HORIZON-EUROHPC-JU-2021-COE-01

Centre of Excellence in Exascale CFD

CEEC – Centre of Excellence in Exascale CFD
Grant Agreement Number: 101093393

D4.1 – Approach to workflows, ML-based submodeling, visualization, uncertainty quantification and dynamic resource management

WP4: Exascale techniques

EuroHPC Joint Undertaking

Document Information

Deliverable Number	D4.1
Deliverable Name	Approach to workflows, ML-based submodeling, visualization, uncertainty quantification and dynamic resource management
Due Date	31/10/2023 (PM 10)
Deliverable lead	FAU, KTH
Authors	Rosa M Badia (BSC) , Marta Garcia-Gasulla (BSC), Josep Pocurull (BSC), Anna Schwarz (USTUTT), Gregor Weiss (USTUTT), Martin Aumüller (USTUTT), Susanne Malheiros (USTUTT), Manuel Münsch (FAU)
Responsible Author	Manuel Münsch (FAU) manuel.muensch@fau.de
Keywords	consortium codes, solvers, requirements analysis, strategies definition
WP	WP4
Nature	R
Dissemination Level	PU
Final Version Date	31/10/2023
Reviewed by	Kajol Kulkarni (FAU-LSS) , Daniel Kempf (USTUTT-IAG)
MGT Board Approval	31/10/2023

Acknowledgment:

Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and Sweden, Germany, Spain, Greece, and Denmark under grant agreement No 101093393.

Document History

Partner	Date	Comment	Version
UmU, DTU, USTUTT	09/04/23	Define questionnaire for the consortium codes	0.1
UmU, KTH, FAU-LSTM	16/05/23	Initial analysis of the questionnaire outputs	0.11
FAU-LSTM	03/08/23	Initial version	0.1
FAU-LSTM, BSC, USTUTT-HLRS, KTH USTUTT-IAG	15/09/23	Analysis of questionnaire output by WP4 partners	0.2
FAU-LSTM, BSC, USTUTT-HLRS, KTH USTUTT-IAG	02/10/23	Draft of the deliverable for proof reading by each LHC representative	0.3
FAU-LSTM	12/10/2023	Incorporation of remarks from first reviewer	0.4
FAU-LSTM	26/10/2023	Incorporation of remarks from second reviewer	0.5
FAU-LSTM	31/10/2023	Final Version	1.0

Executive Summary

This document represents deliverable 1 of work package 4 - 'Approach to workflows, ML-based submodeling, visualization, uncertainty quantification and dynamic resource management' of the Centre of Excellence for Exascale CFD (CEEC). CEEC aims to develop, enhance, and implement exascale ready techniques for addressing relevant challenges with respect to CFD on future exascale systems. These challenges are reflected by the definition of six lighthouse cases (LHC) as core part of the project. Deliverable 1.1 summarized the requirements to tackle the lighthouse cases. Derived from this and by considering the response to a questionnaire provided to the lighthouse case owners, this deliverable provides a documentation of the techniques to be developed and enhanced from a software perspective in work package 4 to meet the lighthouse case requirements. In detail, the required actions refer to an enhance of the support of workflows, the development of solutions for integrated machine learning (ML) based sub-models, the enabling of visualization and efficient management of extreme-scale data sets, uncertainty quantification tools and dynamic resource management.

The software developments and enhancements of technologies reported in this deliverable are in an initial state for the time being. Co-operations among the project partners, detailed analysis, particular work on the tasks in combination with tests on European HPC resources will provide a more and more detailed picture.

Contents

1 Introduction

The Center of Excellence for Exascale CFD aims to develop and advance state-of-the-art computational fluid dynamics (CFD) algorithms and models with the clear goal of enabling exascale performance to compute six lighthouse cases with high relevance to industrial practice. In order to do so, a clear picture of the state of the art with respect to the applied codes and the particular needs to solve the individual lighthouse cases is needed. This picture is provided by deliverable D1.1 whereas the focus is mainly on the computational requirements and exascale readiness of the codes. In work package 4 (WP4) the purpose is to enhance required techniques and technologies to implement the lighthouse cases. A questionnaire was designed (see Appendix) and distributed among the lighthouse case owners to identify necessary developments with respect to the support of workflows, the development of solution strategies for integrating ML-based sub-models, enable visualization and efficient management (compression) of these data of extreme-scale, the adaption of uncertainty quantification tools or techniques for CFD applications and dynamic resource management.

In this regard the document provides a summary on the lighthouse cases in section 2. From D1.1 and the questionnaire required techniques for exascale to be developed or enhanced are derived in section 3. The document closes with a summary in section 4.

2 Lighthouse cases

In the following, the lighthouse cases are summarized in a brief manner.

2.1 LHC1: Shock-Boundary layer interaction and buffet on wings at the edge of the flight envelope

In LHC1, shock/boundary-layer interactions and buffet are investigated to refine and expand the edge of the flight envelope on a 3D wing configuration of a modern prototypical transport aircraft. For this, the high-order solver FLEXI is employed which allows efficient scale resolution of the compressible LES equations closed by wall models. Data-driven shock indicators combined with an h/p-adaptation strategy will allow for a sub-element scale localization of the shock fronts, while providing accurate turbulence resolution. The simulation results will be utilized to investigate the effects of three-dimensionality and wing sweep on the onset and characteristics of buffet and associated aerodynamic loads. For the LHC simulation, the computational kernel of FLEXI will be accelerated and methodological developments to improve the accuracy of FLEXI will be tackled within WP3 and WP4. This includes wall-modelling, mixed-precision approaches for the shock capturing, and the offloading of high-density operations to the GPU. For efficient I/O, the shared memory concept is introduced for solution I/O with zero-copy.

2.2 LHC2: High fidelity aeroelastic simulation of the SFB 401 wing

In LHC2 an aeroelastic simulation of the SFB 401 wing, also known as the HIRENASD model, will be conducted using the multiphysics Alya code. This simulation reproduces typical flight conditions under a transonic regime with high Reynolds numbers and elastic deformations for the structural part. The flow (CFD part) will be solved using turbulence wall-model LES for compressible flows, while the structure will be solved by a solid mechanics framework (computational structure mechanics: CSM) specifically devoted to transient non-linear problems with large deformations. For the dynamic analysis of the structure, the implicit Newmark time-integration scheme will be used together with an iterative solver for the resolution of the algebraic system. With regards to the coupling algorithm, to preserve the advantages of the highly adapted CFD and CSM solvers, a partitioned approach will be conducted with weak coupling. In order to take into account the elastic deformation of the structure, the Alya mesh deformation tool based on ALE formulation will be used. For the LHC simulation machine-learning based wall-modeling for large-eddy simulation will be required. Furthermore, this coupled problem will benefit from dynamic resource management to improve the performance as further described in section 3.

2.3 LHC3: Topology optimization of static mixers

The overall goal of this LHC is to synthesize a static mixer for the increased mixing of species or fluid phases by topology optimization. The transport is convection dominated and the flow is modeled using an Eulerian-Eulerian approach where phases are represented by fractions within the flow solver Neko [9]. The design representation is utilizing the immersed boundary methods developed in WP3 and taking advantage of the research into stable adjoint formulations for turbulent chaotic flows. The design evolution is advanced by a gradient based numerical optimization algorithm. To overcome the intrinsic problem of slow moving geometric features in extreme resolution design optimization, an adaptive and multi-resolution strategy will be employed. This lighthouse case is highly connected to the tasks in WP3. However, dynamic resource management from WP4 might be beneficial to use a sparse representation of the design.

2.4 LHC4: Localized erosion in offshore wind-turbine foundations

In this LHC, a fluid-solid coupled micromechanical simulation of a seabed foundation will be run using the waLBerla framework for the analysis of localized fluidization (so-called piping erosion) during the installation of a suction bucket for the basement of an offshore wind turbine. The fully-resolved simulation will involve a discrete-element (DEM) representation of the solid phase (i.e. the structural element and the single grains of the sandy seabed) and a lattice Boltzmann (LBM) hydrodynamic model of the percolating water. A preliminary coupled model will at first aim to simulate the results of a physical model test in reduced-scale (validation stage with available laboratory tests), while the grand application in 3D will focus on the local phenomena taking place around a representative cut of the full-scale foundation during the first meters of the suction-driven installation. To enable the latter scenario, a major task will concern at first the creation of the initial seabed conditions in terms of a representative granular fabric involving layered deposition, granular cementation and a realistic state of intergranular forces and relative density throughout the embedment depth. The subsequent simulation of the grain-resolved flow will pose a significant computational challenge. The task will be attacked using the Euler-Lagrangian coupling functionality of waLBerla using geometrically resolved particles that can be generated with specified size and shape distributions. The coupling itself will employ the momentum exchange method which must be additionally improved by lubrication models. This lighthouse case is relying on an already developed workflow whereas findings from WP4 can potentially be used to improve the current state. Furthermore machine-learning based submodeling is considered to be interesting in terms of memory traffic reduction. Due to the expected data sizes this LHC will also require enhanced visualization and data management techniques.

2.5 LHC5: Simulation of Atmospheric Boundary Layer flows

A high-resolution LES will be performed to investigate the stable and convective atmospheric boundary layer (ABL) to examine the quality of LES solutions, and in particular their dependence on the mesh, subgrid-scale (SGS) parameters, numerical discretization, and surface boundary conditions. The high-order Nek5000 and NekRS codes will be used, extended with wall models, based on M-O similarity theory for rough walls, appropriate for variational formulation approaches like the SEM. We plan to continue our collaboration with ANL and NREL scientists toward the cross-verification and validation of LES results and corresponding wall models by performing a number of scaling studies to compare the performance of several ABL codes on CPU and GPU platforms. In the stably stratified ABL (e.g., the nocturnal ABL over land), the largest turbulent scales are often much smaller than those seen during neutral or unstable stratification and sensitivity to the SGS model is increased in LES. Although continued improvement of SGS turbulence models is necessary, increasing simulation resolution is one route to decreasing dependence on SGS turbulence models. A well-documented stably stratified atmospheric boundary layer benchmark problem that can be used for these purposes, namely, for model and code inter-comparison, is the Global Energy and Water Cycle Experiment (GEWEX) Atmospheric Boundary Layer Study (GABLS). For the convective ABL we will examine numerical convergence in a sheared daytime convective ABL reported in Sullivan and Patton [16] by tracking low- and high-order statistics and bulk entrainment. Specialized kernels based on OCCA will be developed for the CPU/GPU code NekRS taking into account memory hierarchy and minimizing memory transfer between host and device. Although the LHC owner are not directly involved in WP4, this LHC will potentially benefit from the WP4 findings on machine-learning based sub-models for LES wall-modeling and enhanced techniques on visualization and data management.

2.6 LHC6: Merchant ship hull

For LHC6 the computation of the flow around a ship hull of a merchant ship in model scale, particularly that of the Japan Bulk Carrier, is intended. For that purpose the code Neko [9] will be used to perform large-eddy simulations (LES). The obtained data will be used to analyze the physics of the flow, particularly the streamwise vortical structures forming on the hull and the unsteady characteristics of the turbulent wake in the location of the propeller plane. Additionally, a simulation campaign will be established to test the various methodological developments from WP3 and WP4. Particularly, wall modeling, adaptive mesh refinement, and immersed boundary layer techniques will be applied. Uncertainty quantification techniques will be used for controlling and reporting the errors in statistical quantities, and also for intelligent design of the simulations involving parameter sweeps. In detail, for the investigation of the flow around the Japanese Bulk Carrier, the code Neko [9] has to deliver various functionalities like: wall-resolved as well as wall-modeled LES, wall-modelling via classical wall-models and machine-learning approaches based on supervised or reinforcement learning, uncertainty quantification, advanced insitu-visualization and data management (compression). Furthermore all connected workflows have to be integrated and resources have to be managed in a dynamic manner.

3 Documentation of the requirements for the techniques to be developed or enhanced in WP4

This section deals with the description and documentation of the requirements for the techniques and approaches to be developed within WP4. In an initial step a questionnaire distributed among all lighthouse case partners served as a starting point for the evaluation of needs and requirements to finally predict solutions to the lighthouse cases. Basing on this, table 1 gives a first overview on how the lighthouse case owners classify the proposed techniques addressed in WP4.

WP4	LHC1	LHC2	LHC3	LHC4	LHC5	LHC6
Workflows	(X)			X		X
ML-based sub-models	X	X		(X)	(X)	X
Visualization and data management	(X)			X	(X)	X
Uncertainty quantification						X
Dynamic resource management		X	X			X

Table 1: Techniques to be enhanced or developed in WP4 and their relation to lighthouse cases.

At this stage, i.e., at the beginning of the project it is obvious that lighthouse case owners evaluate their requirement of the proposed techniques to finally complete their cases in a different manner. While LHC6 relies on all developments in WP4, in contrast especially LHC5 seems to not require contributions from WP4. However, in the course of WP4 all findings will be shared proactively among the all LHC owners and an active exchange of information and developments will take place to ensure involvement of all LHCs.

In the following subsection the proposed techniques, their need to be developed or enhanced and their impact is discussed in detail.

3.1 Workflows

Applications are becoming increasingly complex and require the combination of traditional high-performance computing (HPC), data analytics (DA), and artificial intelligence (AI). However, traditionally each of these components are developed in different programming models and environments, i.e., MPI for HPC applications or Python libraries for AI, which are not easily combined. PyCOMPSs [17] has been extended along the years to better integrate task-based workflows with MPI and more recently in the framework of the eFlows4HPC project has been extended to better integrate with DA and AI [6].

In the framework of the CEEC project we plan to develop or port some of the LHCs on top of this environment. In particular, for LHC1, a solution based on SmartSim (Relexi) exists. It will be interesting to analyse the solution and understand if a better one can be obtained with the PyCOMPSs based software stack. For LHC2, a prototype workflow based on SmartSim and SmartRedis has been started, but some functionalities when running multiple ensemble simulations are not fulfilling the requirements of the LHC. An internal collaboration has been kicked-off at BSC to use PyCOMPSs and its possibility of running ensemble MPI simulations or ensemble MPMD simulations in a single workflow at the same time that an AI model is trained. Since we would like to maintain SmartRedis as backend database, the corresponding interfaces to support the integration of PyCOMPSs and StartRedis will be developed.

For LHC4, while there is a workflow already implemented, conversations with FAU and BSC have been started to explore the possible collaborations. FAU is interested in using PyCOMPSs for automating the parameter tuning of an evolutionary algorithm, since right now this step is performed manually. This can be done with dislib [4], a classical machine learning library implemented on top of PyCOMPSs that offer some methods for parameter tuning. We also plan to explore if the task-level fault management mechanism implemented in PyCOMPSs can be used to simulate failures in their algorithm. Another area to explore will be the use of some distributed training algorithms available in dislib to train AI models using GPUs.

For LHC6 a traditional pipeline with steps pre-processing, solver, post-processing will be used. There are two exceptions to this pipeline: in-situ visualization and ML model inference as a solver subcomponent. While this may not be developed in the early stages of the project, exploration of how to integrate the different stages that differ from traditional CFD workflows may be pursued in the near future.

When visualizing an on-going simulation, the visualization pipeline has to integrate with the simulation: either by driving pre-processing and starting the simulation from within the visualization workflow, or by attaching the visualization workflow to the running simulation. Interactive visualization sessions often entail workflows that need to be re-configured dynamically. In order to balance the resource requirements of simulation and remote visualization, we will work on supporting migration of data analysis and visualization modules between systems, e.g. between the system holding the data or running the simulation (in-situ visualization) and a dedicated visualization cluster.

3.2 ML-based sub-models

Machine-learning (ML) approaches for turbulence modeling such as closure models or wall models are required for LHC1, LHC2 and LHC6. A main reason for this is the size of the individual computational problem which renders a wall-resolved DNS or LES impossible. For a flat plate with length L in streamwise direction, the number of grid points (N) required for a DNS is proportional to $Re_L^{37/14}$, a wall-resolved LES still requires $N \sim \mathcal{O}(Re_L^{13/7})$ grid points, whereas a wall-modeled LES scales at most as $\mathcal{O}(Re_L)$ [3]. For the lighthouse cases, the Reynolds numbers are expected to be in orders of $\mathcal{O}(10^6)$ for LHC1 and LHC3 to $\mathcal{O}(10^9)$ for LHC6. Thus, the use of a wall-modeled LES (WMLES) is the route of choice to predict the flow fields for the particular LHCs. A recent trend in WMLES is to combine wall-models with the current advances in machine-learning (ML) for PDEs such that high-fidelity data can be used to develop more elaborate wall-models. Data-driven turbulence closure models have illustrated the potential of machine-learning for free flows and simple canonical test cases. Here, these models will be investigated in complex situations such as transition and shock-boundary layer interactions. The preliminary studies in this direction are promising. Yang et al. [19] successfully utilized neural networks to develop wall-models, although limited to channel flow configurations. Lozano-Duran and Bae [13, 14] used a classifier-predictor technique in which the model is trained with data from different canonical flows. The predictor provides the wall shear stress depending on which canonical flow the input data belong to. This assignment to the canonical flow regime is done by the classifier. Zhou et al. [20] employed a neural network to develop a wall-model using the data from wall-resolved LES of a periodic hill.

For all the cases, a higher quality for the simulation results is envisaged by using machine-learning strategies. Besides hybrid RANS/LES approaches, the strategies mainly adopted so far for wall modeling in LES are: (i) use of analytical laws established from experimental observations of the behavior of turbulent boundary layers, or (ii) coupling with supplementary differential equations that characterize the boundary-layer dynamics. Analytical wall-models encounter difficulties in describing non-equilibrium phenomena, such as adverse pressure gradients, while wall-models solving supplementary equations near the wall are considerably more complex and computationally expensive. ML could improve both approaches, e.g., combine the advantages of both, especially with respect to the more complex flow scenarios (adverse pressure gradients, curved surfaces) to be encountered for the LHCs.

Each group engaged with the LHCs starts from different starting positions as summarized in the following. For LHC1, the group at USTUTT-IAG already developed a data-driven approach for the closure problem of turbulence [10]. This approach is based on a reinforcement learning (RL) strategy since it is highly suitable for devising control strategies in the context of dynamical systems. In contrast to supervised learning where the training data can be generated a priori in an offline manner, RL requires constant run-time interaction and data exchange with the CFD solver during training. Here, Relexi [11] is used as a scalable RL framework that bridges the gap between machine learning workflows and modern CFD solvers on HPC systems, providing both components with its specialized hardware. Relexi allows easy integration of various HPC solvers by means of the in-memory data transfer provided by the SmartSim[1] library. The turbulence closure problem is tackled by finding a control strategy for an optimal eddy viscosity in large eddy simulations.

For LHC1, further work within the CEEC project will be concerned with the development of wall-models based on data-driven approaches to predict the wall stresses or directly the Reynolds stress profiles. Here, previous time sequences, i.e., the history of the flow, will be incorporated into the wall-model to improve the predictive accuracy and robustness of the resulting model. The new model will be trained on canonical flows such as a periodic hill and a shock-turbulence boundary layer interaction and will be finally applied to LHC1.

For LHC2, BSC will follow the RL strategy according to a model which uses limited data as developed by Bae and Komoutsakos [2].

For LHC6, very similar to LHC1, wall-models based on data-driven approaches are envisaged to predict wall shear-stress and from this corresponding source terms to fit the velocity profiles according to the wall model. Based on the Grant Agreement, RL as well as supervised learning techniques were proposed. However, current publications indicate limitations of supervised learning to model near-wall dynamics [2] which has to be further evaluated within the project. During the development and integration period the ML-approach will be initially tested on sub-cases, i.e. canonical flows as channel or flat plate flows. In a next step, since LHC6 is defined by the ship hull: a curved surface inducing adverse pressure gradients, the wall-model will be trained on comparable test cases, like the flow over a hump for example. Afterwards an application to the LHC6 configuration is planned.

In addition to the upper LHCs, ML-based submodeling is also of particular interest for the LHC4 owners. Although LHC4 does not require such modeling approach, there is some interest in a potential runtime reduction while keeping the order of accuracy. Training of any kind of model is preferably done via sub-cases due to computational limitations with respect to LHC4. However, how and if LHC4 will benefit from ML-based submodeling

[1] https://github.com/CrayLabs/SmartRedis

will emerge in the course of the project.

3.3 Visualization and data management

Each LHC will produce several terabytes of data. This will pose challenges for both data management and visualization. For data management, the task compares and identifies data compression algorithms like BigWhoop [18], zfp [12], and Zstandard [8] to evaluate their usability with the LHCs. A focus lies on BigWhoop which can exploit global statistical phenomena of a partitioned simulation. Such global effects can be crucial for visualization without artefacts introduced by partitioning boundaries. BigWhoop is only able to compress data from simulations with structured grids because it is based on the JPEG2000 standard. LHC3 and LHC4 could be primary targets for compression with the BigWhoop library. Both LHCs work with structured meshes with additional complexities that might have to be overcome within the task. LHC3 starts with structured grids with the possibility for unstructured meshes by overlapping domains later in the project. LHC4 uses waLBerla and is structured by definition due to the underlying Lattice-Boltzmann methodology. However, the case can be arranged in an octree-like data structure. Interactive visualization provides the means for getting an overview of these large data sets and for intuitively exploring them. This can be used to validate intermediate results of the simulation, formulate or investigate research questions and, as it is easy to understand even without knowing the physics behind the simulation, as basis for interdisciplinary discussions. The LHCs' domain is 3-dimensional, so 3D visualization in immersive virtual environments such as HMDs (head-mounted devices, e.g. VR glasses) and CAVEs (Cave Automatic Virtual Environment, i.e., a small, walk-in room with 3D images projected onto its walls) is a natural fit for the data. As the LHCs will each generate data with size between 5 and 100 TB, transferring the data to another system for visualization is prohibitively expensive. This makes in-situ and remote visualization necessary. The work will build on the open-source tool Vistle [1] as it is a highly scalable, modular and extensible software for scientific visualization which already includes flexible methods for interactive remote visualization. Additionally, USTUTT-HLRS will draw from the experience from the concurrent project EXCELLERAT P2, where the focus lies on in-situ visualization of simulations.

While some LHCs have already implemented their own visualization workflows, others have expressed the need for visualization for their projects: LHC4 investigates the risk of erosion of soils planned to be used as foundation for offshore wind turbines. For this, they need to visualize the movement over time of a subset of the particles which make up the turbine's foundation. For better understanding the erosion process, tracking the movement of individual particles is a requirement. LHC6 aim to use their ship merchant hull simulation to investigate the structure of turbulent eddies and, thus, plan on, e.g., visualizing the lambda-2 isosurfaces which would need to be calculated during visualization. During the project, it will be investigated how the methods from this task can be extended to other LHCs. Another potential candidate is LHC1 which simulates the shock-boundary layer interaction on wings and stated wanting to visualize volume and surface data on the walls.

As supercomputers are generally heterogeneous machines, most often consisting of CPUs and GPUs, we want to make full use of the available resources by porting computationally expensive parts of the visualization pipeline to the GPU. To avoid losing performance due to costly communication between CPU and GPU, we want to minimize data transfers by keeping data on the GPUs for as long as possible, while still keeping the modularity of Vistle. As the trend in HPC is to move from highly-optimized, vendor-locked GPU programming models to more portable ones, we additionally aim to make Vistle usable for accelerator hardware of multiple vendors.

3.4 Uncertainty quantification

At this stage uncertainty quantification (UQ) is only connected to LHC6 but may become relevant to other LHCs during the project as well. Non-intrusive uncertainty quantification for multiscale problems requires a large number of simulation runs, often with different computational levels of abstraction, mesh resolutions and models. RANS, LES and DNS computations differ greatly in their demands on an HPC system. All of this leads to very heterogeneous HPC and workflow management requirements, as $\mathcal{O}(10^5)$ or more computations of very different complexities, sizes and lengths and HPC requirements need to be handled. In order to perform such a large scale UQ analysis on HPC systems, the POUNCE [5] and UQit [15] frameworks have been developed for non-intrusive UQ simulations on HPC clusters. These tools are envisaged to be used to scale, schedule and handle heterogeneous data sets on the one hand side and to perform multi-level and multi-fidelity analysis on the other hand. The modular design of POUNCE allows it to easily add APIs for other baseline solvers and other clusters, such that efficient non-intrusive UQ simulations are possible in these other environments as well. While POUNCE has been applied successfully on HAWK at HLRS with FLEXI as a baseline solver, it is planned to a) analyze and extend the framework towards LHC6 and b) combine it with the Nek* codes, their I/O requirements (or others) and adapt it to different HPC architectures. In addition, the compression algorithms developed by HLRS are incorporated to safely and reproducible store the large data pools generated through sampling.

UQ methods will be utilized in LHC6 to increase the predictive reliability of deterministic simulations, here the prediction of the flow field around the Japanese Bulk Carrier. The drivers of uncertainty will be identified via sensitivity analysis in combination with a careful analysis of convergence of statistical quantities. In this case, time-averaged quantities (velocities, pressure, shear stress) and model parameters of the wall-models and sub-grid scale models employed in the LES are of particular interest, whereas the number of parameters is of the order $\mathcal{O}(10^1)$.

3.5 Dynamic resource management

All the lighthouse cases rely on the efficient allocation and management of computational resources, so it is crucial to ensure their optimal utilization, particularly for LHC2, LHC3, and LHC6. At the core of our strategy lies the DLB library [7]. Our objective is to utilize the DLB library to improve the load balance of MPI processes for LHC2, LHC3, and LHC6. DLB will allow us to dynamically redistribute computational resources at runtime, addressing sources of imbalance that may arise due to algorithm variations, data distribution, hardware architecture differences, and variations in resource availability.

Moreover, the integration of DLB and PyCOMPSs will allow a holistic dynamic resource management to ensure an efficient use of the computational resources available.

LHC2 involves a fluid-structure interaction problem solved using a weak coupling approach, such as Gauss-Seidel. We will collaborate closely to implement dynamic load balancing strategies that enhance performance by maximizing load occupancy, reducing resource requirements, execution time, and ultimately save energy. This will require continuous monitoring and adjustment of resource allocation based on real-time performance metrics.

LHC3 and LHC6 share common needs related to resource management, specifically in their utilization of sparse representations within their computational workflows. Our goal is to customize our resource management strategies to align with their unique requirements, increasing the efficiency and effectiveness of their computational tasks.

4 Summary

This document describes the requirements for techniques to be developed or enhanced in work package 4 to successfully implement the six lighthouse cases considered by the Centre of Excellence for Exascale CFD (CEEC). After a short introduction to each LHC, the requirements and the need of development in the categories workflows, ML-based submodeling, visualization, UQ and dynamics resource management was presented in detail.

Applications in general, but in particular the LHCs are increasingly complex and require combinations of HPC, DA or AI components. PyCOMPSs by BSC will be used to combine the different programming models and environments (like MPI and Python libraries) to enhance workflows for ML/AI, parameter tuning and in-situ visualization. Development action to orchestrate workflows in such scenarios is in particular required for lighthouse cases LHC4 and LHC6. Here, possible synergies are also expected for LHC1, whereas an workflow basing on the RELEXI tool chain and SmartSim is already available.

For lighthouse cases LHC1, LHC2 and LHC6 wall-modeled large-eddy simulations (WMLES) is the only feasible way to predict the flow fields for the expected Reynolds number ranges which are in orders of $\mathcal{O}(10^6)$ to $\mathcal{O}(10^9)$. To marry WMLES with machine-learning (ML) further development is in particular required on the modeling of complex flow scenarios (non-equilibrium phenomena, adverse pressure gradients and curved boundaries), which are characteristic for the LHCs. For this, Relexi (USTUTT-IAG) is considered as RL framework for modern CFD solvers on HPC systems. The findings are also considered to be potentially of interest for LHC4 as well as for LHC5 in respect to the wall-models considered for the atmospheric boundary layer.

By considering the expected data amount (5-100 TB) produced by the codes when computing the individual LHCs (deliverable D1.1) the challenges for data management and visualization are obvious. Here the focus lies on enhancing the data compression tool BigWhoop and the open-source visualisation tool Vistle (both from USTUTT-HLRS) for the LHC application. In particular development action is needed for LHC4 and LHC6. In addition, work on visualization and data management is also considered to be beneficial for LHC1 and LHC5.

For LHC6 development action is needed in terms of uncertainty quantification (UQ). In particular on exascale level, UQ is considered to be important to finally increase the predictive reliability of such expensive simulations. In order to perform large scale UQ studies on HPC systems, the tools POUNCE (USTUTT-IAG) and UQit (KTH) will be applied for non-intrusive UQ analyses.

Performing efficient computations on an HPC system highly relies on the efficient allocation and management of computational resources, i.e., the load balance of MPI processes. Here the DLB library (BSC) will allow to dynamically redistribute computational resources within the LHC computations which has to be further explored among the lighthouse case partners. Here lighthouse cases LHC2, LHC3 and LHC6 are expected to benefit from development action in respect to dynamic load balancing strategies that enhance performance and increase efficiency and effectiveness of computational tasks.

After evaluating and describing the need of developing and enhancing exascale techniques within this deliverable, the ”Requirements Phase” for WP4 is completed. Within the next step, the ”Development Phase” is entered where the focus will be placed on the development and integration of the above mentioned algorithms and technologies.

A Questionnaire WP4 Exascale Techniques

The WP4 questionnaire which was distributed among all light-house case owners to estimate the needs and requirements for exascale techniques:

WP4: Workflows

- Does the LHC require the development of any workflow that composes multiple invocations to different software components?
  - If yes, briefly describe stages of the workflow and components invoked in each stage.
  - If yes, is the workflow already developed or will be developed during the CEEC lifetime?
  - Additional information: If yes, for each workflow stage describe:
    * Granularity
    * Programming Languages used in this block
    * Required specific software (HPC solvers or other tools currently used or plan to use)
    * Data Analytics, ML requirements: do you need to apply Data analytics or machine learning in this stage?
    * Other required functionalities?
    * Data inputs: File format, Size of data
    * Data outputs: File format, Size of data

WP4: ML-based sub-modeling

- Does your LHC require ML-base sub-modeling?
- How will your LHC benefit from this?
- Are you able to generate training data from the LHC or, due to computational limitations, are sub-cases required?
- If you use ML sub-models, give a short description.

WP4: Uncertainty quantification

- Does your LHC require uncertainty quantification?
- How will your LHC benefit from this?
- Which uncertain parameters are of interest? How many?

WP4: Dynamic resource management

- Does your LHC require dynamic resource management?
- How will your LHC benefit from this?

Bibliography

[1] Martin Aumüller. Hybrid Remote Visualization in Immersive Virtual Environments with Vistle. In Hank Childs and Steffen Frey, editors, *Eurographics Symposium on Parallel Graphics and Visualization*. The Eurographics Association, 2019.

[2] H. Jane Bae and Petros Koumoutsakos. Scientific multi-agent reinforcement learning for wall-models of turbulent flows. *Nature Communications*, 13(1):1–9, 2022.

[3] Haecheon Choi and Parviz Moin. Grid-point requirements for large eddy simulation: Chapman's estimates revisited. *Physics of fluids*, 24(1), 2012.

[4] Javier Álvarez Cid-Fuentes, Salvi Solà, Pol Álvarez, Alfred Castro-Ginard, and Rosa M Badia. dislib: Large scale high performance machine learning in python. In *2019 15th International Conference on eScience (eScience)*, pages 96–105. IEEE, 2019.

[5] Jakob Duerrwaechter, Thomas Kuhn, Fabian Meyer, Andrea Beck, and Claus-Dieter Munz. Pounce: A framework for automatized uncertainty quantification simulations on high-performance clusters. *Journal of Open Source Software*, 8(82):4683, 2023.

[6] Jorge Ejarque, Rosa M Badia, Loïc Albertin, Giovanni Aloisio, Enrico Baglione, Yolanda Becerra, Stefan Boschert, Julian R Berlin, Alessandro D'Anca, Donatello Elia, et al. Enabling dynamic and intelligent workflows for hpc, data analytics, and ai convergence. *Future generation computer systems*, 134:414–429, 2022.

[7] Marta Garcia, Jesus Labarta, and Julita Corbalan. Hints to improve automatic load balancing with LeWI for hybrid applications. *Journal of Parallel and Distributed Computing*, 74(9):2781–2794, 2014.

[8] Meta Platforms, Inc. Zstandard compression. `https://github.com/facebook/zstd`. Accessed: 2023-09-28.

[9] Niclas Jansson, Martin Karp, Artur Podobas, Stefano Markidis, and Philipp Schlatter. Neko: A modern, portable, and scalable framework for high-fidelity computational fluid dynamics. *ArXiv*, 2107.01243, 2021.

[10] Marius Kurz, Philipp Offenhaeuser, and Andrea Beck. Deep reinforcement learning for turbulence modeling in large eddy simulations. *International Journal of Heat and Fluid Flow*, 99:109094, 02 2023.

[11] Marius Kurz, Philipp Offenhaeuser, Dominic Viola, Michael Resch, and Andrea Beck. Relexi—A scalable open source reinforcement learning framework for high-performance computing. *Software Impacts*, 14, 09 2022.

[12] Peter Lindstrom. Fixed-rate compressed floating-point arrays. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2674–2683, 2014.

[13] Adrián Lozano-Durán and Bae. Self-critical machine-learning wall-modeled les for external aerodynamics. Annual research briefs 2020, Center for Turbulence Research, 2020.

[14] Adrián Lozano-Durán and H Jane Bae. Machine learning building-block-flow wall model for large-eddy simulation. *Journal of Fluid Mechanics*, 963:A35, 2023.

[15] Saleh Rezaeiravesh, Ricardo Vinuesa, and Philipp Schlatter. Uqit: A python package for uncertainty quantification (uq) in computational fluid dynamics (cfd). *Journal of*

Open Source Software, 6(60):2871, 2021.

[16] Peter P. Sullivan and Edward G. Patton. The effect of mesh resolution on convective boundary layer statistics and structures generated by large-eddy simulation. *Journal of the Atmospheric Sciences*, 68(10):2395–2415, 2011.

[17] Enric Tejedor, Yolanda Becerra, Guillem Alomar, Anna Queralt, Rosa M Badia, Jordi Torres, Toni Cortes, and Jesús Labarta. Pycompss: Parallel computational workflows in python. *The International Journal of High Performance Computing Applications*, 31(1):66–82, 2017.

[18] Patrick Vogler and Ulrich Rist. Handling large numerical data-sets: Viability of a lossy compressor for cfd-simulations. In Michael M. Resch, Manuela Wossough, Wolfgang Bez, Erich Focht, and Hiroaki Kobayashi, editors, *Sustained Simulation Performance 2019 and 2020*, pages 97–110, Cham, 2021. Springer International Publishing.

[19] XIA Yang, Suhaib Zafar, J-X Wang, and Heng Xiao. Predictive large-eddy-simulation wall modeling via physics-informed neural networks. *Physical Review Fluids*, 4(3):034602, 2019.

[20] Zhideng Zhou, Guowei He, and Xiaolei Yang. Wall model based on neural networks for les of turbulent flows over periodic hills. *Physical Review Fluids*, 6(5):054610, 2021.