

D5.2

Progress report

Project Title	dealii-X: an Exascale Framework for Digital Twins of the Human Body
Project Number	101172493
Funding Program	European High-Performance Computing Joint Undertaking
Project start date	1 October 2024
Duration	27 months



dealii-X has received funding from the European High-Performance Computing Joint Undertaking Programme under grant agreement N° 101172493

Deliverable title	Progress report
Deliverable number	D5.2
Deliverable version	1.0
Date of delivery	30 September 2025
Actual date of delivery	[Actual date]
Nature of deliverable	Report
Dissemination level	Public
Work Package	WP5
Partner responsible	RUB

Abstract	This report summarizes M1–M12 project progress, highlighting advances in scalable numerical methods, multiphysics biomedical applications, and exascale computing, as well as key outcomes in project management, communication, and exploitation activities.
Keywords	progress report; exascale computing; biomedical simulations; management; dissemination; exploitation

Document Control Information

Version	Date	Author	Changes Made
0.1	5.09.2025	I. Prusak	Initial draft
0.2	9.09.2025	G. Stanic	Communication and Dissemination sections
0.3	16.09.2025	I. Prusak	Integration of partners' contribution
0.4	[Date]	dealii-X Consortium	Consortium revision
1.0	[Date]	[Author name]	Final version

Approval Details

Approved by: [Name]

Approval Date: [Date]

Distribution List

- Project Coordinators (PCs)
- Work Package Leaders (WPLs)
- Steering Committee (SC)
- European Commission (EC)

Disclaimer: This project has received funding from the European Union. The views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European High-Performance Computing Joint Undertaking (the “granting authority”). Neither the European Union nor the granting authority can be held responsible for them.

Contents

1	Introduction	5
1.1	Purpose of the Document	5
1.2	Structure of the Document	5
2	dealii-X exascale building blocks and support tools (WP1)	7
2.1	Objectives	7
2.2	Progress Achieved in Year 1	8
2.3	Next Steps	19
3	dealii-X lighthouse Applications for Digital Twins of the Human Body (WP2)	21
3.1	Objectives	21
3.2	Progress Achieved in Year 1	22
3.3	Next Steps	36
4	Co-Design, Technology Exploitation, and Energy Efficiency(WP3)	38
4.1	Objectives	38
4.2	Progress Achieved in Year 1	38
4.3	Connection to Other CoEs	45
4.4	Next Steps	45
5	Dissemination, Communication, and Exploitation (WP4)	47
5.1	Objectives	47
5.2	Progress Achieved in Year 1	47

5.3	Key Performance Indicators	49
5.4	Next Steps	50
6	Project Management (WP5)	51
6.1	Objectives	51
6.2	Progress Achieved in Year 1	51
6.3	Next Steps	52
6.4	Summary	53

1 Introduction

1.1 Purpose of the Document

This document provides an overview of the project's activities, achievements, and challenges during the reporting period M1–M12. Its main purpose is to give a clear and comprehensive account of the work carried out, highlighting both scientific and technical progress as well as management, communication, and exploitation efforts.

The report covers progress in developing an exascale framework for digital twins of the human body. It details advances in scalable numerical methods, high-performance computing strategies, and multiphysics biomedical applications. These technical developments are accompanied by efforts to ensure the reliability, efficiency, and usability of the software, which form the foundation for future scientific and clinical applications.

In addition to the scientific and technical aspects, this document also reports on project management, communication, and exploitation activities. This includes coordination between partners, dissemination of results to the wider scientific community, and strategies to promote uptake and impact of the project's outputs. By integrating these different dimensions, the report not only documents achievements but also provides context for planning the next steps and setting priorities for the upcoming phases of the project.

Overall, this report serves as both a record of past activities and a guide for future work, helping to maintain transparency, accountability, and alignment with the project's overarching goals.

1.2 Structure of the Document

- Section 2: dealii-X exascale building blocks and support tools (WP1)
- Section 3: dealii-X lighthouse Applications for Digital Twins of the Human Body (WP2)
- Section 4: Co-Design, Technology Exploitation, and Energy Efficiency (WP3)
- Section 5: Dissemination, Communication, and Exploitation (WP4)

- Section 6: Project Management (WP5)

2 dealii-X exascale building blocks and support tools (WP1)

2.1 Objectives

The main objective of Work Package 1 (WP1) is to serve as the foundation for the dealii-X Centre of Excellence by enhancing and expanding the capabilities of the deal.II library to address the challenges of exascale computing and facilitate the creation of advanced digital twins of human organs.

The key steps of WP1 include:

- Extending and improving the exascale capabilities of deal.II;
- Improving pre-exascale modules of the deal.II library;
- Developing an experimental polygonal discretization module for deal.II;
- Integrating PSCToolkit within deal.II;
- Integrating MUMPS within deal.II.

Specifically, the sub-work packages aim to:

- **WP1.1 (Lead RUB):** Develop matrix-free computational methods optimized for GPU architectures and enhance the scalability of solvers;
- **WP1.2 (Lead UNIP):** Improve the gmsh API, develop a generalized interface for coupling operators, enhance reduced order modeling capabilities, integrate low-rank approximation methods, and develop block preconditioners;
- **WP1.3 (Lead SISSA):** Introduce and parallelize polygonal discretization methods within deal.II and develop related multigrid techniques;
- **WP1.4 (Lead UNITOV):** integrate PSCToolkit into deal.II, leveraging GPU computing and developing efficient preconditioners for multiphysics problems;
- **WP1.5 (Lead INPT):** Integrate the MUMPS solver directly into deal.II for use in multigrid methods and explore low-rank and mixed-precision techniques;

In summary, WP1 is dedicated to developing and integrating fundamental software components within the deal.II library and external libraries, with a strong emphasis on enabling exascale computation for the digital twin applications in WP2.

2.2 Progress Achieved in Year 1

New deal.II release and enhancement of dealii-X building blocks

The 9.7 release of deal.II has introduced several improvements directly aligned with WP1.1 on GPU-enabled matrix-free solvers and exascale readiness. Key advances concern the matrix-free infrastructure, which was extended with:

- Support for multi-component and vector-valued finite elements in the matrix-free evaluation routines for the Kokkos-based portable branch. This substantially broadens the class of multiphysics applications (e.g. poromechanics in brain and liver) that can be ported to GPUs efficiently.
- Portability layer `Portable::MatrixFree` built on top of `kokkos`, ensuring code can target NVIDIA, AMD, and Intel GPUs with the same interface. The release consolidates temporary storage structures into new `DeviceVector` and `DeviceBlockVector` classes, simplifying GPU memory management.
- Performance optimizations to sum factorization kernels, yielding measurable speedups especially on NVIDIA A100/H100 hardware.
- Extended ghost cell handling and lexicographic patch smoothers for geometric multigrid preconditioners, crucial to scaling implicit solvers to 10^{11} - 10^{12} DoFs on EuroHPC machines.

In addition, new solver strategies were enabled by the `TensorProductMatrixCreator` for diagonalization-based inverses, applicable in the context of multigrid smoothers and coarse solvers. These improvements directly support the objective of saturating node-level bandwidth on GPUs and minimizing data transfer overheads, as required for exascale digital twin workloads.

Ongoing developments for the matrix-free kernels, in particular node-level experiments for various GPU vendors, optimization of Kokkos-based and CUDA-based

algorithms as well as energy efficiency improvements have also been made in the past few months. These contributions are described in more detail, along with an extensive evaluation of performance, in the report on deliverable D3.2: “Co-Design and Efficiency Report I”.

WP1.1: Extending and improving the exascale capabilities in deal.II

Work package 1.1 focuses on enabling efficient finite element computations with the deal.II finite element library, where matrix-free evaluation techniques and multigrid methods are the core scientific components.

The activities of the group at RUB can be summarized as follows:

- Benchmarks to assess the current performance of the core components.
- Exascale readiness of Raviart–Thomas finite element algorithms for high–performance computing.
- Restart capabilities in application solver ExaDG.
- Initial setup of a domain decomposition infrastructure in the deal.II library to establish additional solver paradigms for the exascale era.

Benchmarks for GPU systems

The current status of the implementation focuses on element–wise communication–free algorithms such as mass and stiffness matrix operators.

Implementations are available in the <https://github.com/dealii-X/benchmarks> repository. A brief overview of the kernels is provided below:

- Bake-off Kernel 1 (BK1) for mass-matrix operator with a Gauss–Legendre quadrature formula involving $p+2$ points per coordinate direction for polynomial degree p .
- Bake-off Kernel 5 (BK5) for the stiffness-matrix operator on Gauss–Lobatto–Legendre quadrature points, utilizing the collocation of nodal points of the underlying Lagrange basis with the points of quadrature.

The main challenge in implementing benchmarks on cutting-edge accelerators is the growing number of GPU vendors such as Nvidia, AMD, and Intel, whose non-standard programming models bring an additional burden on developers. To address this challenge, this subproject also aims to ensure portability across different GPUs by employing the Kokkos library, which supports multiple back-ends. Accordingly, current implementations evaluate the portability and performance of Kokkos by comparing it with CUDA kernels. Further activities have also considered Intel GPUs and ARM CPUs; these results are in a preliminary state due to the ongoing collaboration between the RUB and LRZ partners. More detailed preliminary results and implementation details are given in the reports for deliverables D1.2, D1.3 and D3.2.

Raviart–Thomas finite elements

Another string of activities is centered around the application-readiness of Raviart–Thomas elements, where future requirements and potential improvement opportunities are investigated with the ExaDG application code pioneering the use of Raviart–Thomas elements for high-Reynolds incompressible flow problems in a matrix-free setting based on deal.II.

This will also require some debugging in the deal.II implementation of the Raviart–Thomas elements, as these code paths are quite new and have not been extensively used by the community yet due to the fact that the use of Raviart–Thomas elements has significantly increased in the last years. Currently, the time integration and matrix-free solver capabilities in ExaDG have been extended, and further testing regarding the performance will be executed in the near future on the JUPITER system.

Restart capabilities

To reduce simulation costs for large-scale flow problems, the restarting capabilities of ExaDG have also been extended, while the general setup and software design have been kept general, solely based on deal.II data structures for maximal re-usability. The main goal here is to allow for matrix-free mesh-to-mesh projection, such that performance studies of various solver settings and performance tuning can start from a precursor run to face the target physics directly without the need for recomputing the startup phase. This has been realized in a general setting, splitting off the grid-to-grid projection aspect and the changes with respect to the time integrators within ExaDG. Hence, this code can be easily transferred to other codes as

the required routines are based solely on deal.II. This naturally allows considering arbitrary physics and discretizations, various finite elements (polynomial degree, cell type, continuity, etc.), or non-matching grids.

WP1.2: Improvement of pre-exascale modules of the deal.II library

Work package 1.2 focuses on enhancing the existing modules of the deal.II library to prepare them for the challenges of exascale computing. This includes several key activities:

- Improving the gmsh API to support large-scale;
- Developing a generalized interface for coupling operators, which is fundamental for multiphysics and multiscale simulations, including methods for conforming and non-conforming grids and non-local coupling techniques;
- Enhancing the reduced-order modeling (ROM) capabilities of deal.II, implementing algorithms for data analysis and reduced-order geometric modeling;
- Integrating low-rank approximation methods within deal.II, including low-rank and hierarchical low-rank solvers and preconditioners to tackle large-scale computational problems;
- Developing block preconditioners for coupled problems, which are essential for the stability and efficiency of simulations.

The overall objective is to improve the pre-exascale functionalities of deal.II, making it an even more robust and efficient library for a wide range of scientific and engineering applications aiming for the use of exascale computers.

Gmsh API for large-scale distributed meshes

A new reader for *partitioned* Gmsh meshes into `parallel::fullydistributed::Triangulation` is under final review (PR #18759). The interface supports 1D/2D/3D meshes produced with Gmsh's built-in partitioner and automates vertex/cell connectivity and ghost handling across MPI ranks. This removes costly repartitioning inside deal.II and enables streamlined large-scale workflows for realistic organ geometries. The contribution is feature-complete and expected to enter the next release cycle.

Generalized interface for coupling operators

The interface has been stabilized and exercised on non-matching couplings and fictitious-domain settings, with Lagrange-multiplier enforcement used as a working reference. The implementation underpins new solver technology and supports both conforming and non-conforming constraints, preparing the ground for robust multi-physics coupling in organ-scale models.

Block preconditioners for coupled problems

A family of augmented-Lagrangian block preconditioners has been developed and analyzed for interface-constrained Poisson/Stokes problems arising from non-matching and fictitious-domain discretizations. Theory provides mesh-size independent spectral bounds; numerics confirm low, essentially constant FGMRES iteration counts without exact block inverses (loose inner tolerances suffice). A distributed-memory implementation based on deal.II is maintained publicly, and a preprint is available on arXiv [1]. Ongoing extensions target (i) elliptic interfaces with strong coefficient contrasts and (ii) fluid–structure interaction (see the report for deliverable D1.4).

Low-rank and hierarchical techniques via external solvers

deal.II 9.7 integrates MUMPS as a first-class backend, exposing block low-rank compression and mixed-precision options relevant to exascale factorization. This provides a robust path for coarse-grid and ill-conditioned subproblems in multilevel solvers, complementing matrix-free fine-scale kernels. In parallel, hooks for PSC-Toolkit/PSBLAS have been added to enable scalable AMG-based coarse solvers and GPU-capable sparse BLAS, aligning with hybrid matrix-free / matrix-based workflows.

Data pathways and ROM-enabling plumbing

A lightweight VTK interoperability layer (`vtk_utils.h`) has been prototyped to import unstructured meshes and multi-field data, infer suitable finite element layouts (vertex/cell, scalar/vector), and shuttle data between serial and distributed DoF vectors. While not a ROM algorithm per se, this infrastructure closes critical gaps in data ingestion and feature extraction pipelines needed by ROM and data-driven components (see report for deliverable D1.4).

References

- [1] M. Benzi, M. Feder, L. Heltai, and F. Mugnaioni "Scalable augmented Lagrangian preconditioners for fictitious domain problems," *arXiv Preprint arXiv:2504.11339*, 2025.

WP1.3: Experimental polygonal discretization module for deal.II

The primary objective of this sub work package is the development of a polygonal discretization module within the deal.II library. The deal.II library currently supports a wide range of finite element methods, defined on standard simplicial and hexahedral meshes, but does not support polygonal and polyhedral (*polytopic*) meshes.

Planned activities include:

- Develop data structure for polytopic meshes obtained by agglomeration of standard deal.II meshes;
- Exploit R-tree data structures for efficient generation of balanced and nested hierarchies of polytopic meshes and test use of such hierarchies of polytopic meshes to enhance multigrid solvers;
- Develop extensive library of examples of polytopic finite element methods for complex multiscale problems.

Polydeal library

The preliminary version of the polygonal discretization module developed within the first six months of the project was made available at <https://github.com/fdrmerc/Polydeal>. The module has since been extensively tested and parallelised. Milestone M5 has been reached and deployed as deliverable D1.5 (cf. the specific report for more details). The milestone was set to establish the parallel implementation of polygonal discretization methods. The deliverable provides a tutorial program on polygonal discretizations, showcasing the implementation of these methods for the solution of a basic model problem. Aside from the tutorial, a series of problems and tests have been run, confirming the validity and robustness of the approach for

the solution of problems in a dimension-independent setting. In particular, the polytopic discontinuous Galerkin methods have been implemented for the solution of a series of benchmark problems, including fluid mechanics and electrophysiology.

Data structure for efficient generation of balanced and nested hierarchies of polytopic meshes

The results of this early stage of the project are presented in the article “R3MG: R-tree based agglomeration of polytopal grids with applications to multilevel methods” [1], that clearly shows the potential of such discretization methods.

This work introduces a novel approach for the agglomeration of standard deal.II grids into polygonal and polyhedral meshes, based on spatial indices, specifically the R-tree data structure. The “R3MG” article demonstrates how building an R-tree spatial database from an arbitrary fine mesh offers a natural and efficient agglomeration strategy with several key features:

- The process is fully automated, robust, and dimension-independent;
- It automatically generates a balanced and nested hierarchy of agglomerates. This is a crucial property for the subsequent application of multigrid methods;
- The shape of the agglomerates is closely aligned with their axis-aligned bounding boxes and thus are characterized by good and easy to assess shape-regularity properties.

A fundamental aspect of this approach is the ability to automatically extract nested sequences of agglomerated meshes, which can be directly used within multigrid solvers. This is particularly relevant for WP1.3, as one of its objectives is the development of agglomeration-based multigrid methods for single-physics problems. The R-tree based approach, named R3MG (R-tree based MultiGrid), is proposed as a multigrid preconditioning technique with Discontinuous Galerkin methods. The experiments presented in the article, based on polygonal Discontinuous Galerkin methods, confirm the effectiveness of the approach in the context of complex three-dimensional geometries relevant to biomedical applications and in the design of geometric multigrid preconditioners, at least for model problems.

Furthermore, the “R3MG” article highlights how the R-tree-based method preserves mesh quality and significantly reduces the computational cost associated

with the agglomeration process, favorably comparing it with graph partitioning tools like METIS. In particular, it is demonstrated that R-tree-based agglomeration preserves structured meshes, a property not shared by METIS. The ability to generate nested grid hierarchies through R-tree agglomeration simplifies the use of simpler and more economical intergrid transfer operators compared to the non-nested case. In summary, the “R3MG” article represents a significant contribution to the objective of WP1.3 to develop an experimental polygonal discretization module in deal.II, providing an efficient and automated methodology for generating R-tree based agglomerated polygonal grid hierarchies, specifically designed for application with Discontinuous Galerkin methods and multigrid techniques.

Applications to fluid mechanics

The polytopic discontinuous Galerkin method has been extended to the solution of fluid mechanics boundary value problems with the analysis of the method for the solution of the Oseen problem. This required a new analysis which has been carried out for the hp -version of the method, hence permitting arbitrarily high-order implementations. Both stabilised equal order $(\mathcal{P}_k, \mathcal{P}_k)$ and pressure lower-order $(\mathcal{P}_k, \mathcal{P}_{k-1})$ versions have been analysed. New working examples implementing the new method have been committed by the SISSA team within the examples suits of the *polydeal* repository <https://github.com/fdrmerc/Polydeal>. Preliminary results show that the polytopic discontinuous Galerkin (polyDG) is achieving the optimal convergence rate when applied to the Oseen problem.

A publication presenting both the new sharp a priori analysis of the discontinuous Galerkin method for the Oseen problem as well as numerical experiments validating the approach for the solution of the Oseen problem is under preparation. It will be submitted to a top numerical analysis journal within the next semester.

References

- [1] M. Feder, A. Cangiani, L. Heltai: “R3MG: R-tree based agglomeration of polytopal grids with applications to multilevel methods”. *Journal of Computational Physics*, 2025, 526, pp.1-23. doi:10.1016/j.jcp.2025.113773

WP1.4: Integration of PSCToolkit within deal.II

The primary objective of WP1.4 is to integrate the PSCToolkit (Parallel Sparse Computing Toolkit) into the deal.II library. This integration aims to leverage GPU computing to enhance performance and develop efficient preconditioners for multiphysics problems. PSCToolkit provides advanced routines for sparse matrix operations optimized for GPU architectures, which are critical for large-scale simulations.

Planned activities include:

- Developing a GPU-accelerated interface for PSCToolkit within deal.II;
- Implementing efficient preconditioners tailored for multiphysics problems;
- Benchmarking the performance of the integrated toolkit on exascale systems.

Improvement in the new RC for PSBLAS and AMG4PSBLAS

To prepare for the integration with deal.II, we have developed a new release candidate of PSBLAS and AMG4PSBLAS that includes several modifications to the handling of the preprocessor directives. These modifications aim to improve compatibility and ease of use when integrating with other libraries and frameworks. Specifically, we have made the following changes:

- The preprocessor directives have been updated to have a `PSB_` prefix, which helps to avoid conflicts with other libraries and ensures that the directives are clearly associated with PSBLAS.
- The preprocessor directives have been moved to two dedicated header files for both PSBLAS and AMG4PSBLAS which are created at the compile time of the two libraries and are the `psb_config.h` and `amg_config.h` files, respectively. This allows for better organization and management of the preprocessor directives, making it easier to maintain and update them in the future.

As part of an ongoing effort to improve the ease of installation of the PSCToolkit libraries, the new RC also includes a new CMake module that simplifies the process of finding and linking the PSBLAS and AMG4PSBLAS libraries in CMake-based projects. Furthermore, we have added PSBLAS inside the Spack package manager, which allows for easy installation and management of the library in various

environments; see <https://packages.spack.io/package.html?name=psblas>. We expect to add the AMG4PSBLAS library to Spack after the new release candidate is finalized by the end of 2025.

Another addition to the RC stream was the introduction of the C interface to enable the creation of a parallel context in PSCToolkit from a pre-existing MPI communicator. This capability was already natively available in Fortran, but was not available from the C interfaces.

The interface with deal.II

The interface with deal.II we describe in the following is reported in the pull request <https://github.com/dealii/dealii/pull/18662>, which has yet to be merged in the main branch as of writing this document. The bulk of the interface is implemented in the two files

- `dealii/source/lac/psblas.cc`,
- `dealii/include/deal.II/lac/psctoolkit.h`,

in addition to the modification to the `CMakeLists.txt` files and scripts to enable the compilation of the new interface against the new PSCToolkit release candidate.

The interface is organized into four main namespaces: **Communicator** for MPI communication context management and descriptor operations, **Matrix** for sparse matrix creation and manipulation, **PSBVector** for vector operations and data distribution, and **Solvers** for preconditioners and Krylov solvers. The interface is conditionally compiled when `DEAL_II_WITH_PSBLAS` is defined, ensuring compatibility only when PSBLAS is available. A more complete description of these namespace and further information on the interface with deal.II is available in the deliverable D1.1 “External libraries integration”.

WP1.5: Integration of MUMPS within deal.II

WP1.5 aims to integrate the MUMPS (Multifrontal Massively Parallel Sparse) solver directly into deal.II. This task focuses on enabling its use in multigrid methods and exploring advanced techniques such as low-rank approximations and mixed-precision computations to enhance solver efficiency.

Planned activities include:

- Developing a direct interface between deal.II and MUMPS;
- Implementing low-rank approximation techniques to reduce computational costs;
- Exploring mixed-precision strategies to optimize performance on modern hardware.

Implemented improvements in the MUMPS support

The basic deal.II MUMPS interface did not include support for distributed-memory parallelism, which means that the system matrix and right-hand side(s) were entirely assembled on the master process where the subsequent phases (symbolic analysis, factorization and backward/forward substitution) take place; not only this might be infeasible due to memory limitations but it severely limits the performance of the sparse direct solver. The MUMPS integration in deal.II was extended to support distributed memory parallelism through the use of distributed matrix and vector datatypes as defined in the deal.II PETSc and Trilinos wrappers. This allows for a completely parallel initialization of the MUMPS data structure (that includes the system matrix and the right-hand sides) prior to the symbolic analysis, numerical factorization and forward/backward substitution.

Setting up MUMPS internal configuration parameters was not possible in the original deal.II MUMPS interface which, therefore, had to be extended to enable the use of the more advanced features of the MUMPS solver; these include the Block Low-Rank approximations, the GPU support and numerous other parameters that allow for fine-tuning both the performance and the numerical robustness of the solver. This extension was achieved by adding a `AdditionalData` object to the constructor of the MUMPS solver class that takes the desired values for selected solver parameters, following the common design patterns of the deal.II library.

The work related to above-mentioned extension is documented in pull request #18497 (<https://github.com/dealii/dealii/pull/18497>) which was eventually merged in the master deal.II branch.

Furthermore, the deal.II documentation was updated with a detailed description of the MUMPS API within deal.II: [https://dealii.org/developer/doxygen/dealii.classSparseDirectMUMPS.html](https://dealii.org/developer/doxygen/dealii/classSparseDirectMUMPS.html)

Preliminary experimental results

The new MUMPS interface has been validated through a series of tests to verify its correctness, using both serial and distributed sparse matrices. Such tests have been added to the deal.II test suite and are run automatically at each commit. In particular, the following additional steps have been carried out to validate the robustness of the current interface:

- Usage of Block-Low Rank approximations as a preconditioner for a diffusion problem in 2D and 3D, on a sequence of parallel and adaptively refined meshes, both for PETSc and Trilinos distributed matrices, thereby confirming the correctness of the distributed memory parallelism support and the capabilities of BLR as a preconditioner.
- Integration into the *Brain application*, as added in pull request <https://github.com/BRAINIACS-Group/ExaBrain/pull/1>, serving as a drop-in replacement for the previous solver based on Amesos_Superludist from Trilinos. Preliminary results across several parameter configurations show that the new MUMPS solver reduces the total solver time by a factor ranging from 2 up to approximately 3.

2.3 Next Steps

- Enhance the scalability of single-physics solvers in deal.II to exploit the full potential of large-scale exascale computing environments.
- Develop a generalized interface for coupling operators, key for multiphysics and multiscale simulations.
- Generalize and test polygonal discretizations, including coarse-grid hierarchy construction, mesh adaptivity, and multigrid methods, before integration into deal.II.
- Integrate AMG4PSBLAS for algebraic multigrid preconditioning and expose its GPU capabilities.
- Develop a direct interface between deal.II and MUMPS, and explore advanced techniques such as low-rank approximations and mixed-precision computations.
- Validate interoperability between polygonal discretizations, AMG solvers, and MUMPS direct solvers in multiphysics applications.

- Deliver comprehensive benchmarks on EuroHPC systems (LEONARDO, LUMI, JUPITER).

3 dealii-X lighthouse Applications for Digital Twins of the Human Body (WP2)

3.1 Objectives

The overall objective of WP2 is to advance accurate, efficient, and scalable digital twins of major human organs, building on the algorithmic innovations of WP1. Each sub-workpackage targets a specific domain while contributing to a unified multi-organ framework.

The key applications of WP2 include:

- Exascale simulations of the respiratory system;
- Exascale simulations in cardiac computational medicine;
- Exascale simulations of brain tissue mechanics;
- Development of a digital twin for the human Liver;
- Development of a digital twin for cellular Interactions;
- Low-code platform for human organ digital twin creation.

Specifically, the sub-work packages aim to:

- **WP2.1 (Lead TUM):** Develop exascale-ready solvers for respiratory flow and transport, with advanced time stepping, mixed meshes, and fluid–structure interaction, extending later to other biomedical flows;
- **WP2.2 (Lead POLIMI):** Enhance `lifex` for exascale cardiac simulations, including GPU-enabled solvers, high-order FEM, multiphysics integration, and multiscale models of healthy and pathological conditions;
- **WP2.3 (Lead FAU):** Optimize an HPC brain solver with parallelization, GPU acceleration, and nonlinear routines, enabling high-fidelity, patient-specific brain simulations;

- **WP2.4 (Lead FVB-WIAS):** Build and validate a liver simulator prototype with enhanced poro-visco-elastic models, realistic geometries, and coupling with cardiovascular flows;
- **WP2.5 (Lead UNIBS):** Develop cell-scale digital twins using novel space–time FEM and coupling of receptor dynamics with flow, validated against in vitro data;
- **WP2.6 (Lead EXACT LAB):** Deliver a low-code platform and meta-scheduler to simplify digital twin construction and execution across HPC resources.

Together, WP2 subprojects aim to push the boundaries of computational simulations for major human organs, leveraging and integrating the advancements in exascale computing and the deal.II software library produced in WP1.

3.2 Progress Achieved in Year 1

WP2.1 Lungs: Exascale Simulations of the Respiratory System

Application description and relevance

Insights into lung mechanics during breathing, but especially during mechanical ventilation, are crucial for many vital medical questions. One challenge is that those mechanical aspects are relevant on a wide range of length scales. For illustration, hundreds of millions of alveoli have a size in the two to three-digit micrometer range and a wall thickness that is small enough to allow for the necessary gas exchange between the air space and the blood. The overall lung, on the other hand, has a size of four to six liters. To bridge these scales, the trachea, with a diameter of roughly 2.5 cm, bifurcates for up to 24 generations into the parenchyma. The terminal bronchioles, the smallest conducting airways found in the 16th generation of this bronchial tree, are already tens of thousands in number, each being responsible for the supply of air to tens of thousands of alveoli. It is clear from these numbers that a fully resolved simulation of a lung incorporating the important effects on all length scales is not possible. Although simplified, generally reduced order or homogenized models of lungs exist in the literature, these models fail to incorporate some of the critical aspects of respiratory mechanics. With our decade-long expertise in lung modeling, we have identified surfactant dysfunction as one such aspect. Despite the evident

clinical relevance — manifesting itself, for example, as the infant respiratory distress syndrome in newborns — large-scale simulations of the human lungs incorporating dynamic surfactant (and surfactant deficiency) effects have not been possible due to the high interfacial resolution that would be necessary to capture the surfactant effects on the alveolar walls.

Preliminary steps

Together with our project partners, we have concluded that exascale-enabled simulation software is the key to illuminating these open problems via solving the nonlinear dynamics of fine alveolar structures in the presence of surfactants in a large enough domain. Insights gained via many such resolved simulations in various settings will unravel critical insights into surfactant effects. They will also enable the inclusion of such phenomena in more traditional reduced-order models of respiratory mechanics that are applicable at the bedside. We have already substantiated the necessary first steps with our partners in working towards this goal. As a first step, we have identified multiple appropriate geometries for our problem and obtained meshes at different sizes and refinement levels. We have set up a prototype implementation based on deal.II. Specifically, we began implementing the complex mechanics of biological tissue, including a surfactant-based surface energy formulation. Once this initial implementation has been verified, we will integrate it into our established open-source HPC framework ExaDG.

Key achievements

The lung application is built on the high-performance library ExaDG, which itself relies heavily on the deal.II matrix-free framework as its engine. ExaDG implements discontinuous Galerkin as well as continuous Galerkin solvers for engineering problems. Although the main focus of ExaDG has been computational fluid dynamics (also successfully applied previously in respiratory mechanics) with discontinuous, hypercube-shaped elements, more recent developments and improvements [1] have enabled solving structural problems with continuous, simplex-shaped elements, which is more typical and appropriate for the partial differential equations underlying elastodynamics.

In accordance with our goal of simulating the elastodynamic processes in the very fine alveolar structures of the lung parenchyma and understanding how these fine structures behave under the influence of surface tension effects, we are conducting initial simulations with an alveolar geometry using ExaDG. A mesh of a

smaller alveolar geometry that is able to show an exemplary displacement field under axial tension has been successfully generated. For such complex, thin-walled geometries, the construction of linear solver and preconditioner combinations constitutes an essential part of the development process. We have been experimenting with different setups to find the best solution. So far, multigrid preconditioning is the most promising approach.

Parallel to our work on using the alveolar geometry for ExaDG simulations, we have also been evaluating the applicability and implementing the enhanced-surface finite element formulation for the surfactant, which we had reported in deliverables D2.1 and D2.2. We expect this formulation to be incorporated into ExaDG in the very short term.

References

- [1] R. Schussnig, N. Fehn, P. Munch, and M. Kronbichler.: “Matrix-free higher-order finite element methods for hyperelasticity”. *Computer Methods in Applied Mechanics and Engineering*, 2025, 435, pp.1-23. doi:10.1016/j.cma.2024.117600

WP2.2 Heart: Exascale Simulations in Cardiac Computational Medicine

Application description and relevance

Computational models are increasingly used in cardiology to understand the heart's pathophysiology and assess the effectiveness of treatment strategies of therapeutic devices. Notably, numerical simulations have great potential in enabling in-silico clinical trials, complementing and augmenting more traditional population-based studies. To this end, cardiac models must account for the different physical processes underlying the heartbeat. These include electrophysiology, muscular force generation and mechanics, fluid dynamics of the blood, valvular dynamics, myocardial perfusion, as well as the numerous interactions and feedback effects between them. Depending on the specific application, some of these may be neglected or reduced in complexity. Nonetheless, a comprehensive framework for cardiac simulation will inevitably lead to a multiphysics coupled model, entailing partic-

ularly challenging numerical problems and calling for high-performance computing techniques tailored to large-scale computations.

Preliminary steps

Pre-exascale and exascale simulation software are key in allowing large simulations in sufficiently large number to support in-silico clinical trials. In this respect, previous studies on cardiac simulations with the lifex software have highlighted how the performance bottleneck is currently associated to the efficiency and parallel scalability of linear solvers and preconditioners. Therefore, the initial phase of the project will be devoted to migrating from traditional, matrix-based implementations to matrix-free methods, and carefully evaluating the performance in biophysically detailed test cases, comparing with the current state of the simulation software. If successful, this will provide support for exploring high-order discretizations, which particularly benefit from matrix-free implementations, as well as GPU-based computing strategies

Key achievements

the POLIMI team is working on migrating the lifex solvers to deal.II's matrix-free infrastructure, and is extensively refactoring the code organization towards this goal. In the new implementation, each concrete PDE model (such as cardiac electrophysiology, muscular mechanics or myocardial perfusion) builds upon the same abstract software components (i.e., it is derived from the same classes), as opposed to the old implementation, where different models would often reimplement common tasks (such as the initialization of finite element spaces or linear system assembly). Besides significantly improving the code organization and its maintainability, this centralized structure significantly facilitates the implementation of new physical models and their improvement, and overall provides a structured scaffolding for the future library developments.

The abstract physical model classes at the core of this new implementation leverage deal.II's facilities to support many solver configurations in a flexible way: they allow for scalar- and vector-valued problems, linear and nonlinear, steady and unsteady, on hexahedral and tetrahedral meshes, and support both matrix-based and matrix-free solvers. Finite element evaluation tasks are carried out with the `dealii::FEEvaluation` class, instead of `dealii::FEValues` as done in the previous implementation. This enables a significant performance improvement already in the matrix-based solvers, thanks to the use of sum factorization, SIMD vectorization, and improved cache friendliness.

The solver for the monodomain problem of cardiac electrophysiology has been implemented on top of the new infrastructure, and its parallel performance has been compared to the one of the old code by running a strong scalability test based on the Niederer benchmark. The results show a pronounced improvement in both absolute wall times and parallel scalability, with wall times reduced by a factor 1.5–30, depending on mesh type and number of parallel cores. We note in particular that the vectorization of the ionic solver has led to a much clearer scalability. This is even more noticeable when observing that the cost of solving ionic models often dominates electrophysiology simulations.

Additionally, we have migrated to the new implementation of the multicompartment Darcy solver for myocardial perfusion, and the solver for generating rule-based cardiac fiber architectures. Finally, we are in the process of extending this new framework to the cardiac mechanics solver, as well as coupling the latter to the Darcy model to simulate the behavior of a porous medium subjected to fluid flow.

The upcoming work will follow along these lines, by progressively extending the new infrastructure to additional models (e.g. active force generation for muscular contraction) and introducing coupling between the existing models to build multi-physics simulations (e.g. electromechanics or poromechanics for myocardial perfusion). Complementary to this, we will proceed by expanding the set of numerical methods supported by the core solvers. Most notably, the code currently supports only black-box preconditioning methods (AMG, ILU, etc.), which are not well suited for the matrix-free solvers. Since preconditioning is crucial for the robustness and efficiency of cardiac mechanics solvers, this will be addressed shortly, by integrating a multigrid method within the new code.

WP2.3 Brain: Exascale simulations of brain tissue mechanics

Application description and relevance

Despite decades of research, the human brain still poses exciting challenges for researchers from various fields. More recently, there is increasing interest in the role of mechanical signals for brain development, injury, and disease. Modeling based on the theory of nonlinear continuum mechanics proves a valuable tool to computationally test hypotheses that complement experimental findings, to understand processes in the brain under physiological and pathological conditions, and to

assist diagnosis and treatment of neurological disorders through personalized predictions. Depending on the application, mechanical models for human brain tissue need to cover a wide range of time and length scales. Its highly heterogeneous, region-dependent microstructure relates to viscoelastic and poroelastic effects that cannot be neglected for predictions on the organ scale. Viscoelastic models with two relaxation times have been successful in capturing the time-dependent mechanical response of brain tissue under various loading conditions. However, free-flowing interstitial fluid occupies a large fraction of the brain volume and contributes to the biomechanical response of human brain tissue through poroelastic effects. For some applications, e.g., drug delivery in the brain during cancer treatment, it is thus essential to model the porous properties of brain tissue explicitly.

Preliminary steps

Our versatile poro-viscoelastic model is already implemented in deal.II and provides the possibility to describe and explore the underlying physical mechanisms within a biphasic material during mechanical loading. However, identifying the associated model parameters becomes increasingly difficult with increasing model complexity. As the various brain regions show distinct mechanical properties and as it is virtually impossible to achieve homogeneous deformation states during testing due to the ultrasoft nature of brain tissue, an inverse parameter identification algorithm that captures the exact boundary conditions of the test must be run separately for every single brain region based on meaningful experimental data. In this context, the underlying, computationally demanding multiphysics problem needs to be solved hundreds to thousands of times to identify a reliable set of material model parameters for the entire human brain. Therefore, ongoing work focuses on the transfer of our existing code to an HPC environment. Once successful, we will proceed with the parameter identification whilst also setting a first benchmark to identify potential for computational improvements with our project partners within dealii-X. As simulations of the full human brain require very fine resolution due to its complex geometry, exascale-enabled software will be the key to improve our understanding on mechanical mechanisms within the brain and to take advantage of the modeling capabilities, e.g., in clinical applications.

Key achievements

Main activities were directed towards the progress within the ExaBrain library, which is developed to facilitate high-resolution full human brain finite element simu-

lations using a complex nonlinear poro-viscoelastic material model.

On the one hand, we transferred our code to the National High Performance Computing Center (NHR@FAU) at FAU in Erlangen. This allowed us to perform first benchmarks on the code performance and scalability in an HPC environment which will also serve as the baseline for upcoming developments. Overall, we compared the performance for several system sizes and two different direct solvers. Preliminary results were carried out for the – so far – largest system consisting of 24,576 hexahedral Q2P1 (quadratic in displacements, linear in pore pressure) elements and 644,916 degrees of freedom. The simulations were run on HPC system “Fritz” with our baseline parallel direct solver SuperLU-dist and a newly available MUMPS direct solver, which was incorporated into deal.II through our project partners at University of Pisa and the work of the MUMPS team at INPT. The new MUMPS solver decreases the memory requirement by approximately 50 % to 75 %, depending on the number of parallel processes, while the system assembly is independent of the solver and scales perfectly with the number of processes. While the new MUMPS solver is significantly faster than SuperLU-dist, it does not show good scalability through parallelization, and a crucial improvement towards larger systems is required for the full human brain.

Further improvements for the solver phase of the linearized system are possible by employing iterative solvers. Some first comparisons between the direct solvers previously mentioned and GMRES as an iterative solver are available. The simulations were run on the University of Pisa HPC system “Toeplitz”. GMRES is preconditioned with the MUMPS solver using the mixed precision Block Low Rank feature. Multiple strategies can be employed to precondition GMRES with BLR: the same preconditioner can be reused to precondition multiple executions of the GMRES solver, reducing the number of times the preconditioner has to be evaluated and thus reducing the overall time spent in the linear solver. Ongoing work is focusing on optimizing the number of times the preconditioner must be evaluated in relation to the time required for the convergence of GMRES to further increase the speed of the solve phase.

On the other hand, as full brain simulations require proper material parameters, FAU is working on the efficiency and reliability of the inverse parameter identification. Here, our new HPC possibilities allow us to perform numerical experiments that can then help to identify potential experimental and algorithmic improvements. Starting with a known set of poroelastic parameters, we performed a relaxation ex-

periment of a cylindrical specimen and evaluated the total nominal stress as well as the lateral displacement of a point on the lateral surface at half of the specimen height. This numerical “experimental” data was then used as an input for the inverse parameter identification with 16 different sets of initial parameters. In one case, the objective function included only the total nominal stress, while in another case it also incorporated the lateral displacement with equal weighting. Even for this simplified poroelastic example using “perfect” experimental data, the nominal stress response alone does not provide enough information to identify a unique parameter set. In fact, the correct parameter set was recovered only in 12 out of 16 cases. The situation improves significantly if we incorporate the lateral displacement as additional information. For our poro-viscoelastic human brain model, the number of unknown model parameters increases from four to seven strongly coupled material parameters. Therefore, numerical experiments evidence even more importance in understanding the model response and interpreting the results from inverse parameter identifications. With an increasing number of model parameters, we have to cover a significantly larger parameter space, where we will benefit substantially from our improvements on the linear solver and the inverse identification itself.

WP2.4 Liver: Development of a Digital Twin for the Human Liver

Application description and relevance

Computational modeling of liver tissues requires handling several spatial scales, as well as the interplay of both fluid and solid phases at various levels. However, numerical handling at the smallest scales is unfeasible due to the extreme computational costs associated with discretizations.

Effective (homogenized) tissue models can be successfully used to describe macroscale properties of the liver. However, in selected applications, understanding the tight coupling between fluid and solid components of a vascular tissue has relevant implications in the field of medical imaging and diagnosis. In fact, tissue imaging techniques such as Magnetic Resonance Elastography (MRE) are able to characterize mechanical properties from phase-contrast MRI imaging and, in connection with suitable physical models of tissue, can provide insights in pathologically relevant biomarkers. An efficient and robust numerical handling of the scales will therefore also provide insights into how effective tissue properties (such as compressibility or elastic modulus) can be related to interstitial pressure and fluid properties of the

microscale vasculature.

Preliminary steps

As the first step, in collaboration among project partners, a new mathematical model that accounts, in a multiscale fashion, has been proposed for the effect of a thin (one-dimensional) fluid vessel in a three-dimensional elastic matrix. This model can be efficiently used to represent different realizations of vascular structures, as well as the coupling with corresponding flow models. Pre-exascale-enabled simulations will allow to extend the capabilities of the multiscale model and handle liver scale simulations of effective tissues, both for forward mechanics and inverse problems related, e.g., to the identification of effective parameters from MRE measurements. The current multiscale model has been implemented in deal.II and it will be extended with automatic generation of physiological microvasculature, considering different physical models for the solid matrix. The resulting model is being integrated into the open-source HPC dealii-X framework.

Key achievements

The computational model is built on deal.II, and on the implementation of reduced Lagrange multipliers for multidimensional PDEs. An efficient approach for handling the multiple scales of the problem (fluid flow in the vasculature and elastodynamics in the tissue) is necessary to handle large-scale problems.

The reduced Lagrange multipliers approach formulates the weak coupled (fluid-structure) problem in a weak Lagrange multiplier fashion, applying then a model order reduction based on the properties of the relevant physics (i.e., the approximation that flow can be described by one-dimensional Navier-Stokes equations). The work in WP2.4 focused on the application of this framework to multiscale elasticity.

The reduced Lagrange multiplier model extends the previously state-of-the-art multiscale models to the case of Dirichlet boundary conditions enforced at the vessel-tissue boundaries. This condition is necessary to handle the coupling with pulsatile vasculature. To this purpose, a non-standard interface condition has been considered, which removes macroscopic motion (translational and rotational) and only enforces the coupling at the level of local normal displacement of the vessel. The mathematical framework of this model provides the right structure for selecting the proper reduced-order space, and extending these results, we proved well-posedness at the continuous level of the new formulation. The model has been used

to perform *in silico* experiments relating the properties of the microstructure to the effective elastic parameters of a tissue sample. These experiments represent the first step towards the formulation of an inverse problem for the parametrization of effective liver samples. Results have been submitted for publication to the International Journal of Numerical Methods for Biomedical Engineering [1].

Following the development of the model, current work is considering the coupling with an active one-dimensional model for the blood flow, extending previous models. The coupling is implemented, representing the one-dimensional network (described by nodes and edges) within the three-dimensional domain as a discrete set of singular points, on which a normal displacement is imposed on the tissue, based on the vessel pulsation. The resulting tissue pressure is, in turn, considered as an external force in the blood flow model. The coupling was first implemented in a staggered fashion, performing preliminary investigations on the stability of the model. A monolithic implementation is under development.

The upcoming work will focus on the generation of realistic vasculatures using physiological parameters specific for the liver, as well as the connection with data assimilation methods to integrate available data for model parametrization.

References

- [1] C. Belponer, A. Caiazzo, L. Heltai.: “Mixed-dimensional modeling of vascular tissues with reduced Lagrange multipliers”. *arXiv Preprint*, 2025, [arXiv:2309.06797](https://arxiv.org/abs/2309.06797)

WP2.5 Mechanobiology: Development of a Digital Twin for Cellular Interactions

Application description and relevance

Cell motility plays a crucial role in processes such as tumor metastasis and embryogenesis. It results from a complex and continuous cycle of actin polymerization and depolymerization. The interactions between cell surface receptors and ligands in the extracellular matrix trigger these processes, and are vital for various physio-

logical and pathological events. These interactions ultimately lead to the reorganization of the cytoskeleton, including the formation of a new protrusive network at the cell's leading edge and the contraction of the rear through myosin action in stress fibers. Traditional continuum chemo-mechanical theories, such as those developed by Larché and Cahn, fail to fully capture the dynamic interplay between mechanics, chemistry, and species transport that governs this cytoskeletal reorganization. To address this, our approach introduces and implements new partial differential equations (PDEs) that describe cellular motility, extending beyond Larché-Cahn theories.

Preliminary steps

In the starting phase, we have determined that pre-exascale-enabled simulation software is essential for addressing these challenges by modeling the dynamics of cytoskeletal structures in the presence of extracellular ligands across various environments. The new software will enhance the pre-exascale deal.II codes, currently available at The Mechanobiology Research Center (TMRC) at UNIBS, with the overarching goal of advancing mechanobiological research. As an initial step, we are developing a prototype implementation for single-cell motility using the deal.II codes generated at TMRC. The class hierarchy and the code for separated physics (mechanics, chemistry, and transport) have been completed. Soon, coupling strategies of monolithic and staggered nature will be developed for the class of Larché-Cahn and non-Larché-Cahn governing equations.

Applications of this novel code will include the relocation of receptors on advecting lipid membranes, and it will be integrated into our existing, general multiphysics open-source HPC framework.

Key achievements

Traditional continuum theories, such as those proposed by Larché and Cahn and Mixture Theories, fall short in capturing the complex coupling of mechanical, chemical, and transport phenomena that govern cytoskeletal reorganization, as they lack the ability to describe the dynamic creation and disassembly of networks. The mathematical foundations of our approach—including the underlying kinematics and balance equations—are detailed in an upcoming publication [1]. The computational model is implemented in dealii-X and employs staggered algorithms to solve the coupled PDE system involving chemo-transport and mechanics. Manufactured solutions have been tested in parallel computing environments utilizing multiple processors and distributed memory architectures.

The main work was focused on a Lagrangian approach, adopting both global and adaptive local mesh refinement strategies, with the latter guided by the Kelly-ErrorEstimator in deal.II to estimate cell-wise error and improve solution accuracy. Simulations show the ability of the code to generate force-bearing domains upon polymerization of monomers. These simulations apply well in hemostasis, where the growing domain simulates the development of fibrin networks from activated platelets with real data.

The simulations started from an experimentally informed initial configuration of platelets surrounded by thrombin, which drives clot formation. Using adaptive mesh refinement on a parallel computing architecture, the model captured the progressive development of the fibrin network over time. As polymerization proceeded, the load-bearing structure of the network expanded, while the highest fibrin concentrations consistently appeared near the platelet regions where the activation signal was applied, gradually decreasing with distance.

The network concentration reports the following result: Since the activation signal is applied within a fixed region surrounding the platelet in the current configuration, we observe a higher fibrin concentration near the platelet, with decreasing concentrations farther from the activation region.

An Eulerian framework for solids suits well the goal of the dynamic creation and disassembly of networks. The Eulerian perspective is particularly effective in cellular motility, where signaling occurs in localized zones naturally described in the current configuration. Future work will focus on extending the current code, providing realistic mechanisms for the generation and development of the actin cytoskeleton using parameters specific to fibroblasts and potentially other cell types.

References

- [1] A. Salvadori, M. Serpelloni, R. M. McMeeking: “Chemo-thermo-mechanics for mixtures of solids generated by a chemical reaction within a liquid.”. *to be submitted to: Journal of the Mechanics and Physics of Solids*, 2025.

WP2.6 PoC: Low-code platform for human organ digital twin creation

Main objectives and relevance

One of the long-term goals of the dealii-X project is to develop key components for diagnostic and training tools in the fields of personalized medicine. In order for such tools to be used by a large number of professionals, some intuitive yet complete interfaces should be created. Moreover, different levels of interaction should be envisioned: a biomedical engineer should be able to access the mechanical properties of the organs or tissues, while a surgeon could be interested in intuitive ways to drag an MRI scan into the system and have an interactive simulation without having to deal with the technical details of the simulation.

Currently, the typical workflow for a user of the dealii-X libraries to run a simulation starting from medical data involves the following steps:

- Convert a medical input data (e.g. MRI, X-ray scan, etc.) into a mesh using an appropriate software
- Write a C++ program that models the physical process (the digital twin)
- Run the simulation on an HPC cluster
- Fetch the resulting data to a local machine
- Write/Use post-processing code to prepare the results for visualization
- Use an application to visualize the data

The flow described above is rather complex, and generally not suitable for a non-expert user. The goal of the PoC platform is to allow a user to perform all these steps in a single environment, while retaining the possibility of deep customization.

In view of these diverse use cases, we created a list of requirements that the dealii-X PoC platform should sport.

- A collaborative space, in which professionals with different needs can contribute
- A modular design, to allow components to be easily added

- An interaction that allows deep customization without requiring specific coding skills
- A transparent usage of computational resources, ranging from local machines to HPC clusters

Those requirements naturally bring into play the programming approach that goes under the name of "Low-code/No-code".

Low-code approach to deal.II programming

In the present section, we present some details of the design of a node-based low-code interface for an Object-Oriented Library, using deal.II as an operative example.

A minimal set of entities and a control loop that allow the creation of an object-oriented, Turing-complete language is the following:

- Variables and instances of objects
- Functions and member functions
- Loops
- Conditionals

Each of these entities can be represented as a node in a graph, with inputs and outputs. The graph is then traversed in a way that is consistent with the logic of the programming language. To specify the order of the operation, special connectors on the node are used, and special edges are displayed to distinguish instruction flow from arguments. The object works in the same way, but member function nodes can be attached to the object instance node, and the output of the function can then be passed to other nodes.

The power of node-based programming is twofold: on one side, it allows for creating complex operations without knowing the details of a specific programming language. On the other side, it allows for encapsulating a collection of nodes into a single node that can be reused in other parts of the program. These hierarchical features unleash the possibility of creating high-level entities that can be used by non-expert users. The latter is the key feature that links the low-code and no-code

approach, allowing the creation of a single environment that can be used by both expert and non-expert users.

Moreover, the node-based approach naturally leads to the definition of a computational graph that can be used to optimize the execution of the program: by analyzing the graph, it is possible to determine which parts of it can be executed concurrently, which allows a potential second layer of parallelization on top of the one that is already present in the libraries. Operatively, we are currently investigating the use of the [taskflow](#) library to implement graph analysis and node scheduling.

dealii-X Low-code/No-code interface

This subsection reports the progress on the Low-code/No-code interface as well as the related backend. The development follows the approach and the design outlined in Deliverable 2.6 “Design of the PoC platform”. The system is composed of three logical parts, which correspond to the following subsections:

- deal.II-X Client (Local client app) - the frontend and main user interface, used to create the computation graph and transparently submit calculations;
- deal.II-X Cloud (Remote server) - the web server, managing user accounts and communication with the client;
- CORAL - the transducer of the computational graph into executable code.

The client acts as a starting point for every new project. Every new computational graph is created or loaded into the client, choosing from a list of available computational nodes. The client will upload each computational graph to the HPC cluster and send the command to start the computation. The state of the computation will be displayed in the client. The client will also save the graph to the remote server, and the work will be sharable with other users. The HPC cluster and the cloud will never communicate directly, but always through the client. The detailed description of each component is available in Deliverable 2.2 “Report on application improvements - II semester”.

3.3 Next Steps

- Advance ExaDG simulations of alveolar geometries to study elastodynamics under surface tension, integrate surfactant formulations, and develop efficient

multigrid-based solvers for thin-walled structures.

- Continue transition from matrix-based to matrix-free methods, enabling high-order discretizations and GPU acceleration; introduce tailored multigrid preconditioners; and progressively extend the infrastructure to multiphysics models such as muscular contraction, electromechanics, and poromechanics.
- Develop exascale-ready simulations of brain mechanics, supporting high-resolution modeling of complex geometries for clinical and research applications.
- Generate physiologically realistic vasculature models, integrate data assimilation for parameterization, and explore the relationship between tissue mechanics and microscale vascular fluid dynamics.
- Extend the Eulerian framework for solids to model cytoskeletal remodeling in fibroblasts and other cell types, and simulate lipid membrane receptor dynamics within the general multiphysics HPC framework.
- Provide proof-of-concept demonstrators showcasing accessibility and integration of the developed HPC frameworks through low-code/no-code interfaces.

4 Co-Design, Technology Exploitation, and Energy Efficiency(WP3)

4.1 Objectives

WP3 focuses on advancing co-design, technology exploitation, and energy efficiency within the dealii-X project. The objectives are:

- Investigate novel hardware and software solutions to leverage the latest advancements in high-performance computing to achieve optimal performance
- Identify and evaluate emerging technologies that can be exploited to enhance the capabilities of the exascale computing system.
- Explore and implement techniques to optimize energy consumption and power efficiency in the exascale computing infrastructure.
- Identify and address bottlenecks in the system to improve overall computational efficiency.
- Foster collaboration within the CASTIEL2 network and other relevant HPC initiatives to leverage shared expertise and resources.

As part of this work package, the benchmark repository is made publically available at: <https://github.com/dealii-X/benchmarks/>. The preparation of these benchmarks has been the focus of Milestone #3 – Public repository of benchmark sets for performance evaluation – verified by means of runs of the software on at least three different accelerated platforms.

4.2 Progress Achieved in Year 1

CEED Backoff Problems

The CEED Bakeoff Problems are a benchmark to measure the evaluation speed of the basic finite element operators (discretized bilinear forms), posed for problems on hexahedral grids, and the primary focus is on sum factorization techniques. Both scalar and vector problems are included in the benchmarks. So far our efforts have been focused on the scalar problems. Each of the Bakeoff Problems (BP) has an

associated Benchmark Kernel (BK) that evaluates a particular finite element operator and is of high arithmetic intensity. The efficient implementation of these kernels is paramount to obtaining good performance.

The subset we have focused on so far are,

- BK1: scalar PCG with mass matrix, $q = p + 2$
- BK3: scalar PCG with stiffness matrix, $q = p + 1$
- BK5: scalar PCG with stiffness matrix, $q = p + 2$

The benchmark kernels form the algorithm core of the benchmark problems (BP), which include a complete finite element workflow, including mesh setup, partitioning, and parallel computation.

The initial kernels have been implemented in three variants: 1) sequential kernels for verification, 2) using Kokkos Core, a C++ programming model for writing portable applications, part of the larger Kokkos C++ Performance Portability Programming Ecosystem¹ and 3) using the CUDA Driver API. We note the deal.II library currently bases the portable kernels on Kokkos, and the investigations presented in this report have the objective of providing insight into the behavior of the methods on modern hardware. In order to evaluate the impact of parallelization strategies, the Kokkos and CUDA-based kernels have several sub-variants that use different thread-mapping strategies, i.e., the assignment of degrees-of-freedom and quadrature points to GPU threads.

In order to obtain a wider impression of GPU programming models available we have also created variants using OpenCL, OpenMP target offloading and a variant for the Tiny Tensor Compiler, a recently introduced domain specific language for tensor operations from Intel.

Below, we briefly mention the architecture systems and performance results achieved, and refer to Deliverable D3.2 “Co-Design and Energy Efficiency Report I” for more details.

Kokkos and CUDA Performance Results

BK1. Early performance results for the different test cases are evaluated on an

¹Kokkos, <https://kokkos.org>

Nvidia H100 GPU. Each test case uses 3D thread blocks, determines the number of quadrature points at run time, and employs thread blocks for element-wise computations. The only difference between the two cases is the programming model: CUDA for testcase 1 and Kokkos for testcase 2.

The first finding is that polynomial order 1 shows lower performance compared to higher-order elements in both test cases. This behavior, rooted in a first-order polynomial, is configured with 27 threads per block. This leads to under-utilization of the warp (size 32) on the target hardware. The second finding is that Kokkos achieves approximately 90% of the performance observed with the CUDA model, demonstrating that the performance portability model introduces only a minor overhead.

BK5. Unlike the multiple polynomial order evaluations in the BK1 test cases, we analyzed the kernels with a fixed cubic polynomial order ($p=3$) and templated quadrature points. Using templates shifts certain computations from runtime to compile-time, allowing the compiler to apply optimizations. As a result, templated implementations are often significantly faster, in most cases improving performance by 30% at least, compared to their runtime counterparts.

Both CUDA and Kokkos kernels were implemented with three-dimensional thread blocks and a simple data mapping approach, where each thread operates on a single data entry. The total number of degrees of freedom (DOF) in this test case is 27 million. Kernel performance values were measured by NVIDIA Nsight Compute, and, in these measurements, both CUDA and Kokkos executions showed identical performance. The corresponding arithmetic intensity was found to lead to 86% of peak bandwidth utilization (2.88 TB/s out of 3.2 TB/s). The kernels also achieved 75% occupancy, and register usage per thread was identified as the primary limiting factor.

Overall, the results demonstrate that both CUDA and Kokkos BK5 implementations deliver near-optimal performance on the H100, efficiently exploiting available memory bandwidth and achieving performance levels very close to the hardware limits.

OpenCL

For experimentation across a wider array of potential accelerator devices, the CUDA kernels have also been translated to the OpenCL-C language for use with

the vendor-neutral OpenCL framework from Khronos. OpenCL remains widely supported across both CPU and GPU devices from multiple vendors. The verbose setup required via the OpenCL runtime API, the split kernel source model, and restriction to C level language semantics, are some of the main criticisms from scientific software developers. The growing popularity of C++ coupled with powerful syntactic features has shifted the attention to more expressive programming models including SYCL and Kokkos, which provide similar capabilities as OpenCL, but are arguably simpler to use.

OpenMP Offloading

Since its introduction in OpenMP 4.5, offloading has become an integral component of this directive-based parallel programming framework. Adding OpenMP directives for offloading to an existing code can be quite straightforward, assuming the right data structures and loop patterns are already in place. When using OpenMP for GPU offloading, it is important to distinguish between two categories of directives for memory movement and work sharing.

Similar to other GPU programming models, OpenMP offers fine-grained control over the hierarchical parallelism in terms of teams and threads. These are reflected by the two directives, ‘omp teams’ which replicates execution across a league of teams and ‘omp parallel’ which replicates execution across threads of a team. The two directives can also be used to program in a SPMD-like mode (also called “me” mode). Typically, however, the directives are combined with work-sharing constructs like ‘distribute’ and ‘for’ which control the parallelization of (nested) loops. A very useful work-sharing construct is the ‘[teams] loop’ directive, which provides a descriptive form of parallelism in contrast to other prescriptive directives. With the ‘target [teams] loop’ directive, the precise distribution of the loop iteration space across teams and threads is left to the compiler. This kind of descriptiveness can provide superior performance portability, assuming the compiler succeeds in auto-parallelizing the loop well.

Tiny Tensor Compiler

The Tiny Tensor Compiler (TinyTC) is an open-source tensor compiler developed by Intel for efficient execution of tensor computations on CPUs and GPUs. TinyTC compiles programs written in a domain-specific tensor language into OpenCL-C or SPIR-V, supporting runtime environments such as OpenCL, Level Zero, and SYCL. C and C++ APIs are available. TinyTC assumes a batched execution model, where

each kernel is executed by a work-group with concurrent work-items, and the compiler maps tensor operations to efficient GPU instructions, including cooperative GEMMs and subgroup vectorization.

Our first test of TinyTC aims to accelerate the finite element mass matrix evaluation (BK1) using sum factorization. As a first demonstration of tensor language, we present the BK1 kernel using an algorithm, where each tensor product is implemented using the loop over GEMM approach. For simplicity, the implementation uses four temporary arrays, `wsp0-wsp4`, allocated in group local storage. As the `alloca` instruction reserves temporary memory in shared local memory, TinyTC automatically inserts thread barriers between each of the six GEMM passes to ensure functional correctness. The kernel shown above achieved 10 GDoF/s on the Intel Ponte Vecchio GPU, roughly one quarter of the performance we observed using OpenMP. One of several optimizations possible is fusing dimensions, to obtain larger GEMMs with a more favourable operation count.

Other GEMM steps can be fused in a similar manner, however this only delivered a few percent improvement. Upon consultation with the TinyTC author it became apparent a different strategy would be needed to obtain performant execution and optimal SIMD utilization of the PVC architecture. In collaboration with Intel, we are currently investigating the application of index fusion, batch vectorization and memory layout transformations to guarantee optimal register and SIMD usage.

Streaming kernels

Iterative linear system solvers make heavy use of low arithmetic intensity operations, so-called streaming operations. When matrix-free high-order FEM schemes are combined with iterative solvers. Previous research in the case of the conjugate gradient method shows that above a certain problem size, the streaming operations contribute a significant chunk of the solver run-time.

Streaming kernels capture a second important group of low-intensity operations, where memory movement is dominant. The kernels generally take one or two long vectors as arguments. The first four kernels included are: copy (COPY), scaled sum of vectors (AXPBY), squared norm (NRM2), and inner product of two vectors (DOT). The fifth kernel is the fused conjugate gradient update that involves the combination of a scaled sum and norm of a residual vector in a single sweep. The last two kernels

are the gather and scatter kernels that involve packing and unpacking the degrees of freedom associated with a given finite element mesh into a linear vector. The gather and scatter kernels are tied to a particular mesh and finite element space, requiring more extensive driver setup, and they are planned for future work.

As an initial step toward the performance characterization and modelling of streaming operations, we have implemented the streaming kernels as a portable Fortran program using OpenMP for parallel CPU and GPU execution. We have also implemented “backends” using other programming solutions, including C++ array libraries for linear algebra such as Eigen² and/or common linear algebra libraries for Fortran and C such as BLAS and BLIS³. In the remaining months of the work package we also plan to expand the backends to cover Kokkos and C++ standard parallelism for better cross-platform assesment.

Our preliminary work has been concerned about studying the bandwidth scaling with respect to number of threads for the streaming kernels on two recent CPU architectures: an Nvidia Grace CPU (part of the Grace Hopper Superchip) and an Intel Sapphire Rapids (Intel Xeon Platinum 8480+) CPU. We observed the expected saturating behavior. The Grace CPU achieves a throughput of over 300 GB/s over all kernels, exceeding 50% of the peak memory bandwidth (567 GB/s). The BS3 kernel reached almost 450 GB/s, a value close to the maximum memory bandwidth of 467 GB/s. In the case of BS2 and BS5, the saturated regime was reached with only 20 out of 72 cores. Similar saturating behavior was observed for the Intel Sapphire Rapids, with bandwidths in the range of 250–350 GB/s, close to the peak bandwidth. A slightly slower rate can be observed for the BS1 kernel, including a step increase when using more than 46 cores, warranting more investigation. This behavior appears to be linked to the write-allocation evasion mechanism of modern Intel CPUs.

Overall, our initial results confirm that the proposed streaming kernels are able to expose memory bandwidth behavior consistently across different architectures, demonstrating their usefulness as a lightweight proxy for iterative solver workloads. The observed saturation patterns highlight the importance of systematic performance characterization. In the next stage, we aim to extend the benchmarking to a broader set of CPUs and GPUs and integrate the results into predictive performance models for matrix-free iterative solvers.

²Eigen, <https://eigen.tuxfamily.org/>

³BLIS, <https://github.com/flame/blis>

Energy Efficiency

The shift to heterogeneous computing with accelerators is driven not only by the demand for higher performance but also by the increasing power and thermal constraints of modern semiconductor technology. These pressures have led to hardware specialization for more energy-efficient computing, as reflected in the Green500 list, where GPU-accelerated systems dominate. As shown by [1] in a case study of an astrophysical simulation, accelerators can deliver a 5–10× improvement in energy efficiency (flops per watt) over CPUs. This advantage stems from their throughput-oriented parallel design, which contrasts with the latency-focused nature of traditional CPUs.

Maximizing these benefits requires energy-aware practices from both HPC practitioners and application developers. Since power and energy usage vary across devices, workloads, and runtime conditions, precise modeling is difficult, making empirical measurement essential. System-level monitoring frameworks such as Intel's RAPL, AMD uProf, NVIDIA's NVML, or Arm's Streamline provide access to hardware counters, while job- and process-level tools such as the Energy Aware Runtime, perf, likwid-powermeter, and PAPI enable finer-grained insights. For kernel-level breakdowns, vendor tools like NVIDIA Nsight or Intel VTune are available, but they often involve heavy instrumentation and vendor lock-in. In practice, lightweight and portable solutions remain highly attractive.

To this end, [1] proposed a process-based approach in which applications call an external “energy-meter” script during execution. The script samples instantaneous power through utilities such as ‘nvidia-smi’, ‘xpu-smi’, or ‘perf’, and integrates the readings to estimate total energy consumption. A key advantage is that programmers can call the meter only around the sections of interest, thereby excluding initialization or I/O phases and focusing on the algorithmic kernels. The method is portable, simple to maintain, and adaptable to a variety of devices or custom meters.

In the remaining period of this work package, we plan to adopt this strategy to measure the energy usage of the core algorithmic kernels and gain an overview of their power behavior. Although we have not yet carried out such experiments, we take inspiration from a recent presentation by S. Cielo (LRZ), who demonstrated the benefits of these techniques, and we intend to implement them as part of our future performance and energy-efficiency evaluations.

References

- [1] S. Cielo, A. Pöpl, I. Pribeć.: “ SYCL for energy-efficient numerical astrophysics: the case of DPEcho.”. *arXiv Preprint*, 2025, [arXiv:2508.14117](https://arxiv.org/abs/2508.14117)

4.3 Connection to Other CoEs

The dealii-X project has actively engaged with CASTIEL2 to foster collaboration and knowledge exchange among European HPC Centres of Excellence. In June 2025, deal.II was presented as the “Code of the Month” by CASTIEL2, highlighting its impact and relevance within the European HPC ecosystem. As part of CASTIEL2 WP2, dealii-X contributed code descriptions, use cases, and technical information, as documented in Deliverable D3.1 (M6).

The consortium has actively participated in PMT-CoE leader meetings, joint workshops, and Kokkos programming sessions to align strategies and share best practices across CoEs. Collaboration on continuous integration and deployment (CI/CD) has been initiated through the European Environment for Scientific Software Installations (EESSI), enabling multi-system deployment of dealii-X’s three core codes: deal.II, PSCToolkit, and MUMPS.

To strengthen knowledge transfer and community engagement, dealii-X will participate in the NCC-CoE all-hands meeting in Tallinn, Estonia (September 2025) and has planned an online seminar series to showcase research outcomes and invite guest speakers from other CoEs. The first dealii-X Research School is scheduled for December 2025 in Trieste, focusing on hands-on HPC training, co-design methodologies, and accelerator programming. These activities ensure that dealii-X remains closely connected to the European HPC ecosystem and actively contributes to its development.

4.4 Next Steps

- Expand benchmarking to additional prototype hardware and emerging platforms.
- Continue supporting accelerator programming paradigms and ensure full compatibility across WP1 applications.

- Implement performance portability improvements based on collected benchmarks and profiling data.
- Complete systematic energy and performance profiling to guide optimized configurations for simulations.

5 Dissemination, Communication, and Exploitation (WP4)

5.1 Objectives

The objective of WP4 is to ensure that the results of dealii–X are communicated, disseminated, and exploited effectively to achieve maximum impact. Specifically, WP4 aims to:

- Build awareness of the project among industry, academia, clinicians, regulators, policymakers, patients, and the general public.
- Promote project outcomes to the HPC, scientific, and healthcare communities.
- Track and facilitate exploitation of results by partners and third parties.
- Align with relevant European initiatives such as EuroHPC JU Pilots, Virtual Human Twins, and NextGenerationEU Consortium iNEST.

5.2 Progress Achieved in Year 1

WP4.1: Develop a data management plan

The Data Management Plan (DMP) was prepared under the leadership of RUB in close collaboration with all partners and submitted as Deliverable D4.3 in M6. The DMP defines the project-wide framework for managing data, including data formats, storage, access policies, and preservation procedures, and it provides concrete strategies to ensure compliance with FAIR principles (Findable, Accessible, Interoperable, Reusable).

The process of drafting the DMP included extensive partner consultations to capture data needs across work packages. This allowed the consortium to identify suitable formats, common repositories, and licensing approaches that facilitate both intra-consortium sharing and secondary use by the broader community.

In addition, the consortium has benefited from the expertise of WIAS, which is leading the German National Research Data Infrastructure (NFDI) consortium for mathematics (MaRDI). This connection has provided valuable input and best prac-

tices on standards, metadata, and interoperability relevant to computational science.

WP4.2: Communication and dissemination strategy, branding, and tools

- A comprehensive Communication and Dissemination Strategy was developed and approved in M4.
- Stakeholder mapping was carried out, identifying six primary target groups (clinicians, ICT researchers, industry, regulators, patients, policymakers/public).
- A full visual identity was created, including project logo, templates, infographics, and EU acknowledgement banner.
- Communication channels established:
 - Project website (live since M6; regularly updated).
 - Social media (LinkedIn; active since M5 with 219 followers). More detailed stats will be provided in the first dissemination and communication report.
- First communication materials produced: project flyer, presentation template, social media kit and a project poster.

WP4.3: Dissemination activities

- Project Partners started disseminating the project at conferences/workshops they were attending. The events were also disseminated via project social media channels.
- Dissemination was supported via VPH and partner institutional channels (websites, newsletters, social media).

WP4.4: Stakeholder outreach and alignment with relevant EU initiatives and projects

- Initial contacts established with key stakeholder groups, including clinicians, patients, industry, regulatory bodies, HTA organisations, payers, citizens, and policymakers.
- Project objectives presented through direct exchanges and dissemination via VPHI and partner channels.
- First links established with complementary EU initiatives to ensure alignment, coherence, and maximise impact.

- Preparation of structured engagement activities (focus groups, surveys, interviews) to be launched in the next reporting period.

WP4.5: Exploitation

The first step toward systematic exploitation was the preparation and submission of Deliverable D4.4 (Table of Exploitable Results) in M6, coordinated by RUB with contributions from all partners. This deliverable established an initial inventory of project outputs with potential for exploitation, including software, workflows, training resources, and integration pathways into partner tools.

The exploitation framework is designed to ensure that results are captured early and monitored throughout the project lifetime. Key measures include:

- Regularly updating the exploitable results table as new outputs emerge.
- Supporting open–source releases under appropriate licenses, to maximise uptake and community impact.
- Facilitating integration of developed tools into existing partner platforms and software products.
- Engaging with external stakeholders (academia, HPC centres, industry, and clinical users) to explore adoption and sustainability pathways.

Moving forward, WP4.5 will develop a detailed exploitation roadmap to secure the long-term sustainability of project outputs, including software maintenance, training material reuse, and possible transfer of knowledge and tools to industrial or clinical partners.

5.3 Key Performance Indicators

- Website launched (M6), achieving ~2600 visits and ~800 unique visitors in the first 6 months.
- Social media growth: ~200 followers on LinkedIn; reach of ~25,000 impressions to date, with steady monthly growth.
- Data management: Data Management Plan submitted (M6); 100% of partners consulted in its preparation. Initial data formats and repositories aligned with FAIR principles.

- Exploitation: Initial Table of Exploitable Results submitted (M6), with 20 identified results covering software, workflows, and training resources.
- The 2025 SIAM/ACM Prize in Computational Science and Engineering was awarded to the principal authors (among whom are dealii-X PCs M. Kronbichler and L. Heltai) of the deal.II Project in recognition of their contributions to finite element calculations.
- deal.II was presented by the dealii-X consortium as the “Code of the Month” by CASTIEL2, highlighting its impact and relevance within the European HPC ecosystem.
- Dissemination: Strong presence at international conferences and workshops, including SIAM CSE 2025, Coupled Problems 2025, Large-Scale Scientific Computations (LSSC 2025), Domain Decomposition Methods (DD29), and ICOSA-HOM 2025, among others.

5.4 Next Steps

- Strengthen communication campaigns and widen dissemination to additional HPC and clinical communities, with targeted communication material and enhanced social media presence.
- Organise the first project workshop (dealii-X Research School in Trieste in December 2025), providing hands-on training and fostering early user engagement.
- Launch structured stakeholder engagement activities (focus groups, surveys, interviews) targeting regulators, clinicians, and patients to collect feedback on project outputs.
- Update the Data Management Plan to reflect new data types generated in WPs 2–3, and refine strategies for long-term preservation and reuse.
- Develop a detailed exploitation roadmap, including sustainability measures for software maintenance, data curation, and training materials reuse.
- Deepen collaborations with EuroHPC JU and Virtual Human Twins initiatives to maximise impact and synergies with related European efforts.

6 Project Management (WP5)

6.1 Objectives

WP5 is devoted to project management and administrative tasks. The main objective is to facilitate technical work, encourage synergy among partners, ensure consistency across tasks, and guarantee that expected results are achieved on time, of high quality, and compliant with the Ethics Requirements. Specifically, WP5 aims to:

- Target- and result-oriented management of the project, including risk, conflict, and IP management.
- Effective communication within the consortium and with the European Commission, ensuring deliverables and periodic reports are submitted on time.
- Quality control of results and deliverables according to the defined procedures, ensuring internal review and approval prior to submission.

6.2 Progress Achieved in Year 1

WP5.1: Management

- Monthly consortium meetings held to provide updates, discuss progress, and address technical or administrative issues.
- Deliverable D5.1 “Project Management Handbook” submitted in M3 by RUB, based on PM² methodology, establishing procedures for decision-making, risk management, and deliverable approval.
- Decision-making procedures and escalation mechanisms established to monitor risks and issues effectively.
- Internal documentation repository set up to ensure transparency and easy access for all partners.

WP5.2: Communication

- PMT acts as the interface between the consortium and the EC, preparing periodic and midterm reports (M12, M22) for submission.

- Internal communication channels established, including mailing lists, collaborative platforms, and a secure workspace for sharing project information, deliverables, and meeting minutes.
- Coordination with WP4 to align communication and dissemination activities.

WP5.3: Quality Control

- Project results and formal deliverables undergo internal quality control prior to submission.
- Reviewers assigned for all deliverables to verify content and compliance with project standards.
- Deliverable D1.1 "External Libraries Integration" postponed from M6 to M10; revised timeline described, motivated, approved by the EC, and successfully submitted.
- Templates and procedures for deliverable review made available to the consortium via the internal workspace.

Critical Risks and Mitigation

- Key risks identified included potential delays in deliverable submissions, coordination challenges among distributed partners, and technical dependencies between WPs.
- Mitigation measures included monthly monitoring, proactive communication, and formal approval of revised deliverable timelines by the EC.
- Residual risks will continue to be tracked, with contingency measures updated as needed.

6.3 Next Steps

- Continue monthly consortium meetings and strengthen risk monitoring and mitigation practices.
- Maintain effective internal and external communication, including coordination with WP4 on dissemination and timely submission of reports to the EC.

- Periodically review and refine quality control procedures for deliverables and internal documentation.
- Ensure timely updates of the risk register and implement any necessary corrective actions.
- Support upcoming workshops and consortium-wide events, facilitating coordination and logistics.
- Monitor deliverable timelines to anticipate potential delays and communicate with the EC proactively.

6.4 Summary

WP5 has established a robust management framework that ensures the smooth operation of the dealii–X project. Through regular consortium meetings, structured decision-making, and effective internal and external communication, the consortium maintains alignment across work packages and partners. Quality control procedures, including internal reviews of deliverables and monitoring of critical risks, support the timely and high-quality production of project outputs. Together, these measures foster synergy among partners, mitigate potential delays, and contribute to achieving the project's objectives efficiently and in compliance with all ethical and contractual requirements.