The Energy-Oriented Centre of Excellence

Invited Paper

Elisabetta Boella
E4 Computer Engineering SpA
Scandiano, Italy
elisabetta.boella@e4company.com

Richard Angersbach
Friedrich-Alexander-Universität
Erlangen, Germany
richard.angersbach@fau.de

David Anton
Barcelona Supercomputing Center (BSC)
Barcelona, Spain
david.anton@bsc.es

Emmanuel Agullo
INRIA
Bordeaux, France
emmanuel.agullo@inria.fr

Edouard Audit
CEA
Gif-sur-Yvette, France
edouard.audit@cea.fr

Matias Avila
Barcelona Supercomputing Center (BSC)
Barcelona, Spain
matias.avila@bsc.es

Massimo Bernaschi
CNR-IAC
Rome, Italy
massimo.bernaschi@cnr.it

Francesco Buonocore
ENEA
Rome, Italy
francesco.buonocore@enea.it

Alfredo Buttari
IRIT-CNRS
Toulouse, France
alfredo.butari@irit.fr

Beatrice Calosso
ENEA
Rome, Italy
beatrice.calosso@enea.it

Mauro G. Carrozzo
CNR-IAC
Rome, Italy
maurogiovanni.carrozzo@cnr.it

Marco Catillo
ENEA
Rome, Italy
marco.catillo@enea.it

Alessandro Celestini
CNR-IAC
Rome, Italy
alessandro.celestini@cnr.it

Massimo Celino
ENEA
Rome, Italy
massimo.celino@enea.it

Christoph Conrads
Forschungszentrum Jülich
Jülich, Germany
c.conrads@fz-juelich.de

Pasqua D'Ambra
CNR-IAC
Naples, Italy
pasqua.dambra@cnr.it

Edoardo Di Napoli
Forschungszentrum Jülich GmbH
Jülich, Germany
e.di.napoli@fz-juelich.de

Fabio Durastante*
University of Pisa
Pisa, Italy
fabio.durastante@unipi.it

Muhammad Fahad
Forschungszentrum Jülich GmbH
Jülich, Germany
m.fahad@fz-juelich.de

Salvatore Filippone†
University of Rome Tor Vergata
Rome, Italy
salvatore.filippone@uniroma2.it

Luc Giraud
INRIA
Bordeaux, France
luc.giraud@inria.fr

Simone Giusepponi
ENEA
Rome, Italy
simone.giusepponi@enea.it

Virignie Grandgirard
CEA, IRFM
Saint Paul-lez-Durance, France
virginie.grandgirard@cea.fr

Daniele Gregori
E4 Computer Engineering SpA
Scandiano, Italy
daniele.gregori@e4company.com

Harald Koestler
Friedrich-Alexander-Universität
Erlangen, Germany
harald.koestler@fau.de

Stefan Kollet
Forschungszentrum Jülich GmbH
Jülich, Germany
s.kollet@fz-juelich.de

Carola Kruse
Cerfacs
Toulouse, France
carola.kruse@cerfacs.fr

Bruno Majone
University of Trento
Trento, Italy
bruno.majone@unitn.it

Gilles Marait
INRIA
Bordeaux, France
gilles.marait@inria.fr

Rafael Mayo-García
CIEMAT
Madrid, Spain
rafael.mayo@ciemat.es

José A. Moríñigo
CIEMAT
Madrid, Spain
josea.morinigo@ciemat.es

Stojche Nakov
INRIA
Bordeaux, France
stojche.nakov@inria.fr

Kevin Obrejan
CEA, IRFM
Saint Paul-lez-Durance, France
kevin.obrejan@cea.fr

Herbert Owen
Barcelona Supercomputing Center (BSC)
Barcelona, Spain
herbert.owen@bsc.es

Alessandro Pecchia
CNR-ISMN
Rome, Italy
alessandro.pecchia@cnr.it

Matthieu Robeyns
IRIT-CNRS
Toulouse, France
matthieu.robeyns@irit.fr

Roberto Rocco
E4 Computer Engineering SpA
Scandiano, Italy
roberto.rocco@e4company.com

Helen Schottenhamml
IFP Energies nouvelles
Rueil-Malmaison, France
helen.schottenhamml@ifpen.fr

Ani Anciaux Sedrakian
IFP Energies nouvelles
Rueil-Malmaison, France
ani.anciaux-sedrakian@ifpen.fr

Philipp Suffa
Friedrich-Alexander-Universität Erlangen, Germany
philipp.suffa@fau.de

Petr Vacek
IFP Energies nouvelles
Rueil-Malmaison, France
petr.vacek@irit.fr

Abstract

EoCoE is a EuroHPC Centre of Excellence contributing to Europe transition to sustainable energy through High Performance Computing. Now in its third phase, the project harnesses exascale technologies, massively parallel simulations, and advanced computational methods to tackle some of the most pressing challenges in renewable energy. At the heart of EoCoE are five lighthouse applications spanning four strategic domains: fusion, photovoltaics, water, and wind energy. By enhancing these applications, making them exascale-ready, and developing and optimising the underlying software tools and mathematical libraries, the project contributes to both the green energy transition and the advancement of scientific computing methodologies in Europe. This paper presents the five lighthouse applications and highlights the key achievements in performance, portability, and productivity. It further describes the development of enabling tools and libraries, and outlines the project broader efforts to maximise scientific, technological, and societal impact.

CCS Concepts

• **Computing methodologies** → **Massively parallel algorithms**; **Modeling and simulation**.

Keywords

EoCoE, Centre of Excellence, Energy oriented Centre of Excellence, HPC, exascale, wind energy, fusion energy, hydropower, photovoltaics

ACM Reference Format:
Elisabetta Boella, Richard Angersbach, David Anton, Emmanuel Agullo, Edouard Audit, Matias Avila, Massimo Bernaschi, Francesco Buonocore, Alfredo Buttari, Beatrice Calosso, Mauro G. Carrozzo, Marco Catillo, Alessandro Celestini, Massimo Celino, Christoph Conrads, Pasqua D'Ambra, Edoardo Di Napoli, Fabio Durastante, Muhammad Fahad, Salvatore Filippone, Luc Giraud, Simone Giusepponi, Virignie Grandgirard, Daniele Gregori, Harald Koestler, Stefan Kollet, Carola Kruse, Bruno Majone, Gilles Marait, Rafael Mayo-García, José A. Moríñigo, Stojche Nakov, Kevin Obrejan, Herbert Owen, Alessandro Pecchia, Matthieu Robeyns, Roberto Rocco, Helen Schottenhamml, Ani Anciaux Sedrakian, Philipp Suffa, and Petr Vacek. 2026. The Energy-Oriented Centre of Excellence: Invited Paper. In *Proceedings of the 23rd ACM International Conference on Computing Frontiers (CF Companion '26), May 19–21, 2026, Catania, Italy.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3801488.3806243

1 Introduction

The Energy-oriented Centre of Excellence (EoCoE) is a project supported by the EuroHPC Joint Undertaking (JU) whose primary objective is to accelerate Europe transition to a decarbonised and sustainable energy system by fully leveraging the capabilities of High Performance Computing (HPC) technologies, particularly the

[*] Also with CNR-IAC.
[†] Also with CNR-IAC.

CF Companion '26, Catania, Italy

ACM ISBN 979-8-4007-2569-2/26/05
https://doi.org/10.1145/3801488.3806243

exascale systems deployed by EuroHPC. In this context, EoCoE highlights the strategic role of computational science and large-scale numerical simulations as key enablers to achieve Europe net-zero energy ambitions. Building on a strong legacy established since its inception, the current phase, EoCoE III (January 2024 - December 2026), focuses on five exascale lighthouse applications spanning four major energy domains: energy materials, water, wind, and fusion, each addressing critical scientific, engineering, and technological challenges faced by the energy sector.

The overarching objective of EoCoE III is the systematic improvement of the lighthouse codes, together with the development and optimisation of a set of transversal software tools on which they rely, in order to significantly enhance scalability, performance, portability, and energy efficiency on modern HPC platforms. Particular emphasis is placed on preparing the applications for execution on pre-exascale and exascale systems, ensuring their robustness and sustainability in the rapidly evolving European supercomputing landscape. Through these efforts, EoCoE contributes not only to Europe green energy transition, but also to the advancement of state-of-the-art scientific computing methodologies.

This paper is organised as follows. Section 2 presents the five lighthouse applications at the core of EoCoE, describing the underlying models, the numerical algorithms employed, their implementation languages, as well as their performance and scalability. Section 3 discusses the strategies adopted to ensure code portability across diverse HPC architectures and to fully exploit modern heterogeneous systems, including GPUs and other accelerators. It also details the hardware–software co-design principles that guided the development process. A key component of EoCoE is the development of highly optimised, cross-cutting libraries and tools, particularly in the area of linear algebra, which are essential for achieving extreme scalability. These tools are designed not only to support the EoCoE applications, but also to be reusable by a broader range of scientific communities; they are described in Section 4. Section 5 provides an overview of the dissemination, training, and community-building activities carried out within EoCoE III, aimed at maximising the project impact and ensuring the long-term sustainability of the software, expertise, and ecosystems developed. Finally, Section 6 summarises the paper and outlines future activities.

2 The EoCoE flagship applications

EoCoE III advances five flagship applications in energy materials, water, wind, and fusion. These applications are already in production and used to tackle complex and critical problems in the energy domain, contributing the European transition towards a carbon free energy. With large user bases and proven petascale scalability, these codes are being modernised to fully harness the power of heterogeneous exascale systems across Europe and worldwide.

2.1 Fusion for energy: Gysela-X++

The Fortran code Gysela [5] is one of the European flagships for global flux-driven gyrokinetics simulations of plasmas confined within a nuclear fusion reactor. However, porting such a long-lived code, from its initial petascale CPU architectures where it can scale up to 500k cores to the heterogeneous hardware of exascale supercomputers while expecting similar or better performance poses a major challenge [12]. The Gysela-X project[1] therefore consists in the development of a series of C++ codes, with the final goal of creating a successor to Gysela, called Gysela-X++, which will feature new physics capabilities, including magnetic configurations with an X-point. To ensure performance and portability, these codes are written in C++20, parallelised using MPI and Kokkos, and take advantage of exascale-oriented libraries such as Kokkos-kernels, DDC[2] or PDI[3]. In addition, development is split between 2 parts: on the one hand Gyselalib++ [3], an open-source back-end library[4] which contains all the core kernels, and on the other hand, a set of application codes combining them into specialised simulations (e.g., simplified model geometries, X-point tokamaks, stellarators).

A major milestone of the project is the axisymmetric (2D-2V) code, Gysela-Axi, which serves as a proof of concept for our development strategy, and as a test-bed for new components (multi-patch interpolations, advanced geometries, in-situ diagnostics). Figure 1 reports a preliminary benchmark for weak scaling performed on the MI250X partition of Adastra (CINES, France), showing a final overall weak efficiency of 36% on 128 nodes or 1024 GPUs. The simulations included 2 plasma species on a grid going from $(N_r, N_\theta, N_v, N_\mu) = (256, 512, 128, 64)$ up to $(1024, 1024, 512, 256)$ where we alternated between increasing a velocity dimension (v, μ) or a spacial one (r, θ), hence the stair steps pattern on some of the kernels. Since this is the first functional version of Gysela-Axi and has not yet been optimised, certain inefficiencies were expected. For instance, the "QN solver" currently lacks parallelisation, with each GPU redundantly solving the same $N_r \times N_\theta$ problem, while other inefficiencies remain to be investigated. The benchmark main objective was to stress-test the code and identify bottlenecks as well as good/bad practices in our development strategy to guide our work towards the 5D code Gysela-X++, targeting the exascale with larger meshes on even more GPUs.

2.2 Materials for energy: libNEGF

The Non-equilibrium Green's Functions (NEGF) formalism is one of the most general approaches to treat a many-body quantum system under non-equilibrium conditions, e.g. in the presence of light illumination, applied electrochemical potentials, and in the presence of carrier-phonon or carrier-carrier interactions. The most common field of application of the formalism has been in modelling nanoelectronics components, e.g., nanotransistors, resonant tunnelling devices, interband tunnelling diodes or molecular electronics. It was also applied in modelling optoelectronic devices such as LEDs or quantum cascade lasers, nanophotonic, and photovoltaic devices based on III-V heterostructures. The libNEGF project[5] started from the observation that the underlying formalism can be generalised to any local basis-set formulation of the quantum problem, hence allowing the development of an abstract interfacing to a NEGF solver [14]. In abstract sense, the required NEGFs can be written as two-dimensional tensors, e.g., $G^{r,<}_{ij}(k, E)$, corresponding to the

[1] Gysela-X: https://gyselax.github.io/
[2] DDC website: https://ddc.mdls.fr/
[3] PDI website: https://pdi.dev/main/
[4] Gyselalib++: https://github.com/gyselax/gyselalibxx/
[5] libNEGF: https://github.com/libnegf/libnegf

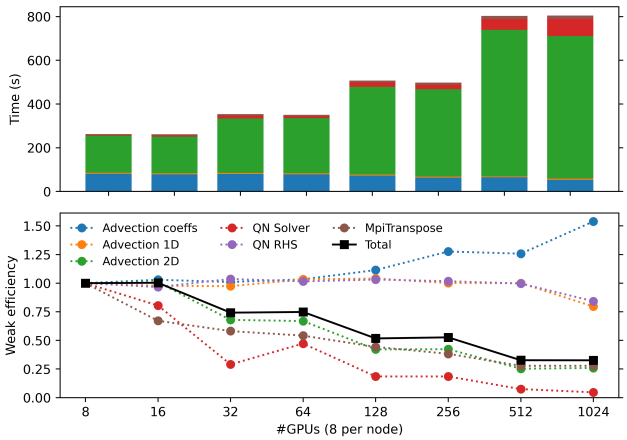

Figure 1: Preliminary benchmark for weak scaling of the first working version of Gysela-Axi on the Adastra supercomputer: walltime (upper panel) and parallel efficiency (lower panel). Different colours denote distinct kernels.

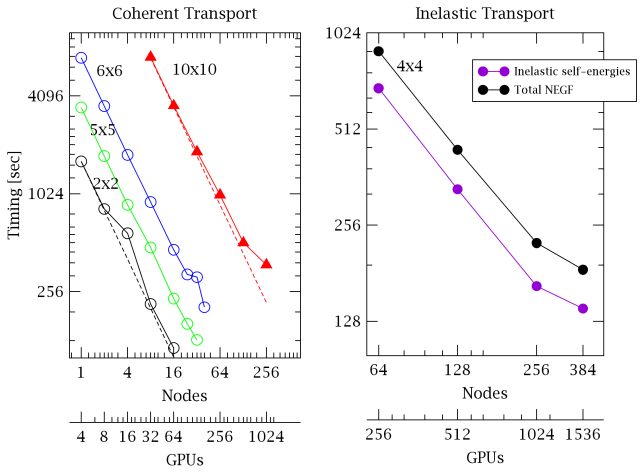

Figure 2: Benchmarks of libNEGF on JUWELS Booster (JSC, Germany) for coherent transport (left panel) and inelastic transport (right panel) for Si wires of increasing size.

retarded and *lesser* GF, respectively. The local basis indices, ij, encode spatial information, whereas (k, E) are momentum and energy variables. Solving the NEGF equations requires the solution of large dense linear systems of the form

$$\left[E\mathbf{S}(k) - \mathbf{H}(k) - \Sigma^r(k,E)\right]\mathbf{G}^r(k,E) = \mathbf{I}, \tag{1}$$

$$\left[E\mathbf{S}(k) - \mathbf{H}(k) - \Sigma^r(k,E)\right]\mathbf{G}^<(k,E) = \Sigma^<(k,E)\mathbf{G}^{r\dagger}(k,E), \tag{2}$$

where we have used bold-face matrix notation for G_{ij}. The so-called interaction *self-energies*, $\Sigma^{r,<}(k,E)$, are obtained as energy and momentum convolutions of the GFs and coupling matrices. In case of ballistic (coherent) transport, interactions are neglected and the GFs at different (k,E) are independent offering an obvious parallelisation scheme. In libNEGF we discretise and distribute k and E over an MPI cartesian grid and solve (1) and (2) with recursive algorithms. On the other hand, the convolutions involved in the computation of the self-energies lead to a communication bottleneck. Coherent and inelastic types of calculations applied to systems used for code benchmarking are shown in Figure 2. In the left panel, the systems have increasing cross-sections, from Si 2x2 with total matrix size, M=2880, corresponding to a thin wire to Si 10x10, with total matrix size, M=72000 and a cross-section of 5 nm. An example of inelastic calculation is shown on the right panel for a Si 4x4 (M=11520). Being able to address non-equilibrium carrier transport in nanophotonic devices opens the way to better understand carrier dynamics and potentially improve the engineering of novel photovoltaic materials or interfaces. In EoCoE III, we study MoS_2/WS_2 bilayers, a type-II interface that showed efficient exciton splitting. NEGF is also a valuable tool to study interfaces, such as Si/αSi in Silicon Heterojunction Solar Cells or the disordered interface of hybrid Perovskites with thin electron and hole transport layers or with metallic electrodes.

2.3 Water for energy: ParFlow

ParFlow[6] is a physics based, open source, massively parallel three dimensional hydrologic model that simulates coupled surface and subsurface flow across the terrestrial water cycle [7, 9, 10, 13, 22]. It solves variably saturated groundwater flow using Richards equation and overland flow using a simplified shallow water formulation, discretised on structured grids with conservative finite difference and finite volume schemes. The resulting nonlinear system is solved with implicit time integration using Newton Krylov methods, preconditioned by multigrid solvers. The model represents groundwater, soil moisture, and surface water as a fully coupled system, enabling consistent simulation of infiltration, runoff, river flow, and water table dynamics under heterogeneous subsurface properties and complex topography. It supports steady and transient conditions and can be coupled with land surface, atmospheric, and geochemical models to resolve cross scale interactions.

ParFlow is applied from watershed to continental and global scales, including kilometre scale simulations of the water continuum. This capability is critical for the energy transition, where hydropower and water resource management depend on reliable prediction of water availability and variability. ParFlow enables high resolution forecasts and scenario analysis of streamflow, reservoir inflows, and groundwater contributions under climate change, supporting grid stability, risk assessment of extremes, and allocation of water across competing sectors.

ParFlow is primarily written in C, with components in C++ and high level interfaces in Python and Tcl for workflow and model configuration. The code is designed for HPC with a modular solver stack built on Newton Krylov methods and scalable multigrid preconditioners, using KINSOL[7] and Hypre[8].

[6]ParFlow: https://github.com/parflow/parflow
[7]KINSOL: https://computing.llnl.gov/projects/sundials/kinsol
[8]Hypre: https://computing.llnl.gov/projects/hypre-scalable-linear-solvers-multigrid-methods

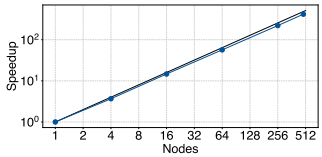

Figure 3: ParFlow speedup in weak scaling tests on LUMI-G.

Parallelisation is based on distributed memory domain decomposition with MPI. The computational grid is partitioned into subdomains assigned to processes, with communication at subdomain boundaries. The nonlinear solve is handled by the Scaled Preconditioned Generalised Minimal Residual (SPGMR) Krylov solver within KINSOL, including Jacobian evaluations and global reductions.

Performance portability is achieved through a macro based abstraction layer that decouples numerical kernels from hardware specific implementations. This enables execution across CPUs and GPUs using CUDA and Kokkos, with HIP support for AMD GPUs. Memory management is optimised through portable allocators such as Umpire[9] and RMM[10]. The build system supports configurable backends while maintaining a single code base.

ParFlow demonstrates scalable performance on leadership class systems, supporting both strong and weak scaling at extreme problem sizes. Strong scaling on JUWELS Booster reaches full system runs with $O(10^{11})$ degrees of freedom, maintaining high parallel efficiency and effective node level utilisation. These results were recognized through JUREAP certification[11], highlighting strong scaling efficiency, sustained performance, and readiness for deployment on JUPITER Booster (JSC, Germany). Weak scaling on LUMI-G (CSC, Finland), based on ClayL, a representative ParFlow benchmark simulating variably saturated infiltration into clay soil, shows nearly constant solver time and sustains 86% efficiency from 1 to 484 nodes, with a stable per GPU memory footprint (Figure 3). Performance portability is validated across heterogeneous architectures, including NVIDIA A100 (JUWELS Booster), AMD MI250X (LUMI-G), and NVIDIA Grace Hopper GH200 (JEDI, JSC, Germany). On JEDI, ParFlow achieves up to 2.8× speedup compared to JUWELS Booster for identical problem sizes, using half of the nodes.

These capabilities enabled a recent proof-of-concept study demonstrating, for the first time, integrated hydrological simulations of the terrestrial water continuum at kilometre-scale resolution on a global domain [8] (Figure 4). Together, the results establish ParFlow as a scalable and performance portable framework for hyper-resolution hydrologic modelling in energy and climate applications targeting exascale and post-exascale architectures.

2.4 Wind for energy: waLBerla-wind & SOD2D

Wind energy research relies heavily on large-eddy simulation (LES) to investigate atmospheric flows. The strong coupling between atmospheric flow, wind turbine wakes, and wind farm dynamics involves a wide range of length scales: from kilometre-scale atmospheric eddies to metre-scale wake structures. Capturing these phenomena requires both large computational domains and fine

[9]Umpire: https://github.com/llnl/Umpire
[10]RMM: https://github.com/rapidsai/RMM
[11]JUREAP: https://www.fz-juelich.de/en/jsc/jupiter/jureap

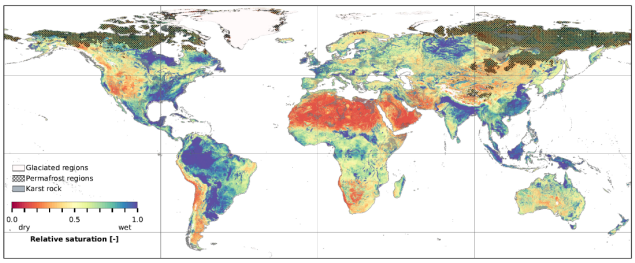

Figure 4: ParFlow hydrological simulation of the terrestrial water continuum at kilometre-scale resolution on a global domain.

spatial resolutions, making highly efficient numerical solvers essential.

In this context, two main numerical approaches are commonly used for turbulent-flow simulations. The first is based on mesoscopic methods, among which the lattice Boltzmann method (LBM) has attracted increasing interest due to its outstanding computational performance and scalability, as well as its suitability for modern HPC architectures. The second approach relies on the direct discretisation of the Navier–Stokes equations using classical computational fluid dynamics (CFD) methods, such as finite-volume, finite-difference, or finite-element schemes.

In this paper, we first present the developments carried out within the lattice Boltzmann framework, followed by those related to the Navier–Stokes-based solution. For wind energy applications using the LBM, we employ the solver waLBerla-wind[12], which was created during the EoCoE II and extended during the EoCoE III European CoE project [16, 17]. Based on the multi-physics framework waLBerla [1], waLBerla-wind benefits directly from its code-generation capabilities for highly optimised compute kernels targeting CPUs, GPUs, and APUs, with support for SIMD vectorisation and multiple floating-point precisions. waLBerla-wind contains performance-portable wind turbine models and wall-boundary treatments for neutral atmospheric boundary layer (ABL) simulations, with and without Coriolis forcing.

To alleviate grid-resolution constraints, actuator-based turbine models are used instead of explicitly resolving turbine geometries. In particular, the actuator line model (ALM) provides a more realistic representation of turbine aerodynamics but requires finer resolutions, whereas the actuator disk model (ADM) enables coarser grids at the expense of near-wake accuracy. Building on the developments initiated in EoCoE II, EoCoE III placed particular emphasis on support for AMD GPUs and APUs, as well as on improving node-level performance and parallel scalability on modern supercomputing platforms.

Mesh refinement is another key ingredient for improving computational efficiency on both CPU- and GPU-based architectures, which is addressed in waLBerla-wind. By refining the mesh only in regions of interest, such as near turbines, in wakes, or near boundaries, one can maintain accuracy while limiting computational cost.

Performance and energy consumption are evaluated on multiple HPC systems, using up to 1024 NVIDIA and AMD GPUs or

[12]waLBerla-wind: https://i10git.cs.fau.de/walberla/walberla

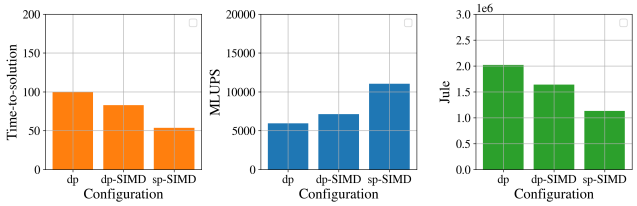

Figure 5: waLBerla-wind benchmarks on nine nodes of the scalar partition of the Adastra supercomputer. Time-to-solution (left panel), performance in Million Lattice Updates Per Second (central panel), and energy consumption (right panel) for different combinations of floating-point precision and vectorisation.

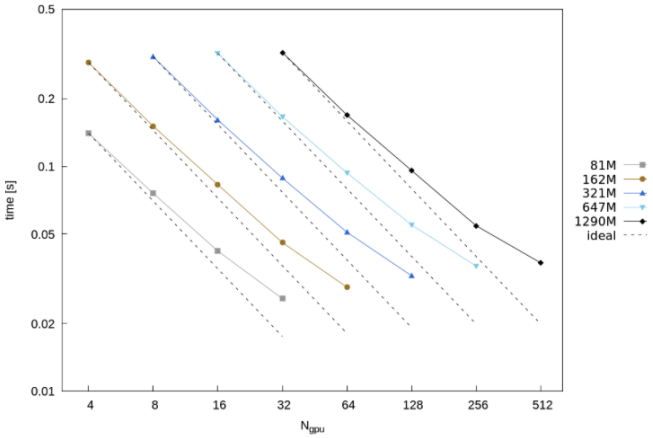

Figure 6: Scaling tests of SOD2D on MareNostrum 5 - accelerated partition. Colors indicate simulations with progressively increasing numbers of DoFs.

76800 CPU cores across test cases of increasing domain size and complexity. These cases range from a single wind turbine, namely an isolated NREL 5 MW turbine, to full wind farm simulations, including the Westermost and Anholt wind farms, which comprise 35 and 111 turbines, respectively, as well as wind farm interaction studies based on the Danish Energy Island test case, consisting of 10 wind farms with 67 turbines each. Figure 5 shows the computational performance and energy consumption for a simulation of the Anholt wind farm using nine nodes of the scalar partition of the Adastra supercomputer. The ability to perform faster-than-real-time simulations is particularly appealing, as it significantly accelerates analysis and opens the door to a wide range of investigations. As an example, one configuration related to the Danish Energy Island case, involving 670 wind turbines with 16 cells per diameter and a domain size of approximately 2.2×10^9 cells, achieves faster-than-real-time performance on 50 compute nodes using the Adastra supercomputer with reduced energy consumption. This result is especially remarkable given the complexity of the setup.

For the Navier–Stokes-based approach, we rely on SOD2D[13] (Spectral high-Order coDe to solve partial Differential equations), a high-order CFD solver for the incompressible Navier–Stokes equations based on a fractional-step scheme and spectral-element discretisation on unstructured grids. SOD2D is written in Fortran with OpenACC and was designed from the outset as a GPU-first code for modern accelerator-based architectures. It has been benchmarked on several European supercomputers and has demonstrated excellent scalability for more than 5 million degrees of freedom (DoFs) per GPU, as illustrated in Figure 6, which shows scalability up to 512 GPUs on MareNostrum 5 (BSC, Spain) for meshes ranging from 81 million DoFs to 1290 million DoFs. In particular, the code has been used in production on more than 80% of the accelerated partition of MareNostrum 5, corresponding to 4000 NVIDIA H100 GPUs. SOD2D is fully open source and distributed under the MIT license.

For wind energy applications, SOD2D couples the incompressible flow equations with a thermal transport equation to represent stable and convective atmospheric regimes. This coupling is essential, since atmospheric stratification strongly affects turbulence intensity, wake recovery, and the vertical wind profile. In stable conditions, less turbulent mixing leads to significantly longer wakes, which may extend from one wind farm to neighbouring ones. To support realistic wind farm simulations, SOD2D also includes ADMs, canopy models for forests, damping layers to account for gravity waves, and precursor simulations to generate turbulent inflow conditions. The use of unstructured grids allows to easily deal with complex terrain and has also been useful in offshore wind farms to better refine regions of interest. The use of unstructured grids further enables the treatment of complex terrain and local mesh refinement in regions of interest, both onshore and offshore.

By combining advanced physical modelling with a high-order discretisation optimised for GPU execution, SOD2D provides an accurate and scalable tool for the analysis of atmospheric boundary layer flows in wind farms. Although wind energy is already a mature technology, wind plant losses remain substantial because of the complexity of turbulent flow, especially over complex terrain. Improved numerical modelling can therefore contribute to a more profound understanding of wake interactions, blockage effects, and power distribution within wind farms, ultimately supporting more efficient wind energy production. The SOD2D team has collaborated with Iberdrola for more than 15 years and has recently started working with Ocean Winds. Figure 7 shows pressure results for the Mermaid wind farm operated by Ocean Winds: the horizontal plane highlights the impact of individual turbines on the pressure field, while the vertical cuts reveal gravity waves generated by the interaction between the flow and the farm.

3 Co-design & Performance Portability

3.1 Co-Design

Performance co-design across multiple application codes is inherently challenging due to their diverse computational characteristics. To address this, we adopt a measurement-driven approach within the co-design process, evaluating both energy consumption and time-to-solution across different hardware architectures. Our analysis spans single-node and scaling performance to capture both architectural efficiency and distributed behaviour. Within EoCoE III, these efforts are embedded in a structured co-design framework

[13] SOD2D: https://gitlab.com/bsc_sod2d/sod2d_gitlab

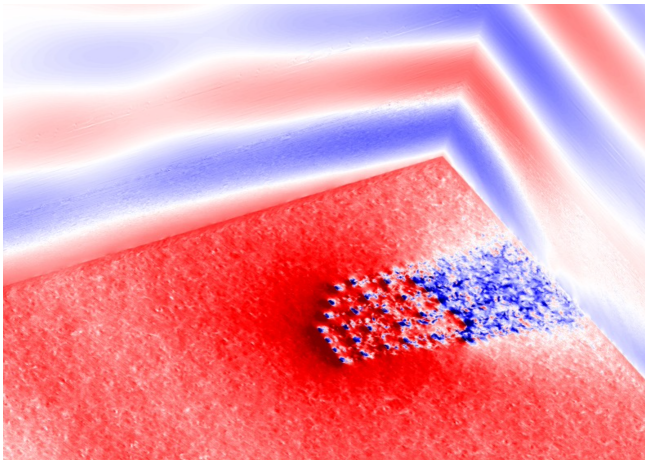

Figure 7: SOD2D Mermaid wind farm (OceanWinds) simulation including gravity waves.

that tightly couples application-oriented work packages with performance engineering activities. The co-design process itself can be viewed as a multi-parameter optimisation problem, where hardware, algorithms, and implementations are continuously refined through collaboration between application developers, mathematical algorithm specialists, HPC experts, and technology providers. In this context, performance engineering serves as the central driver, informed by systematic measurements and guided by application requirements and architectural constraints. Concretely, co-design activities are tailored to the needs of each flagship code while following common principles. For ParFlow, kernel code generation using pystencils[14] and matrix-free algorithms are explored. In SOD2D, the focus is on analysing memory-bound OpenACC kernels, improving memory access patterns, and kernel fusion. For waLBerla-wind, co-design centres on adaptive refinement and GPU-specific optimisations of the LBM. Gysela-X++ is undergoing early-stage performance analysis alongside its ongoing development to guide design decisions from the outset. In libNEGF, co-design includes the exploration of new algorithms and programming models through a transition to Julia, enabling rapid prototyping and improved portability, particularly for GPU execution. Importantly, energy efficiency is a cross-cutting concern throughout this process. Performance optimisations identified through co-design not only reduce runtime but also directly translate into lower energy consumption. This tight coupling of performance and energy considerations enables a unified perspective on optimisation, ensuring that advances achieved for individual codes contribute to broader exascale readiness across the application portfolio.

3.2 Performance Portability

Modern HPC increasingly depends on performance-portable programming models that allow scientific applications to run efficiently across heterogeneous architectures without major code changes. Abstraction frameworks such as Kokkos, RAJA, SYCL, and OpenMP target offloading enable a single codebase to exploit CPUs, GPUs, and emerging accelerators while maintaining long-term sustainability. Pystencils contributes to this landscape by generating optimised, architecture-specific kernels from a unified high-level stencil description. Its symbolic transformations, backend-aware optimisations, and just-in-time compilation deliver competitive performance across diverse hardware with minimal developer effort. This capability is already leveraged in our large-scale simulation frameworks waLBerla-wind and ParFlow, where pystencils helps ensure portability, maintainability, and adaptability on rapidly evolving HPC systems.

[14] pystencils: https://pypi.org/project/pystencils/

4 Tools and libraries for the energy transition

HPC is a key enabler for advanced applications in the energy sector, supporting the simulation, analysis, and optimisation of complex multi-physics and multi-scale systems. From materials design and fusion energy to wind modelling and water management, applications rely on increasingly accurate and large-scale numerical simulations, often coupled with data-driven methodologies. In this context, the efficiency and scalability of the underlying computational methods are critical to the full exploitation of emerging exascale architectures. In particular, linear algebra operations, including sparse and dense matrix computations, iterative solvers, and preconditioners, represent the computational backbone of most energy-related applications, while data analytics introduces additional challenges related to data movement, I/O, and real-time processing. The development of advanced, scalable, and performance-portable software libraries is therefore essential to translate mathematical innovations into effective solutions for next-generation energy applications.

Within this framework, significant efforts have been devoted to the design and development of scalable algorithms and software libraries for linear algebra and in situ data analytics, targeting both extreme scalability and node-level performance on heterogeneous architectures. The activities span communication-avoiding Krylov methods, advanced multigrid preconditioners, randomised linear algebra, and domain decomposition techniques. A central outcome is the evolution of the `PSCToolkit` framework [4], which has been enhanced with new kernels supporting communication-reduced Krylov solvers, advanced polynomial smoothers, and multi-GPU support, and demonstrating excellent weak scalability up to tens of billions of DoFs on thousands of GPUs (Figure 8).

Another central outcome is the evolution of the Composyx framework, which has been enhanced with hybrid multigrid/domain decomposition methods, mixed-precision support, and full multi-GPU and multicore capabilities, and demonstrating the excellent weak scalability on the Jean Zay supercomputer (CNRS, France). Complementary developments include GPU-enabled AMG solvers (AGMG), randomised SVD-based preconditioners integrated into PETSc, all demonstrating strong scalability on large European HPC systems such as Leonardo (CINECA, Italy) and MareNostrum 5.

At the single-node level, significant advances have been achieved in exploiting GPU architectures and mixed-precision techniques. In particular, this work investigates level-dependent mixed-precision formulations of AMG-preconditioned Krylov subspace methods, in which numerical precisions are selectively assigned across the multigrid hierarchy according to the role and sensitivity of each level. Its main contribution is to show that using lower precision on

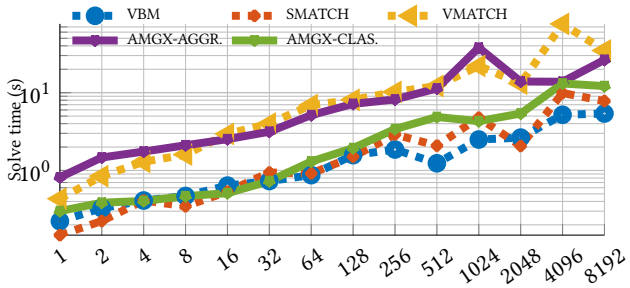

Figure 8: Solve time as a function of the number of GPUs for the preconditioned conjugate gradient method using different AMG preconditioners (weak scaling tests). The reported time corresponds to the total time to convergence. The tested preconditioners include three configurations from `PSCToolkit`: VBM, SMATCH, and VMATCH. These are compared with two configurations from the NVIDIA AMGX library: AMGX-AGGREGATION and AMGX-CLASSICAL. VBM and SMATCH achieve the best overall time-to-solution at medium-to-large scales, outperforming both AMGX variants on the Leonardo machine.

carefully selected multigrid levels, while retaining higher precision where required for stability and accuracy, can improve performance, memory efficiency, and energy consumption without compromising convergence to double-precision accuracy. More broadly, optimised GPU kernels, improved OpenMP support, and adaptive AMG strategies have led to enhanced performance and energy efficiency, with gains over state-of-the-art libraries. At the inter-node parallel level, the development of mixed-precision multigrid methods and task-based implementations of direct solvers provides new pathways to balance accuracy, performance, and energy consumption.

These advances are tightly coupled with application-driven developments in key scientific domains. For instance, `PSCToolkit` has been interfaced with ParFlow through Sundials/KINSOL [6] to enhance the robustness and scalability of nonlinear solvers for the Richards equation, enabling more efficient simulations of subsurface flow [2]. In the context of fusion plasma simulations, the development of geometric multigrid preconditioners tailored to the Gysela-X++ code provides scalable solvers adapted to complex geometries and anisotropic discretisations [11]. Furthermore, novel strategies to introduce hybrid parallelism in linear algebra kernels are being investigated for libNEGF, with the aim of improving the efficiency of block-structured solvers arising in quantum materials simulations.

Beyond their immediate application context, these developments are conceived as reusable and extensible building blocks with the potential to impact a broad range of scientific computing applications. By promoting performance portability, algorithmic scalability, and interoperability with existing software ecosystems, these libraries contribute to strengthening the HPC software stack and fostering the adoption of advanced numerical methods within the wider scientific and engineering community.

In addition to simulation-oriented kernels, we also address the growing importance of data analytics in HPC workflows through the DEISA framework for in situ data processing. The transition to a distributed runtime based on Ray enables scalable task scheduling and efficient integration with flagship applications, supporting large-scale data analysis pipelines tightly coupled with simulations. Overall, these activities demonstrate how advanced numerical libraries can bridge the gap between mathematical innovation and practical HPC applications, delivering tangible performance improvements and enabling new scientific capabilities at the exascale.

5 Dissemination and Impact for the Long-Term Sustainability of the EoCoE Project

A targeted dissemination strategy has been designed and implemented to ensure the long-term sustainability and impact of EoCoE III outcomes, while promoting their effective uptake. This strategy relies on a diversified set of actions, including the project website, newsletters, LinkedIn, YouTube, scientific publications, workshops, webinars, and training activities, alongside continuous dialogue with stakeholders from both the HPC and energy communities.

Industrial collaborations have supported the development of energy-related use case studies, demonstrating the efficiency of the developed codes in reducing energy consumption. In parallel, dissemination efforts have facilitated broader adoption: the lighthouse codes are publicly available on GitLab/GitHub, enabling open and collaborative development.

The project has reached both scientific and policy audiences through joint webinars and knowledge-exchange activities with other CoEs and National Competence Centres. Additional initiatives include roundtables with HPC vendors (e.g., NVIDIA, SiPearl) and workshops at major conferences such as HiPEAC 2026. Collaborations with EU initiatives, including EuroCC, CASTIEL-2, CEEC, POP [21], MaX [15], Cheese, MultiXscale, Excellerat P2, Hidalgo 2, SPACE [18–20], and Esiwace, have also been strengthened.

Training has played a central role in fostering a future community of skilled users and developers. In this context, EoCoE has contributed to high-level educational events such as the 3rd CINI Summer School on HPC, as well as hackathons and activities involving PhD students.

Engagement with energy stakeholders has been consolidated through participation in the European Sustainable Energy Week (EUSEW, as in 2025), involvement in the SET Plan Task Force 3, and collaboration with EERA.

The EoCoE III project continues to actively disseminate its results and solutions across the HPC and energy communities. Traditional communication channels, such as publications, conference presentations, seminars, and online content, have been systematically employed over the years, as documented on the project website[15]. At the same time, particular emphasis has been placed on expanding the user community through dedicated training activities and joint initiatives with the European Energy Research Alliance (EERA), the largest energy research and development network in Europe, comprising more than 250 institutions.

EERA's role as a high-level advisor to the European Commission and its involvement in initiatives such as the SET Plan provide an effective framework for disseminating EoCoE III results. This

[15] EoCoE website: https://www.eocoe.eu/publications/

collaboration has led to contributions to position papers on the exploitation of HPC[16] and AI[17] in the energy domain, participation in advisory boards for cross-cutting technologies (e.g., digitalisation) within SET Plan Task Forces, and the organisation of workshops and seminars on HPC applications in energy. A recent example is the workshop held in Vienna in February 2026 on photovoltaics and digitalisation[18].

6 Summary & Perspectives

We presented an overview of the five lighthouse applications at the core of the EoCoE III project, highlighting the refactoring, algorithmic, and software engineering advances that are transforming legacy codes into exascale-ready applications. These developments enable efficient use of European pre-exascale and emerging exascale systems to address complex challenges in fusion, materials, water, and wind energy with unprecedented levels of resolution and fidelity.

We also outlined the strategies adopted to ensure code portability across diverse HPC architectures and to fully exploit modern heterogeneous systems, including GPUs and other accelerators. These efforts are guided by hardware–software co-design principles, carried out in close collaboration with technology providers, and aligned with European objectives on technological sovereignty. In addition, we reported on the development of highly optimised, transversal libraries and tools, particularly in the area of linear algebra, that are essential for achieving extreme scalability.

Furthermore, we discussed the dissemination, training, and community-building initiatives undertaken within the EoCoE project to maximise its impact. These activities promote knowledge transfer, reinforce collaboration between the energy and HPC communities, and contribute to the long-term sustainability of the software, expertise, and ecosystems developed.

Finally, we emphasize that all EoCoE flagship applications are already deployed in production to address mission-critical energy challenges, contributing to Europe transition toward carbon-free energy. For example, Gysela supports turbulence studies for ITER, ParFlow is used by hydropower companies, libNEGF enables research on photovoltaic materials, and wind codes are used for advanced industrial analyses. Building on this foundation, EoCoE III is enhancing these applications to fully leverage upcoming exascale systems. Therefore, its societal impact is already tangible and will continue to grow, consolidating a robust European ecosystem for exascale computing in energy and enabling lasting scientific and industrial advances beyond the end of the project in December 2026.

Acknowledgments

This project has received funding from the European High Performance Computing Joint Undertaking under grant agreement n. 101144014.

[16] EERA–EoCoE position paper 1: https://www.eera-set.eu/component/attachments/?task=download&id=771

[17] EERA–EoCoE position paper 2: https://zenodo.org/records/15105498

[18] EERA Workshop on Photovoltaics & Digitalization: https://luma.com/pvdigi26

References

[1] M. Bauer et al. 2020. waLBerla: A block-structured high-performance framework for multiphysics simulations. *Computers & Mathematics with Applications*. doi:10.1016/j.camwa.2020.01.007

[2] D. Bertaccini et al. 2024. Why diffusion-based preconditioning of Richards equation works: Spectral analysis and computational experiments at very large scale. *Numerical Linear Algebra with Applications* 31, 1 (2024), e2523. doi:10.1002/nla.2523

[3] E. Bourne et al. 2025. Gyselalib++: A portable c++ library for semi-lagrangian kinetic and gyrokinetic simulations. *Journal of Open Source Software* 10, 113 (2025), 8582.

[4] P. D'Ambra, F. Durastante, and S. Filippone. 2025. PSCToolkit: Solving Sparse Linear Systems with a Large Number of GPUs. In *Emerging Technologies in Computational Sciences for Industry, Sustainability and Innovation*, M. Giacomini et al. (Eds.). Springer Nature Switzerland, Cham, 271–285.

[5] V. Grandgirard et al. 2016. A 5D gyrokinetic full-f global semi-Lagrangian code for flux-driven ion turbulence simulations. *Computer physics communications* 207 (2016), 35–68.

[6] A. C. Hindmarsh et al. 2005. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Software* 31, 3 (2005), 363–396.

[7] Jim E. J. and Carol S. W. 2001. Newton–Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems. *Advances in Water Resources* 24, 7 (2001), 763–774. doi:10.1016/S0309-1708(00)00075-0

[8] S. Kollet et al. 2026. Global groundwater modeling: Proof-of-concept of 3D variably saturated flow simulation at kilometer resolution. *Journal of Hydrology X* 30 (2026), 100213. doi:10.1016/j.hydroa.2025.100213

[9] S. J. Kollet and R. M. Maxwell. 2006. Integrated surface–groundwater flow modeling: A free-surface overland flow boundary condition in a parallel groundwater flow model. *Advances in Water Resources* 29, 7 (2006), 945–958. doi:10.1016/j.advwatres.2005.08.006

[10] B. N. O. Kuffour et al. 2020. Simulating coupled surface–subsurface flows with ParFlow v3.5.0: capabilities, applications, and ongoing development of an open-source, massively parallel, integrated hydrologic model. *Geoscientific Model Development* 13, 3 (2020), 1373–1397. doi:10.5194/gmd-13-1373-2020

[11] J. Litz et al. 2026. Memory - and compute-optimized geometric multigrid GMG-Polar for curvilinear coordinate representations – Applications to fusion plasma. *J. Comput. Appl. Math.* 481 (2026), 117308. doi:10.1016/j.cam.2025.117308

[12] E. Malaboeuf et al. 2025. Porting the GYSELA code on GPU enabled supercomputers. In *COMPAS*.

[13] R. M. Maxwell. 2013. A terrain-following grid transform and preconditioner for parallel, large-scale, integrated hydrologic modeling. *Advances in Water Resources* 53 (2013), 109–117. doi:10.1016/j.advwatres.2012.10.001

[14] A. Pecchia et al. 2008. Non-equilibrium Green's functions in density functional tight binding: method and applications. *New Journal of Physics* 10, 6 (jun 2008), 065022. doi:10.1088/1367-2630/10/6/065022

[15] L. Riha et al. 2025. MaX - Materials design at the exascale - a EuroHPC Centre of Excellence: Recent selected results: Invited Paper. In *Proceedings of the 22nd ACM International Conference on Computing Frontiers: Workshops and Special Sessions (CF '25 Companion)*. Association for Computing Machinery, New York, NY, USA, 150–156. doi:10.1145/3706594.3727577

[16] H. Schottenhamml. 2026. *Lattice Boltzmann Methods for Large-scale Wind Energy Applications: Wind Turbine Modelling in an Atmospheric Boundary Layer*. Ph. D. Dissertation. Friedrich-Alexander-University Erlangen-Nuremberg.

[17] H. Schottenhamml et al. 2024. waLBerla-wind: A Lattice-Boltzmann-based High-Performance Flow Solver for Wind Energy Applications. *Concurrency and Computation: Practice and Experience* 36, 16 (July 2024), e8117. arXiv:2402.13171 [physics] doi:10.1002/cpe.8117

[18] N. Shukla et al. 2025. EuroHPC SPACE CoE: Redesigning Scalable Parallel Astrophysical Codes for Exascale (Invited Paper). In *Proceedings of the 22nd ACM International Conference on Computing Frontiers Workshops and Special Sessions (CF '25 Companion)*. ACM, 177–184. doi:10.1145/3706594.3728892

[19] N. Shukla et al. 2025. Towards Exascale Computing for Astrophysical Simulation Leveraging the Leonardo EuroHPC System. *Procedia Computer Science* (2025), 112–123. doi:10.1016/j.procs.2025.08.238267

[20] N. Shukla et al. 2026. Exascale Computing to Accelerate Discoveries in Astrophysics and Space Plasma Physics. *Nature Astronomy* (2026). doi:10.1038/s41550-026-02807-8

[21] R. Vavrik et al. 2025. POP3: Advancing HPC Performance and Productivity: Invited Paper. In *Proceedings of the 22nd ACM International Conference on Computing Frontiers: Workshops and Special Sessions (CF '25 Companion)*. Association for Computing Machinery, New York, NY, USA, 157–162. doi:10.1145/3706594.3727579

[22] C. S. Woodward. 1998. A Newton-Krylov-multigrid Solver For Variably Saturated Flow Problems. *WIT Transactions on Ecology and the Environment* 24 (1998). doi:10.2495/CMWR980752