Scaling Up Simulations

The Open-Source Advantage

Open-source codes: the easiest way to scale up?

Whether in the design phase or in the finalization phase of a project, mechanical engineering design departments must launch numerical simulation campaigns that are often weeks up to months long. Computational costs arise from two sources, most of the time combined:

- the desire to vary a large number of parameters in order to find an optimal system configuration (e.g. motor blade geometry);
- the need to process large-scale numerical models, to capture the various physical phenomena realistically (e.g. stress concentration in a part).

Design offices therefore need to find ways of speeding up the time set aside for these studies, which are often associated with customer milestones. This is where the natural approach comes in: launch several calculations at the same time and accelerate the resolution time of each calculation. The proprietary codes used to process these simulations offer with token extensions or special licenses to accelerate calculations via parallelism options. Running simulations using fully parallel studies is therefore a major financial investment, all the more so as the licenses concerned are on a different price scale to sequential ones.

In this context, the use of open-source simulation codes appears to be an attractive solution. Since no licenses are required, design offices can run a multitude of calculations in parallel. Once this initial consideration has been made, however, questions arise for companies wishing to migrate from proprietary codes:

- Does the chosen open-source code offer the same versatility of models and functionalities as proprietary code? Engineers wants to rely on a solution that is able to handle various advanced use cases.
- Has open-source code been validated as thoroughly as proprietary code? It's mandatory in an industrial process to use a solution that is proven robust.
- Is the performance of open-source code comparable to that of proprietary code? The benefits of not requiring a license must be combined with a number of hours of calculation within the time allowed by project requirements.

While switching to open-source tools offers undeniable advantages, all these legitimate questions can be a risk factor that puts the brakes on migrations from proprietary code. So, what are the best ways to manage these risks and benefit from all the advantages of open-source?

EDF simulation software, open-source AND industrially recognized calculation codes

Ensuring the safety and availability of mechanical installations and civil engineering structures at EDF requires thorough justification for operational, repair, and replacement decisions. This implies that all codes used for simulation should be certified by an independent French technical authority associated with the nuclear domain, ensuring their reliability and compliance with stringent safety standards.

These justifications heavily depend on non-linear mechanical modeling. To achieve this, code_aster, the finite element software developed by EDF, stands out by integrating various scientific research efforts focused on addressing these safety challenges. It effectively manages this research and seamlessly incorporates it into engineering practices.

The ambition for code_aster is twofold:

- To deliver an up-to-date, powerful simulation software that is stable and robust for expert-level studies, all within a rigorous quality assurance framework.

- To facilitate the integration and consolidation of numerical mechanical models from EDF's R&D efforts.

Developing its own code aligns with these objectives, ensuring the capitalization of R&D investments and enabling swift knowledge transfer to the engineering team. This process can be complex with commercial codes. As EDF focuses on operations rather than manufacturing, its R&D efforts are unique. EDF must validate the lifespan of its equipment and infrastructure from both economic and regulatory standpoints. The primary goal of code_aster is to continually enhance the code's quality and usability by expanding its applications, positioning EDF as a key contributor in simulation partnerships.

Code_aster includes distinct numerical models for:

- Simulating the aging of materials and structures, such as creep, fatigue, damage, and fracture mechanics.
- Addressing specific challenges related to nuclear operations, including soil-structure interaction, seismic analysis, regulatory computations, and fuel assembly modeling.

Used daily by EDF's engineering teams, code_aster is a continuously certified software. It includes more than 3,500 tests, from unit testing to replication of data from large-scale experiments. Each modification in the code is therefore tested from an IT point of view and compared with the physics to be represented.

How to benefit from the best code performance

The EDF solvers therefore answer the first two questions posed by open-source codes: versatility and robustness. This makes them major candidates for companies wishing to migrate from proprietary codes. While they are used daily on EDF's HPC clusters, the question of how to access the best numerical performance for third-party companies was still open until recently. With this in mind, EDF launched Simvia, its new subsidiary, at the end of 2024. Simvia's main purpose is to enhance the value of EDF's open-source codes by accelerating their availability, increasing users' skills and optimizing their use in various industrial sectors.

The question of the performance of a simulation code is obviously linked to the infrastructures on which it runs. Medium-sized simulation teams generally own computing servers of sufficient size for standard studies (32 to 64 cores). However, for parametric studies or calculations involving more degrees of freedom, these resources may not be sufficient. In this context, as in more intensive remote applications, the remote computing solutions offered by Rescale are a major time-saving opportunity.

From the start of 2025, Rescale and Simvia naturally joined forces to enable platform users to benefit from the powerful tools developed by EDF. The main idea behind this joint effort is to enable greater fluidity between work done on local machines and work on remote servers. Simvia has therefore crafted various images of open-source code enabling jobs to be run in parallel on the Rescale platform.

Illustrating the scalability of EDF codes: the case of code_aster finite element software

High-performance computing (HPC) is a critical area of ongoing development for code_aster. Significant advancements have been made in both meshing and solver technologies. By partitioning meshes and implementing "ghost" elements at interfaces, the HPC mode enables the computation of large-scale models with high computational demands, reaching hundreds of millions of degrees of freedom (DoF), while minimizing memory usage.

Key HPC functionalities include:

- Parallel mesh partitioning during mesh reading, if activated.
- Parallel I/O capabilities for storing data in a single file.
- Fully HPC-compatible mechanical and thermal analysis.
- A preconditioner specifically designed for managing large linear systems.
- Use of intensively parallel linear system solvers, to further optimize performance.

It is worth noting that in all these aspects, code_aster is integrated with other open-source projects. Therefore, performance enhancements are mutualized at every step of the analysis.

Since earlier versions, the code_aster solver has been able to run parallel solutions of systems of equations using the OpenMP protocol. This option is already available to all users on the annual release offered by EDF[1]. In addition, images of the HPC version mentioned above are distributed by Simvia for various industrial uses. These two versions are compared here on a real-world use case.

As already mentioned, the physics covered by code_aster is quite broad. We have chosen here to illustrate the scalability of the code in the context of civil engineering. The study consists in determining the behavior of a tunnel, located under different soil layers, during the passage of a train. To do this, we model the tunnel components (made of concrete) and each of the soil layers volumetrically. The vibrations generated by the train are represented here by a point source excited at different frequencies.

To correctly represent wave propagation in the various components, the mesh produced must be sufficiently fine. Thus, the mesh on which we are establishing this rapid benchmark is composed of 1,102,235 tetrahedral elements, representing a total problem with 616,266 degrees of freedom. Figure 1 shows the volume mesh produced for the study, shown in an exploded view. Here, the colored areas represent the breakdown into subdomains as proposed by the partitioner integrated into code_aster. In an MPI-based computation, each subdomain is primarily handled by a dedicated process, with inter-process communication occurring as needed (particularly at subdomain interfaces). This parallelization strategy provides significant gains over the OpenMP version.

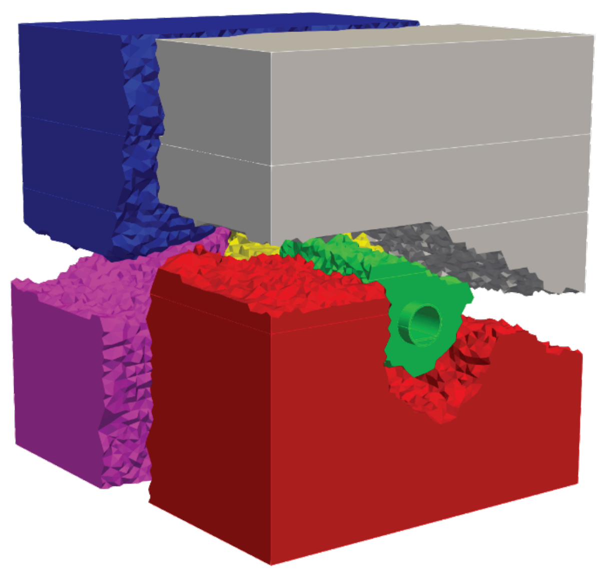

Figure 1: Exploded view of the partitioned volume mesh used for MPI-based computation.

To illustrate the gains observed when using code_aster with the different options, we ran a dozen simulations of the study on Rescale's *Emerald Max* machine[2]. The reference result is obtained for an OpenMP protocol on an instance of 4 CPUs. The

[1] Find out more about how to benefit on Simvia's YouTube channel: [Tips & Tricks] Install Salome_Meca on Windows using WSL

[2] Intel(R) Xeon(R) Platinum 8124M CPU @ 3.00GHz

number of CPUs is then varied between 8 and 36, with a breakdown between MPI processes and the number of associated threads. Figure 2 shows the computation times as a function of the number of CPUs for the different configurations.

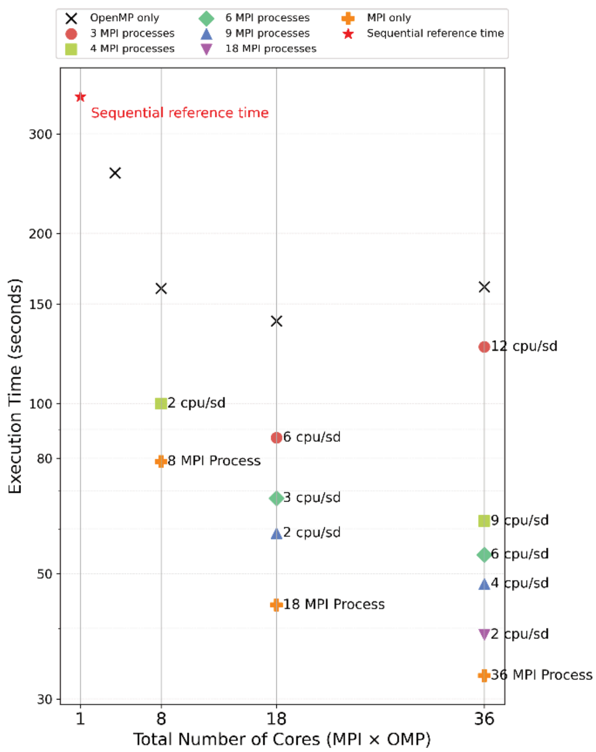

Figure 2 : Execution time for a given number of cores, exhibiting potential computational acceleration with code_aster.

The main findings are that, based solely on the OpenMP version, increasing the number of CPUs is no longer beneficial after 8. On the other hand, the greater the number of MPI processes with fewer cores, the greater the gains observed. In this case, the calculation time between the sequential reference solution and the best configuration is divided by more than 10.5 (from 350 seconds to 33 seconds).

Ready to Accelerate Your Simulations and Reduce Costs?

Transitioning smoothly to open-source FEM software like EDF's code_aster requires careful planning and robust execution tailored to your unique engineering workflows. Simvia simplifies this migration with expert support and semi-automated tools, ensuring that your shift from proprietary solutions is efficient, risk-controlled, and optimized for maximum performance. Combined with Rescale's flexible, cloud-based HPC platform, your simulations benefit from on-demand scalability, advanced hardware selection, seamless software integration, and enhanced collaboration capabilities.

Rescale streamlines R&D by unifying disparate workflows, integrating advanced simulation tools, and leveraging data for AI-driven methods. Engineering teams can accelerate end-to-end processes, reduce manual and duplicate work, and gain insights from historical data to discover and validate new products. Rescale accelerates product development for engineers while improving IT operational efficiency.

In leveraging the powerful synergy between Rescale and Simvia, you can significantly cut licensing costs and rapidly execute large-scale simulations, enabling your engineering teams to focus resources on innovation and finding the optimal solutions.

Take the first step today—explore how Simvia and Rescale can transform your simulation process into a strategic advantage.