D5.6 Implementation Report on Pilot Applications

Date: July 18, 2024

Document Identification			
Status	Review	Due Date	30/06/2024
Version	1.0	Submission Date	18/07/2024

Related WP	WP5	Document Reference	D5.6
Related Deliverable(s)	D5.7, D5.8	Dissemination Level (*)	PU – Public
Lead Participant	FAU	Lead Author	Harald Koestler
Contributors	UNISTRA, MTG, PSNC, FAU, SZE	Reviewers	Piotr Kopta (PSNC)
			Petros Anastasiadis (ICCS)

Keywords:
Pilot applications, Implementations, initial development actions, use case definition, potential software coupling, challenges faced, implementation plans

Document Information

List of Contributors	
Name	**Partner**
Michael Zikeli	FAU
Ravi Kiran Ayyala Somayajula	FAU
Harald Koestler	FAU
Michał Kulczewski	PSNC
Wojciech Szeliga	PSNC
Krzysztof Kotecki	PSNC
Christophe Prud'homme	UNISTRA
Mátyás Constans	SZE
József Bakosi	SZE
László Környei	SZE
Luis Torres	MTG
David Caballero	MTG
Ángela Rivera	MTG

Document History			
Version	**Date**	**Change editors**	**Changes**
0.1	12/06/2024	FAU	ToC, first draft
0.15	14/06/2024	Marcin Lawenda (PSNC) Harald Koestler (FAU)	ToC, timeline and responsibilities approved
0.2	02/07/2024	FAU	Second draft (after initial review)
0.3	03/07/2024	UNISTRA	updates from reviewer feedback
0.4	03/07/2024	SZE	Include content for UAP
0.5	05/07/2024	FAU	Prepare file for second reviewing round
0.6	15/07/2024	FAU	Version after second review
0.7	17/07/2024	FAU	Prepare a file with right formatting

0.8	17/07/2024	FAU	Insert content in formatted file
0.99	17/05/2024	Harald Koestler	Quality assurance check
1.0	18/07/2024	Marcin Lawenda	Final check

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable Leader	Harald Koestler (FAU)	17/07/2024
Quality Manager	Harald Koestler (FAU)	17/07/2024
Project Coordinator	Marcin Lawenda (PSNC)	18/07/2024

Table of Contents

List of Figures

List of Acronyms

Insert here all the acronyms appearing along the deliverable in alphabetical order.

Abbreviation / acronym	Description
AI	Artificial Intelligence
CD	Continuous Delivery or Continuous Deployment
CI	Continuous integration
CI/CD/CD	Continuous integration/Continuous Delivery/Continuous Deployment
Dx.y	Deliverable number y belonging to WP x
EC	European Commission
Feel++	Finite Element Embedded Library in C++
HPC	High-Performance Computing
MTW	Material Transport in water
PDE	Partial Differential Equation
RES	Renewable Energy Sources
UAP	Urban Air Project
UB	Urban Building
UBM	Urban Building Model
UBEM	Urban Building Energy Modeling
WP	Work Package

Executive Summary

As part of the HiDALGO2 project, D5.6 marks a critical milestone in our pursuit of harnessing High-Performance Computing (HPC) to address global challenges through our pilot applications. This deliverable documents the initial outcomes of our pilot implementations, setting a foundation for subsequent advancements and integration strategies. Covering the period up to Month 18, D5.6 provides the first insight into the development of our pilot applications, namely the Urban Air Project (UAP), Urban Buildings (UB), Renewable Energy Sources (RES), Wildfires (WF), and Material Transport in Water (MTW).

This deliverable demonstrates the capabilities of our innovative computational solutions which are developed to address urban air quality, energy efficiency in buildings, renewable energy forecasting, wildfire risk assessment, and pollution transport in water. Each pilot has implemented initial scenarios to address support for city policymakers, energy producers, and disaster management authorities.

On top of general advancements in the development of these pilots, this deliverable also outlines a strategic roadmap for the ongoing research and development activities for the forthcoming deliverable D5.7 and D5.8. It sets the stage for the next steps in simulation enhancements, by introducing various simulation tools employed and the potential couplings between them. This deliverable only aims to describe couplings between the various software tools to get a better understanding of the pilots workflow. It does not aim to describe the coupling between pilot scenarios, as this is already described in D5.1 [7].

1 Introduction

1.1 Purpose of the document

The primary purpose of this document is to meticulously record the detailed implementations of each task under Work Package 5 (WP5). WP5 encompasses several critical use cases, namely Urban Air Project (UAP), Urban Building (UB) Model, Renewable Energy Sources (RES), Wildfires, and Material Transport in Water (MTW). Each use case leverages sophisticated computational models, High-Performance Computing (HPC), and Artificial Intelligence (AI) technologies. The objective is to demonstrate the scalability and practical application of these technologies in real-world scenarios. This document aims to provide a comprehensive overview of the current implementation status of these use cases, detailing the various software packages employed and exploring potential couplings between these software solutions. Additionally, it outlines the anticipated technical advancements and developments expected to be achieved by the time of the forthcoming deliverable 5.7, due in month 29 of the project timeline.

1.2 Relation to other project work

This document is integrally connected to other aspects of the broader project, serving as a crucial component that aligns with and supports various other work packages. The implementations and findings reported here feed into the overall project objectives by providing essential work flows and insights that contribute to the development and optimization of the project's computational models and technologies. The synergies between WP5 and other work packages ensure a cohesive approach to achieving the project's goals. For instance, the integration of HPC and AI technologies in these use cases not only showcases their potential in specific applications but also provides a foundation for their application in other areas of the project. The collaborative nature of this work ensures that advancements in WP5 can be leveraged to enhance the overall project outcomes, facilitating knowledge transfer and the sharing of best practices across different domains.

1.3 Structure of the document

The document is organized to present a clear and coherent flow of information, enabling a thorough grasp of the implementation processes and results of each HiDALGO2 pilot application. The pilot applications are detailed in their own section, with the Urban Air Project (UAP) in section 2, Renewable Energy Sources (RES) in section 3, Urban Buildings (UB) in section 4, Wildfires (WF) in section 5, and Material Transport in Water (MTW) in section 6.

Each pilot section comprises five subsections. These subsections cover the pilot's definition and motivation (subsection x.1), specific implementations and used software packages (subsection x.2), computational models and potential connections between different software packages (subsection x.3), the current development status (subsection x.4), and an outlook on the expected technical advancements (subsection x.5).

These sections are designed to provide a comprehensive overview of the methodologies and technologies utilized, along with the encountered challenges and devised solutions strategies, providing a guide for future work and emphasizing key milestones and objectives that lead to the upcoming deliverable 5.7.

2 Urban Air Project Implementation status

2.1 Use case definition and motivation

The primary aim of the Urban Air Project is to create a realistic model for microscale urban climate which enables modelling the spread of air pollution, and so assessing the air quality of a city on a microscale level. Having a well validated climate model also enables wind comfort modelling and forecasting the spread of hazardous aerosols.

Creating and using reduced models drastically increases computational performance and forecasting capabilities of the models. The UAP workflow also showcases several tools to facilitate mesh generation, simulation deployments and visualization.

2.2 Description of Software Packages

The Urban Air Project pilot within the HiDALGO2 initiative utilizes a range of advanced software packages to meet its objectives. These tools are essential for the modelling, simulation, and analysis needed to evaluate and enhance urban air quality. There are three primary simulation tools employed in the pilot.

REDSIM

RedSIM is a finite-volume method CFD solver, with an emphasis on high performance in GPU environments, multi node CPU environments and a model order reduction for up to 50x speedups in computation time. It solves the compressible Euler equations and aims to be a general CFD solver; it can currently solve a wide range of problems, ranging from acoustic problems such as exhaust pipes, to urban air pollution problems on multiple kilometres. Most compressible FVM codes are targeted at multi-node CPU architectures with MPI; we have developed REDSIM with an emphasis on GPU architectures as our priority (we achieve over 90% efficiency for meshes with over 100M cells), although we also heavily optimized our MPI CPU implementation. RedSIM can simulate on arbitrary polyhedral meshes, both in 2D and 3D. Thanks to a heavy amount of metaprogramming, the 2D and 3D codepaths are identical without any runtime cost; some codepaths are manually optimized with SIMD. It is currently a closed-source software; we do not distribute source-code directly, but instead distribute an MPI and CUDA version (redsim_mpi and redsim_cuda respectively) compiled on most major EUROHPC machines: LUMI, KAROLINA and VEGA. REDSIM has a LUA scripting interface; every config file given to RedSIM is a LUA script, allowing for a lot of expressiveness (mesh generation, boundary and initial condition, etc.). The documentation for REDSIM's LUA API is available for the general public on our webpage, https://redsim.mathso.sze.hu/.

XYST

Xyst is a Navier-Stokes solver for engineering flows. The goal of Xyst is to simulate engineering problems with a production-quality code that is extensible and maintainable, using hardware resources efficiently, even for problems with unpredictable, heterogeneous, and dynamic compute-load distribution. The software implementation facilitates the effective use of hardware of any size, from laptops to the largest distributed-memory clusters, by combining data-, and task-parallelism on top of the Charm++ runtime system. Charm++'s execution model is asynchronous by default, allowing arbitrary overlap of computation and communication. Built-in automatic load balancing enables redistribution of heterogeneous computational load based on real-time CPU load measurement at negligible cost. The runtime system also features automatic checkpointing, fault tolerance, resilience against hardware failure, and supports power-, and energy-aware computation. Xyst's fundamental choices of data structures, algorithm, and software design specifically target engineering resolution and accuracy requirements. Some of these choices are: unstructured grids (to explicitly resolve complex 3D geometries), tetrahedra-only computational elements (to enable automatic mesh generation), edge-based finite element scheme (to reduce indirect addressing for increased performance), distributed-memory parallel computing paradigm (to enable large problems), and Charm++ as the runtime system (to effectively use computing resources even in the presence of hardware heterogeneities and dynamic application requirements).

UAP-FOAM

UAP-FOAM is an OpenFOAM based simulation framework for microscale (1 meter) urban air flow and air pollution analysis with multicomponent pollutant tracking, atmospheric flow simulation. It is integrated into UAP with automated post-processing, portal based deploying on several HPC systems. It has been developed since 2019. It is deployed and is highly scalable on multiple EuroHPC systems.

Additional tools

There are additional tools used in the UAP framework for simulation workflow:

UAP-SNAPPY is an OpenFOAM based workflow for creating octree and polyhedral meshes for urban area climate models. It facilitates usage of the snappyHexMesh mesher of OpenFOAM and integrates it into the UAP workflow.

OCTREEMESHER is an SZE-developed JAVA based application for octree mesh generation for urban area climate models. It is integrated into the UAP workflow.

POLYTOPE is an interface developed in the previous HiDALGO project. It interfaces the pollution spread simulation with global scale weather simulations of ECMWF

SUMO is a traffic simulation software for estimating vehicle movement within a city based on predefined parameters

CFDR We also developed REDSIM with digital-twin functionalities in mind, integrating it completely into our visualization software CFDR: REDSIM handles some visualization computations, such as slices, and can change the way those visualizations are done through our web-based visualization tool (thus, making it a digital twin). UAP-FOAM can also take advantage of CFDR functionalities.
MATHSO portal is an integrated webui-orchestrator-simulation deployment system, which provides a web-based user interface for configuring simulations, submitting them to EuroHPC machines and presenting results. MathSO also supports integration of new HPC workflows, integration and deployment with its blueprint system.

Additional tools that are used but are not developed by our teams, like singularity or ReFrame are not described in this section.

2.3 Established and Potential Software Couplings

The workflow of the UAP application is implemented in TOSCA language. It orchestrates the pre-processing, simulation and post-processing. For the simulation component, the OpenFOAM-based UAP-FOAM solver is fully integrated, RedSim partially done for the wind comfort simulations. SZE has started integrating Xyst since its entering to the HiDALGO2-project, after the amendment in M15. The UAP workflow is running from the HiDALGO2-portal, see https://portal.hidalgo2.eu/ and from the MathSO-portal (https://portal.mathso.sze.hu/#/). In this way, we can set up the UAP-application's parameters, execute the jobs in EuroHPC machines, and visualize the results in a unified way, independently from the particular CFD-solvers.

XYST

Xyst is not currently coupled to any other software. One future idea is to use the fluid solvers in Xyst to enable the simulation of engineering problems involving interacting fluids and solids. Various computational technologies for fluid-fluid and fluid-structure interaction exist and are used in both academia and industry. However, current large simulations typically target only single physics, such as fluids or solids, and industrial codes with engineering capability, involving coupled physics and complex geometries, rarely scale beyond O(10) compute nodes with problem sizes of O(10^9) degrees of freedom.. The unique load balancing of Charm++ and Xyst would enable the effective use of larger hardware for this area of coupled physics.

UAP-FOAM

The UAP-FOAM application is coupled with ECMWF global scale weather simulations with polytope and SUMO traffic simulations for calculating emissions from vehicle traffic.

2.4 Current status of software capabilities as of M18

RedSIM

As of M18, as planned in the milestones, REDSIM supports running on a single node with multiple CUDA capable GPU-s, which was tested on KAROLINA and VEGA and also running on multiple CPU nodes thanks to OpenMPI, which was tested on LUMI. RedSim supports an expressive scripting interface in LUA, publicly accessible on RedSim's webpage. A digital twin mode with CFDR is also developed for visualization, so slices can be dynamically moved and computed on the fly.

XYST

Xyst currently contains multiple distributed-memory-parallel fluid solvers for complex 3D engineering geometries. Computational domains of arbitrary shapes are discretized into tetrahedron elements and decomposed into small chunks assigned to different CPUs. The number of chunks may be more than the number of CPUs, allowing over decomposition, useful for effective cache utilization and automatic load balancing. The solution along partition boundaries, that exists on multiple processing elements, is made consistent with asynchronous communication which hides latencies by enabling overlapping of computation and communication.

The numerical methods belong to the family of continuous Galerkin finite element methods storing solution values at nodes of the computational mesh. Currently, there are 4 fluid-density-based solvers, intended for the mathematical modelling of energetic, high-speed, compressible flows, each with its unique features (advantages and disadvantages compared to each other). Details on the methods and their characteristics, together with verification and validation examples are documented at https://xyst.cc/inciter_main.html and in the peer-reviewed journal papers listed at https://xyst.cc/papers.html .The full source code is available under the GNU General Public License. Scalability and performance of two of the solvers in Xyst are quantified at https://xyst.cc/riecg_performance.html and https://xyst.cc/zalcg_performance.html.

Recent work in Xyst has concentrated on the following two main areas. First, develop, verify, validate, document, and publish a novel "partition deactivation" algorithm, implemented in Xyst's ZalCG solver. ZalCG is the fastest solver in Xyst, featuring a unique partition deactivation procedure, which increases performance for propagation phenomena, such as urban-scale (e.g., contaminant) dispersion scenarios. Further details on this work are available at https://xyst.cc/inciter_zalcg.html and https://xyst.cc/vnv.html#zalcg_vnv, while documentation of the algorithm is currently under review in Journal of Computational Physics. Second, implement, verify, validate, and document a time-derivative preconditioning technique in Xyst as a new solver, LaxCG. LaxCG enables the effective computation of flows at all Mach numbers. See https://xyst.cc/inciter_laxcg.html for more details.

UAP-FOAM

The UAP-FOAM has the following major capabilities. Some of these are inherent in the OpenFOAM toolbox, however, the majority is developed by the UAP team. It supports steady and unsteady simulation of incompressible air flow in urban scale. These both are coupled, so unsteady calculations start from steady state flow. Both unsteady and steady support multi component pollution spread simulation with convection-diffusion equations. Equations are coupled one way, so pollution spread does not influence air flow. With regards to coupling, boundary conditions follow steady and time dependent atmospheric boundary layer profile. Custom, height and time dependent profile based on ECMWF data is also supported. Cell based, time varying volumetric source terms for pollutants are supported. These source terms may come from traffic simulation or external pollution data. With regards to sampling, surface sampling of velocity, pressure and pollutant are supported at predefined heights. Point sampling of the same variables are supported at predefined locations. Calculation of the total amount of pollutants is supported for predefined volumes. Most of the features are continuously integrated into the MathSO portal, including visualization with CFDR. There are various additional features, where point source terms can be specified with pollutant, start time, stop time and intensity and is integrated into the MathSO CI/CD system. The simulation is validated with a 1-year scenario of the city of Antwerp of 2016.

2.5 Challenges faced and future plans

RedSIM

Our main challenge that we have started tackling with REDSIM is an MPI-based multi-GPU solution. Currently, when running redsim_cuda, we can only run REDSIM on an individual GPU node, on multiple GPU-s. We are working on an implementation, where we can leverage MPI in order to run REDSIM on as many GPU-s as available for a given HPC machine.

A secondary future plan would be to implement an OpenCL version of the GPU code, in order to run on the AMD GPU-s available on LUMI. Finally, we are also in the middle of working on a concentration computation module, with a linear solver, to be able to approximate pollution levels properly. A version with advection only is already functional.

XYST

Current work in Xyst is focused on developing a projection-based fluid solver for constant-density (incompressible) flows. Components of this solver will include semi-implicit time integration and a preconditioned conjugate gradients method for the solution of linear systems for pressure and momentum. As all solvers in Xyst, the implementation will be parallel targeting the largest distributed-memory machines with asynchronous communication and automatic load balancing.

Projection-based algorithms are among the most effective and efficient ones for the time-accurate simulation of constant-density flows as required in applications of urban-scale air flow calculations. A unique challenge of projection solvers is to achieve an efficient and scalable performance of the solution of the elliptic equation required to compute the fluid pressure. To attack this problem-area we are planning to explore various options for its linear solver, such as preconditioners of different complexity and performance, e.g., algebraic multigrid, using the asynchronous parallel programming paradigm provided by Charm++.

UAP-FOAM

The following features are planned for UAP-FOAM, excluding coupling and integration: simulation and validation of chemical reactions between pollution species, thermal convection and Solar radiation. The development and validation of our models on additional cities is also planned. There are some new versions of OpenFOAM available, that inherently support GPU, integrating this version into UAP-FOAM is among the next steps.

3 Renewable Energy Sources Implementation status

3.1 Use case definition and motivation

The Renewable Energy Sources (RES) pilot is driven, among others, by the requirements of one of the largest DSO in Poland. It addresses DSO's needs by providing a high-resolution, detailed weather prediction, which can be then correlated with data from wind farms and photovoltaic panels to predict how much energy can be produced. The understanding of physical phenomena behind renewable energy sources is the key to control and increase productions of energy from these kind of energy sources. Such gain is important for individual users to optimise their energy consumption between traditional sources and RES, but also for DSOs to stabilise the grid and optimise incomes from trading energy in the energy market. The detailed use definition and motivation was described in deliverable D5.3.

3.2 Description of Software Packages

RES is a multiscale application which combines mesoscale and regional weather forecast model with a local detailed one for accurate modelling of wind flows over complex terrain topography and through building structures for urban areas. The same models are of interest of solar energy plants owners and operators as they can be applied for the prediction of solar energy acquisition using detailed sun/shade forecasts. Yet another scenario RES implements is the prediction of damages to infrastructure. The aforementioned models are **WRF** and **EULAG**. The former is an open-source, community-based model. The latter is an all-scale geophysical flow solver, which in RES is tailored towards simulating flows over complex terrain topography and in urban areas. The models are encapsulated into RES-runner, a framework written in Python to manage the whole workflow - starting from pre-processing of the data and setting up the initial and boundary conditions, through execution of the models in HPC environment, ending with post-processing of the data and visualisation. More details are given in deliverable D5.3. For each scenario there is a separate version available: i) RES-wind for wind farms forecasts, ii) RES-solar for photovoltaic systems forecast, and iii) RES-damages to prediction of damages to overhead electrical network or RES infrastructure due to extreme weather events.

3.3 Established and Potential Software Couplings

RES is already coupled with WRF mesoscale weather prediction model. There are plans for an additional coupling with WRF-MG provided by MeteoGrid. Moreover, sun/shade model available in UB pilot could be useful in RES. These potential coupling with other pilots are described in details in deliverable D5.1. Here focus is given on technological coupling with other software.

Weather Research and Forecasting model

The computational core of RES are two independent weather forecasting applications, each providing results for different scale range. The first, WRF, is a mesoscale solver capable of providing a spatial resolution of the forecast up to scale of ca. 1000 meters, considering its internal models applicability. The second one, EULAG, is an all-scale geophysical flow solver, which can be applied to more fine-grained scenarios and therefore it can provide a forecast of finer resolution and including multiple local factors, such as city buildings, forests, detailed topography etc. In RES, the output of WRF is automatically passed as input data to EULAG. The combination of both gives an opportunity to receive results which on one hand are detailed and on the other hand take into account the bigger image. What is more, while a single pipeline includes a single WRF simulation, there can be multiple instances of EULAG run in parallel to enhance multiple areas of the domain covered by WRF simulation.

ReFrame

For the benchmarking tasks to be done in the HIDALGO2 project, an integration of RES framework with ReFrame benchmarking toolkit was done. Modifications were made both to RES orchestrator and ReFrame input scripts. In order to ensure a simple and straightforward data exchange between both applications the logging functionality of RES was refactorized and additional benchmarking messages were added to the code. The ReFrame input scripts were adjusted specifically for the RES application. It was ensured that every benchmarking results are automatically uploaded to the appropriate project repository. A description of the execution workflow using ReFrame is described within section 2.2.5 below**Error! Reference source not found.**.

mUQSA and QCG Portal

mUQSA is multiscale uncertainty quantification (UQ) and sensitivity analysis (SA) toolbox. Written in Python and based on SEVEA toolkit, it allows in ease manner to run UQ and SA analysis for any application. In turn, QCG-Portal provides a GUI to run computational experiments on HPC resources, including uncertainty quantification and sensitivity analysis scenarios defined in mUQSA. RES-damages has been integrated with both mUQSA and QCG Portal. A generic interface and batch-job template are used for this purpose, and for the future it is planned to provide RES-specific template in the portal so that input parameters, model configuration and all workflow can be defined by the user, with visualisation of simulation results presented directly in the user’s web browser. mUQSA is also used as a standalone toolbox to apply UQ and SA to RES-damages without the QCG-Portal as a GUI frontend.

Visualisation

There is potential coupling planned with HiDALGO2 visualisation software, in particular with CFDR (SZE), UnrealEngine (MeteoGrid) and CAVE/Covise (HLRS), details of which can be found in deliverable D4.6.

3.4 Current status of software capabilities as of M18

Automatic workflow with pre- and post-processing

The main purpose of the RES-runner is providing an orchestrated multiscale weather forecasting tool dedicated for renewable energy sources operators. The workflow implemented in the RES framework is designed to provide to a user a complete pipeline orchestration required to obtain ready-to-use weather simulation results. RES binds multiple software packages and functional libraries. It is designed in a way which significantly simplifies obtaining the data required and reduces the effort necessary to execute the workflow. RES also provides solutions for raw data post-processing, improving the overall user-friendliness and offering a holistic environment.

The input data required by WRF and EULAG models, key RES components, is rather complicated and requires extra care with multiple physical and numerical parameters and data sources which can easily become a tedious task. Besides the physical correctness and ensuring numerical stability, one needs to correctly set up simulation details for in order to ensure successful data exchange. The RES framework wraps both solvers in a way which allows to automatize setting up majority of the required data in a way suitable for renewable energy sources-related scenarios. The user can describe the simulation with a minimalist data provided in a single JSON file. The data includes such information as location, terrain size, spatial resolution and date range. If EULAG stage is to be included, then paths to mesh file and a CSV file with locations of renewable energy sources facilities is also required. If necessary, a lot of additional flags and options are available for the user.

RES is designed to be executed on an HPC environment controlled with SLURM queuing system. The framework includes a dedicated module for communication with SLURM in order to send jobs, control their status and administer all the stages of the workflow. In RES every computationally-intensive process is realized as a separate SLURM job in order to ensure optimal HPC resources usage and possibly parallelize the pipeline when possible to shorten the overall RES instance execution time. If necessary, the user can switch off this functionality and realize the whole pipeline with a single task if, for example, waiting time for every job is especially long.

In order to ensure a deep understanding of the realized stages by the user, RES provides deep and detailed integrated logging functionality which is done with a dedicated Python library. When possible, if any error occurs, the framework provides an informative description of the possible cause. The user can also use a dedicated flag to turn on logging benchmarking information for the performance analysis.

Additional effort was made to ensure customized post-processing stage, thanks to which the user is not left only with large volumetric files which contain a lot of data to be processed. RES uses dedicated libraries to generate PNG images of the domains' cross-sections including the most important physical variables. The images are also merged into GIF animations. RES also extracts point data for predefined locations of renewable energy sources facilities.

The whole RES workflow includes various stages depending on the choice of the user (some stages can be skipped if not necessary). The first step is a detailed input data and input flags check before more computationally intensive processes are started. The framework checks types, values and ranges of all the given parameters to ensure that the user-defined problem forms numerically and physically reasonable setup which can be successfully calculated. After the sanity check is done, RES initializes the working directory and downloads from the public server of NOAA the most up-to-date global GFS forecast for the user-defined time range in order to provide input data required by WRF. Afterwards, pre-processing tools of WRF, known as WPS, are executed to preprocess the input data. Then the actual solver of WRF is executed as a separate SLURM job. Parallel to WRF execution, compilation of all the user-defined EULAG instances is also done with separate jobs. When WRF simulation is done and RES confirms that all the required files are saved, it executes all the EULAG instances creating multiple SLURM jobs. After all the simulations are done, the last stage - post-processing starts and generates PNG, GIF and CSV files.

RES-damages, uncertainty quantification and sensitivity analysis

In RES-damages pilot, uncertainty quantification and sensitivity analysis is applied, by the means of the aforementioned mUQSA toolbox, to estimate how much extreme weather events may impact the overhead electrical network. The exemplary study was conducted in one of Polish cities, focusing on excessive wind speed and wind gusts. In the workflow, three levels of domains nesting were used, where the outer domain of the scale of the country had a resolution of 3.6 km, while the innermost domain was discretised with the spacing of just 100 meters. In this study, custom overhead electrical network was prepared for the sake of this study (at the moment real data covering existing electrical network is not to be published). The high level of detail allowed the simulation to provide meteorological results individually for each component of the infrastructure. Different weather conditions such as wind speed, wind directions and wind gusts are considered uncertainties for the ensembles generations and further analysis. With the mUQSA toolkit we are able to select different minimum, maximum and mean value for each of the uncertainty alongside their distribution model. A user can provide details on UQ method to be applied with different parameters related to it. UQ results, provided in graphical form, present how much each of the site is prone to excessive wind speed and gusts. Sensitivity analysis results, also presented in graphical form, give a user an information whether given site is more vulnerable due to wind speed or wind direction. Details of the solution are given in the paper "Fostering uncertainty quantification in Global Challenges with mUQSA toolkit" submitted to 15th International Conference on Parallel Processing & Applied Mathematics (accepted).

3.5 Challenges faced and future plans

In deliverable D5.3, challenges and plans were outlined. This section provides an update to those.

Dockerised version

The complex structure, multilevel workflow and different programming languages and libraries used in RES were the first reason to introduce the containerization. The second one is portability of the system. The third one are some limitations present at some EuroHPC JU machines, like a limited number of files that can be installed by a single user, which are obstacles that cannot be avoided without a container. The workflow of the RES is following:

1. Reframe (python code) executed as 1 task
2. RES (python code) executed by Reframe as 1 task
3. WRF (Fortran code) executed by RES as multi-task
4. Compilation of the Eulag (Fortran code) executed by Reframe as 1 task
5. Eulag (Fortran 77 code) executed by Reframe as multi-task

The challenge here is to run this non-MPI and MPI applications inside a single container on HPC machine with SLURM queueing system because of some execution issues. Different approaches were tested, including the hybrid model (MPI is available on host machine and container), bind model (MPI from host in mounted to the container) and PMI model (MPI available only in container). The probable solution is to divide the workflow between different executions of container.

HPDA

RES runs daily to produce damage predictions on the electrical overhead network in one of the largest Polish cities. Forecasts on the amount of energy produced will be run similarly. In either case, the data produced does not require HPDA techniques. These will be required once ensemble runs are conducted. The RES pilot plans to run ensembles with different sets of parameters – different mesoscale weather prediction models, different parametrisation, different input parameters, etc. Ensembles are now used in RES damages scenario to check how wind speed and direction affect the probability of the damages to the overhead electrical network. For efficient parameter value generation and orchestration of ensembles runs within the HPC environment, the mUQSA toolkit is used (described in Deliverable D2.1). While it provides basic data analytics capabilities, such as calculating the average value of the damages probability, a proper HPDA solution is required to analyse dozens and hundreds of 3D and 4D data in the most efficient way to provide, e.g. mean results or probability distribution. We plan to start using HPDA tools for the ensembles by M24.

Wind and solar energy modules

Applying AI is one of possible techniques to find a correlation between weather forecast and amount of electrical energy produced. AI can be used in a form of predictive model based on the analysis of sequential data, i.e. so-called time series, to enable more effective forecasting in the field of demand-supply. Such information would improve the way the network is managed by energy system operators, improve network security and enable the transformation of the network to a flexible model, where it would be possible to introduce new prices for energy based on the knowledge of what is happening in the network. Both wind and photovoltaics will require separate models.

The input data (weather forecast) to the AI model is already available and produced on a daily basis, though access to real data for verification is required. The process of obtaining this data is ongoing.

The output data is historical data regarding energy generation. For initial work, synthetic data will be used. As of real data, PSNC is finalising the installation of a large PV system (almost 1MWp) and will start collecting and storing the data for further processing. For wind farms, the data is planned to be obtained through cooperation with one of the largest DSOs in Poland.

We aim at providing first version of the energy module by M24.

Uncertainty quantification and sensitivity analysis

Applying UQ and SA will help in:

- Determining which input data and/or model parameters can be omitted to lower the number of ensembles needed;
- Improving the results by lowering the uncertainties;
- Analysing if and to what extent domain size and mesh resolution impacts the results.

These activities are planned and pending. The mUQSA can also help in accelerating the computations by replacing certain parts of the EULAG model with surrogates. We started to work on surrogates by applying EasySurrogate (https://github.com/wedeling/EasySurrogate). Based on the results we will continue with EasySurrogate or use other techniques based on AI.

4 Urban Building Implementation status

We detail in the following sections the implementation status of the Urban Building pilot. More details are available in the reference [6].

4.1 Use case definition and motivation

The Urban Building pilot within the HiDALGO2 project aims to create a detailed and scalable simulation framework to accurately assess and manage the energy performance of buildings within urban environments. The pilot focuses on several key areas: energy performance assessment, simulating energy consumption and efficiency at both the individual building level and the broader urban scale, and evaluating thermal comfort and indoor air quality. It also includes building stock integration by incorporating various data sources, such as GIS data, Building Information Modeling (BIM), and OpenStreetMap, to generate accurate 3D models of urban buildings and integrating building simulations with environmental models to assess the impact on urban air quality and heat islands. Optimization and planning are addressed by identifying potential areas for energy savings and efficiency improvements, comparing and evaluating different renovation and energy management strategies, and providing data-driven insights for urban planners and policymakers to optimize urban development and renovation projects. The pilot employs advanced simulation techniques, such as radiative heat transfer modelling and computational acceleration strategies, to enhance the precision and scalability of simulations, and it includes coupling with the Urban Air Pollution (UAP) model to provide a comprehensive environmental impact assessment.

The motivation for the Urban Building pilot stems from the significant role that buildings play in energy consumption and greenhouse gas emissions within the European Union. Key motivations include reducing environmental impact, as the building sector is responsible for 40% of final energy consumption and 36% of greenhouse gas emissions in the EU. Achieving the EU's Horizon 2050 objectives, which aim to double annual energy renovation rates over the next decade, requires innovative and scalable solutions. The pilot supports policy objectives by providing accurate data and simulations to support the EU's energy efficiency and renovation targets, enabling policymakers to make informed decisions based on comprehensive simulations that incorporate both energy performance and environmental impact. Enhancing urban sustainability is also a key motivation, as improving the energy efficiency of buildings contributes to the overall sustainability of urban areas, and simulations can help identify the most effective strategies for reducing energy consumption and improving air quality. The pilot addresses urban challenges related to energy use, air quality, and thermal comfort by providing tools and insights to tackle these issues through detailed and accurate simulations. Finally, advancing scientific and technological innovation is a motivation, as developing advanced simulation methodologies and integrating various data sources push the boundaries of current scientific and technological capabilities, creating a scalable and robust simulation framework that can be applied to different urban contexts and scenarios.

In summary, the Urban Building pilot is driven by the need to reduce the environmental impact of buildings, support policy objectives, enhance urban sustainability, address complex urban challenges, and advance scientific and technological innovation. Through detailed simulations and data integration, the pilot aims to provide valuable insights and tools for optimizing energy performance and improving the quality of urban environments.

Ktirio is the name of our advanced simulation platform developed in parts within the HiDALGO2 project, designed to enhance urban building modelling and energy performance analysis. We call Ktirio Urban Building the Hidalgo2 pilot later on. Ktirio offers a comprehensive suite of tools and functionalities that enable detailed simulations of building energy consumption, thermal comfort, and environmental impact at various levels of detail (LOD). The platform integrates advanced modelling techniques, data management capabilities, and high-performance computing resources, making it a robust and scalable solution for urban planners and policymakers seeking to optimize urban development and energy efficiency.

Finally let's describe the level of details that we use: LOD-0 (Level of Detail 0) is the most basic level of detail in modelling, representing buildings and urban elements with minimal complexity. It features simplified geometries, often depicted as block shapes without detailed architectural features, making it suitable for large-scale city-wide simulations where high detail is not necessary. LOD-1 (Level of Detail 1) offers an intermediate level of detail, including more geometric and architectural features than LOD-0. This level provides basic building shapes with some architectural elements such as roofs and external walls, making it suitable for simulations that require a moderate level of detail, such as initial energy performance assessments and urban planning. LOD-2 (Level of Detail 2) represents a higher level of detail, providing a more accurate and detailed representation of buildings and urban environments. This level includes detailed geometries featuring individual building components such as windows, doors, and structural elements, making it ideal for precise simulations of energy consumption, thermal comfort, and environmental impact assessments.

4.2 Description of Software Packages

The Urban Building pilot within the HiDALGO2 project leverages a variety of advanced software packages to achieve its objectives. These software tools are integral to the modelling, simulation, and analysis processes necessary for assessing and improving the energy performance of urban buildings. **Feel++** is a comprehensive C++ library for finite element analysis, enabling high-performance computing (HPC) applications in various scientific and engineering domains. It provides advanced numerical methods for solving partial differential equations (PDEs), supports scalable simulations with parallel computing capabilities, and integrates seamlessly with other software tools for multi-physics simulations. Feel++ is used for detailed finite element modelling and simulation of building energy performance, thermal comfort, and radiative heat transfer. **Modelica** is an open-standard, object-oriented language for modelling complex physical systems, particularly those involving multi-domain simulations. It facilitates the creation of component-based models for various physical systems, supports the integration of thermal, electrical, mechanical, and fluid systems, and enables the development of modular and reusable simulation models. Modelica is employed for multi-zone building energy simulations, allowing for detailed analysis of energy consumption and indoor environmental quality.

Apptainer (formerly Singularity) is a containerization solution designed specifically for HPC and scientific applications. It enables the creation of portable and reproducible computing environments, provides a secure way to package and distribute software dependencies and applications, ensures consistent execution across different computing environments, and supports HPC workloads without requiring root privileges. Apptainer is used to containerize the various software packages and their dependencies, ensuring reproducibility and ease of deployment across (Euro)HPC systems. Docker is a platform for developing, shipping, and running applications in containers. It provides a lightweight, consistent environment for software development and deployment, enables the creation of isolated environments for applications and their dependencies, facilitates continuous integration and continuous deployment (CI/CD) workflows, and provides the containers that are then converted to Apptainer for deployment on (Euro)HPC systems. Docker is used for containerizing applications and managing CI/CD pipelines, ensuring that the software stack remains consistent and up-to-date.

ReFrame is a regression testing framework designed for HPC systems. It simplifies the process of writing and running performance and scalability tests, provides a flexible, Python-based framework for defining and managing tests, supports systematic benchmarking across different HPC environments, and facilitates the collection and analysis of performance data. ReFrame is employed for benchmarking and validating the performance of simulation models on various (Euro)HPC systems, ensuring they meet the required standards for reproducibility. **SLURM** (Simple Linux Utility for Resource Management) is a highly scalable and flexible workload manager for HPC environments. It manages job scheduling, resource allocation, and workload balancing, provides a REST API for programmatic job submission and monitoring, supports complex job dependencies and workflows, and is used to manage and schedule the execution of simulations on HPC systems, optimizing the use of computational resources via the CI/CD system. The pilot makes advanced use of SLURM to submit and monitor jobs in the CI/CD system.

GitHub Actions is a CI/CD platform integrated with GitHub repositories, enabling automated workflows for building, testing, and deploying code. It automates the software development lifecycle, including testing, building, and deployment processes, integrates seamlessly with GitHub repositories for version control, and supports custom workflows to meet specific project needs. GitHub Actions is used to automate the CI/CD pipeline on standard resources as well as (Euro)HPC compute resources, ensuring continuous integration, testing, and deployment of the Urban Building pilot applications, enhancing productivity and reliability.

These software packages form the backbone of the Urban Building pilot, enabling comprehensive simulations that integrate energy performance assessment, environmental modelling, and advanced computational techniques. By leveraging these tools, the pilot aims to provide actionable insights and solutions for sustainable urban development and energy management.

4.3 Established and Potential Software Couplings

The Urban Building pilot within the HiDALGO2 project leverages several advanced software packages to achieve comprehensive modelling, simulation, and analysis of urban environments. Establishing and exploring potential software couplings between these packages and other pilot applications can enhance the accuracy and scope of simulations.

Established software couplings include the integration of Feel++ and Modelica. Feel++ provides advanced numerical methods for finite element analysis, while Modelica supports multi-domain simulations. Coupling these tools enables detailed multi-zone building energy simulations, incorporating complex physical interactions such as thermal and fluid dynamics, to provide more accurate assessments of energy performance and indoor environmental quality. Additionally, Docker is used for containerizing applications, while Apptainer ensures reproducibility in HPC environments. Applications containerized with Docker are converted to Apptainer containers for deployment on EuroHPC systems, ensuring consistent execution and facilitating easy updates and scalability across different computational environments. Furthermore, ReFrame provides a framework for performance and scalability testing, while SLURM manages job scheduling and resource allocation. Integrating ReFrame with SLURM allows for benchmarking and validating simulation models' performance on various HPC systems, ensuring efficient resource management and reliable execution of large-scale simulations. The profiling and tracing tools Extrae and Nsight are also integrated to provide detailed performance analysis, helping to identify bottlenecks and optimize computational workflows.

The Ktirio platform, when coupled with High-Performance Data Analytics (HPDA) and Machine Learning (ML) tools, can facilitate the analysis of large datasets, uncover hidden patterns, and improve predictive models for energy demand response and urban planning.

These established and potential software couplings highlight the collaborative and integrative approach of the Ktirio platform , aiming to develop holistic and sustainable solutions for urban planning and environmental management through advanced simulations and data analysis.

4.4 Current status of software capabilities as of M18

The Urban Building pilot's physical modelling framework is progressing well, with LOD-0 energy simulation fully operational at the city scale, and LOD-1 for solar masks operational with energy simulation implemented and currently undergoing tests. The integration of coupling LOD-0 (E/S) and LOD-1 (S/M) is being finalized, and city geometry reconstruction has improved with initial support for vegetation, watertight meshes, and work on roof support. New building models now support structures with more than 10 floors and heating systems such as boilers or heat pumps.

Several use cases have been identified, including city-scale energy simulation (UB UC#1), solar masks computation for optimal solar cell location (UB UC#2), and diffuse energy demand response at scale (UB UC#3). The CI/CD pipeline has seen significant developments, with Apptainer deployed on EuroHPC systems like LUMI, Karolina, Discoverer, Meluxina, Vega, Leonardo, and PSNC/Altair. The pipeline is integrated and automated for container updates and releases, ensuring performance and verification tests at a large scale for quality assurance. Codes and reports are automatically deployed to HiDALGO2 and Castiel2 repositories, with added checks for runner status to ensure deployment only if the system is online. The EuroHPC systems have expanded to include new platforms.

Benchmarking has been enhanced with the finalization of an initial benchmark dataset, including physical and performance reports for various cities, and more detailed profiling data has improved the accuracy of timing information. The pilot application has been updated to support High-Performance Data Analytics (HPDA) and Machine Learning (ML), providing basic tools for data processing.

Reporting capabilities have been extended to multi-scale reporting at city and district levels, covering energy consumption, comfort indicators, and building types. The Ktirio Urban Building platform has been released, comprising components like kub, kub-cicd, ktirio.gui, and ktirio.cases.

These advancements demonstrate significant progress in the Urban Building pilot, enhancing capabilities in energy simulation, solar mask computation, and energy demand response, while integrating robust CI/CD practices and preparing for comprehensive deployment and coupling with other pilot applications.

4.5 Challenges faced and future plans

The Urban Building (UB) pilot within the HiDALGO2 project has encountered several challenges in its journey towards enhancing urban simulation capabilities. The complexity of developing and integrating multi-fidelity models, such as coupling LOD-0 and LOD-1, requires precise coordination and data handling. Geometric reconstruction, particularly creating watertight meshes and incorporating detailed urban elements like roofs, vegetation, and urban furniture, remains a significant challenge. Efficiently managing input/output operations with large datasets is another critical area needing improvement. Ensuring the CI/CD pipeline is robust and can handle various tasks, from client applications to visualization and reporting, while expanding to additional EuroHPC platforms, has demanded substantial effort. Integrating with other pilots, such as the Urban Air Pollution (UAP) pilot, and ensuring seamless data exchange, particularly in generating common building surface meshes, pose challenges. Implementing hybrid computing using CPU and GPU, as well as improving profiling and partitioning for better performance, are advanced computational challenges. Additionally, ensuring sustainable data management practices while automating the CI/CD processes is an ongoing concern.

To address these challenges and further advance the Urban Building pilot, several plans have been outlined. In modelling, the integration of multi-fidelity models, focusing on coupling LOD-0 and LOD-1, will be finalized. Geometric reconstruction will be enhanced, improving support for vegetation, watertight meshes, roofs, and LOD-2, and adding urban elements such as roads and urban furniture. View factors will be computed to enhance the interaction between building surfaces, and scenarios considering building occupancy and various energy policies will be incorporated. Improved building energy simulation will add support for HVAC systems, further develop LOD2 support, and visualize metrics for cities in Ktirio.Gui.

In computing and performance, strategies to improve I/O performance and enhance profiling capabilities will be implemented. Ensemble runs will be utilized with Feel++ for multiple simulation scenarios, and initial support for hybrid computing using both CPU and GPU, as well as advanced partitioning techniques, will be developed. For the CI/CD pipeline, processes will be automated as much as possible, from client applications to visualization and reporting. The CI/CD pipeline support will be expanded to remaining EuroHPC platforms and integrated with GitHub and GitLab. Sustainable data management practices will be a focus to ensure longevity and reliability of data handling.

Coupling efforts will include generating a common building surface mesh for data exchange between the UB and UAP pilots. AI and ML will be utilized to discover useful information in massive datasets, and two-way coupling between models will be supported, with initial experiments using frameworks like MUSCLE3. Pre-processing pipelines will be developed to generate training datasets for AI and ML applications.

These future plans aim to overcome current challenges and push the boundaries of urban simulation, making the Urban Building pilot a robust, scalable, and efficient tool for urban planners and policymakers.

5 Wild Fires Implementation status

5.1 Use case definition and motivation

The Wildfires (WF) pilot focuses on the study and modelling of complex interactions between fire and its environment (atmosphere) that are not generally addressed operationally in wildland firefighting.

Two use cases have been identified, with two specific scales and scenario definitions, which may be of interest: the landscape level and the settlement level.

Use Case 1: Landscape level. The main objective is to provide simulations of wildfire progression, the energy released and coupled atmosphere-fire interactions, particularly the disturbance of wind fields and the generation of pyro-convective movements, as well as smoke release and dispersion. For this purpose, coupled fire-atmosphere models, such as **WRF-SFIRE**, has been used, which also calculates smoke dispersion.

Use Case 2: Urbanization level. The main objective is to provide simulations of fire behaviour at a flame scale (a few meters). This includes consideration of atmospheric interactions, including airflow disturbance and the local effect of buildings (geometric bodies) and vegetation (diffuse porous media). Also considered are the generation, transport and emission of flying embers, as well as the production and dispersion of local smoke. Comprehensive CFD models such as **OpenFOAM** and the specific combustion module **fireFOAM** are applied.

The integration of atmosphere and wildfire simulations to analyse the interactions between the lower atmosphere and fire generated energy in the Landscape Scale is of great interest in fires that develop in areas with steep orography and in fires that develop with high energy emission.

The fire-atmosphere interaction is increasingly relevant in a type of fire that goes beyond fire-fighting capabilities and poses a risk to both responders and the population. This interaction is complex, dynamic, and occurs in three-dimensional spaces that are challenging to model with numerical solutions. Indeed, more energetic forest fires inject energy and a significant amount of gasses, particles, and water vapour, altering the local circulation of the local atmosphere. In the most dramatic cases, these disturbances create convection cells that induce winds dragging the fire itself, thus creating feedback phenomena that are challenging to control. It is also relevant to consider the production and dispersion of smoke, which often affects urban areas that are sometimes located far from the main front.

Fire growth models such as SFIRE integrate spatial information on fuel types and terrain factors, along with temporal data on evolving weather conditions and fire risk levels, to replicate the propagation and evolution of fires across the terrain.

5.2 Description of Software Packages

To better characterise the scenarios already identified, the WF pilot has selected a set of models that target the scale and processes described. These libraries and tools are key to the modelling and analysis required.

WRF-SFIRE is a coupled fire-atmosphere model based on the Weather Research and Forecast model (WRF). It uses WRF model to simulate meteorology and it is combined with a semi-empirical fire spread model that includes a fuel parametrization (SFIRE) and description of initial ignition points.

OpenFOAM (Open Field Operation And Manipulation) is a tool used to model and analyse continuum mechanics problems, such as fluid flow phenomena. It has a suite of pre-processing, processing and post-processing tools, and libraries for simulating diverse scenarios, such as incompressible and compressible flows, turbulence and heat transfer, optimisation, acoustics, chemical reactions, solid mechanics and electromagnetics.

fireFOAM is a module within OpenFOAM, it is a transient solver for fires and turbulent diffusion flames with reacting particle clouds, surface film and pyrolysis modelling.

ndown.exe is a tool of the WRF model that allows running a nested domain independently of the preceding domain. When WRF runs with multiple domains, they interact bidirectionally, i.e., domain n serves as boundary conditions to n+1 but the latter returns feedback that modifies the next time step of the former. The ndown.exe program allows executing a domain using the output of the previous domain already calculated and without feeding back the execution of this one. This allows to change some parameters that governing the physics of the model or to modify the number of cores with which the process is launched.

5.3 Established and Potential Software Couplings

The simulation of wildfires and fire-atmosphere interaction is carried out at different scales according to the objective pursued.

At the landscape scale (several tens of kilometres), the goal is to obtain the expansion of a flame front and the production and dispersion of generated smoke, identifying which populated areas may be affected.

At the urban core scale (city, town, residential area, several hundred meters), the goal is to identify the potential pattern of fire propagation in the vicinity and within the urbanization, and the potential impact on homes from flames and on the population from smoke due to the combustion of forest vegetation, garden vegetation, and residential fuels.

This multi-scale approach requires using and coupling two simulation schemes with different models, data, and spatial and temporal resolutions. One of which is WRF-SFIRE which is used for wildfire simulation at the landscape scale. The other is OpenFOAM-fireFOAM which is used for combustion and fire propagation simulation in the vicinity and within urban areas.

The results of **WRF-SFIRE** are delivered in the form of grids with variable resolution, but in HIDALGO2, horizontal resolutions of 200 m or less are proposed for the atmospheric grid. Additionally, fire propagation is calculated on grids 10 times finer, that is, with a resolution of 20 m.

In the OpenFOAM-fireFOAM fluid dynamics model, much finer three-dimensional adaptive grids are used, on the order of centimetres or sometimes smaller. The coupling of the WRF-SFIRE and OpenFOAM-fireFOAM models requires careful selection of the grids and their exact referencing so that the results of one can be resampled in the matrix of the other as boundary conditions.

In particular, the wind vector field, calculated in WRF-SFIRE and including the feedback due to the fire, the flame front position, the amount of heat emitted (heat flux), and the smoke density at each point in space.

Both models are coupled through their files via resampling and following the recommendations of good practices as proposed in [9].

The workflow for coupling both models is as follows:

1. Simulation of fire propagation and smoke with WRF-SFIRE
2. Extraction of the variables of interest at time intervals:
    a. Wind vector components (3D)
    b. Smoke column density (3D)
    c. Fraction of burned area (flame front position) (2D)
    d. Heat flux (2D)
3. Description of the OpenFOAM computational domain and mesh resolution
4. Resampling of WRF-SFIRE variables onto the OpenFOAM mesh
5. Description of the geometry in the OpenFOAM scenario
6. Parameterization of the OpenFOAM simulations
7. Incorporation of the resampled grids into the OpenFOAM simulation at time intervals as boundary conditions (e.g., wind, smoke concentration, combustion points)
8. Execution of the OpenFOAM-fireFOAM simulations

5.4 Current status of software capabilities as of M18

The WRF-SFIRE model [8] has been selected to simulate these wildfires in the central region of Spain due to its suitability and its long history of development and validation.

The model has been compiled on different machines within the EuroHPC network, each with its particularities, namely EAGLE at PSNC Poznan (Poland), VEGA at IZUM Maribor (Slovenia), LUMI at the CSC data centre in Kajaani (Finland), and MELUXINA at the LuxConnect data centre in Bissen (Luxembourg).

WRF-SFIRE operates on two separate meshes: the lowest resolution mesh (Geo2D atmospheric mesh) and the highest resolution grid (2D fire mesh).

Four nested domains (D01-D04) with increasing resolutions have been established, allowing for the downscaling of atmospheric data provided by ERA5 reanalysis.

The simulations performed with WRF-SFIRE are:

Design of **WFR-SFIRE** simulation scenarios, with up to four nested domains and the use of WRF-SFIRE coupled models for real fires simulation. This has been installed, tested, and used at Eagle HPC and EuroHPC servers Vega, Meluxina, and Lumi.

Design of a benchmark scenario and test environment setup, **ReFrame** with WRF-SFIRE has been installed and tested on local infrastructure Eagle and EuroHPC server Vega with on-going testing and configuration in LUMI and Meluxina.

Results for real wildfire/weather scenarios for the simulation in WRF/SFIRE have been obtained and simulations with up to 55 nodes have been performed on Eagle, LUMI and Meluxina.

The first simulation, conducted on VEGA, used four domains and specified the use of the LES model for vorticity computation. The second simulation, conducted on LUMI, used four domains and parameterized the planetary boundary layer (PBL) to avoid large-eddy simulation (LES) computation, allowing for result comparison. The third simulation, conducted on EAGLE, used four domains, parameterized the PBL, and simulated the atmospheric dynamics with and without fire to estimate the feedback effect of the fire-atmosphere interaction. Finally, the fourth simulation, conducted on LUMI, used four domains, parameterized the PBL using the Yonsei University (YSU) scheme for D01, D02 and D03 and the Shing-Hong scheme for D04, and performed simulations with and without fire-atmosphere interaction feedback to precisely estimate the magnitude of this effect.

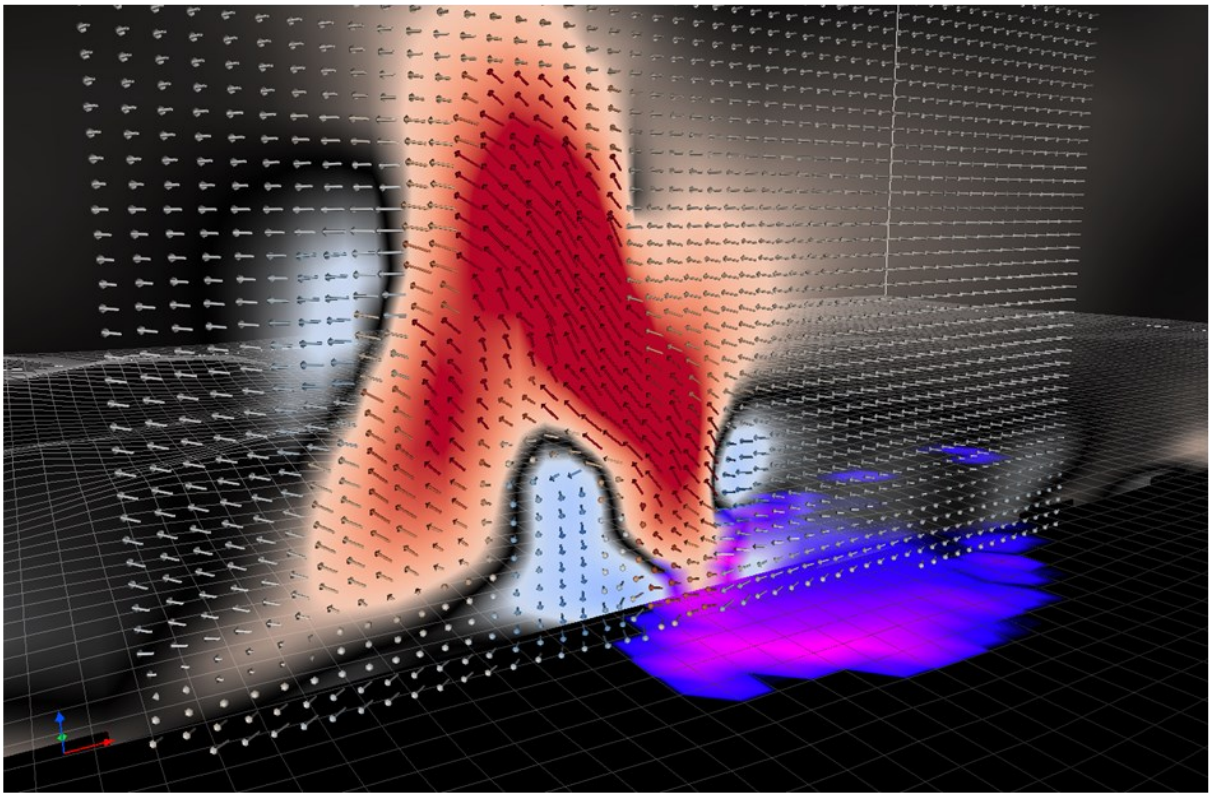

Figure 1 Detail of a fire-atmosphere interaction. The heat release in the ground due to the forest fire spread entails wind flow perturbation and this is modifying fire behaviour in a loopback fashion.

The results of the WRF wind modelling, based on ERA5 reanalysis data, show good consistency in both magnitude and direction compared to the values observed on the same day at the automatic stations in the fire area. the WRF-SFIRE simulation with fire-atmosphere feedback does develop the main axis practically to where the fire reached on the first day, although it underpredicts the expansion of the flanks, probably due to the feedback effect of the wind induced by the fire itself. The calculated heat flux for the surface fuel models yields values between 0 and 43,400 W/m2 for this simulation, which are injected into the atmospheric dynamics computation, representing medium or low values compared to other forest fires with higher shrub loads or crown fires. Despite this, these simulations show a clear fire-atmosphere feedback, making the propagation, especially at the head, faster, resulting in a more pronounced length-to-width ratio (more elongated shape).

Additionally, a simulation of smoke production and dispersion was conducted using passive tracers.

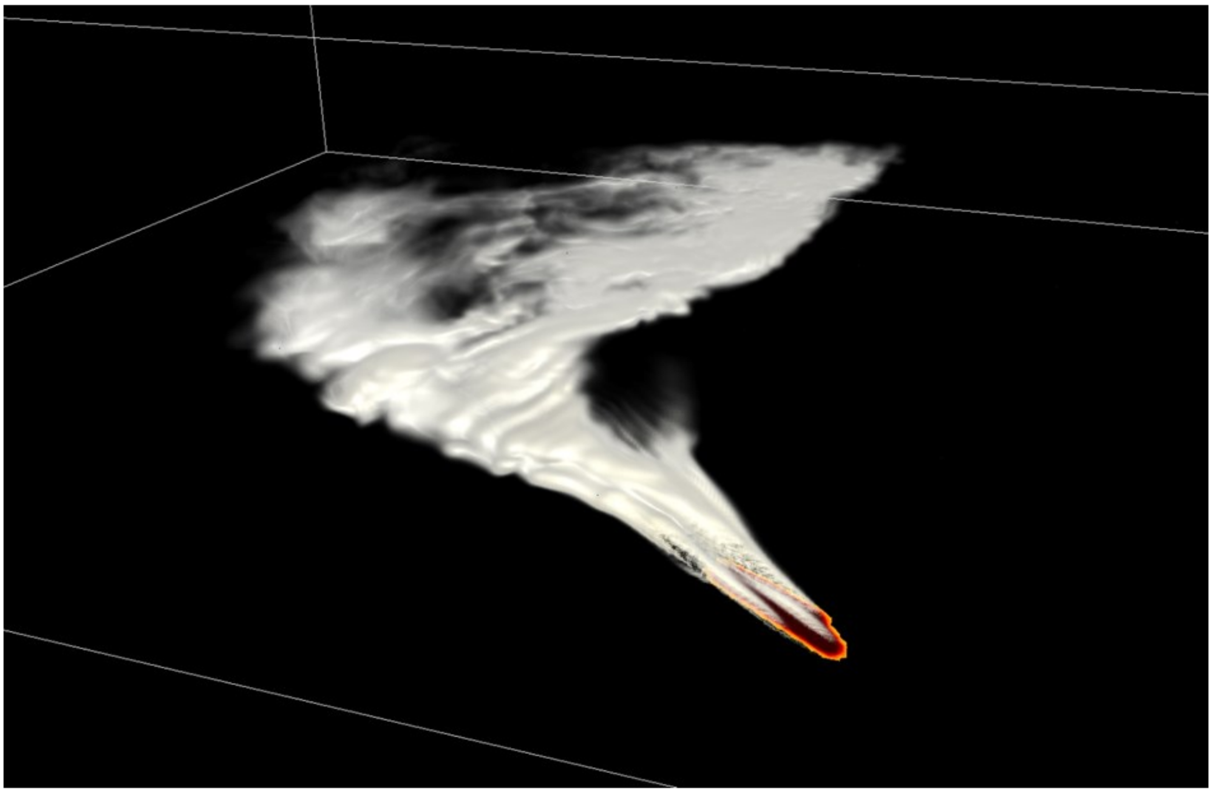

Figure 2 Example of a WRF-SFIRE simulation, including fire spread and associated smoke production and dispersion using passive tracers.

Visualization

For the forest fire pilot in Hidalgo2 project, the use of Unreal Engine (UE) has been proposed as an integrating platform for scenarios and visual simulation of forest fires at various scales. The proposed methodology includes the integration of external files using universal interchange formats along with other objects into UE environments such as FGA, VDB, or more generally ASCII and RAW binary formats. IN particular, MTG has developed a software which translates the resulting netcdf files into FGA and VDB files to be directly imported in Unreal Engine.

The Digital Terrain Model (DTM) can be replaced by the CESIUM service, which injects 3D geometry from the Google Tiling Service. All data is interpreted as assets in UE. Optical response of light to smoke plumes has been simulated, resulting in photo-realistic, real-time visual simulations to be used in VR experiences.

Figure 3 Example of integrating a CFD simulation done in WRF-SFIRE of volumetric smoke and fire from a wildfire into a photo-realistic immersive environment using Unreal Engine. The digital terrain model has been replaced by the Cesium-Google Tiles service. Some vector GIS layers and background volumetric clouds have been added.

5.5 Challenges faced and future plans

Challenges and limitations

The advantage of having an HPC facility with thousands of available cores for running simulations is undeniable. However, in practical operation, it is not without limitations, which can sometimes be difficult to resolve, delaying or even preventing the completion of tasks.

For example, it is important to emphasize the limitation that choosing an appropriate number of processors requires considering the decomposition of processes in relation to the size of the domains. During processing, each domain will be divided into tiles, and the number of tiles will depend on the total number of processors used, with one tile per processor. Besides, each tile will also have a minimum of 5 rows and columns on each side (referred to as ‘halo regions’), which are used to transfer information from each cell/processor to neighbouring tiles.

Using too few processors can result in very slow execution or even failure. When domains have significantly different sizes, and the number of processors is either insufficient for one or excessive for another, the *ndown* program is recommended. For example, if D01, D02, and D03 run well with a certain number of processors but D04 is too large, the first three domains should be run as a single process, and then ndown should be used to run D04 separately.

The number of variables written to each output file is also a crucial factor to consider. The more variables that are written, the larger the files become, including both the output files and the logs (rsl.error.* and rsl.out.*). This can impact the post-processing of simulations and strain the available resources, which are often limited. In the specific case of VEGA, this has been problematic, as running simulations with a large number of points and outputting all available variables can quickly exhaust the available workspace storage.

Furthermore, it is relevant to highlight that the HPC resources allocated were not always available for the simulations, resulting in jobs being queued. At times, the wait time for the allocation of requested resources exceeded 24 hours, thus noticeably slowing down the work.

Future work

To enhance **WRF-SFIRE simulations**, we have identified several key areas of focus. First, we will improve input data by utilizing predictions from the ECMWF to obtain more accurate and reliable information. We also plan to implement new parametrizations for the PBL and LES to achieve a more precise representation of atmospheric processes. Additionally, we will investigate the use of GPU technology and enhance the portability of WRF-SFIRE to boost computational efficiency. We are organizing a hackathon by the end of 2024 to foster innovation and collaborative development within the WRF-SFIRE community, bringing together experts and enthusiasts to advance its capabilities. To further improve model accuracy, we plan to redesign tests using ReFrame, incorporating ECMWF data with simulations that utilize two and three domains. For data assimilation in WRF-SFIRE, we will integrate weather data—including wind, temperature, and moisture—and incorporate satellite-based products for accurate fire front representation. Finally, we aim to enhance fire behaviour modelling by transitioning from SFIRE to include the Fire Spread Engine (FSE) and the Balbi model, and by incorporating the dynamics of crown fire spread to improve the realism of our fire behaviour simulations.

To enhance our **CI/CD capabilities**, we are implementing several strategic initiatives. First, we are integrating the WRF installation process with GitLab CI/CD pipelines to streamline and automate deployments, ensuring consistent and reliable updates. Recognizing the importance of a solid foundation, we have postponed code development until the third year, allowing us to thoroughly plan and prepare before beginning coding tasks. Additionally, in preparation for future developments, we will create containers for WRF-SFIRE and ReFrame. These containers will facilitate easier deployment and scaling of our simulations across various environments.

To improve **OpenFOAM simulations**, we are implementing several key initiatives. We will develop a wrapper to enhance the functionality and usability of OpenFOAM and fireFOAM, making integration with other tools and systems easier. Inspired by the work of W. Mell, we are exploring the development of a solver that models particle-based forest fuel, aiming to provide more accurate and detailed simulations of forest fire dynamics, as previously implemented in FDS. Additionally, we are investigating the use of GPU technology in OpenFOAM to accelerate computational processes and improve simulation efficiency. Finally, we will integrate ReFrame with OpenFOAM-fireFOAM on HPC servers to monitor and measure the scalability and performance of our simulation tasks.

For **uncertainty analysis**, we have identified several key areas of focus. First, we will use UQ tools to systematically analyse and quantify uncertainties in our simulations, ensuring more robust and reliable results. We will also incorporate data assimilation techniques and create variations to the input parameters, improving the accuracy and reliability of our models and allowing for better decision-making and forecasting, particularly for operational simulations. Additionally, we will integrate atmospheric ensembles from ECMWF to enhance our uncertainty analysis, providing a broader and more comprehensive understanding of potential outcomes and scenarios.

6 Material Transport in Water Implementation status

6.1 Use case definition and motivation

River pollution poses significant risks to the environment, with contaminants such as microplastics and biochemical substances adversely affecting aquatic ecosystems. To better understand and mitigate this environmental threat, researchers are turning to advanced numerical simulations. Developing a numerical simulation provides a deeper insight into the complex process of pollution transport in rivers, offering a means to enhance control and prevention strategies.

Simulating **river pollution transport**, however, is no easy feat due to the involvement of intricate physics that must be coupled. This includes fluid simulation (ideally with a free surface) to model river flow, particle simulation for the river bed (sediment transport), and the coupling between the fluid and particles. Additionally, the simulation of pollution is described by an advection-diffusion partial differential equation, adding another layer of complexity. Addressing these challenges is essential for creating an accurate and comprehensive model that reflects real-world scenarios.

Development of such complex codes would pave the way for environmentalists and physics experts to find effective solutions concerning the mitigation of river pollution, which in the modern world has been an ever increasing global challenge.

6.2 Description of Software Packages

Several software packages and frameworks have been employed to solve various parts of the current use case which is a multiphysics simulations involving complex physics of fluids, solids and concentration.

waLBerla

Widely applicable Lattice Boltzmann Erlangen (waLBerla) [1] is an open source multiphysics framework used for CFD simulations where the Lattice Boltzmann method is the underlying physic method to solve the fluid fields. Examples of some application scenarios include rigid-particle dynamics, free-surface flows, thermal problems, flows with phase change and phase-field models etc.

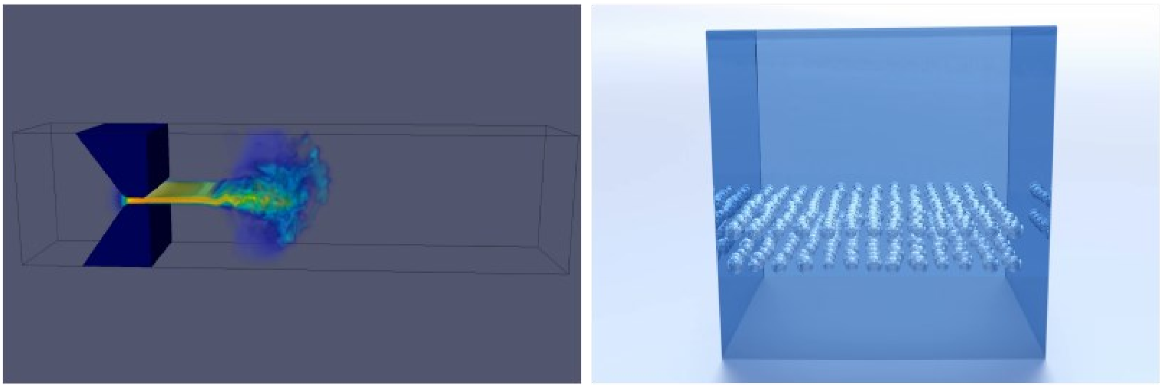

Figure 4 Application scenarios simulated using waLBerla

HyTeG

Hybrid Tetrahedral Grids (HyTeG) [4] is a C++ framework for extreme-scale finite element simulations with a strong focus on matrix-free geometric multigrid. With its block structured tetrahedral discretion (Figure 5), it ideally combines flexibility with performance especially for circular domains. One use case where this capability is employed is the large scale simulation of earth mantle convection as shown in **Error! Reference source not found.**. For this pilot, HyTeG might be used as a tool to solve partial differential equations on the imposed flow field using an

implicit time stepping method.

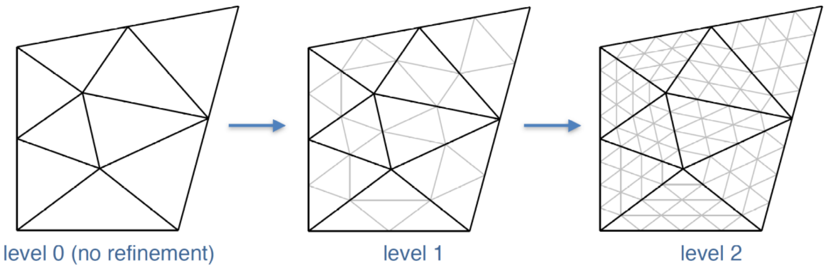

Figure 5 Grid structure and hierarchy in HyTeG

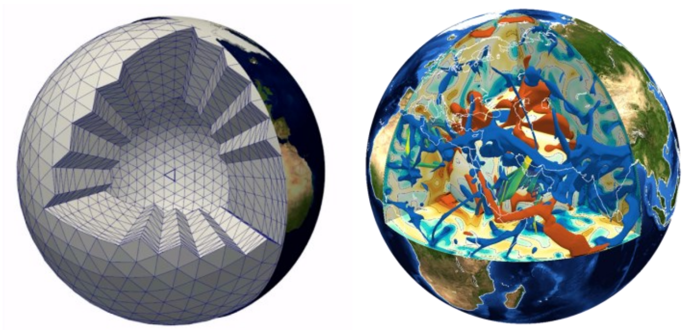

Figure 6 Simulation of Earth mantle convection in HyTeG

ExaStencils

The ExaStencils [5] code generation framework processes input in its own multi-layered domain-specific language (DSL) ExaSlang to compose highly optimized stencil codes in general and geometric multigrid solvers in particular. While most other code generation tools generate kernels or classes for existing frameworks, ExaStencils is a whole program generated that allows for generation of whole C++ simulation frameworks. These generated frameworks provide automatic parallelization capability with OpenMP, MPI and CUDA.

MESA-PD

The Modular and Extensible Software Architecture for Particle Dynamics (MESA-PD) [3] is a particle dynamics library that combines extensibility, flexibility, performance, and scalability. MESA-PD supports molecular dynamics and the discrete element method (DEM). To achieve highly performant simulations on modern HPC systems without the need for low-level code optimizations, code generation is used to separate the parallel computing functionalities from the physically complex modelling of each particle.

6.3 Established and Potential Software Couplings

When solving multiphysics problems, several functions and kernels come together to provide a reliable approximation that should also be optimized to perform fast on given hardware architecture, e.g., on GPUs. For our use case a fluid-solid-concentration coupling has to be established. Since all parts of this application are physically complex problems of their own, we intent to utilize and couple the software frameworks from chair the computer science chair 10 of FAU that are developed for each specific sub-domain. To simulate the fluid dynamics within the system the waLBerla framework is utilized and coupled with MESA-PD which is used to simulate the physical effects of the fully resolved solid particles within the fluid. For describing the concentration development within the simulation, a partial differential equation (PDE) is solved. This PDE can either be approximated using an explicit or implicit method. For communication between these frameworks, we rely on a strong coupling, so instead of communicating intermediate results between our frameworks, interfaces are defined to directly access and modify inter-framework data. To enhance flexibility between the continuously developed software frameworks, we rely on code generation to optimize the abstract description of these interfaces on a hardware level. Furthermore, various target hardware poses different challenges for the development of performant software tools dependent on the architectures. Here we make use of code generation to allow for fast and scalable approximations over a large variety of machines.

One important software coupling is the coupling between waLBerla and ExaStencils. The native codes of waLBerla are generated using ExaStencils which paves way for highly efficient codes that can potentially save a lot of computational effort.

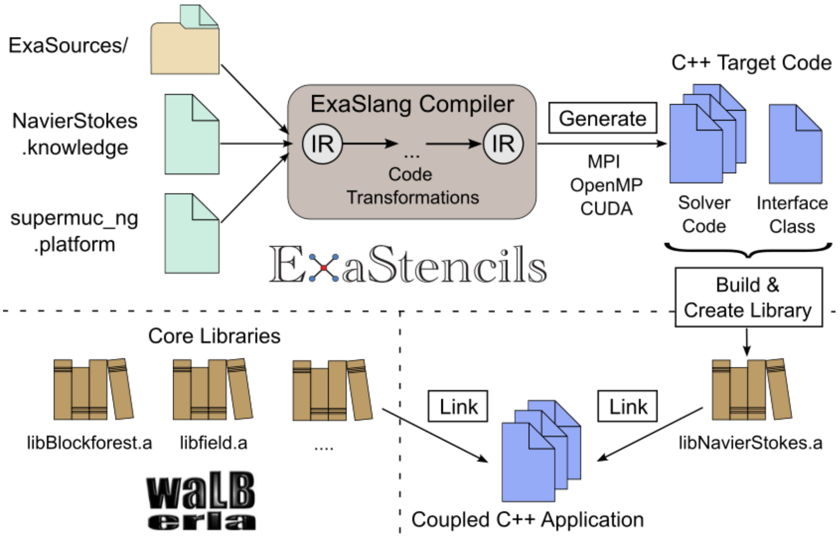

Figure 7 Schematic diagram representing coupling of waLBerla and ExaStencils

6.4 Current status of software capabilities as of M18

The **Fluidized Bed application,** which is a two way coupled fluid-particle multiphysics simulation [2], has been developed, tested and validated.

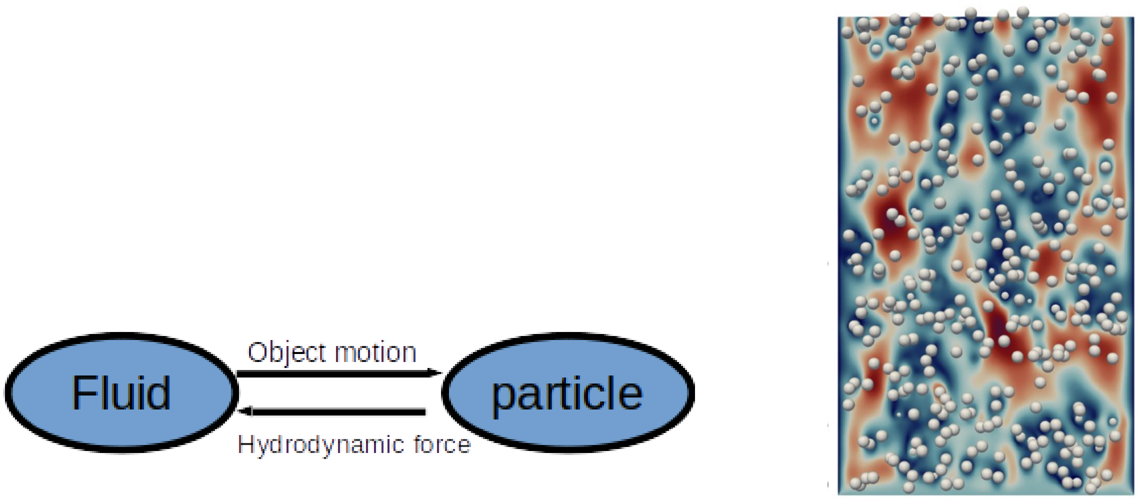

Figure 8 The figure on the left represents a block diagram for the coupled physics effects in a fluidized bed application. The figure on the right shows a VTK visualization of the velocity field in a fluidized bed simulation.

As a continuation work, **charged particles application** has been developed. In this application, in addition to the fluid-particle coupling, the particles are having an electrostatic charge and therefore electrostatic effects on the particles persist which affects the particles behaviour. In this application, the coupling of waLBerla-ExaStencils has been established to solve the electrostatic Poisson equation using the generated geometric multigrid solver. This application although has no physical significance to the current use case, in a computer science/programming aspect is similar to the current use case for the fact that in both scenarios, a partial differential equation is being solved.

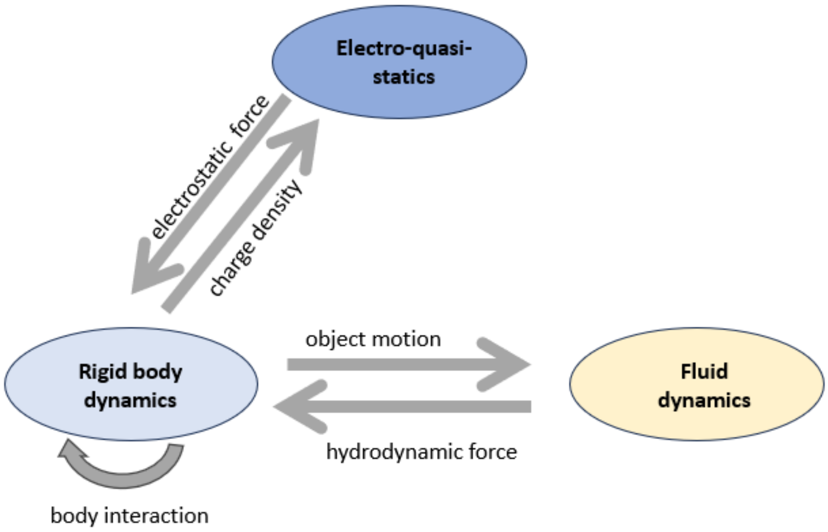

Figure 9 Block diagram of the various physical effects in a charged particles application.

Coupled 3D transport application has been developed in which a one way coupling has been established between concentration and fluid. The fluid is simulated using LBM and the resulting velocities of the fluid domain are sent as an input to the advection-diffusion equation. The advection-diffusion equation is solved using the geometric multigrid solver in ExaStencils.

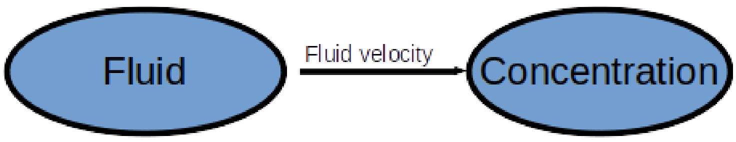

Figure 10 Block diagram of the various physical effects in a coupled 3D transport application.

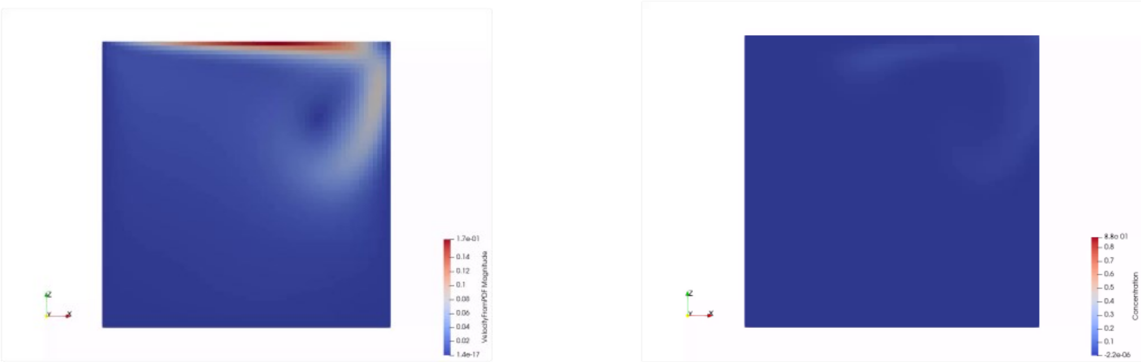

Figure 11 The coupled 3D transport simulation showcase with the fluid velocity field on the left and the concentration field on the right.

6.5 Challenges faced and future plans

The current applications have codes working for fluid-particle coupling (as for the fluidized bed application) and fluid-concentration coupling (as for coupled 3D transport application). The task ahead in the use case is to establish a **fluid-particle-concentration coupling**. To do this, we have several ideas that are depicted in **Error! Reference source not found.** and Figure 13.

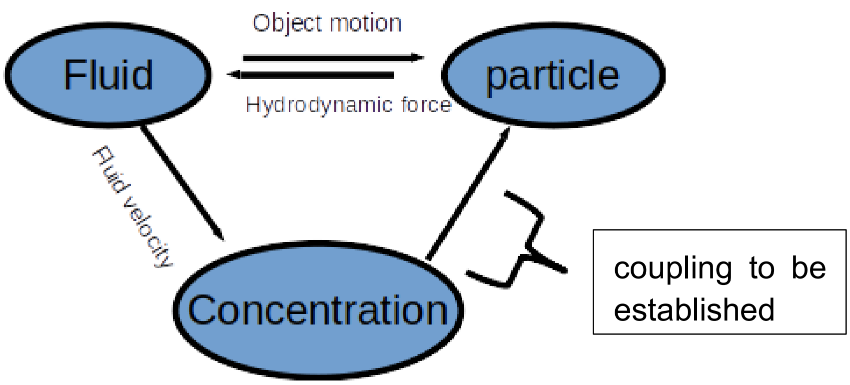

Figure 12 Block diagram showing the various couplings in MTW use case.

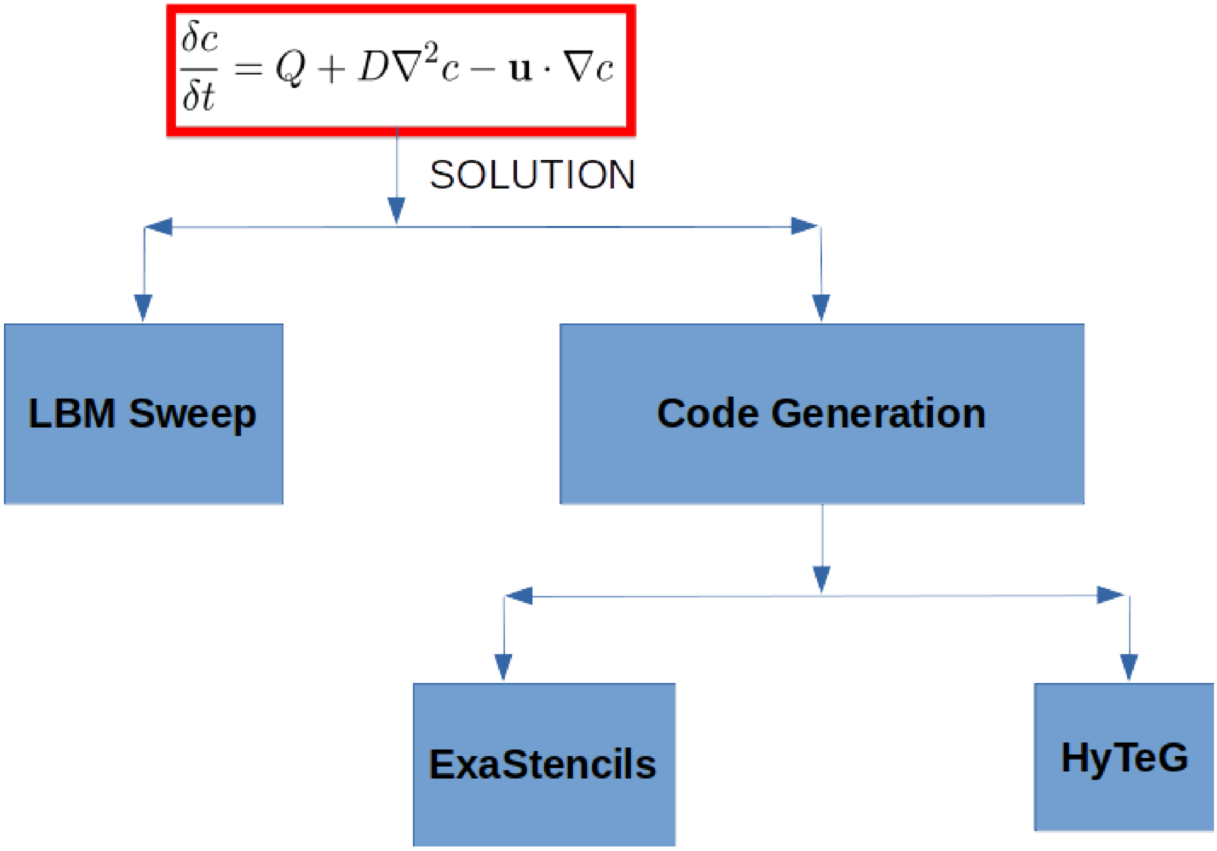

Figure 13 Potential strategies to solve the transport equation.

The LBM sweep is an explicit time stepping scheme to solve the advection-diffusion equation. ExaStencils and HyTeG solve the partial differential equations using implicit time stepping. The ExaStencils way of doing this has already been done for the coupled 3D transport application as discussed in the previous section. However there is further scope for improvement especially with regards to the code performance. In the coming months, the main focus would be to try out the implementations using LBM sweep and also think about potential strategies to use HyTeG for our use case.

7 Conclusions

In the D5.6 "Implementation Report on Pilot Applications", we outline the progress of the HiDALGO2 project up to Month 18. The report evaluates the initial development actions of WP5 and assesses the initial status and planned application couplings of the pilot applications: Material Transport in Water (MTW), Urban Air Pollution (UAP), Urban Buildings (UB), Renewable Energy Sources (RES), and Wildfires (WF).

The report provides a comprehensive overview of the capabilities, current development status, and planned next steps for each pilot application, as well as the software frameworks used. It also details the technical couplings between these frameworks.

The future steps involve continuous development of each pilot application to realize their intended capabilities in supporting city policymakers, energy producers, disaster management authorities, and environmental scientists. Additionally, efforts will be made to enhance scalability and applicability across a wider range of hardware architectures, reduce compute time for each pilot, and improve the potential to execute them on any available architecture. These efforts will be continued and expanded in subsequent deliverables such as D5.7 and D5.8 to ensure that the HiDALGO2 project achieves its goals of developing holistic and sustainable solutions for urban and environmental issues.

References

[1] First name initials (i.e. Cain, A., & Burris) Author Surname. (Year of publication or most recent update (i.e. 1999)) Name of Web Site (i.e. Research WebSite). [Online]. Website URL (i.e. http://www.cutr.eng.usf.edu/its/mobile_phone_text.htm)

[2] Web Site Title (i.e. WITDOM). [Online]. web url (i.e. http://www.witdom.eu/deliverables)

[3] First name initials (i.e. Levenstein, H. A.) Author surname, *Title (i.e. Revolution at the table: The transformation of the American diet)*. City (i.e. Berkeley): Publisher (i.e. University of California Press), Year of publication (i.e. 2003).

[4] First Name Initials Lead Author Surname, "Project name, deliverable number and title (i.e. WITDOM. D2.2 - Functional analysis and use cases identification)," Year (i.e. 2015).

[5] First Name Initials Surname, "Title (i.e. The power of peers)," vol. Volume (i.e. 2), no. Issue and pages (i.e. 2 - 57-63), Year of publication (i.e. 2002).