR framework in Nek5000 turned out to be very successful, allowing us to perform previously unaffordable high-fidelity simulations of a NACA0012 aerofoil with a rounded wing tip (C1U1) and a simplified rotor in the rotating reference frame (C1U3).

During the third year of the project, we focused on the parallel performance of the AMR solver carrying out strong scaling tests on Dardel (PDC, Sweden) and LUMI (CSC, Finland). The plot shows an initial super-linear scaling caused by improved cache usage which is followed by performance degradation mostly due to increasing cost of a coarse grid solver.

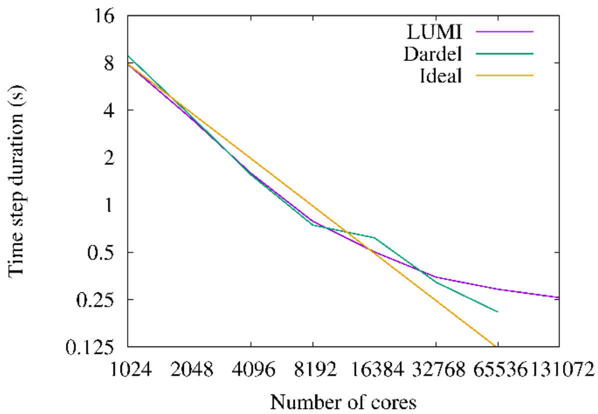


Figure 1 Strong scaling results for nonconforming Nek5000 solver performed on Dardel and Lumi using C1U3 test case.

Uncertainty Quantification

We finalised a Catalyst interface for in-situ visualisation and UQ estimation. This task was performed in collaboration with WP4 and it allows us to estimate during the run the uncertainties of the time-averaged quantities. We continued development of pressure preconditioners for nonconforming meshes and generation of high order meshes for a relatively complex drone rotor using gmsh fortran interface. In addition, there was work done on integration of AMR Nek5000 with SENSEI and Vistle (in collaboration with WP4). We performed the set of the full-scale high fidelity runs of both use cases collecting turbulence statistics and analysing complex flow dynamics. The short description of the preliminary results of C1U1 follows.

Three dimensionality effects in general and wingtip vortices in particular are known to impact the aerodynamic efficiency of wings quite significantly. This part of the project focuses on understanding the impact of these vortices on flow physics, including the near wall turbulence, the wake, and lift and drag coefficients. Six different configurations are considered, as three angles of attacks (0, 5, 10 degrees) and two geometries (infinite-span wing and a rounded wingtip geometry). The flow physics in the infinite-span case is less complex and thus the production runs are performed on grids of 500 million grid points or less. The rounded wingtip cases require a finer grid and a larger domain, requiring 951 million grid points for α=0° (without a wingtip vortex), 1.58 billion grid points for α=5° (with a wingtip vortex), and an estimated 2 billion grid points for α=10° with a strong wingtip vortex (not at production yet).

Our analysis of lift and drag shows that the lift-to-drag ratio of the wing drops from 24.4 for the periodic (infinite-space) case at α=5° to 7.4 for a rounded wingtip at the same angle of attack, which is caused by a decrease in lift coefficient from 0.49 to 0.19 as well as an increase in drag coefficient from 0.020 to 0.025.

Our results also analyse the impact of the presence of wingtip vortices on flow physics and turbulence. Of particular interest is the relaminarization of the turbulent boundary layer near the tip, as a result of the rotation and pressure gradient imposed by the wingtip vortex, and the extra turbulent kinetic energy (TKE) production both at the wingtip and the near wake due to the presence of wingtip vortex see Figure 2

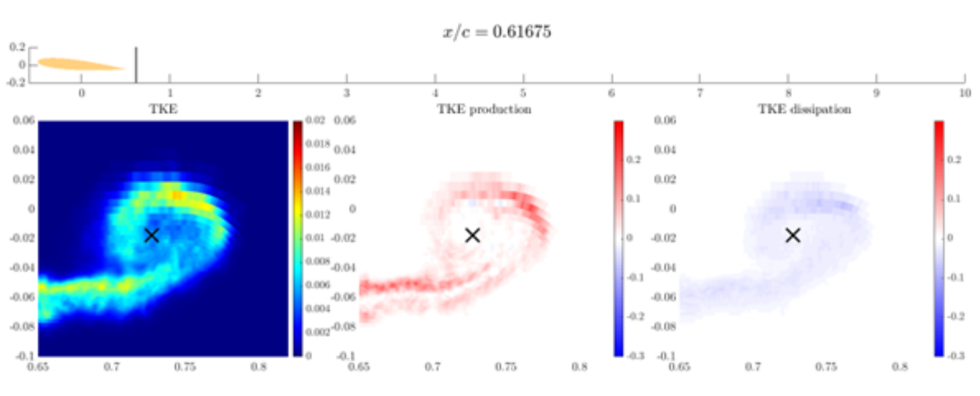


Figure 2 Turbulent kinetic energy (TKE) and its production and dissipation at a location near the trailing edge of a rounded wingtip case at 5 degrees angle of attack. The results are from the production run on 1.58 billion grid points.

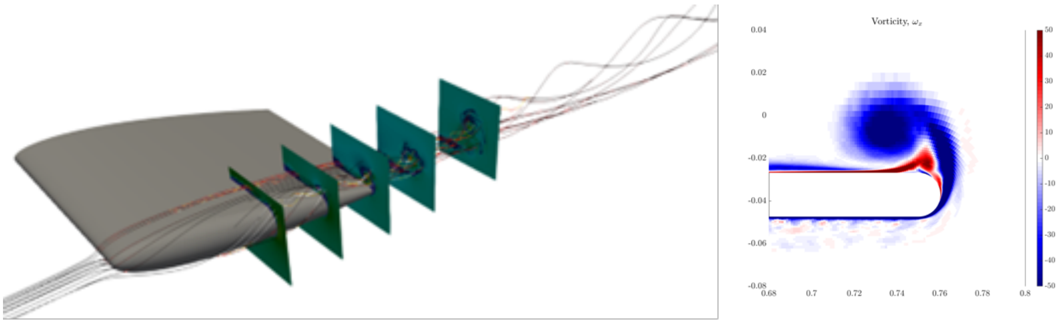


Figure 3 . (left) velocity streamlines colored by vorticity at an angle of attack of 10 degrees on a preliminary grid and (right) mean streamwise vorticity for α=5º showing the presence of secondary and tertiary vortices beside the primary wingtip vortex performed on a pre-production grid with 500 million grid points.

3.1.2 Interaction with other work packages

The result of collaboration with WP3 and WP4 are the important contributions to the core code of AMR Nek5000 solver. The most crucial one developed during the last year of the project is the in-situ UQ analysis done with WP4 and Sensei instrumentation allowing to use of Vistle with the code. There was an interaction with WP3 related to pressure preconditioners and grid partitioning and this effort will be continued after the project end.

3.1.3 Future work

After the project end, we will continue work on high order hex-based meshing using gmsh fortran interface, development of the efficient pressure preconditioners for nonconforming meshes, "hardware aware" mesh partitioners (e.g., TreePart) and communication kernels. We will perform studies as well of the flow around the drone rotor with E63 aerofoil.

Some simplifications have been adopted for use-case C1U3, and the present rotor mesh is not the exact counterpart of the experimental geometry. Currently we perform studies based on a simplified (small) four blade rotor, and after the project end, we will continue with the drone rotor with E63 aerofoil.

3.2 Alya

Alya is a parallel multi-physics / multi-scale simulation code developed by the CASE department from BSC, which has been designed to run efficiently in supercomputing environments, with optimal performance on heterogeneous architectures. Alya is one of the two engineering codes (CFD) of the Prace Benchmark Suite and has been tested showing excellent parallel performance on most European supercomputer platforms. It was specially designed for massively parallel supercomputers, and the parallelization is *hybrid*. A substructuring technique with MPI as the message passing library is used for distributed memory supercomputers. At the node level, both *loop* and *task* parallelisms are considered, using both OpenMP and OmpSs. *Dynamic load balance* techniques have been introduced to better exploit computational resources at the node level. Accelerators like GPUs are also exploited at the iterative solver and assembly levels to further enhance the performance of the code. The numerical strategy is based on the Finite Element Method using linear and quadratic elements, and currently being extended to spectral elements.

In EXCELLERAT, the main activities related to Alya have been to extend the parallel performance of the code for extreme-scale applications for current and emerging HPC technologies through the setup of two use-cases that represents fundamental challenges in the aeronautical sector. The cases are shortly described here:

1. The first use-case corresponds to the prediction of emissions in combustion applications, which nowadays is leading the design process of next generation of engines for road transportation and aviation.
2. The second use-case targets the investigation of active flow control for full aircraft aerodynamics, which is a very relevant topic for the development of the new ultra-high bypass ratio engines.

During the first two years of the project, the activities were focused on solving the identified bottlenecks found when running the two aforementioned applications. It includes the development of a fully parallel workflow, load balancing at node and system level, porting to GPUs and emerging technologies and developing an adaptive mesh refinement (AMR) capability to increase the efficiency of the simulations.

3.2.1 Achievements

The achievements of the third year of the project in terms of applications are related to the use-cases of combustion (C2U1) and aerodynamics (C2U2). Combustion simulations are focused on two fundamental problems related to the transportation sector, high-pressure spray flames at internal combustion (IC) engines-relevant conditions and swirl-stabilized flames of an aeroengine-model combustor at atmospheric pressure, while aerodynamics are focused on full-aircraft simulations.

As a continuation activity from the second year after validation, the focus was given to investigate the combustion characteristics of high-pressure spray flames of renewable fuels. A family of novel synthetic fuels, so called Oxymethylene Dimethyl Ethers (OMEx), have shown high potential for their use as liquid fuels for low-carbon transport applications due to their capacity to avoid soot formation among other pollutants [6]. Intense research activities have been carried out spanning from fundamental works to direct application in engines [6]. The use of numerical simulations can help to better understand the performance of these fuels at engine-relevant conditions, though it brings fundamental challenges due to the need for detailed descriptions of both the chemical and flow processes [7]. The modelling approach employed to describe the conditions of the high-pressure spray flame is based on the use of an Eulerian-Lagrangian framework. The gas is represented by the Eulerian phase, while the liquid phase is described by the Lagrangian particles tracking (LPT) with heat and mass transfer models. The gas phase is described by the use of filtered equations in the Large Eddy Simulations (LES) framework and include continuity, momentum and enthalpy. The investigation includes the analysis of OMEx spray flame at ECN Spray A nominal conditions using a reduced mechanism developed in-house using Path Flux Analysis (PFA) technique, which is compared with a detailed and reduced mechanism from the literature [7]. The spray A is one of the reference cases under investigation of the Engine Combustion Network (ECN) (https://ecn.sandia.gov). Sample results of the spray flame for the different reaction mechanisms are given in Figure 4.

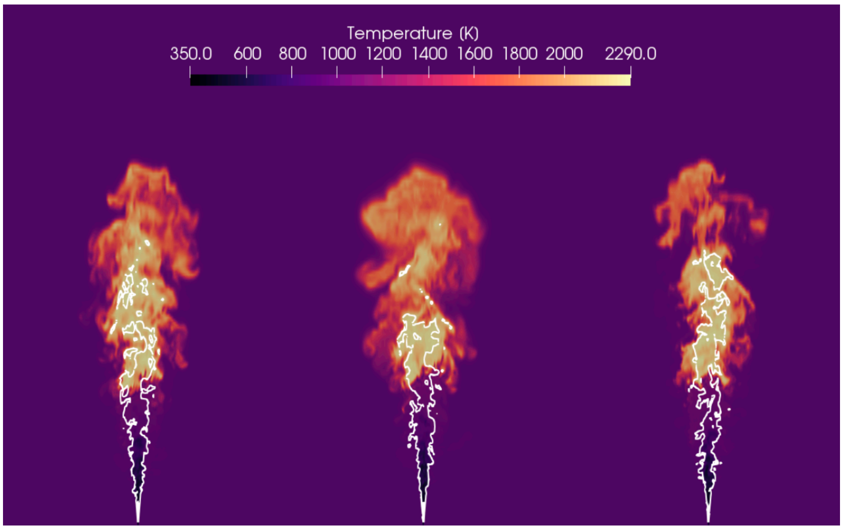


Figure 4 Temperature distribution of OMEx spray flame under ECN Spray A nominal conditions using three reaction mechanisms detailed (left), BSC 168 (center), Aachen (right) at 1.875 ms.

The second application case is the reacting flow field of a turbulent swirl-stabilized burner that is used as a demonstrator of a load balancing strategy for reaction rate evaluation and chemistry integration in reacting flow simulations developed in WP3. The large disparity in scales during combustion introduces stiffness in the numerical integration of the PDEs and generates load imbalance during the parallel execution. The strategy is based on the use of the DLB library to redistribute the computing resources at node level, lending additional CPU-cores to higher loaded MPI processes. This approach does not require explicit data transfer and is activated automatically at runtime. Two chemistry descriptions, detailed and reduced, are evaluated for this flame. A description of the mesh and sample results of this are shown in Figure 5.

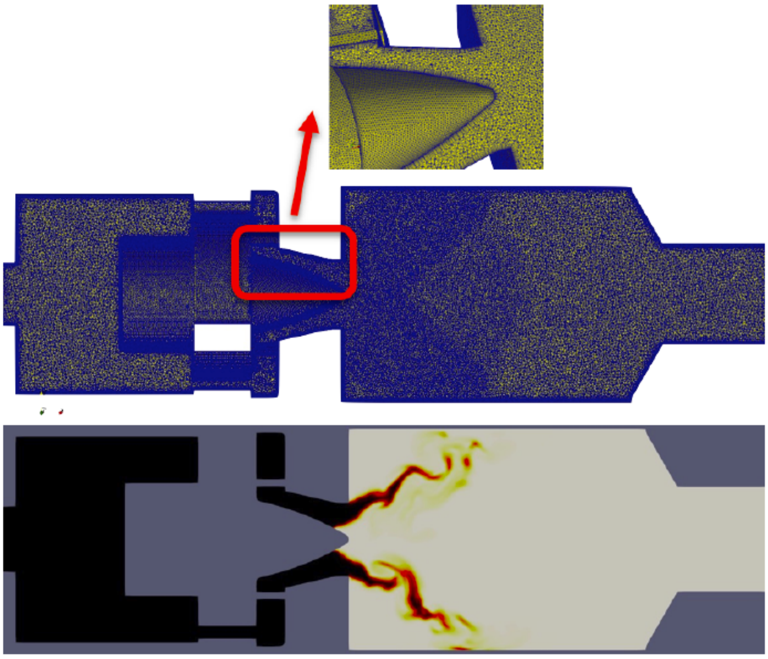


Figure 5 Swirl-stabilized turbulent premixed flame using the detailed reaction mechanism: mesh resolution (top) and temperature (bottom) using a 24 million cells in a hybrid mesh.

The developments related to the use-case C2U2 on external aerodynamics and active flow control during Year 3 were focused on the evaluation of the aerodynamic performance of active flow control on wings using synthetic jets with zero net-mass flow.

The aerodynamic performance of an aircraft wing is significantly affected by the interaction of the flow with the nacelle and brackets, as well as by boundary layer flow separation, specially at the high angles of attack (AoA) typically encountered during take-off and landing operations. At high angles of attack, the loss of momentum across the boundary layer leads to the flow separation and eventually the wing stall. In such situations, adding momentum to the flow might prevent the flow detachment, thus increasing the aerodynamic performance of the wing. In the present work, the use of synthetic jets for AFC of the wing boundary layer of a full aircraft in stall is explored by means of wall-modeled large-eddy simulations (WMLES). Slotted synthetic jets are located at both the main and slat of the JAXA Standard Model (JSM). This geometry was the subject of study in the recent 3rd AIAA CFD High Lift Prediction Workshop [8], where RANS-based methodologies, challenged to predict the onset of stall and maximum lift, had difficulties to predict the lift coefficient at high angles of attack AoA. A landing configuration with the high-lift devices (slat and flap) deployed in the absence of nacelle/pylon is considered at Re_c = 1.93e6 and Mach number M_0 = 0.15. Here, the Reynolds number is defined in terms of the free stream velocity U_0 and the wing mean chord C. Ten different actuation strategies are explored to assess the impact of the different control parameters in stall conditions. The computational domain is shown in Figure 6, which shows the position of the aircraft and the wing (Figure 6 left), and the position of the synthetic jets on the wing (Figure 6 right).

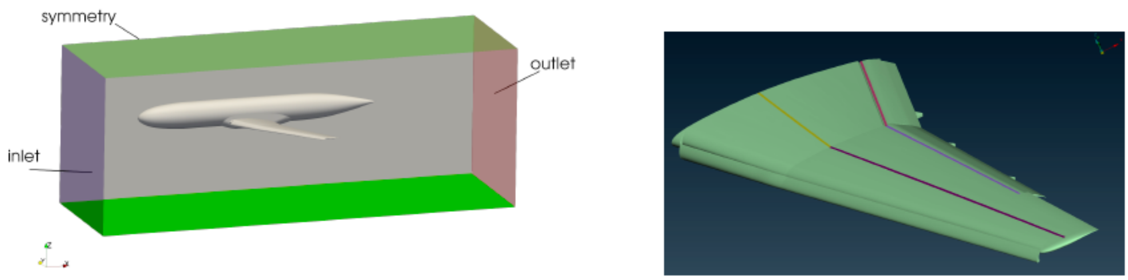


Figure 6 Computational domain (not-to-scale) (left) and location of the slotted synthetic jets (main at x=C = 0:5 and flap at x=C = 0:1) (right).

To understand the effect of the actuation on the main wing, we inspect the flow patterns in the presence and absence of actuation. The visualizations suggest that the main losses in the lift force are due to the formation of large separated zones. In the uncontrolled case, a large recirculation region occurs behind the flap, which lifts up the boundary layer in the rear end of the main wing right after the flow changes direction as induced by the flap deployment. When the actuation in the main wing is suppressed, the variations in the lift is very small, and the jet located at the flap can barely act on the recirculation zone. The actuation strategies, which approximates the best the ideal linear behavior include the large recirculation zone behind the flap is almost suppressed and the streamlines evolve parallel to each other behind the flap (see Figure 7).

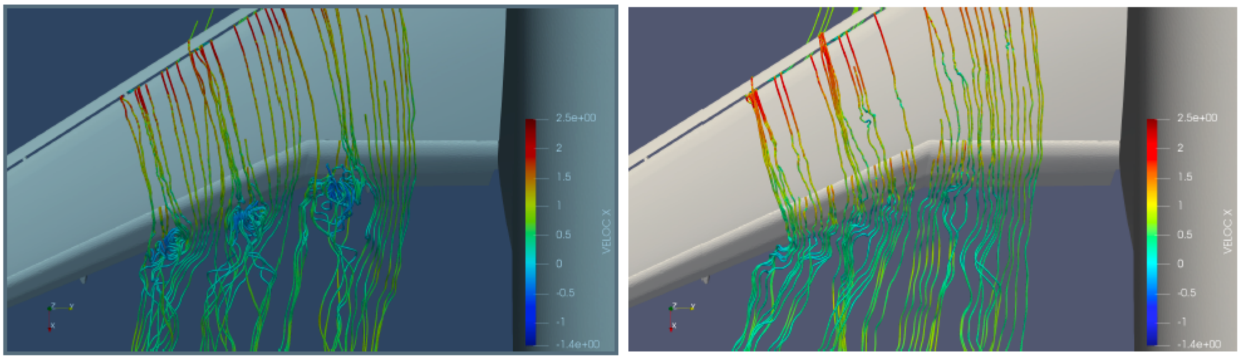


Figure 7 Instantaneous vortical structures identified by the Q-criterion.

3.2.2 Interaction with other work packages

The main interaction with other WPs is by the HPC developments conducted in WP3, which came as requirements on the code development roadmap. The main aspects that enabled to run the simulations of the two use cases (C2U1 and C2U2) in Year 3 correspond to the co-execution for heterogenous architectures, and the development of an Eulerian-Lagrangian framework to conduct spray flames. The first data transfer and dispatching demonstrator was achieved and is reported in D3.2, while the *in situ* visualization demonstrator is reported in D4.2.

3.2.3 Future work

Considering the achievements conducted in EXCELLERAT for the two applications, a robust and reliable platform for high-fidelity simulations has been developed and optimized. However, the needs of the aeronautical industry to obtain detailed information of the relevant parameters from the aircraft and propulsive systems still demands the generation of advanced digital tools that can be used for the design of low-emissions and ultra-silent aircrafts.

An efficient strategy to develop these digital tools is the use of data-driven methods based on deep learning, where a neural network is fed with data coming from experiments and/or high-fidelity simulations. These tools can be integrated in design and optimization workflows in order to reduce the computational cost, the turnaround time and the energy consumption.

In this context, the code Alya will aim to extend this multiphysics simulation platform in the future to integrate simulations of the aircraft and propulsive system with advanced HPC algorithms and data-driven methods based on machine learning. This will permit obtaining detailed predictions of aerodynamic parameters and emissions characteristics including non-CO2 effects like soot and particulate matter and contribute to the design and development cleaner and more efficient aircrafts.

3.3 AVBP

AVBP is a compressible Navier Stokes solver for 3D unstructured meshes capable of simulating two-phase flow combustion.

The prediction of these unsteady turbulent flows is based on the Large Eddy Simulation (LES) approach where small scales are modelled and large scales are resolved. Arrhenius reduced law and analytically reduced chemistry models allow investigating combustion for complex configurations. AVBP is based on a cell-vertex finite volume approximation, leveraging a Lax-Wendroff or a finite element type low-dissipation Taylor-Galerkin scheme in combination with a linear-preserving artificial viscosity model. Two-phase flows are accounted for using the Eulerian or Lagrangian approaches.

Within the framework of EXCELLERAT two use cases have been showcased: C3U1: Safety application; and C3U2: Pollutant prediction using LES for aeronautical applications;

C3U1 consists on the simulation of a confined explosion on the Sydney configuration [1].

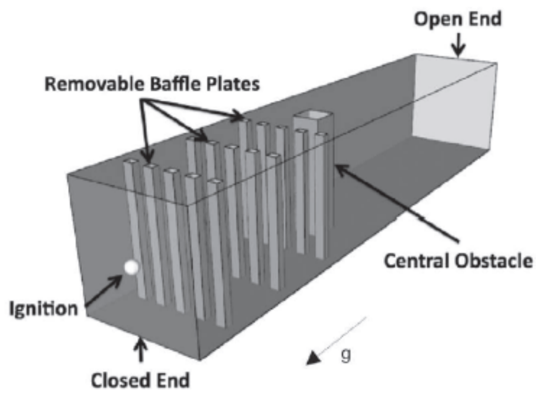


Figure 8: Sydney configuration for flame propagation past repeated obstacles [1]

The Sydney geometry (also called MASRI) consists of a confined combustion chamber with repeating obstacles. For EXCELLERAT we chose the most complex one with three rows.

C3U2 consists of the simulation of the combustion chamber of the DGEN-380 from Price Induction, INC and Akira Technologies [2].

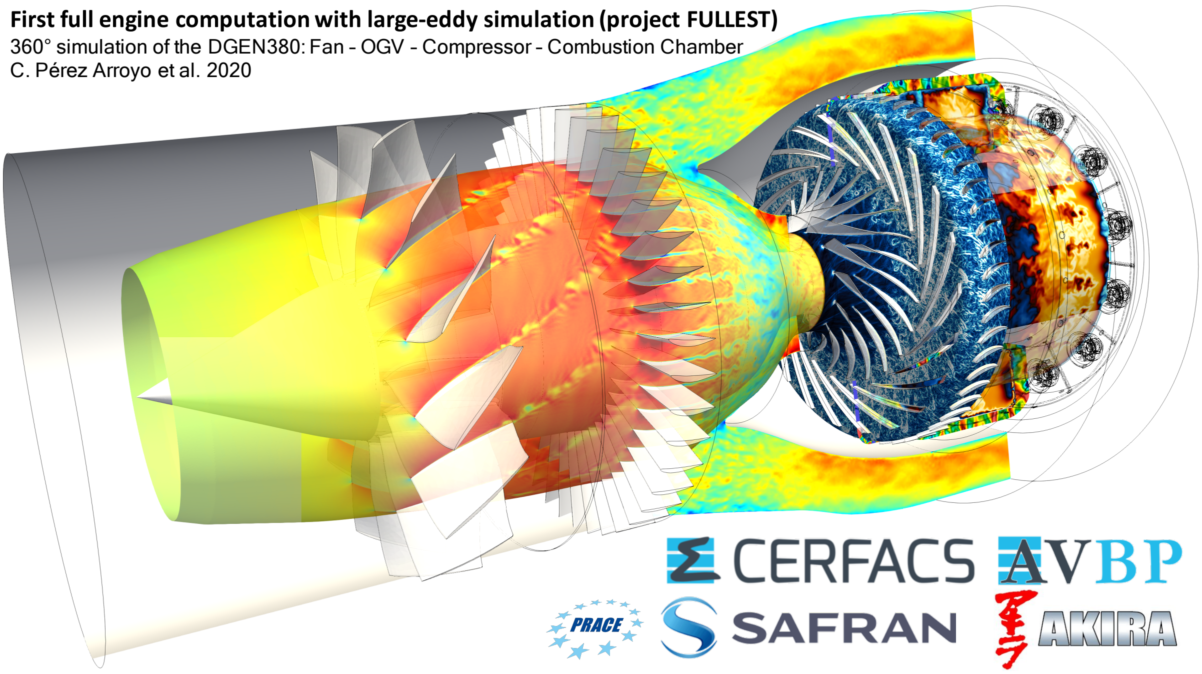


Figure 9: Snapshot of the FULLEST project including the DGEN combustion chamber (right extrema).

3.3.1 Achievements

Here we summarize the principal achievements of use case C3U1.

During the project we have successfully simulated the Sydney use case and compared it to experimental data. Part of the challenge was to improve the computational efficiency of such a case by introducing fast and accurate automatic mesh refinement techniques (WP3). Three simulations were performed, a reference static mesh case with 20 million tetrahedras, and two dynamic mesh cases using AMR with ten and five million elements each. Simulation conditions can be found here [3].

The main success criterium for the simulation is capturing the overpressure over time due to the explosion. This single parameter controls the safety and correctness of the modelling. Figure 3 shows the overpressure over time for the simulations versus the experimental data demonstrating the success of the techniques.

The performance graph (Figure 10) singles out the fact that using automatic mesh refinement greatly improves the efficiency of the solver specially on the early stages of the simulation where the explosion is limited spatially and we don't need a finer mesh elsewhere in the domain. Measured speed up using 540 cores is at least two-fold compared to the static refined case when considering the overall cost of the simulation when using the 10 million element mesh, 2.5 when using the 5 million element case.

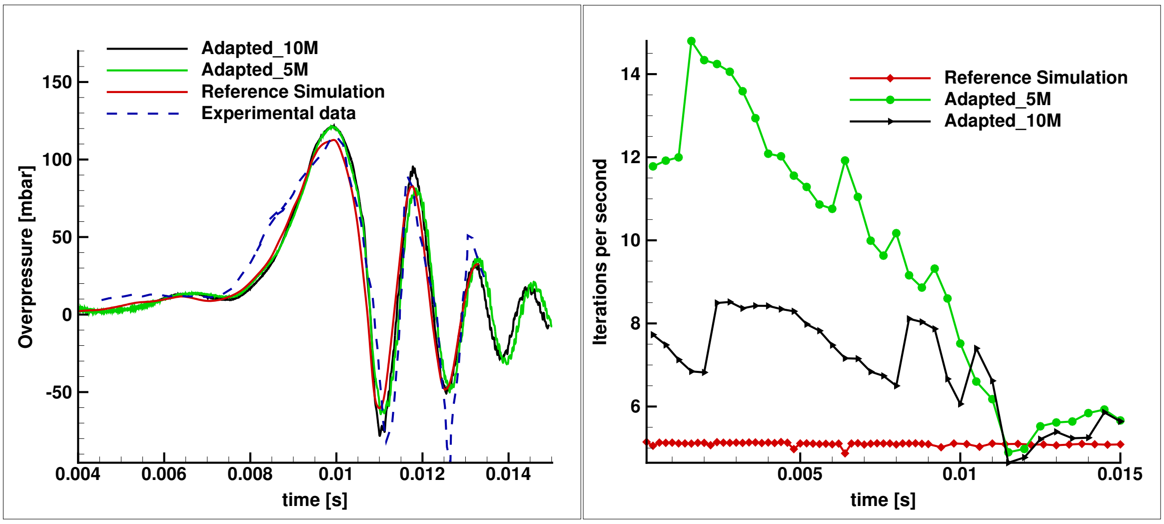


Figure 10: Left: Measured overpressure compared to numerical results using AVBP: static mesh (reference) , AMR (5M and 10M elements respectively). Right: solver efficiency measured in iteration per second using 540 on the Joliot Curie system from PRACE at TGCC

The second use case focused on the DGEN-380 combustion chamber simulation using accelerated hardware.

The combustion chamber is part of the DGEN-380 gas turbine demonstrator and the simulation parameters for the setup can be found here [4] and here [5].

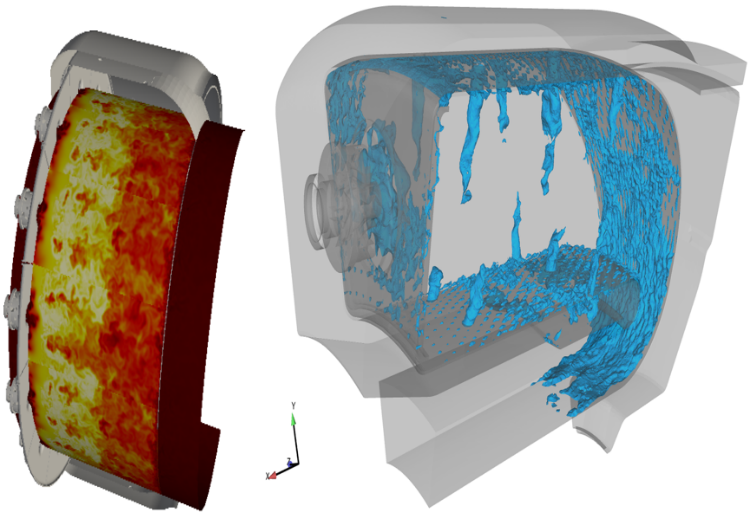


Figure 11: snapshots of the DGEN combustion chamber simulation: temperature field (left) isosurface of cooling on a clipped sector (right)

The simulation was performed successfully using 1024 NVIDIA V100 GPUs on the Tier 1, JeanZay System from GENCI at IDRIS [http://www.idris.fr/jean-zay/]. This work required the porting and validation of the workflow using GPUs (WP3).

Figure 11 shows snapshot of the full 360 case (left) and a clipped via for the cooling in the system (right).

3.3.2 Interaction with other work packages

As detailed in the previous section, tasks in WP2 have interacted heavily with WP3 and WP4. First, ensuring that node and system performance of the code is not impacted by the changes required for a dynamic mesh structure and automatic mesh refinement. Also, the development of the said mesh refinement framework. Secondly, testing and porting the code and the use case to alternative architectures. The static mesh C3U1 case has been used as a reference workflow for the GPU port until the definite C3U2 setup was chosen and C3U2 was used since as a benchmark to optimise GPU acceleration.

GPU optimisation and C3U2 have been the occasion to closely collaborate with NVIDIA within WP4.

3.3.3 Future work

The work from this deliverable will be included in the PhD defence of S. Sengupta expected in September 2022 and published in the 8th European Congress on Computational Methods in Applied Sciences and Engineering, June 2022, Oslo Norway.

3.4 TPLS

The use case for TPLS is counter-current flow in a vertical column with mass transfer. Liquid flows down the column whilst gas flows upwards. Both are driven by the gravitational force and a pressure gradient (the pressure is higher at the bottom than the top). The pressure gradient is such that for the gas it overcomes the downward gravitational force whilst for the liquid (which is denser) the gravitational force dominates. The gas is made up of two components: vapour of the same material as the fluid and a second, immiscible with the fluid, gaseous component. The temperature of the fluids is not necessarily uniform and the liquid can evaporate or condense leading to changes in the volumes of the gas and liquid, and in the composition of the gas.

3.4.1 Technical work and achievements

The public version of TPLS current at the start of the project had a number of issues that prevented it from simulating this use case. Most of these were associated with omissions in the modelling capability of TPLS, however there was one purely technical deficiency which prevented the simulation of large problems: not all of the I/O was done in parallel.

The issues the code had that related to modelling were.

- The direction of gravity was fixed to act perpendicular to the flow.
- Temperature was not modelled.
- The gaseous phase consisted of a single component (vapour was not present).
- There was no model of evaporation/condensation.

Derivatives of TPLS existed that addressed each of these problems but they could not be used directly for computational reasons: for example either they used a two dimensional decomposition of the domain or they used OpenMP rather than MPI both of which severely limit the number of processors that can be applied to simulating the use case. Consequently effort was put into amalgamating various aspects of these codes into the publicly available version in order to meet the demands of the use case.

Achievements in the initial stages

Early achievements in the project included:

- Modification to allow for gravity to act at arbitrary angle to the flow, rather than being perpendicular.
- Modifying timestep and increasing the resolution to allow for the density contrast factor of 1000 which is encountered when the flow consists of a gas and a liquid phase. Previously TPLS was typically used with density contrast factors of about 10.
- Improving the I/O by using HDF5 features supported by PETSc, to overcome the problem of a single MPI process having to read in all the data and then distribute it. This work enabled us to scale up to 24,000 cores (the largest known configuration for TPLS at that time).
- Implementing a re-start capability so that a single simulation could be carried out over multiple runs. This is an important development for the usability of the code and its use in simulation workflows.

- Investigating the performance of the code which revealed that the majority of the processing time (around 85%) was spent in the PETSc solvers. Given that these are third party highly optimised pieces of code, there was little scope for improvement by modifying the existing TPLS code. Therefore we produced a prototype native GPU implementation (i.e. not relying on the built-in GPU capability of PETSc which was found to give poor performance). The approach of using Open-CL allows greater portability across architectures, an important consideration for future-proofing the code.

Diffuse Interface Model (DIM)

The model used to represent the fluids is the so-called “single fluid model’. It gets its name from the fact that the multiple components involved are modelled as spatially and temporally varying properties of a single fluid. The equations for the liquid and gas phases have the same form and the transition between them is smooth. The only difference between them is their composition in terms of liquid, immiscible gas and vapour. There is no sharp boundary between the liquid and the gas: the interface is defined as a small region around those locations filled with half liquid and half gas (by volume). This model of the interface is termed the Diffuse Interface Model, or DIM.

The implementation of the DIM, which was provided by a previous project, was completed, tested and debugged. It was needed as a supplement to the existing Two Phase, Level Set (TPLS) method of defining the interface in order to accommodate the model of mass transfer (evaporation/condensation) that was imported from existing code.

The model of evaporation/condensation is one that is suitable for use far from the saturation temperature (boiling point) of the liquid. It assumes that the liquid and vapour are in quasi-equilibrium in the interface layer and that the gaseous phase is saturated with vapour there. (by contrast, a boiling liquid is definitely not in equilibrium).

The original code had five degrees of freedom: the proportion of liquid in a cell, the pressure and the three components of the bulk velocity of the fluid. During the course of the modifications (i.e. importing of code) two further degrees of freedom were added: the temperature and the proportion of vapour (by volume) in the gaseous phase.

The following graph illustrates the performance of the new code with all of the modelling complete. As can be seen the code, in this case, scales well up to almost 33,000 cores (Figure 12).

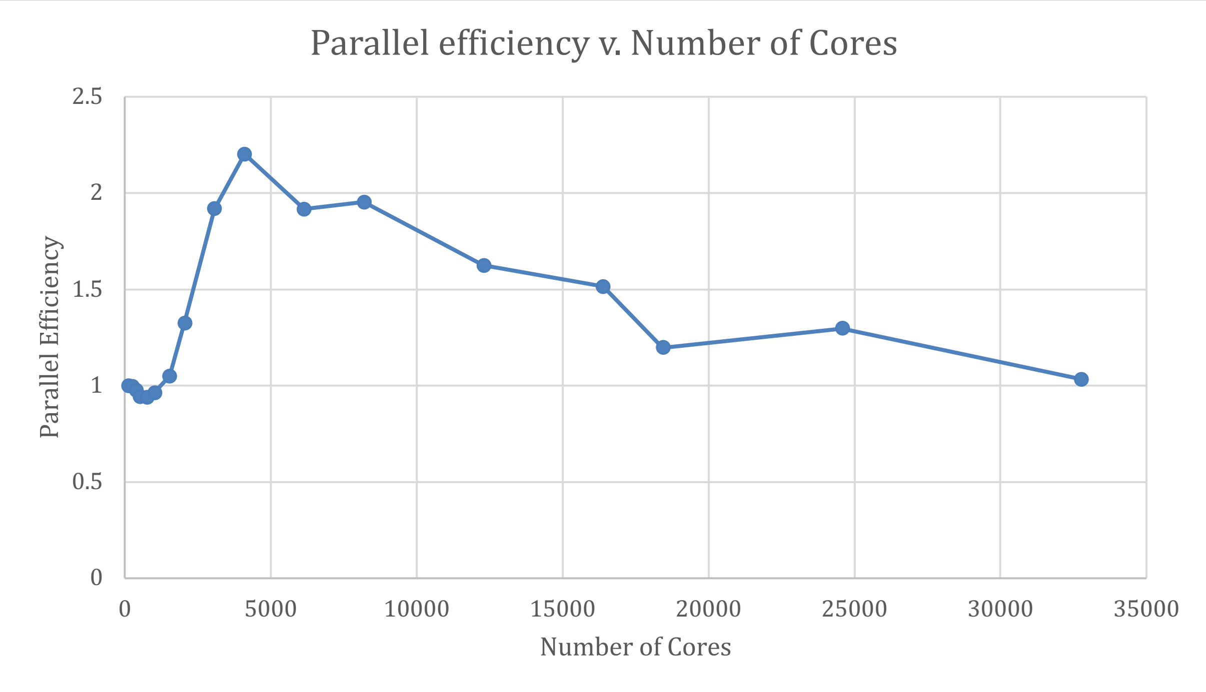


Figure 12 Parallel efficiency of TPLS versus number of cores relative to 1 node (128 cores). The domain has 2688x336x336 cells

Industry perspective

TPLS has been under sustained development for over 10 years and this will continue. The software is, and will continue to be, made publicly available under a BSD style licence. Before the end of the project a new public release of TPLS will be made available via SourceForge.

TPLS is currently in use in the industry-academia EMBOSS project funded by the EPSRC and is a crucial part of a bid for further research funding. It is also being used by industry and academic participants of the €1.2M EC funded ThermaSMART programme.

3.4.2 Interaction with other workpackages

The work done to implement parallel I/O has been done in collaboration with WP4. The single node and system level optimisation that came about as a result of developing a GPU-enabled version provides the link with work done in WP3.

3.4.3 Future work

After EXCELLERAT finishes TPLS will continue to be developed. In EMBOSS a CFD model of boiling has been developed and integration with Lattice Boltzmann and Molecular dynamics codes is being investigated. The model of boiling will appear in a further public release of TPLS. This release (including other improvements) will be funded by the project ARCHER2-eCSE06-3 which will start in June 2022.

TPLS also features in the SANGRIA project which is currently being considered for funding by EPSRC in the UK. SANGRIA will look at spray-cooling. PETSc's support for GPUs has improved considerably since it was last looked at in EXCELLERAT so, when the opportunity arises, the use of GPUs to execute TPLS will be revisited.

3.5 FEniCS

3.5.1 Achievements

FEniCS-HPC is the HPC flavour of the high-level problem-solving environment FEniCS for automated solution of partial differential equations (PDEs) by the finite element method. To manage the complexity of multiphysics problems FEniCS takes the weak form of a PDE as input in a near mathematical notation and automatically generates low-level source code, abstracting away implementation details and HPC concepts from domain scientists. Unicorn is the CFD solver enabled by the FEniCS-HPC API and implements the solution of the incompressible Navier – Stokes equations for high Reynolds regimes, where viscosity can be neglected, and with a special treatment of turbulent dissipation [9]. It also provides an adjoint-based, *a posteriori* error estimation strategy, used to drive the adaptive mesh refinement. Within EXCELLERAT, the use case C5U1 is a study of the external flow for the DrivAer model [1] a modular realistic car CAD available for research purposes defined by the Technische Universität München and studied by using the Windi Tunnel Test facility.

Two different geometries with increasing level of details have been studied:

baseline geometry (fastback rear shape, simple under body, no wheels, no rear mirrors) used to test the direct and adjoint solution, and tune the meshing strategy

complex geometry to apply the adjoint strategy to drive the mesh refinement on a realistic case

The code features require a tetrahedral mesh, but do not need boundary layers to be built on surfaces representing solid bodies, hence tetrahedral cells can be used in the whole domain. Different meshing strategies and tools (snappyHexMesh, cfMesh, Tetgen) have been investigated to produce an optimal quality mesh, aiming at the following:

- Provide meshes with small values of cell radius ratio
- Try to limit volume differences between adjacent cells

The workflow implemented to mesh the DrivAer case via different open source meshers has also been developed a python tool (called *checkMesh*) based on PyVista [11] APIs to automate the mesh quality evaluation, that produces quality maps of an input mesh. It allowed the choice of the best meshing strategy. All the input files and the developed utilities have been collected in a gitlab repository, to ease the process replication and to track the progress of the work.

SnappyHexMesh, the OpenFOAM [13] mesher, produces polyhedral mesh that has been later converted into the tetrahedral one, via the pyVista tool. The steps to produce the mesh can be summarized as follows:

1. Geometry has been defined via stl files
2. Mesh generation has been performed via castellation + snappy meshing. Starting from a template mesh the mesh has been refined to meet the geometry described in the stl files. Low resolution stl files could be used.
3. Quality assessment has been obtained by using the *checkMesh* tool
4. Conversion to vtk format has been performed via foamToVTK

[1] https://www.epc.ed.tum.de/en/aer/research-groups/automotive/drivaer

5. Conversion from polyhedral mesh to tetrahedral via PyVista

Tetgen [13] produces a native tetrahedral, boundary conforming Delaunay mesh, driven by triangulated surfaces provided as input geometries:

1. Geometry has been defined via stl files. Tetgen needs high quality stl inputs
2. Mesh generation via Tetgen: the mesh growed directly from boundary triangulation enabling a perfect match of the provided boundaries.
3. Quality assessment obtained by using *checkMesh*

Due to best matching of the boundaries described in the stl files, and no conversion step needed to generate the tetrahedral cells, Tetgen showed best quality results (see Figure 13 and Figure 14 for a comparison of the obtained meshes in the baseline case), and has been adopted for both baseline and complex simulations meshing.

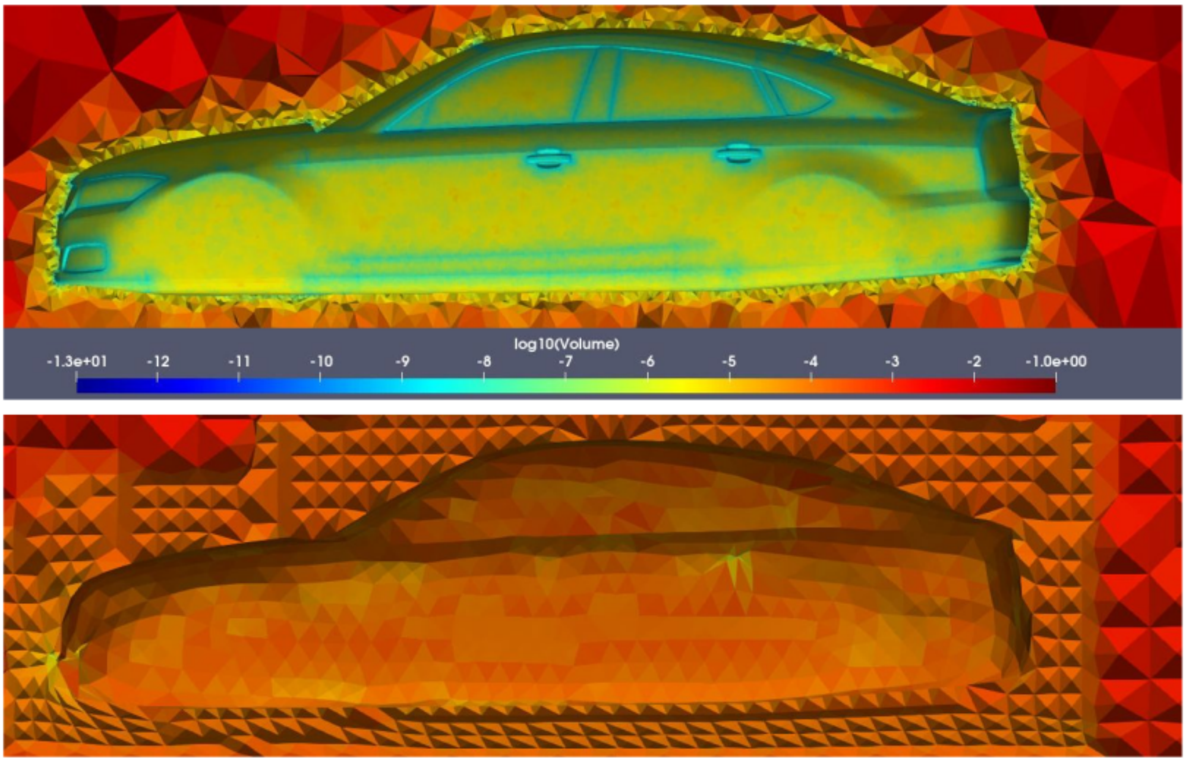


Figure 13: Meshes generated for the DrivAer baseline geometry by Tetgen (above) and SnappyHexMesh (below).

snappyHexMesh model:

- 96% of cells with 1 < Rr < 2
- Rr max = 2.7 E4

tetgen model:

- 80.8% of cells with Rr < 2, 15.8% Rr < 3
- Rr max = 24

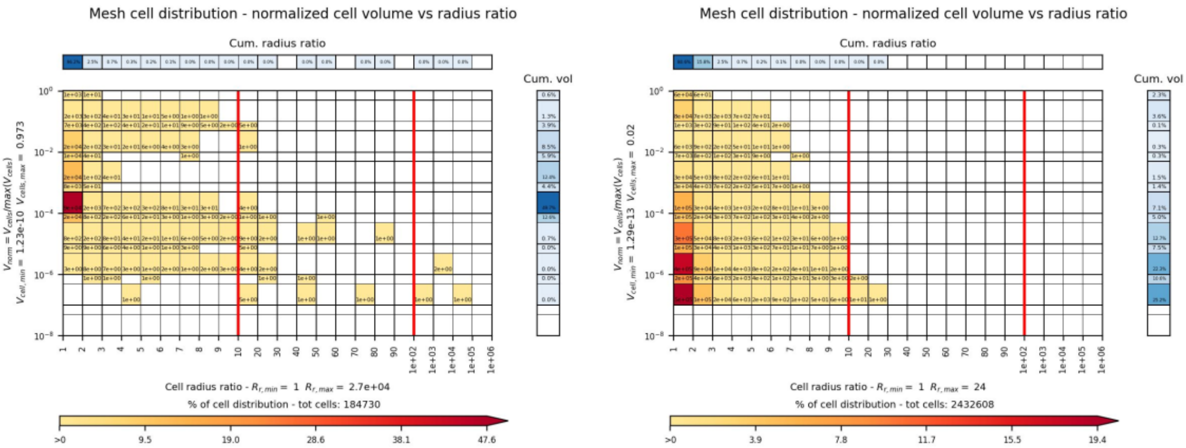


Figure 14: Comparison of the quality maps for the meshes generated for the DrivAer baseline geometry by SnappyHexMesh (left) and Tetgen (right).

The mean pressure and velocity, and the force coefficients behaviour during the simulation for the baseline geometry are shown in Figure 15. The primal and adjoint fields computed are shown in Figure 16. The figures show the distribution of the mesh elements' radius ratios versus the cell volumes. The optimal quality element has a low volume and radius ratio near one, meaning a small, non-degenerate element. The color map represents the density of the mesh cells in the given interval of radius and volumes: darker color, higher density. It can be noted, by comparing left and right side of the figure, that the Tetgen model has higher density in the desired regions and no cells exceeding ratios of 24: it shows therefore the best quality.

The obtained mesh refinement, which correctly matches the flow characteristics for the base line model is illustrated in Figure 17.

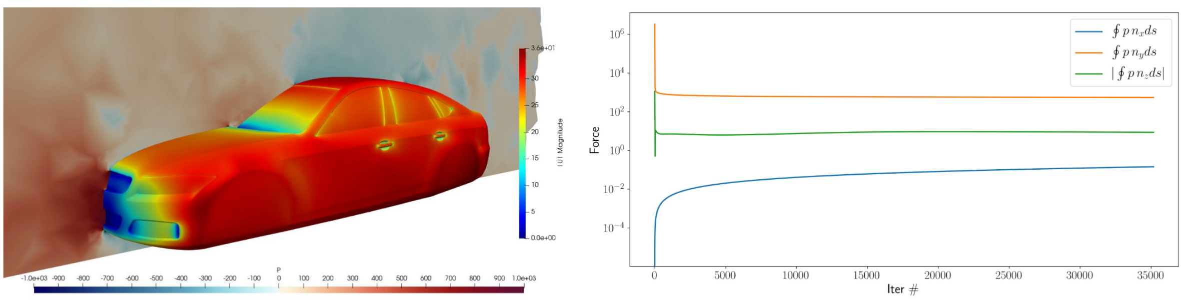
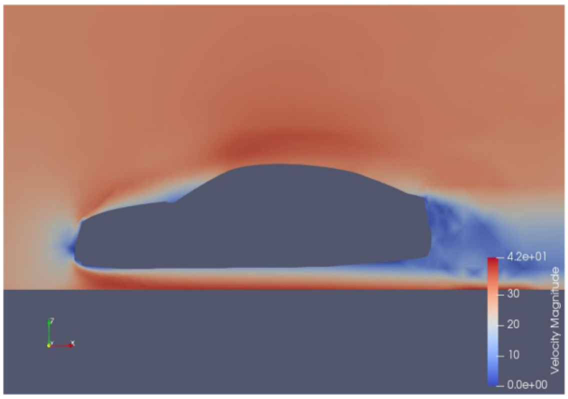


Figure 15: Mean pressure and velocity fields simulated for the baseline geometry (left), and force coefficients trend during the baseline simulation (right).



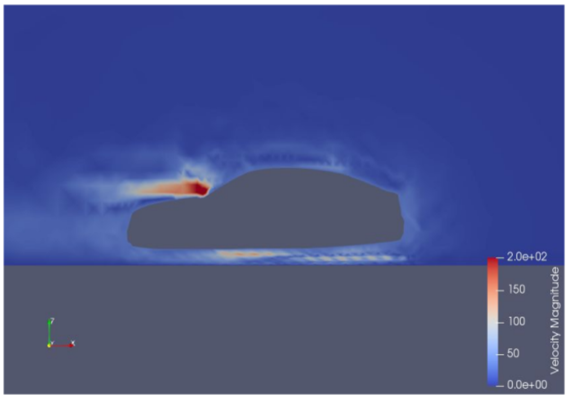


Figure 16: Primal and adjoint velocity magnitude for the baseline case.

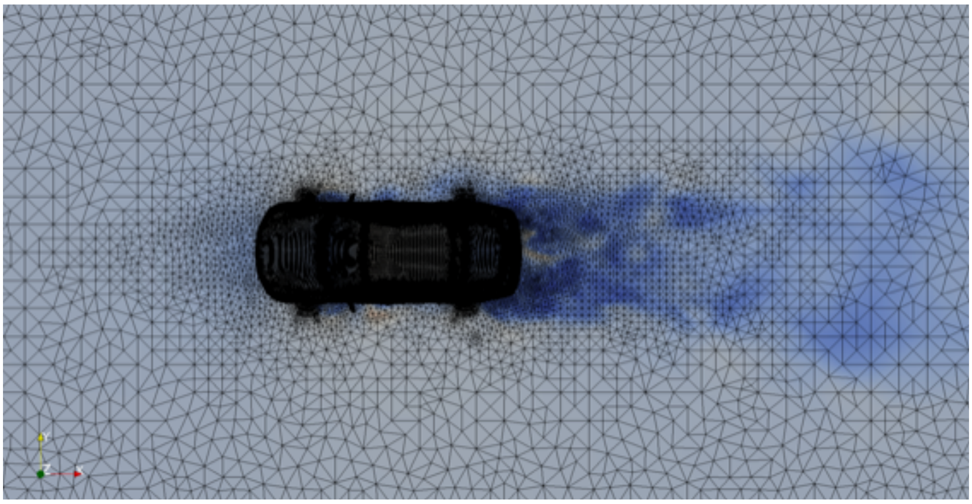


Figure 17: Refinement correctly matches the flow characteristics for the baseline model.

The following step was the meshing of the complex geometry, sketched in Figure 18. The mesh has been obtained using Tetgen, for the reasons outlined above.

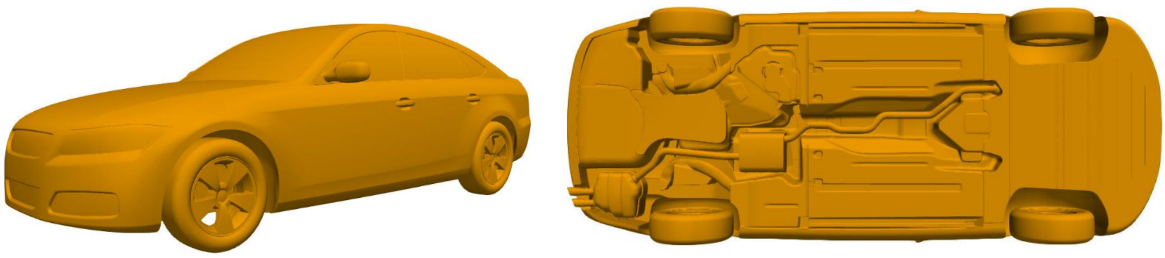


Figure 18: Complex geometry for the DrivAer test case: fastback rear shape, rear mirrors, wheels, detailed under body.

A mandatory requirement for the Tetgen tool is that input triangulation geometries cannot have intersections. For the case of the DrivAer model, this particular requirement was not straightforward to fulfil since the car wheels are in contact with the ground. The problem of the contact between the wheels and the ground has been solved by building a support for each wheel, providing a smooth transition between the ground and the wheels, avoiding any intersection. The steps used to build the wheel supports are presented in Figure 19**.**

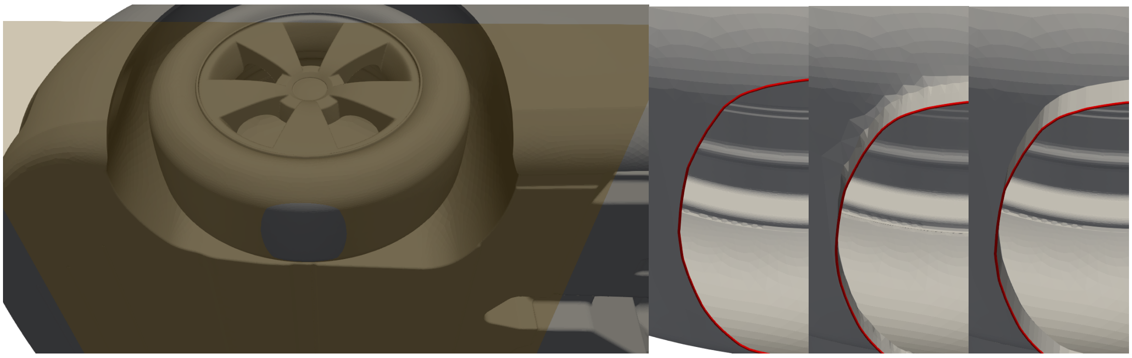


Figure 19: Detail of the cutting plane (left), and detail of the cutted wheel (right), extrusion of the cutting profile towards the ground and effect of the smoothening process (from left to right)

First, the wheels have been clipped using a plane that is parallel to the ground. The cut profiles, shown in red in Figure 19, are then extruded towards the ground. Finally, a smoothening filter is applied to the triangulation. This came out to be a very important step since after the smoothening of the triangulation, significantly lower values of triangle aspect ratios have been obtained. It can be explained since the Delaunay tetrahedralization generated by Tetgen is boundary conformal, high radius ratio values in the input triangulation produce therefore high radius ratio values in the volume mesh. As mentioned above, low tetrahedral radius ratio values are fundamental to ensure the convergence of the simulations with FEniCS, so we focused on keeping them as low as possible. The effect of the smoothening filter can be noted in Figure 20, where a comparison of the radius ratio values, close to the wheel support, is shown before and after the smoothening process.

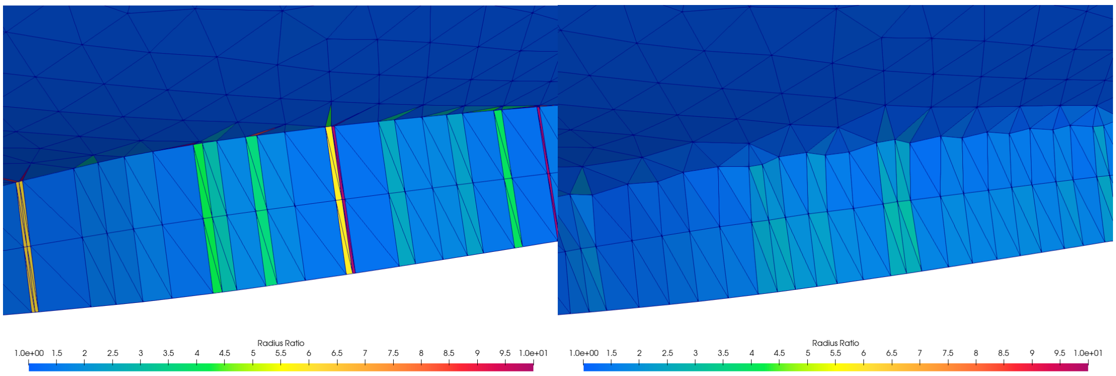


Figure 20: Comparison of radius ratio values, on the extruded profiles, before (above) and after (below) the smoothening process.

Finally, we were able to obtain a mesh (approximately 11 million cells) of this DrivAer model setup. In Figure 21, we show the distribution of the cells volume versus the cells radius ratio: it can be noted that the final mesh has a very good quality, the maximum radius ratio value being equal to 13.

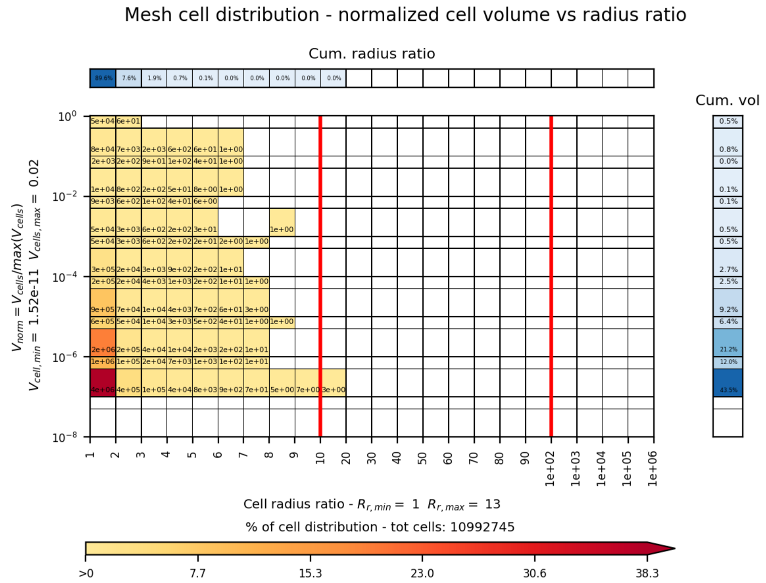


Figure 21: Quality map (cell volume vs cell radius ratio) distribution of the computed mesh for the complex geometry.

The high-quality mesh made it possible to obtain a set of first results for the pressure field and a sketch of the velocity streamlines, as shown in Figure 22.

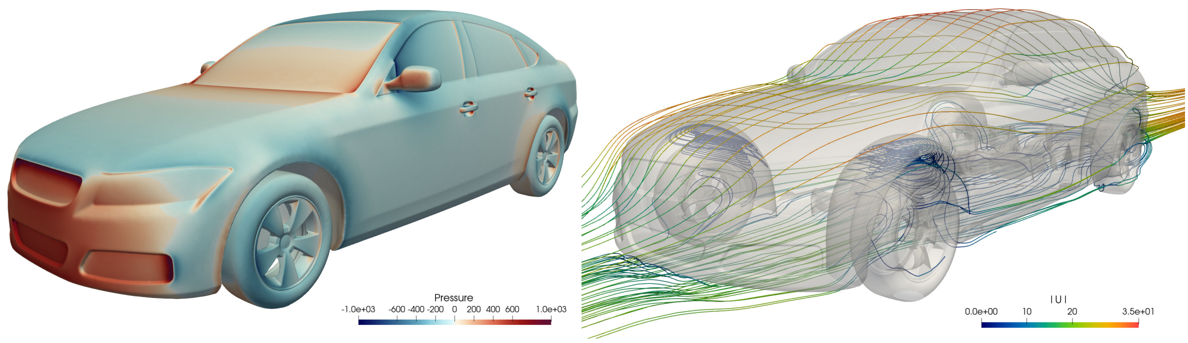


Figure 22: Pressure field (left) and velocity streamlines (right) computed for the complex DrivAer geometry.

However, the increased complexity exposed a bug in the mesh refinement algorithms inside DOLFIN HPC (the core finite element library in FEniCS), which caused the solver to diverge or crash. EXCELLERAT had a close collaboration with the DOLFIN HPC developers to address these issues, providing information and scaled-down reproducers to aid in the resolution of these bugs in recent DOLFIN releases.

Once all the mesh refinement bugs had been resolved, the next challenge was the robustness of the flow solver. Again, the fine details of the complex model caused either instability or required very small time steps to avoid divergence. Since FEniCS is a problem-solving framework, it does not provide a dedicated flow solver. Instead, there are several open-source solvers built on top of FEnICS. Within EXCELLERAT, we based our work on the old Unicorn solver [14] updated to support the recent version of DOLFIN. A large amount of time in the project were spent on making the solver robust enough to run the complex case. Figure 23 shows the results of a fully automated adaptive run. The left part of the figure shows the initial mesh at the final time, while the right part of the figure shows the last, refined mesh at the same simulation time.

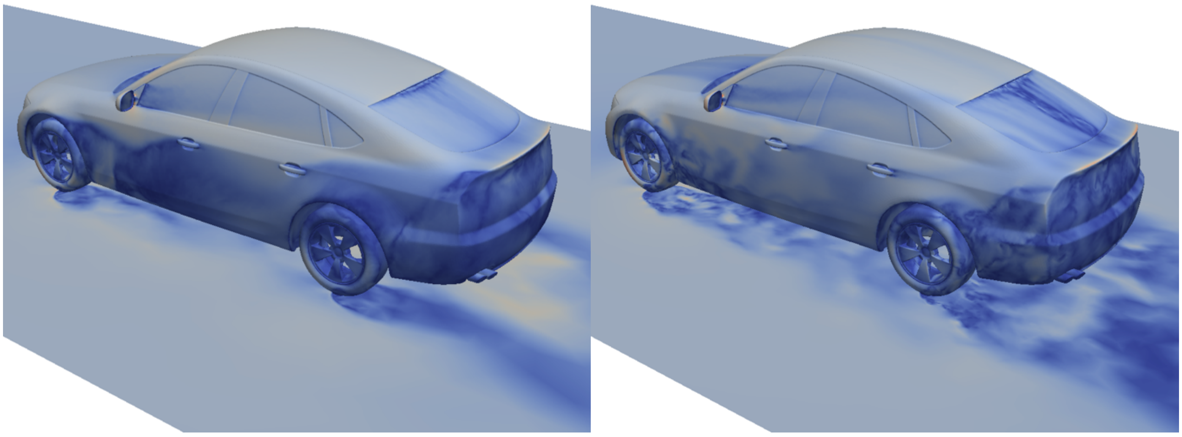


Figure 23: Snapshot of the velocity at the final time-step for two different adaptive iterations, first (left) and final (right).

One of the downsides of using goal-oriented adaptive mesh refinement based on adjoints is the increased cost of first running the primal problem forward in time, followed by the adjoint solver backwards in time. Since this has to be done for each adaptive iteration, it can quickly become more costly to run the whole adaptive process than designing an "overrefined" mesh by hand. To mitigate this problem, we used the projection scheme within Unicorn where the primal solution is projected from the coarser mesh onto the refined mesh before starting the next adaptive iteration. Using this scheme, only the first and final iterations need to be computed over a long time interval, while the intermediate iterations can be relatively short, only to provide enough sensitivity information to build the optimal mesh. The entire scheme is illustrated in Figure 24.

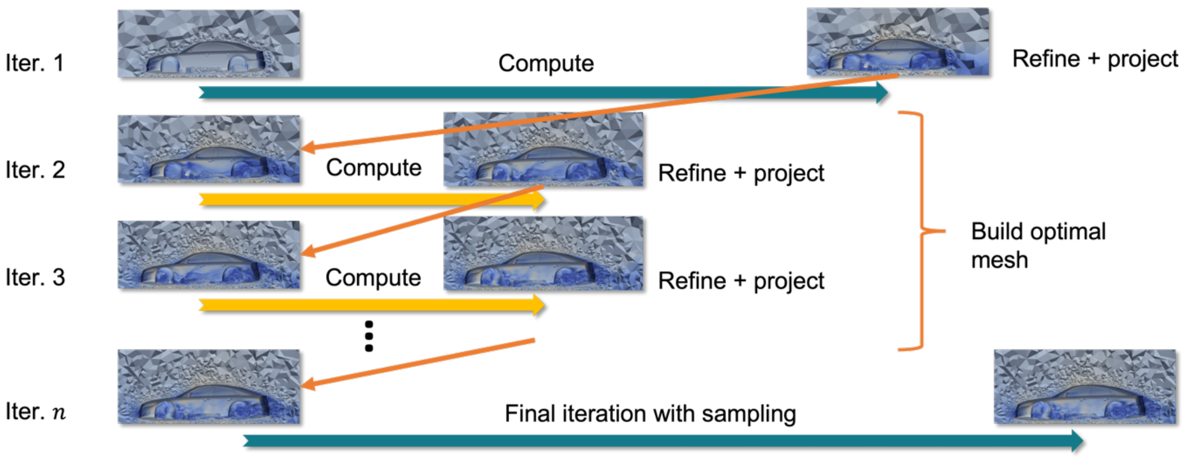


Figure 24: An illustration of the optimised adaptive iterations using projections.

3.5.2 Interaction with other workpackages

As part of task 3.2 in work package 3, the hybrid MPI+PGAS parallelisation of FEniCS has been further developed and optimised. A performance evaluation of the new hybrid linea algebra solvers and its use in a full fluid solver was performed and later published in the proceedings of IEEE Cluster'2020 [10]

3.6 CODA (formerly FLUCS)

CODA is a next-generation Computational Fluid Dynamics (CFD) software for the simulation of aircraft aerodynamics. It solves the Reynolds-Averaged Navier-Stokes equations on unstructured grids based on second-order finite-volume and higher-order Discontinuous-Galerkin (DG) discretization**.** The implementation addresses the efficient usage of current and upcoming high performance computing (HPC) systems and emerging technologies such as GPUs. CODA is developed in a joint effort of the German Aerospace Center (DLR), the French Aerospace Lab (ONERA) and Airbus. FlowSimulator is a multi-disciplinary analysis (MDA) framework developed by DLR. CODA uses FlowSimulator's core component, the open-source FlowSimulator DataManager (FSDM) for pre- and post-processing tasks such as I/O and mesh partitioning. FSDM also provides a mesh container class for the exchange of data among multi-disciplinary simulations coupled via FlowSimulator. The mesh container instance is a distributed representation of the data, containing information on the geometry, the mesh, flow fields and solutions, as well as coupling strategies.

3.6.1 Achievements

During the project, for CODA six main tasks were carried out: First, CODA and the surrounding FlowSimulator framework were ported, tested and evaluated on the AMD Naples and the Intel Cascade Lake architecture. Second, the scalability on the largest available partition of DLR's main production system CARA was assessed with the NASA common research model in a strong scaling scenario. Third, the Sparse Linear Systems Solver (Spliss) was fully incorporated into CODA, which allows the transparent use of emerging technologies via the linear solver. Fourth, several improvements have been made to improve the efficiency of CODA and Spliss on CPUs and on GPUs. Fifth, CODA with Spliss running on GPUs was evaluated on the Nvidia V100 and A100 architectures. Finally, the interfacing with the FlowSimulator framework has been extended.

Evaluation on the AMD Naples and Intel Cascade Lake Architectures

After DLR's new HPC cluster CARA went operational in February 2020, CODA and the surrounding workflow FlowSimulator were installed and intensively tested. The AMD Epyc processor introduces new architecture features that need to be considered in CODA, such as two-level NUMA domains and the L3-cache distribution. Thus, the performance of CODA was evaluated on the new architecture with particular focus on the hybrid setup of MPI ranks and OpenMP threads. The same was done for an Intel Cascade Lake test system. The initial results were compared and we identified the ideal hybrid MPI-OpenMP setup for both architectures. Furthermore, we found a limitation in the AMD Epyc architecture that limits the efficient hybrid usage to four OpenMP threads per MPI. This restricts CODA in utilizing its full hybrid capabilities and, thus, also hinders CODA in reaching its full scalability potential. The Intel Cascade Lake architecture does not impose these limitations.

Scalability Assessment on DLR's Main HPC Cluster CARA

After identifying the ideal hybrid setup and adapting all workflow components to CARA, efforts were focused on evaluating the scalability of CODA on CARA using a test case that solves the Reynolds-averaged Navier-Stokes equations (RANS) with a Spalart-Allmaras turbulence model in its negative form (SA-neg). The test case runs on a unstructured mesh from the NASA Common Research Model (CRM) with about 5 million points and 10 million volume elements. The mesh is a rather small mesh, which has been chosen for a strong scalability analysis (fixed problem size) of CODA. Production meshes are at least 20 times larger and accordingly achieve comparable efficiency on much higher scales.

Figure 25 highlights the scalability of CODA on the CARA HPC system as well as the progress that has been made during the project. For the rather small mesh CODA achieves about 71% parallel efficiency on the largest available partition on CARA with 512 nodes and 32,768 cores.

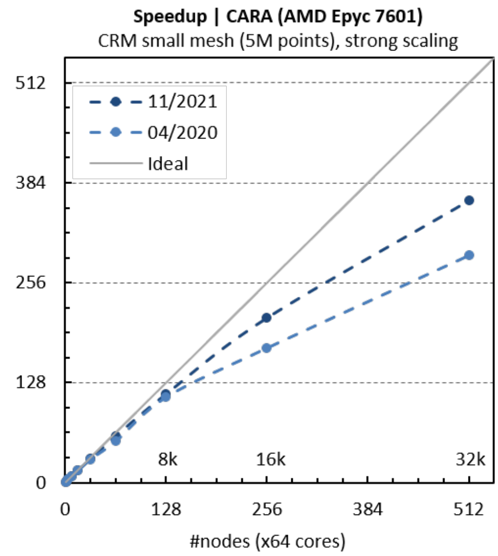


Figure 25: Scalability of CODA on CARA, DLR's HPC cluster based on the AMD Naples architecture.

Incorporation of the Sparse Linear Systems Solver (Spliss)

During the project the introduction of the Sparse Linear Systems Solver (Spliss, developed by DLR) into CODA was finalized. Spliss aims to provide a linear solver library that, on the one hand, is tailored to the wide requirements of CFD applications but, on the other hand, independent of the particular CFD solver. A key design aspect of Spliss is computational efficiency and parallel scalability for current and emerging HPC technologies.

One key success criterion for the introduction of Spliss was that the additional abstraction and flexibility does not introduce significant additional overhead. To ensure this, CODA with and without Spliss was evaluated on the HPC system CARA. The measurements found that Spliss does not introduce any additional overhead compared to the previously used internal linear algebra solver.

Improvements to CODA and Spliss on CPU and GPU

Focusing on the specific task of linear-system solving allows for integrating more advanced, but also more complex, hardware-adapted optimizations, while at the same time hiding this complexity from the CFD software CODA. One example is the usage of GPUs. Spliss enables the execution of the computationally intensive linear solver on GPUs. However, the Spliss interface design provides this capability to a user in a transparent way. By that means, CODA can leverage GPUs without the necessity of any code adaptation in CODA. Next to the main achievement of GPU support, several other optimizations and improvements have been included in Spliss.

Evaluation and Optimization on Nvidia V100 and A100 Architectures

After the introduction of Spliss, CODA with Spliss executed on GPUs was evaluated. First, the numerical stability and correctness was validated. Second, the initial performance was evaluated on up 16 Nvidia V100 GPUs. Third, a detailed performance analysis was carried out to identify potential performance bottlenecks.

After that, Spliss has been optimized and re-analyzed multiple times. Optimizations include among others the inclusion of CUDA-aware MPI and GPUDirect, the improvement of host-to-device copies and the inclusion of CUDA multi-process service (MPS) to allow more flexibility in the allocation of CPUs and GPUs.

Finally, CODA with Spliss executed on GPUs was evaluated again. The current version achieves a speedup of 4.2 to 4.4 in a node-wise comparison of two Intel Xeon 6230 (40 cores) vs. 4 Nvidia V100 GPUs, which is a speedup of 2.4 to the initial version. In addition, CODA with Spliss was evaluated on the Juwels Booster System at Julich Supercomputing Center and achieved a very good parallel efficiency of 63% on 128 Nvidia A100 GPUs.

Interfacing with the FlowSimulator Framework

The CODA CFD solver is not operated as a stand-alone application but rather as a plugin to the multi-disciplinary analysis (MDA) framework FlowSimulator. In particular, CODA uses FlowSimulator's core component, the FlowSimulator DataManager (FSDM) for I/O and FSDM provides the FSMesh class, which is the preferred container for the exchange of data among FlowSimulator plugins; i.e. among different multi-disciplinary sub-components. Several improvements to the interfacing with the FlowSimulator framework have been made and are particularly important for the execution of coupled multi-disciplinary simulations.

3.6.2 Interaction with other work packages

The porting of CODA to the CARA HPC system and the scalability evaluation on the CARA has been done cooperatively with WP3.

The identification of best practices when using the AMD Naples (CARA) and Intel Cascade Lake architectures have been carried out together with WP3 and close cooperation with the respective hardware vendors within WP4. Furthermore, the gained expertise on best practices was made available to WP4.

The porting of the Sparse Linear Systems Solver (Spliss) to GPUs as well as its optimization for CPU and GPU have been achieved cooperatively with WP3 and WP4.

The results were released in publications and presented at conferences with WP7.

3.6.3 Future work

The CODA CFD software and the workflow framework FlowSimulator have achieved a high level of code maturity and usability during the project and are about to replace the predecessor TAU (developed by DLR) in production in the European aircraft industry, research organizations and academia, most prominently at DLR for aerodynamics data production and validation and at the Airbus Group. The final steps to release CODA into production will be one of the main next steps.

In addition, CODA will be continuously analyzed and improved in numerical as well as computational performance. CODA will be adapted and evaluated on new architectures and systems (e.g. DLR's new HPC system based on the AMD Rome architecture) as well as emerging hardware technologies.

3.7 Reference Applications Outcomes

The results that will be produced within EXCELLERAT from the work on the reference applications will be considered from two perspectives:

- To feed in the defining potential service offerings
- To provide technical material for dissemination by EXCELLERAT (for example use cases, best practices, etc.)

In collaboration with WPs 1, 2, 4, and 7, the findings of the work on the reference applications will be used to support and promote the offerings from EXCELLERAT, giving real examples of the achievements available via applied expertise.

The outputs for this task will gather all the results in single points of reference, while it is expected that results will also be reported internally as and when they are produced

3.7.1 Achievements

Definition of services and products offering

According to the definition of this task, in the last period the focus was to have a clear overview of all services and products developed during EXCELLERAT P1 to be able to improve in WP5 the content management for the EXCELLERAT Service Portal, to be able in WP6 to support the sustainability strategy in terms of exploitation strategy and eventually in business plan development plan for the most exploitable key results and finally to improve the communication and dissemination strategy for the most exploitable key results in collaboration with services and products owners.

This overview of services and products has been compiled on the basis of all the expertise that was defined in the proposal and Grant Agreement, summarised and adjusted in the D1.6 Report on the Service Portfolio. Following type of services and products were selected: tools like code and software, consultancy like consulting topics, training, best practise guide and data access. A service portal (https://services.excellerat.eu/) has been established through which services can be accessed.

	Consultancy	Training
Nek5000	High fidelity CFD simulations with Nek5000	High fidelity CFD simulations with Nek5000
Alya	Meshing and re-meshing techniques, methodologies and Software	Use of Alya with mesh optimization
AVBP	AVBP with optimized scaling	Numerical methodes for Large Eddy Simulations
TPLS	Use/set-up of TPLS	
FEniCS	Use of FEniCS	
CODA		
Vistle	Use of Vistle in situ	Use of Vistle in situ

Table 1 Initial consultancy and training services derived from core codes. Further services will be added to the portal as they mature.

Monitoring of technical material for EXCELLERAT dissemination

Based on the exploitation strategy documents like characterisation table and exploitation roadmap, this task developed in collaboration with WP7 template for the following dissemination material like success story, blog and white paper. Specially for the success stories the focus is to explain the stakeholders what is the challenge, the solution, the benefit and the value proposition to potential customers. See table 5 the summary of the dissemination and communication material for the use cases

Communication and Dissemination Material	Partners involved	Name of the success stories (all published at https://www.excellerat.eu/success-stories/
C1U1: Aerospace - Flow around aerofoil with rounded wing tip	**KTH**	Enabling Nek5000 on GPU systems
		Accelerating engineering codes using reconfigurable architectures
C1U3: Aerospace - High fidelity simulation of rotating parts	**KTH**	A novel framework for online estimation of the uncertainties in turbulent flow statistics
		Accelerating engineering codes using reconfigurable architectures
		Enabling Nek5000 on GPU systems
C2U1: Automotive/ Aerospace - Emission prediction of internal combustion and gas turbine engines	**BSC**	A POP proof-of-concept allows a Bunsen flame use case from EXCELLERAT to run two times faster
		Accelerating Alya engineering simulations by using FPGAs
		Accelerating engineering codes using reconfigurable architectures
		Enabling High Performance Computing for Industry through a Data Exchange & Workflow Portal
C2U2: Aerospace - Active flow control of aircraft aerodynamics including synthetic jet actuators	**BSC**	Full Airplane Simulations on Heterogeneous Architectures
		Accelerating Alya engineering simulations by using FPGAs
		Accelerating engineering codes using reconfigurable architectures
		Enabling High Performance Computing for Industry through a

Communication and Dissemination Material	Partners involved	Name of the success stories (all published at https://www.excellerat.eu/success-stories/
		Data Exchange & Workflow Portal
C2U3: Transport systems - Coupled simulation of fluid and structure mechanics for fatigue and fracture	**BSC**	Accelerating Alya engineering simulations by using FPGAs
		Accelerating engineering codes using reconfigurable architectures
		Enabling High Performance Computing for Industry through a Data Exchange & Workflow Portal
C3U1: Aerospace and energy - Combustion instabilities and emission prediction	**CERFACS**	Bringing industrial end-users to Exascale computing: An industrial level combustion design tool on 128K cores
		Enabling sustainable GPU acceleration on a Fortran legacy code
		Running AVBP Industrial code on Arm architectures
C3U2: Safety applications - Explosion in confined spaces	**CERFACS**	Enabling parallel mesh adaptation with Treeadapt
		Bringing industrial end-users to Exascale computing: An industrial level combustion design tool on 128K cores
		Running AVBP Industrial code on ARM architectures
C4U1: Bi-directional two-phase thin film flow – chemical engineering processes	**UEDIN**	Modelling techniques for the chemical and oil and gas industry
C5U1: Aerospace and Automotive - Adjoint optimization in external aerodynamics shape optimization	**CINECA, KTH**	Mesh optimising by using an a posteriori adjoint based error estimation
		Accelerating engineering codes using reconfigurable architectures
C6U1: Aerospace - Design process and simulation of full equipped aero planes	**DLR**	Transparent Integration of Emerging HPC Technologies into the Computational Fluid Dynamics Software CODA

Table 2: Summary of dissemination and communication material for the use cases

3.7.2 Interaction with other work packages

Task 2.4 is working together with members of WP1, WP2, WP4, WP5, WP6 and WP7 to select together the main priorities and develop the activities plan in terms of definition of potential products/ services portfolio, to complete the business plan, the dissemination and communication plan and optimize the content management of the EXCELLERAT Service portal.

3.7.3 Future work

The next steps for EXCELLERAT could be planned as follows

- **Classification of EXCELLERAT services/products:** the next steps will be optimising the classification according to the stakeholders, industrial sectors, applications sectors to be able to do a market analysis according to the market need and offering and adapt the potential services offering from EXCELLERAT
- **EXCELLERAT services and products competitor's analysis:** to finalize a deep analysis of the products/services from EXCELLERAT compared to the products/services of the competitors
- **Potential new EXCELLERAT services/products:** to define the potential products/services from the market need to adapt eventually the EXCELLERAT products/services to the different potential stakeholders needs
- **Content of use cases:** to define the content of some use cases including the challenges, solutions, benefits for each potential stakeholder to deliver material content for the dissemination
- **EXCELLERAT potential stakeholders:** to optimise each potential type of stakeholders lists for each use cases to develop potential business case for new potential stakeholders

4 Conclusion

Many significant advances have been made in taking the six reference application codes towards exascale in the EXCELLERAT project. The work done on the codes has led to technical advances in generic HPC technology carried out elsewhere in the project, and has led to the development of services that are available through the EXCELLERAT Service Portal, https://services.excellerat.eu/, with a view to making the centre sustainable in the future.

5 References

[1] Kent, J. E., et al. "A new chamber to study premixed flame propagation past repeated obstacles." *5th Asia-Pacific conference on combustion*. The University of Adelaide Adelaide, Australia, 2005.

[2] Nagai, Kenichiro and Oinuma, Hideshi and Ishii, Tatsuya. “Acoustic liner test of DGEN 380 turbofan engine” I*NTER-NOISE and NOISE-CON Congress and Conference Proceedings 259* n°3 (2019): 6078-6088

[3] Vermorel, Olivier, Pierre Quillatre, and Thierry Poinsot. "LES of explosions in venting chamber: A test case for premixed turbulent combustion models." *Combustion and Flame* 183 (2017): 207-223.

[4] Pérez Arroyo C, Dombard J, Duchaine F, et al. Towards the Large-Eddy Simulation of a full engine: Integration of a 360 azimuthal degrees fan, compressor and combustion chamber. Part I: Methodology and initialisation. Journal of the Global Power and Propulsion Society. 2021;(May):1-16. doi:10.33737/jgpps/133115.

[5] Pérez Arroyo C, Dombard J, Duchaine F, et al. Towards the Large-Eddy Simulation of a full engine: Integration of a 360 azimuthal degrees fan, compressor and combustion chamber. Part II: Comparison against stand-alone simulations. Journal of the Global Power and Propulsion Society. 2021;(May):1-16. doi:10.33737/jgpps/133116.

[6] J. V. Pastor, J.M. García-Oliver, C. Micó, A.A. García-Carrero, and A. Gómez. Experimental study of the effect of hydrotreated vegetable oil and oxymethylene ethers on main spray and combustion characteristics under engine combustion network spray a conditions. Applied Sciences, 10(16):5460, 2020.

D. Mira, E.J.Pérez-Sánchez, A.Surapaneni**,** J.Benajes. J.M García-Oliver, Pastor, J.M. and De León, D., 2021, October. LES Study on Spray Combustion with Renewable Fuels Under ECN Spray-A Conditions. In *Internal Combustion Engine Division Fall Technical Conference* ASME-ICEF2021, Vol. 85512, p. V001T06A004.

[7] L. Cai, S. Jacobs, R. Langer, F. vom Lehn, K.A. Heufer, and H. Pitsch. Auto-ignition of oxymethylene ethers (OMEn, n=2-4) as promising synthetic e-fuels from renewable electricity: shock tube experiments and automatic mechanism generation. Fuel, 264:116711, 2020.

[8] Christopher L Rumsey, Jeffrey P Slotnick, and Anthony J Sclafani. Overview and summary of the Third AIAA High Lift Prediction Workshop. AIAA P., 2018. 2018-1258. https://hiliftpw.larc.nasa.gov/.

[9] Hoffman, Johan & Jansson, Johan & Jansson, Niclas & Abreu, Rodrigo & Johnson, Claes. (2017). Computability and Adaptivity in CFD. https://doi.org/10.1002/9781119176817.ecm2057.

[10] N. Jansson, 'A Hybrid MPI+PGAS Approach to Improve Strong Scalability Limits of Finite Element Solvers', In proceedings of IEEE International Conference on Cluster Computing, 2020, pp. 303–313
[11] Sullivan and Kaszynski, (2019). PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK). Journal of Open Source Software, 4(37), 1450, https://doi.org/10.21105/joss.01450
[12] Weller, Henry G., Gavin R. Tabor, Hrvoje Jasak and Christer Fureby. "A tensorial approach to computational continuum mechanics using object-oriented techniques." Computers in Physics 12 (1998): 620-631.
[13] Hang Si. 2015. TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator. ACM Trans. Math. Softw. 41, 2, Article 11 (January 2015), 36 pages. https://doi.org/10.1145/2629697
[14] Hoffman, J., J. Jansson, C. Degirmenci, N. Jansson and M. Nazarov (2012). Unicorn: a unified continuum mechanics solver. Automated Solution of Differential Equations by the Finite Element Method: 339--361.