D1.2
Report on deal.II improvements

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	Report on deal.II improvements
Deliverable number	D1.2
Deliverable version	v1
Date of delivery	March 31, 2025
Actual date of delivery	March 28, 2025
Nature of deliverable	Report
Dissemination level	Public
Work Package	WP1
Partner responsible	UNIPI

Abstract	This report details the progress made during the first semester (Months 1–6) of the dealii-X project in enhancing the deal.II finite element library. The focus is on improvements for exascale readiness, specifically in high-performance matrix assembly, matrix-free GPU integration, as well as the initial performance evaluation of the polygonal discretization module (to be integrated in the library). Furthermore, this report outlines other relevant enhancements to deal.II undertaken within the project's work packages during this period.
Keywords	deal.II library; finite element method; Krylov subspace methods; sparse direct solver; matrix-free algorithms; mesh generation; CUDA; Kokkos

Document Control Information

Version	Date	Author	Changes Made
0.1	March 17, 2025	Luca Heltai	Initial draft
0.2	March 24, 2025	Luca Heltai	Gathered information from all partners
1.0	March 27, 2025	Luca Heltai	Final adjustments between partners

Approval Details

Approved by: M. Kronbichler
Approval Date: March 28, 2025

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

1.1 Overview of the dealii-X Project and Objectives

The dealii-X project is a pioneering initiative dedicated to developing a high-performance and scalable computational platform based on the deal.II finite element library. The project directly addresses the HORIZON-EUROHPC-JU-2023-COE-03-01 topic, specifically focusing on the "Personalised Medicine / Digital twin of the human body" as an Exascale Lighthouse application area. The overarching goal of dealii-X is to advance existing pre-exascale digital twin applications for human organs, such as some deal.II based applications dedicated to the simulation of the brain, heart, lungs, liver, and cellular interactions, to achieve exascale readiness.

The project aims to enable real-time simulations of intricate biological processes, thereby contributing to personalized medicine and cutting-edge healthcare research. Ultimately, this enhanced simulation capability holds the potential to significantly improve medical diagnostics and treatment planning.

1.2 Objectives of Work Package 1 (WP1)

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 UNIPI)**: 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.

1.3 Purpose and Scope of this Report (Deliverable D1.2)

The purpose of this initial report on Work Package 1 is to provide a comprehensive overview of the foundational activities and progress achieved during the early stages of the dealii-X project. Specifically, it aims to document the strategies and initial advancements made in enhancing the deal.II library to meet the demands of exascale computing and to support the development of sophisticated digital twins of human organs. This report serves to establish a baseline understanding of the technical developments within WP1, including the optimization of computational methods, the integration of key external libraries, and the exploration of novel discretization techniques. The scope of this report encompasses the activities undertaken within each of the sub-work packages of WP1.

The report will highlight the initial outcomes, any challenges encountered, and the solutions identified, thus providing a clear picture of the progress towards achieving the objectives of WP1 and its contribution to the overall goals of the dealii-X project. It will serve as an initial benchmark for assessing the development of the exascale building blocks and support tools for the subsequent work packages.

A quantitative assessment of the overall advancement of WP1 will be provided in the form of a list of pull requests (PRs) and issues opened in the deal.II GitHub repository, along with a summary of the improvements made in satellite repositories.

2 Improvements for Exascale Readiness

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.

2.1 Benchmarks for GPU systems

In order to assess the core matrix-free algorithms on GPU systems, we have worked towards a set of benchmarks.

As a benchmark, we consider a matrix-free operator using the CEED benchmark BP5[1], which uses a 3D Poisson equation on a mesh of deformed elements and solving the linear system with a conjugate gradient iterative solver and diagonal preconditioner. We list the throughput in terms of degrees of freedom (DoFs) per second and CG iteration. Figure 1 lists the measured performance in February 2025. The CPU code uses advanced data locality optimizations established in our previous research[2], and therefore the CPU codes reach a relatively high throughput despite a

[1] https://ceed.exascaleproject.org/bps/, see the benchmark description at https://doi.org/10.1177/1094342020915762

[2] https://doi.org/10.1177/10943420221107880

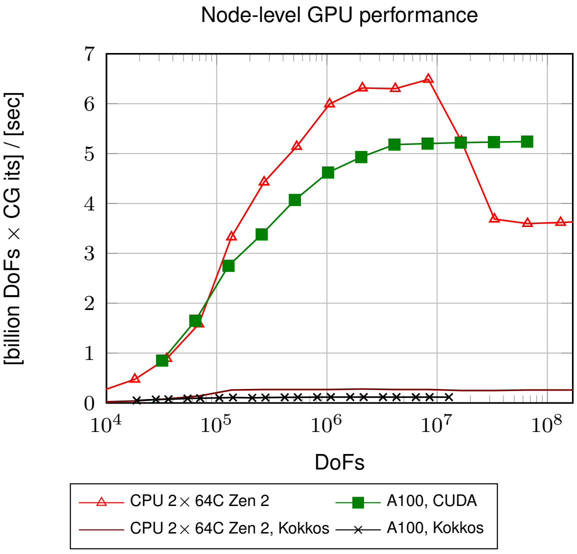

Figure 1: Throughput of CEED BP5 benchmark case with deal.II using Kokkos and native CUDA/CPU implementations.

formally much slower memory subsystem compared to the A100, where the CUDA code reaches 1.1 TB/s of memory bandwidth. The Kokkos code currently integrated in deal.II was found to be really slow in this experiment, reaching 0.12 GDoF/s with the code from February 2025 vs. 5.2 GDoF/s with CUDA (code established in 2016–2022). Also for the CPU, the Kokkos abstractions give poor performance with 0.26 GDoF/s vs. 3.6 GDoF/s.

In order to better understand the performance gap and mitigate performance differences, we have worked on more low-level sum Factorization benchmarks to assess the performance of the Kokkos programming model in comparison. The algorithm successively applies matrix-vector products on finite elements in 3D with polynomial degree 5, with a focus on optimizing the use of shared memory and ensuring coalesced memory access patterns. The objective of the benchmark is to compare the throughput, measured in degrees of freedom per second, for each programming model, highlighting performance differences between the vendor-neutral Kokkos library and the native programming model CUDA. Fig. 2 shows the results of this experiment.

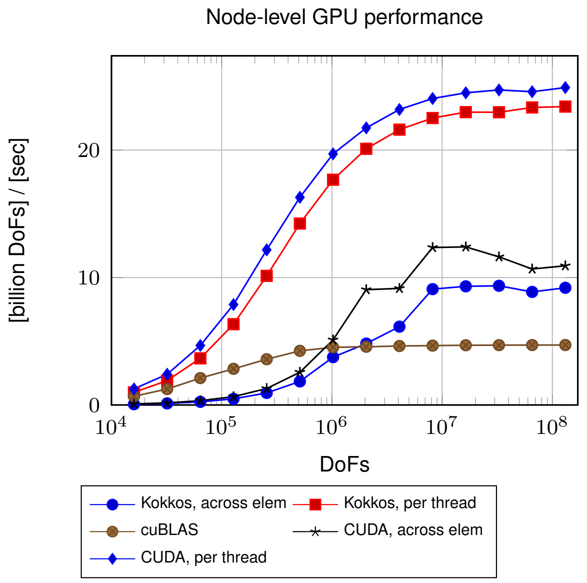

Figure 2: Throughput evaluation of sum factorization kernels for forward evaluation (basis functions → values at quadrature points) with CUDA and Kokkos on an A100 GPU.

2.2 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.

- Improvements of Raviart-Thomas incompressible flow solvers in ExaDG: https://github.com/kronbichler/exadg/tree/martin

2.3 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.

- Grid-to-grid projection:
  https://github.com/exadg/exadg/pull/735
- Restarting including grid-to-grid projection and rollout to all time integrators (implicit and explicit, various single step and multi step schemes) and physics (convection-diffusion, Navier-Stokes equations for compressible and incompressible flow, acoustic conservation equations) in ExaDG:
  https://github.com/exadg/exadg/pull/729
  https://github.com/exadg/exadg/pull/731
  https://github.com/exadg/exadg/pull/732
  https://github.com/exadg/exadg/pull/734

3 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.

3.1 Extension of current gmsh API for exascale simulations

The key researcher in this task is a postdoc researcher, Dr. Devi Raksha, who has been hired on the project, and will start her position in April 2025. The task is to extend the gmsh API to support large-scale simulations, with a focus on biomedical applications, and to improve the gmsh deal.II interface to enable the generation and handling of large-scale meshes.

The task has not been started yet, and it is postponed to the second semester of the project.

3.2 Generalized Interface for Coupling Operators in deal.II

3.2.1 Objectives and Planned Activities (Task WP1.2.2)

The objective of this task is to develop a generalized interface for coupling operators in deal.II, which is essential for multiphysics and multiscale problems. The focus is on two different types of coupling:

- Coupling between different physical processes, such as fluid-structure interaction or heat transfer in porous media;
- Coupling between different discretization schemes, including non-matching and non-overlapping discretizations.

This subtask was supposed to start on the second semester of the project, but an early version of the generalized interface has been developed and merged into the deal.II repository in April 2024, before the official start of the dealii-X project, making it more convenient to start the development of the rest of this task in the first semester.

The developed approach encompasses two main aspects: a high-level interface for defining coupling operators and a low-level interface for implementing efficient evaluation of coupled bilinear forms (on possibly non-matching or non-nested grids). The high-level interface enables the computation of abstract bilinear forms for coupling between geometric entities, supporting various kernels and use cases like BEM and bulk-surface coupling. The low level interface needs to be flexible enough to allow the use of different discretization schemes, including non-conforming discretizations, and possibly different parallel distributions of the meshes.

3.2.2 Implementation Details

The two pull requests, “FECouplingValues” (`https://github.com/dealii/dealii/pull/15773`) and “FEValuesViews::RenumberedView” (`https://github.com/dealii/dealii/pull/15819`), directly address the objective of developing a “Generalized interface for coupling operators in deal.II”, as outlined in Work Package 1.2.2 of the dealii-X project, by providing the high level interface discussed in the introduction.

Pull request #15773 introduces a new class called ‘FECouplingValues‘, specifically designed for the computation of general coupling operators. The primary goal of this class is to enable the computation of abstract bilinear forms of the type:

$$\int_{T_1}\int_{T_2} K(x_1,x_2) f(x_1) g(x_2)\, dT_1\, dT_2,$$

where T_1 and T_2 represent two arbitrary sets (cells, faces, edges, or their combinations), and K is a coupling kernel (potentially singular). The class is designed to support various types of kernels K. The pull request includes several tests demonstrating how the class can be used for different types of coupling:

- Coupling for Boundary Element Methods (BEM);
- As a replacement for ‘FEInterfaceValues‘, integrating discontinuous Galerkin flux terms;
- For bulk-surface coupling computations.

Pull request #15819 introduces the class ‘FEValuesViews::RenumberedView‘. This component was developed as part of the work for #15773 and is intended for use by ‘FECouplingValues‘. ‘RenumberedView‘ allows the creation of a reordered view of an ‘FEValuesBase‘ object, where both degrees of freedom and quadrature points are renumbered according to provided renumbering vectors.

In summary, these two pull requests jointly implement the ”Generalized interface for coupling operators in deal.II” envisioned in WP1.2.2:

- ‘FECouplingValues‘ provides the framework for defining and computing general coupling operators between different geometric entities and discrete spaces. This addresses the need for a versatile interface to manage various coupling techniques, including methods for non-matching grids and non-local coupling.
- ‘FEValuesViews::RenumberedView‘ provides a fundamental tool for managing the correspondence (or lack thereof) between the degrees of freedom and quadrature points of the different spaces being coupled via ‘FECouplingValues‘.

The integration of these two functionalities into the deal.II library provides users with powerful and flexible tools to tackle complex coupling problems essential for the multiphysics and multiscale simulations at the core of the dealii-X project.

The two pull requests were merged into the main deal.II repository on April 11, 2024, and included in the Release 9.6 milestone, before the official start of the dealii-X project. As part of the WP1.2 tasks, a biologically relevant application that uses the new functionality is being developed and will be integrated in the deal.II repository as a new tutorial program. The tutorial will demonstrate the use of the new ‘FECouplingValues‘ class for the computation of a bulk-surface coupling operator.

A low level interface for coupling operators has been developed and merged into the deal.II repository in pull request #17919, which introduces a new abstract base class ‘MGTwoLevelTransferBase‘ to enable non-nested multigrid transfers. The primary goal of this class is to generalize the interface for managing transfers between non-nested grids, which is a crucial step towards addressing the challenges of coupling operators on non-matching grids.

The ‘MGTwoLevelTransferBase‘ class provides a flexible framework for implementing matrix-free transfer operators between different levels of a multigrid hierarchy. This development aligns with the objectives of WP1.2.2, as it contributes to the creation of a generalized interface for coupling operators in deal.II. Additionally, the techniques developed for non-nested multigrid transfers could offer valuable insights and algorithms for tackling coupling problems on non-matching grids in more general contexts.

The implementation of this low-level interface is expected to enhance the scalability and efficiency of multigrid methods in deal.II, particularly for applications involving complex grid hierarchies or non-conforming discretizations. This work also complements the high-level interface provided by ‘FECouplingValues‘, creating a comprehensive toolkit for addressing a wide range of coupling and multigrid challenges in the dealii-X project. The work has been submitted, and is currently under review. A preprint is available at [2].

A tutorial program that illustrates the use of the non-nested multigrid transfer has been proposed and is currently under review in the deal.II library `https://github.com/dealii/dealii/pull/17919`.

4 Polygonal Discretization Methods

Work Package 1.3 of the dealii-X project is dedicated to the introduction and development of an experimental module for polygonal discretization within the deal.II library.

4.1 Objectives and Planned Activities (Task WP1.3.1)

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.

4.2 Progress and Current Status

A Preliminary version of the polygonal discretization module has been developed as part of this work package, and it is currently available in the repository `https://github.com/fdrmrc/Polydeal`. The module is designed to be integrated into the deal.II library, and its integration is expected to be completed by the end of the project.

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.

4.3 Challenges and Future Plans

The repository `https://github.com/fdrmrc/Polydeal` is being enriched with a range of examples for comprehensive benchmarking of both polytopic discretization methods and multigrid solvers, before integration within the deal.II library. This include the implementation of polytopic discontinuous Galerkin methods for fluid flow problems and a multigrid solver for accelerating the solution of complex electrophysiological models.

5 Integration of PSCToolkit into deal.II (WP1.4)

5.1 Objectives and Planned Activities

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.

5.2 Progress and Current Status

Initial efforts have focused on designing the interface between PSCToolkit and deal.II's existing data structures. A pull request has been opened in the deal.II repository to support detection and linkage with the PSBLAS library (one of the two core libraries identified for interfacing PSCToolkit with deal.II; the other is AMG4PSBLAS, whose integration will follow in a subsequent pull request) `https://github.com/dealii/dealii/pull/18199`. A prototype of the interface has been planned in collaboration with the MUMPS team to ensure seamless integration with deal.II's current data structures of both interfaces.

Several enhancements have been made to the PSCToolkit library itself to facilitate its integration with external libraries (such as deal.II), including:

- Fine control and tuning of compile definitions to simplify the detection of the library and its components, including support for external libraries, ensuring compatibility at linking time with deal.II and its dependencies;
- Adding support for CMake to streamline integration with external libraries;
- Exporting the library's macro definitions in a dedicated header file to simplify detection of the library and its components, regardless of the build system used;
- Adding PSBLAS to the Spack package manager to simplify the installation of the library and its dependencies.

5.3 Challenges and Future Plans

The integration of PSCToolkit into deal.II is still in its early stages, requiring careful planning to ensure compatibility and performance. The primary challenge lies in ensuring that the interface between PSCToolkit and deal.II is robust, efficient, and seamlessly communicates with existing deal.II data structures without introducing additional ones. This challenge stems from the proliferation of slightly incompatible data structures in existing external linear algebra packages integrated with deal.II, which has led to significant maintenance efforts and user confusion. Addressing this issue is a key focus to ensure a streamlined and user-friendly integration.

6 Integration of MUMPS Solver into deal.II (WP1.5)

6.1 Objectives and Planned Activities

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.

6.2 Progress and Current Status

A preliminary interface for MUMPS was present in the deal.II library until version 9.2. Such interface had been removed, in favor of using MUMPS through external linear algebra packages such as PETSc and Trilinos.

A careful planning of the interface between deal.II and MUMPS is currently ongoing, with the goal of ensuring that the interface is robust and efficient, and that it seamlessly communicates with existing deal.II data structures, extending, integrating, and modernizing the former MUMPS interface in deal.II. An initial pull request has been opened in the deal.II repository to support detection and linkage with the MUMPS library, reverting the removal of the MUMPS interface in deal.II, and re-introducing the original MUMPS interface, including some simple tests to ensure that the interface is working correctly. The pull request is available at `https://github.com/dealii/dealii/pull/18255`.

6.3 Challenges and Future Plans

The integration of MUMPS into deal.II is currently working only with the serial version of the deal.II data structures, and it is still in its early stages, requiring careful planning to ensure compatibility and performance. Similarly to what has been discussed with the PSBLAS library, the primary challenge lies in ensuring that the interface between MUMPS and deal.II is robust, efficient, and seamlessly communicates with existing deal.II data structures. Ensuring compatibility and avoiding redundant data structures is crucial to streamline integration and reduce maintenance efforts.

References

[1] M. Feder, A. Cangiani, and L. Heltai. R3MG: R-tree based agglomeration of polytopal grids with applications to multilevel methods. *Journal of Computational Physics*, 526:113773, Apr. 2025.

[2] M. Feder, L. Heltai, M. Kronbichler, and P. Munch. Matrix-free implementation of the non-nested multigrid method. Technical report, University of Pisa, 2024.