Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) iii-iv

Procedia Computer Science

www.elsevier.com/locate/procedia

Table of Contents

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 1–2

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

Preface

The EuroHPC Joint Undertaking (EuroHPC JU) is delighted to present its 2024 EuroHPC User Day Book of Proceedings, showcasing some of the remarkable scientific and societal advancements achieved by the EuroHPC user community through European supercomputing resources.

The second EuroHPC User Day took place on 22-23 October 2024 at the Eye Film Museum in Amsterdam, Netherlands. Organised by the EuroHPC JU, in collaboration with EuroCC Netherlands, this two-day event brought together around 200 participants from across Europe.

The event served as a dynamic platform for showcasing cutting-edge scientific projects leveraging EuroHPC supercomputers and played a crucial role in strengthening the EuroHPC user community. Attendees shared best practices, presented results, gathered feedback, and engaged with both established and potential users of EuroHPC supercomputers. The event also offered participants the opportunity to explore a range of topics and deepen their understanding of the resources and support available through the EuroHPC JU.

The EuroHPC JU continues to play a pivotal role in driving Europe's leadership in supercomputing, enabling the European Union and 35 participating countries to pool resources and expertise. Its mission includes deploying a world-class supercomputing infrastructure, fostering research and innovation, and nurturing a skilled and vibrant European supercomputing ecosystem. These efforts are vital to advancing Europe's digital transformation, scientific excellence, and technological sovereignty.

The 2024 EuroHPC User Day featured 16 state-of-art projects showcasing scientific breakthroughs across a range of disciplines, from computational physics to earth sciences and climate, astrophysics, resources optimisation and software management, artificial intelligence, engineering and humanities. These projects have been selected following submission of scientific papers demonstrating the transformative potential of EuroHPC supercomputers. The review selection of the received manuscripts was performed by an editorial board of scientific reviewers.

EuroHPC JU extends its gratitude to the reviewers and projects who contributed to the success of the EuroHPC User Day 2024. Special thanks are also due to the EuroHPC Hosting Entities for the excellent support of the users.

The EuroHPC JU remains committed to empowering researchers and innovators through supercomputing. The projects presented here are a testament to the incredible potential of HPC to address complex challenges, drive scientific discovery, and support Europe's leadership in the field.

We look forward to the continued growth of this vibrant community and to celebrating future successes in the years to come.

Anders Dam Jensen – Executive Director of the EuroHPC JU
Lilit Axner – Chair of the EuroHPC User Day Programme Committee
& Programme Officer Infrastructure at the EuroHPC JU

Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day
10.1016/j.procs.2025.02.254

Program Committee

Lilit Axner EuroHPC JU (Chair)
Anders Dam Jensen EuroHPC JU
Josephine Wood EuroHPC JU
Beatrice Rossi EuroHPC JU
Pauline Gounaud EuroHPC JU
Love Börjeson - KBLab at the National Library of Sweden
Fredrik Heintz - University of Linköping
Adrian Jackson - EPCC
Michele Weiland - EPCC
Jean-Philippe Nomine - CEA
Stéphane Requena – GENCI
Xavier Besseron – University of Luxembourg
Jonas Latt – University of Geneva
Estela Suarez – SiPearl and FZJ
Andrius Popovas – University of Oslo
Rossen Apostolov – KTH Royal Institute of Technology
Zoe Cournia - Academy of Athens

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 3–12

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

Fast, and Accurate Radiative Transfer for Land Surface Models

Kazem Ardaneh[a,*], Fabienne Maignan[b], Sebastiaan Luyssaert[c], Philippe Peylin[b], Olivier Boucher[a]

[a]*Institut Pierre-Simon Laplace (IPSL), Sorbonne Université / CNRS, Paris, France*
[b]*Laboratoire des sciences du climat et de l'environnement (LSCE), IPSL, CEA/CNRS/UVSQ, Gif-sur-Yvette, France*
[c]*Section Systems Ecology, Amsterdam Institute for Life and Environment (A-LIFE), Vrije Universiteit Amsterdam, the Netherlands*

Abstract

Land surface models (LSMs) simulate processes occurring at the Earth's surface (including those related to vegetation, soil, and hydrology) and their interactions with the atmosphere. LSMs are crucial for environmental monitoring, weather forecasting, and climate studies. The radiative transfer (RT) through vegetation canopies is an important process that determines photosynthesis, and the land surface energy budget. Conventional multilayer iterative solvers for RT through the canopy are computationally demanding. Here, we develop a multilayer matrix-based RT solver for vegetation canopies. The results show that the solver matches the accuracy of existing models and significantly reduces the computational time for RT, highlighting its potential for practical applications.

Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day

Keywords: Land surface modeling; ORCHIDEE; Vegetation canopies; Radiative transfer; Two-stream model; Matrix-based solver

1. Introduction

Climate models help to understand and predict the Earth's future climate by simulating the interactions between the atmosphere, oceans, and land surfaces. Within these models, land surface models (LSMs) represent the exchange of carbon, water, and energy, between the land and the atmosphere. ORCHIDEE (Organising Carbon and Hydrology In Dynamic Ecosystems) [1] is a process-based LSM developed at the Institut Pierre-Simon Laplace (IPSL). This model is widely used in Earth System Modeling and integrates experimental, inventory, and monitoring data from field sites, atmospheric stations, and satellites to refine predictions of land-atmosphere exchange and feedback in different climates and ecosystems.

Radiative transfer (RT) through the vegetation canopy is important when modeling the land surface as it influences photosynthesis, the energy balance of the land surface, and the climate system. The vegetation canopy interacts with

* Corresponding author
E-mail address: kardaneh@ipsl.fr

1877-0509
Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day
10.1016/j.procs.2025.02.255

incoming solar radiation through absorption, reflection, and transmission by the vegetation elements consisting of leaves, branches, and stems and their background consisting of litter and soils. A two-stream approximation in the RT model is widely used to simulate radiation's interaction with vegetation canopy. It simplifies the RT equation by considering only two radiation directions: direct (downward) and diffuse (upward and downward). In spite of the simplification, such an approach captures the basic characteristics of radiation interactions within the vegetation canopy while remaining simple.

In considering existing approaches to RT within vegetation canopies, Ref. [2] provided analytical solutions of the two-stream equations for vegetation canopies as presented in Ref. [3], which is used in the Community Land Model (CLM) as detailed in Ref. [4]. The Sellers model, although pioneering, is limited by its single-layer structure. Additionally, Ref. [5] presented a single-layer two-stream model incorporating the black background equations of Ref. [6] and considering successive reflections from the surface. The multilayer version of Ref. [5], presented in Ref. [7], addresses the limits of the single-layer approach by considering the vertical heterogeneity of the vegetation canopies. However, it relies on an iterative approach to find the correct reflection, transmission, and absorption for each layer, making it computationally demanding as it currently accounts for almost 30% of the computing time of the ORCHIDEE LSM.

In this study, we developed a multilayer matrix-based two-stream RT solver for vegetation canopy based on Ref. [8]. We formulate the RT model as a system of linear equations, which can then be solved using efficient matrix operations. This allows us to harness the power of linear algebra algorithms and exploit parallelism more effectively than traditional iterative methods. The solver is validated against the models of Refs. [2, 5, 7]. While producing almost the same results, the new solver considerably reduces the computational cost for RT in the ORCHIDEE code and argues in favor of adopting matrix methods in practical applications.

2. Matrix-based RT

The key to our approach is the matrix-based RT, which deals with the interactions of radiative fluxes within and between multiple layers. This approach is particularly well suited for complex vegetation canopies for which their vertical heterogeneity can have a significant impact on RT. In the two-stream approximation, the basic equations for the upward ($F^{\uparrow}$) and downward ($F^{\downarrow}$) diffuse fluxes are given by [8]:

$$\begin{cases} \dfrac{\mathrm{d}F^{\uparrow}}{\mathrm{d}\tau} = \gamma_1 F^{\uparrow} - \gamma_2 F^{\downarrow} - \gamma_3 \omega \pi F_{\odot} \exp(-\tau/\mu_0) \\ \dfrac{\mathrm{d}F^{\downarrow}}{\mathrm{d}\tau} = \gamma_2 F^{\uparrow} - \gamma_1 F^{\downarrow} + \gamma_4 \omega \pi F_{\odot} \exp(-\tau/\mu_0) \end{cases} \tag{1}$$

The gamma coefficients (γ_1, γ_2, γ_3, and $\gamma_4 = 1-\gamma_3$) are key parameters in the two-stream approximation that prescribe the scattering and absorption properties of the medium and are thus essential in determining the radiative fluxes within each layer. In particular, γ_1, and γ_2 relate to the interactions of upward and downward fluxes, while γ_3 and γ_4 represent the source terms associated with scattering and absorption processes. The values of these coefficients can be derived from various approximation methods such as the Eddington approximation, quadrature, and δ-methods [6]. In Eqs. (1), τ is the optical depth, ω is the single-scattering albedo, $\pi F_{\odot}$ is the incident flux at the top of the canopy, and μ_0 is the cosine of the solar zenith angle.

When in Eqs. (1) γ_1 is substituted with $[1-\omega(1-\beta)]/\bar{\mu}$, γ_2 is replaced by $\omega\beta/\bar{\mu}$, γ_3 is taken as β_0, μ_0 is redefined as $1/K$, $\pi F_{\odot}$ is set equal to K, and τ is replaced with L (leaf area index) the two-stream RT equations for the vegetation canopies are obtained [2, 3, 5]:

$$\begin{cases} \dfrac{\mathrm{d}F^{\uparrow}}{\mathrm{d}L} = \dfrac{[1-\omega(1-\beta)]}{\bar{\mu}} F^{\uparrow} - \dfrac{\omega\beta}{\bar{\mu}} F^{\downarrow} - \beta_0 \omega K \exp(-K L) \\ \dfrac{\mathrm{d}F^{\downarrow}}{\mathrm{d}L} = \dfrac{\omega\beta}{\bar{\mu}} F^{\uparrow} - \dfrac{[1-\omega(1-\beta)]}{\bar{\mu}} F^{\downarrow} + (1-\beta_0)\omega K \exp(-K L) \end{cases} \tag{2}$$

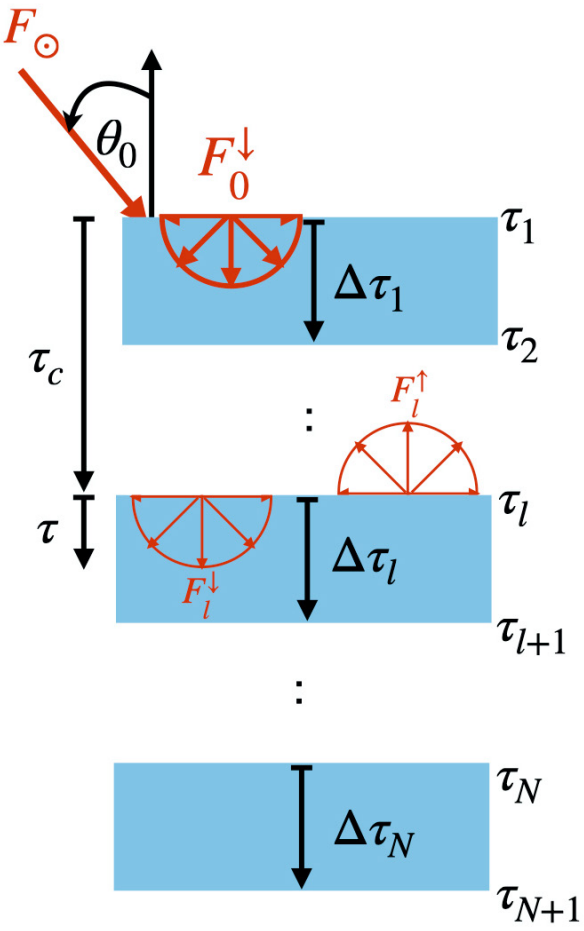


Fig. 1. RT through a multilayer medium. The incoming solar flux $F_\odot$ is incident at an angle θ_0 with the surface normal. The medium is split into N layers, with optical depths of $\Delta\tau_l$. The total optical depth above the lth layer is τ_c, while τ gives the optical depth within a layer measured from the top of that layer. For each layer l, the downward diffuse flux $F_l^\downarrow$ is partially scattered and absorbed, giving an upward diffuse flux $F_l^\uparrow$.

Extending the single-layer two-stream model [Eqs. (1)] to multiple layers involves calculating the fluxes within each layer, considering the cumulative optical depths (τ_c). In a given layer l, the fluxes are expressed as follows:

$$
\begin{cases}
F_l^\downarrow(\tau) = \Gamma_l\, B_{l2}\, \exp(\lambda_l\,\tau) + B_{l1}\, \exp(-\lambda_l\,\tau) + C_l^\downarrow\, \exp[-(\tau+\tau_c)/\mu_0] \\
F_l^\uparrow(\tau) = B_{l2}\, \exp(\lambda_l\,\tau) + \Gamma_l\, B_{l1}\, \exp(-\lambda_l\,\tau) + C_l^\uparrow\, \exp[-(\tau+\tau_c)/\mu_0] \\
\lambda_l = \left(\gamma_{l1}^2 - \gamma_{l2}^2\right)^{1/2} \\
\Gamma_l = \dfrac{\gamma_{l1} - \lambda_l}{\gamma_{l2}} \\
C_l^\uparrow = \dfrac{\omega_0\,\pi\,F_\odot[(\gamma_{l1} - 1/\mu_0)\,\gamma_{l3} + \gamma_{l4}\,\gamma_{l2}]}{(\lambda_l^2 - 1/\mu_0^2)} \\
C_l^\downarrow = \dfrac{\omega_0\,\pi\,F_\odot\,[(\gamma_{l1} + 1/\mu_0)\,\gamma_{l4} + \gamma_{l2}\,\gamma_{l3}]}{(\lambda_l^2 - 1/\mu_0^2)}
\end{cases}
\tag{3}
$$

Equations (3) gives the analytical solution for Eqs. (1), where τ is the optical depth within each layer, τ_c is the cumulative optical depth at the top of each layer (Fig. 1). The coefficients B_{l1} and B_{l2} are derived from the boundary conditions and the continuity at the interfaces between the layers. For numerical stability and to keep all arguments to the exponential terms negative, the variables A_l and B_l as in Eqs. (4) are introduced. These variables represent the modified flux amplitudes within each layer, derived such as to maintain consistent sign conventions between the exponential terms.

$$\begin{cases} A_l = \dfrac{B_{l1}\exp(\lambda_l \Delta\tau_l) + B_{l2}}{2} \\ B_l = \dfrac{B_{l1}\exp(\lambda_l \Delta\tau_l) - B_{l2}}{2} \end{cases} \tag{4}$$

In these equations, $\Delta\tau$ represents the total optical depth of the layer (Fig. 1). The downward and upward fluxes then read:

$$\begin{cases} F_l^{\uparrow}(\tau) = A_l\{\exp[-\lambda_l(\Delta\tau_l - \tau)] + \Gamma_l \exp(-\lambda_l\tau)\} + B_l\{\exp[-\lambda_l(\Delta\tau_l - \tau)] - \Gamma_l \exp(-\lambda_l\tau)\} + C_l^{\uparrow}\ \exp[-(\tau+\tau_c)/\mu_0] \\ F_l^{\downarrow}(\tau) = A_l\{\Gamma_l \exp[-\lambda_l(\Delta\tau_l - \tau)] + \exp(-\lambda_l\tau)\} + B_l\{\Gamma_l \exp[-\lambda_l(\Delta\tau_l - \tau)] - \exp(-\lambda_l\tau)\} + C_l^{\downarrow}\ \exp[-(\tau+\tau_c)/\mu_0] \end{cases} \tag{5}$$

In multilayer RT, flux continuity and boundary conditions are maintained using a matrix of coefficients. The continuity equations for upward and downward fluxes at the interfaces of layers l and $l+1$ are given by:

$$\begin{cases} F_l^{\uparrow}(\tau = \Delta\tau_l) = F_{l+1}^{\uparrow}(\tau = 0), \text{ for } l = 1, 2, \cdots, (N-1) \\ F_l^{\downarrow}(\tau = \Delta\tau_l) = F_{l+1}^{\downarrow}(\tau = 0), \text{ for } l = 1, 2, \cdots, (N-1) \end{cases} \tag{6}$$

Using Eqs. (5), Eqs. (6) can be expressed as a system of linear equations:

$$\begin{cases} e_l A_l + d_l B_l - f_{l+1} A_{l+1} + g_{l+1} B_{l+1} = C_{l+1}^{\uparrow}(0) - C_l^{\uparrow}(\Delta\tau_l) \\ f_l A_l + g_l B_l - e_{l+1} A_{l+1} + d_{l+1} B_{l+1} = C_{l+1}^{\downarrow}(0) - C_l^{\downarrow}(\Delta\tau_l) \end{cases} \tag{7}$$

where the coefficients e_l, d_l, f_l, and g_l are defined as:

$$\begin{cases} e_l = 1 + \Gamma_l \exp(-\lambda_l \Delta\tau_l) \\ d_l = 1 - \Gamma_l \exp(-\lambda_l \Delta\tau_l) \\ f_l = \Gamma_l + \exp(-\lambda_l \Delta\tau_l) \\ g_l = \Gamma_l - \exp(-\lambda_l \Delta\tau_l) \end{cases} \tag{8}$$

We derive a tridiagonal system of equations to reduce computational cost. The following steps outline the derivation process. Multiply the first equation in Eqs. (7) by d_{l+1} and the second by g_{l+1}, subtract the resulting equations to eliminate B_{l+1}, leading to a new equation that relates A_l, B_l, and A_{l+1}. Multiply the second equation in Eqs. (7) by $d_{l+1}e_l - g_{l+1}f_l$ and first by f_l, subtract these results to eliminate A_{l+1}, leading to another new equation that relates A_l, B_l, and B_{l+1}. The final system of equations reads:

$$\begin{cases} \overbrace{[d_{l+1}\, e_l - f_l\, g_{l+1}]}^{\alpha_{l1}} A_l + \overbrace{[d_{l+1}\, d_l - g_l\, g_{l+1}]}^{\beta_{l1}} B_l \quad \underbrace{+\overbrace{[e_{l+1}\, g_{l+1} - f_{l+1} d_{l+1}]}^{\gamma_{l1}} A_{l+1} = [C_{l+1}^{\uparrow}(0) - C_l^{\uparrow}(\Delta\tau_l)]\, d_{l+1} + [C_l^{\downarrow}(\Delta\tau_l) - C_{l+1}^{\downarrow}(0)]\, g_{l+1}}_{\chi_{l1}} \\ \overbrace{[d_l\, f_l - e_l\, g_l]}^{\alpha_{l2}} B_l \quad + \overbrace{[e_{l+1}\, e_l - f_l\, f_{l+1}]}^{\beta_{l2}} A_{l+1} \underbrace{+\overbrace{[f_l\, g_{l+1} - e_l\, d_{l+1}]}^{\gamma_{l2}} B_{l+1} = [C_{l+1}^{\uparrow}(0) - C_l^{\uparrow}(\Delta\tau_l)]\, f_l + [C_l^{\downarrow}(\Delta\tau_l) - C_{l+1}^{\downarrow}(0)]\, e_l}_{\chi_{l2}} \end{cases} \tag{9}$$

where the coefficients $\alpha_{l1}, \beta_{l1}, \gamma_{l1}$ and $\alpha_{l2}, \beta_{l2}, \gamma_{l2}$ are derived from the original coefficients e_l, d_l, f_l, g_l and the source terms $C_l^{\uparrow}$, $C_l^{\downarrow}$.

The downward diffuse flux at the top of the multilayer structure is equal to the incident diffuse flux. In addition, the upward flux at the bottom of the multilayer structure is equal to the product of the downward flux and the reflectance of the surface (R_S), i.e. the reflected incident flux at the surface is all diffuse. Hence, boundary conditions at the top and bottom boundaries are as follows:

$$\begin{cases} F_1^{\downarrow}(\tau = 0) & = A_1\{\Gamma_1 \exp[-\lambda_1(\Delta\tau_1)] + 1\} + B_1\{\Gamma_1 \exp[-\lambda_1(\Delta\tau_1)] - 1\} + C_1^{\downarrow}(0) \\ & = \text{Downward diffuse flux} \\ F_N^{\uparrow}(\tau = \Delta\tau_N) & = A_N\{1 + \Gamma_N \exp(-\lambda_N\Delta\tau_N)\} + B_N\{1 - \Gamma_N \exp(-\lambda_N\Delta\tau_N)\} + C_N^{\uparrow}(\Delta\tau_N) \\ & = R_S F_N^{\downarrow}(\tau = \Delta\tau_N) + R_S F_{\odot}(\tau = \Delta\tau_N) \end{cases} \tag{10}$$

We need to solve a tridiagonal system ($2N - 2$ equations) to determine A_l, B_l, A_{l+1}, B_{l+1} [Eqs. (9)] and the related variables such as $F_l^{\downarrow}$ and $F_l^{\uparrow}$. We use the standard method of tridiagonal solving, known as the Thomas algorithm. This algorithm is efficient for solving tridiagonal systems, as it reduces computational complexity compared with general matrix solvers.

Several key parameters need to be precalculated to model RT in vegetation. These are $G(\mu)$, $K(\mu)$, $\bar{\mu}$, $\omega\beta$, and $\omega\beta_0$. The function $G(\mu)$, known as the asymmetry factor of the phase function, is given by [9]:

$$G(\mu) = \frac{1}{2\pi} \int_{\Omega'} \mathrm{d}\hat{\mathbf{\Omega}}'\, g(\theta') h(\phi') \left| \hat{\mathbf{\Omega}} \cdot \hat{\mathbf{\Omega}}' \right|, \tag{11}$$

where $g(\theta)$ and $(1/2\pi)h(\phi)$ are the probability density functions of leaf normal inclination and azimuth, respectively. Several common functions exist for the probability distribution $g(\theta)\sin\theta$. For instance, the function is $\sin\theta$ in the spherical case while the uniform function is $2/\pi$. Once the G function has been determined, the other parameters can be directly derived. These are:

$$\begin{cases} K(\mu) & = \frac{G(\mu)}{\mu} \\ \bar{\mu} & = \int_0^1 \mathrm{d}\mu' \frac{\mu'}{G(\mu')} \\ \omega & = (r + t) \\ \delta & = (r - t) \\ \omega\beta & = \frac{1}{2}\left(\omega + \delta \int_0^{\pi/2} \mathrm{d}\theta \sin\theta \cos^2\theta\, g(\theta)\right) \\ \omega\beta_0 & = \frac{1}{2}\left(\omega + \frac{\mu_0}{G(\mu_0)}\delta \int_0^{\pi/2} \mathrm{d}\theta \sin\theta \cos^2\theta\, g(\theta)\right) \end{cases} \tag{12}$$

where r is the reflectivity and t is the transmissivity of vegetation (leaf). The last two equations in Eqs. (12) represent the back-scattering parameters (β, β_0) for the diffuse and direct beams, respectively, as defined in Ref. [5]. However, these two parameters need to be adjusted for other cases. In Refs. [3, 2, 4], the back-scattering parameter for the diffuse beam is defined as

$$\omega\beta = \frac{1}{2}\left(\omega + \delta\cos^2\Theta\right) \tag{13}$$

where Θ is the mean leaf inclination angle relative to the horizontal surface. The back-scattering parameter for direct beam, β_0 in Refs. [3, 2, 4] reads:

$$\omega\beta_0 = \frac{1 + \bar{\mu}\,K}{\bar{\mu}\,K}\alpha_{ss}(\mu), \tag{14}$$

where $\alpha_{ss}(\mu)$, the single scattering albedo of the canopy, is given by

$$\alpha_{ss}(\mu) = \omega \int_0^1 \mathrm{d}\mu' \frac{\mu'\,G(\mu)}{\mu\,G(\mu') + \mu'\,G(\mu)}. \tag{15}$$

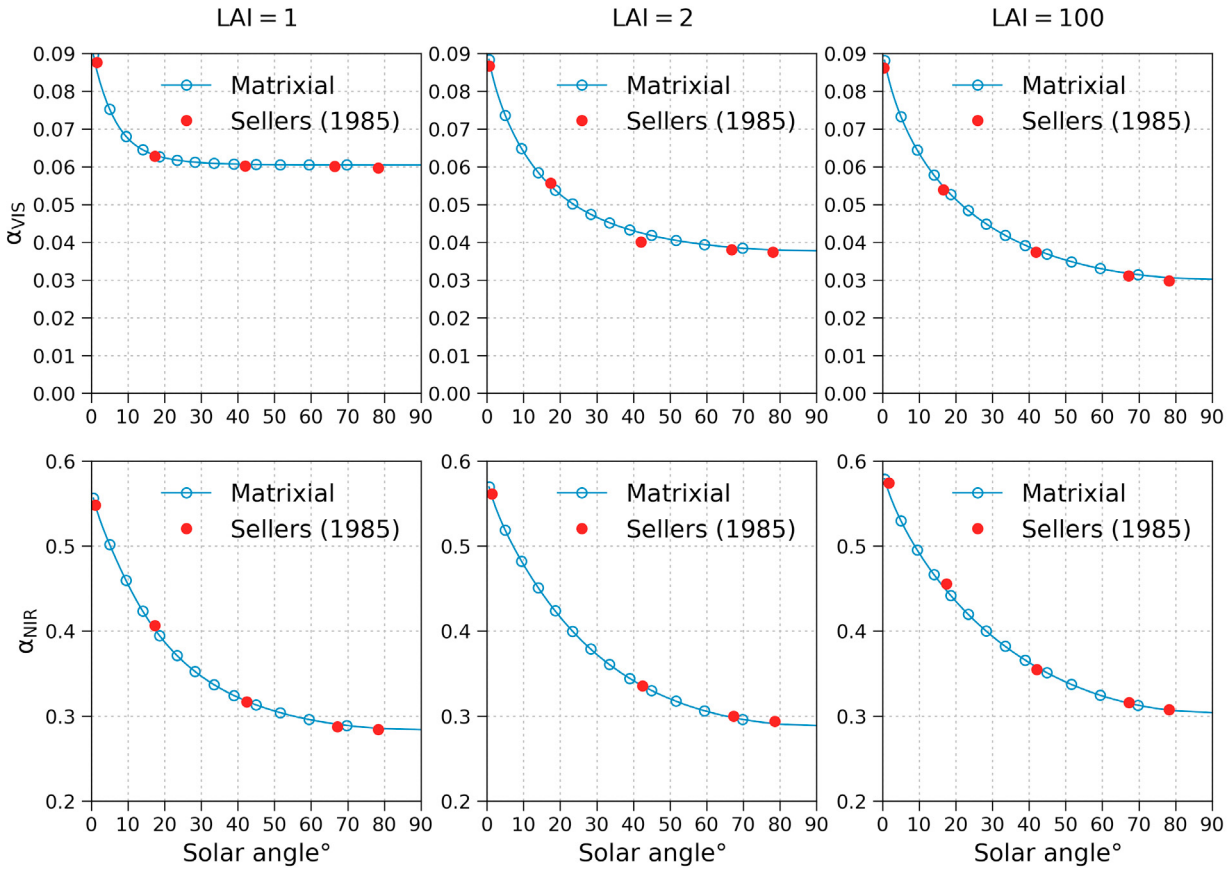


Fig. 2. Comparison of visible albedo (α_{VIS}) and near-infrared albedo (α_{NIR}) as a function of the solar zenith angle for different leaf area index (LAI). The blue circles represent the matrix-based solver results and the red circles represent the results of the model by Ref [2].

3. Verification and validation

The matrix-based RT solver has been verified to ensure its accuracy and reliability. The verification process confirms that the solver produces the same results for single-layer and multilayer configurations, provided the same properties are applied to all layers. The solver was validated to reproduce the results presented in Ref. [8]. Specifically, the results of the solver were compared to the data in Tables (4), (5), and (6) of the referenced article and were found to be in exact agreement. We further evaluated the solver against Ref. [2], Ref. [5], and the multilayer iterative radiation solver in the ORCHIDEE model as presented in Ref. [7].

Fig. 2 shows the comparison between the matrix solver and the canopy reflectance model developed by Ref. [2] for different leaf area index (LAI) values. We note that an LAI value of 100 is excessive and was chosen to represent the infinite case in Ref. [2]. The top row shows the visible albedo (α_{VIS}) while the bottom row shows the near-infrared albedo (α_{NIR}) for solar zenith angles ranging from 0° to 90°. For each LAI value, the matrix solver results (blue circles) are plotted next to the Sellers model results (red circles). The matrix solver tends to follow the general trend of the Sellers model with a maximum relative difference of 8%.

Fig. 3 illustrates the predicted absorption of photosynthetically active radiation (PAR) for a growing wheat canopy, compared between the matrix-based solver and the Sellers model for different solar angles (10°, 60°, and 80°). Absorption percentages are plotted against LAI. At lower solar zenith angles, both models show high absorption rates, which saturate as the LAI increases. At higher solar zenith angles, the absorption decreases with increasing LAI. There is a maximum relative difference of 5% between the Sellers model (red circles) and the matrix solver (blue circles).

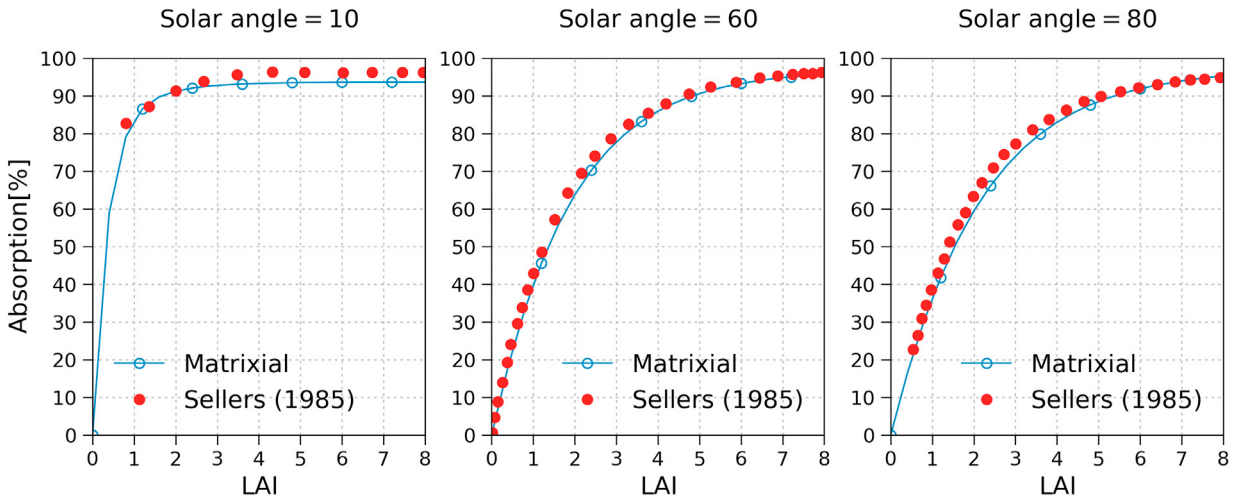


Fig. 3. Predicted absorption of photosynthetically active radiation (PAR) for a growing wheat canopy, compared between the matrix-based solver and Ref. [2] model at solar zenith angles of 10°, 60° and 80°. The blue circles represent the matrix-based solver results while the red circles represent the results of the Ref. [2] model.

Fig. 4 shows a comparison of the directional hemispherical reflectance (DHR) calculated using the matrix-based solver against the results of Ref. [5]. The comparison is performed for different vegetation densities (sparse, medium, dense) and spectral bands (near-infrared, red, snow cover condition in the near-infrared), with each row corresponding to a different vegetation density and each column representing a different spectral band. DHR values are plotted against the solar zenith angle, ranging from 0° to 75°.

In the top row, for sparse vegetation, the near-infrared band shows increasing DHR with the solar zenith angle for both solvers although the matrix solver maximally gives 2% higher DHR at higher angles. For the red band, the DHR decreases with an increasing solar zenith angle, with a maximum difference of 3% between the two approaches. For snow cover, the DHR remains high and relatively constant, with the matrix solver results being at most 2% higher than those in Ref. [5], particularly at lower solar zenith angles.

The middle row, representing the average density of vegetation, shows a similar trend in the near-infrared band with an increase in DHR with a maximum difference of 2% between the two solvers. The red band also shows a decrease in DHR with increasing solar zenith angle, with both methods producing similar results, with a maximum difference of 10% at higher angles. For snow cover, the DHR values are lower than for sparse vegetation but remain relatively stable, with the matrix solver results being at most 5% higher at lower solar zenith angles.

In the bottom row, for dense vegetation, the near-infrared band shows a sharper increase in DHR with solar zenith angle compared to sparse and medium vegetation, with the matrix solver closely matched to Ref. [5], although with a maximum difference of 3% at higher angles. The red band shows an almost stable DHR with increasing solar zenith angle, and the matrix solver results have a maximum difference of 7% relative to those in Ref. [5]. For snow cover, the DHR values also show a maximum difference of 5% between the two solvers.

The matrix-based RT solver has been further compared with the ORCHIDEE iterative solver [7] across all land points, Plant Functional Types (PFTs), spectral bands, and for different simulation durations. For the meteorological forcing data, we used the CRUJRA dataset [10], which provides 6-hourly, gridded data for 1901–2018. This dataset integrates the observational data from the Climatic Research Unit (CRU) with the Japanese Reanalysis (JRA). It includes variables such as temperature, and precipitation at a 0.5° spatial resolution. The comparison was configured following a default ORCHIDEE set-up called FG_CRUJRA_SPIN. This setup uses 2° ×2° CRUJRA forcing, repeatedly cycling over 1901–1910 for a 320-year long-term spinup. It begins from scratch, with 15 PFTs, no land cover changes, fixed nitrogen input (1850, 1860, or 1900 depending on nitrogen species), and CO_2 concentrations fixed to the year 1860. Following spinup the simulation continues with a default transient set-up called FG_CRUJRA_TRANS. FG_CRUJRA_TRANS uses CRUJRA forcing data over 1861–1900, restarting from December 31st of the last year

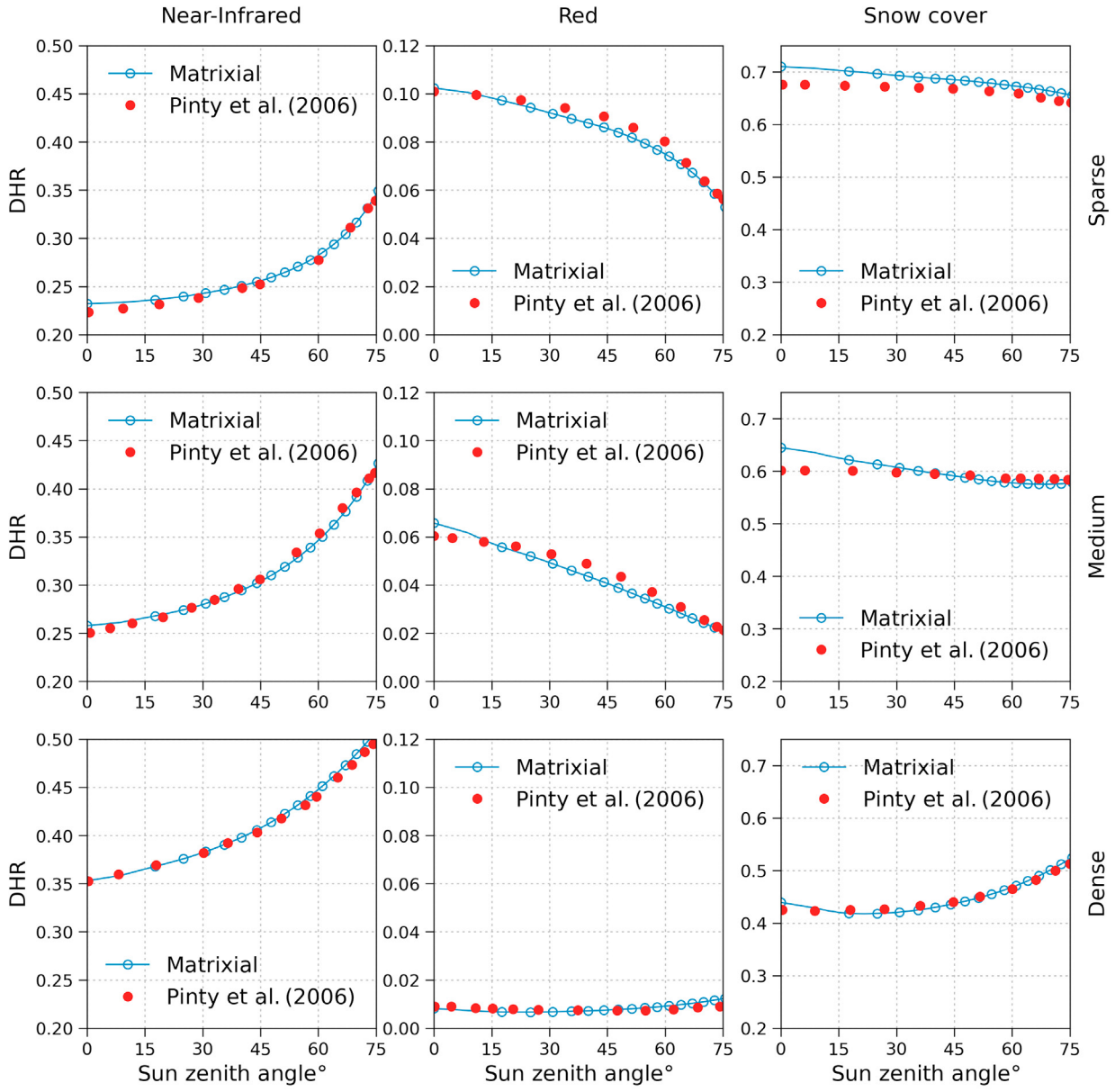


Fig. 4. Comparison of the directional hemispheric reflectance (DHR) for three spectral bands (Near-Infrared, Red, and snow cover condition in the near-infrared) and three vegetation densities (sparse, medium, and dense) across varying solar zenith angles. Blue circles represent results from the matrix-based solver while red circles represent results from Ref. [5].

of FG_CRUJRA_SPIN. It includes 15 PFTs, fixed land cover, nitrogen input for 1900, and annually varying CO_2 concentrations. The simulation is concluded with a historical default run. For ORCHIDEE this set-up is called FG_CRUJRA_HIST and applies the annual CRUJRA forcing from 1901 to 2010. The set-up starts from the last day of the FG_CRUJRA_TRANS, it includes annual updates for land cover, nitrogen input, and CO_2 concentrations, reflecting dynamic historical conditions.

The comparison is based on three main parameters: surface albedo, absorption, and transmission to the ground. Fig. 5 shows α_{VIS} [panels (a) and (b)] and α_{NIR} [panels (c) and (d)], for the iterative (top row) and the matrix solver

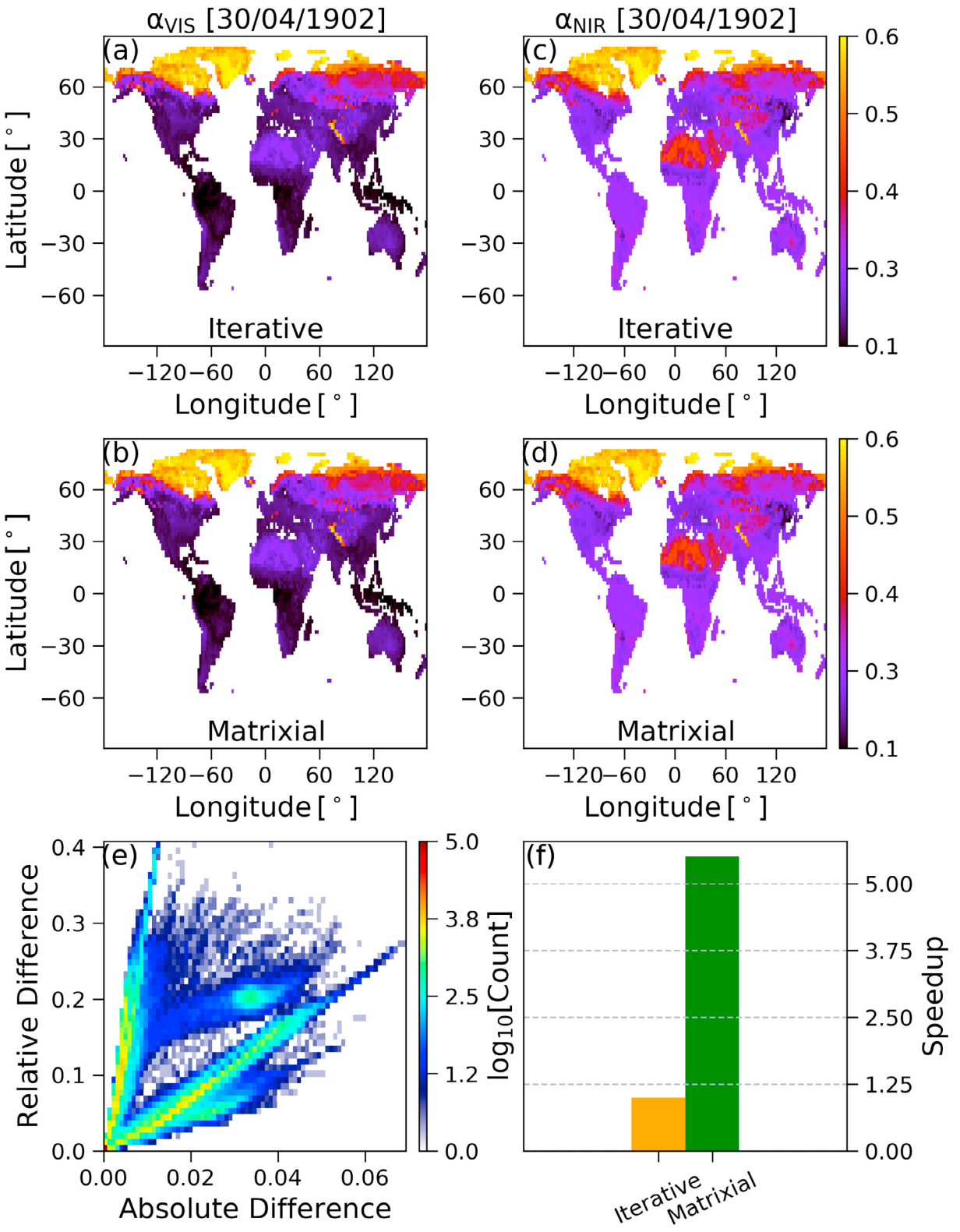


Fig. 5. Visible albedo α_{VIS} [(a) and (b)] and near-infrared albedo α_{NIR} [(c) and (d)], for the iterative solver (top row) and the matrix solver (middle row) at the snapshot of 04/30/1902 for a two-year FG_CRUJRA_HIST simulation, the density plot (logarithm base 10) for the absolute and relative differences between the albedo of the matrix solver and the iterative one on all data points and the full two-year FG_CRUJRA_HIST simulation [(e)], comparison of the computation speed between the two solvers [(f)].

(middle row) at the snapshot time of 30/04/1902 for a two-year FG_CRUJRA_HIST simulation. Both solvers give almost the same results. Fig. 5e shows the density plot of relative and absolute differences for comparisons between the albedos of the two solvers (all land points, all spectral bands, and all PFTs accumulated over two years). The scatterplot highlights the presence of a dense cluster of points near the origin, indicating that in many cases both the absolute and relative differences are small and the two solvers give very similar results. However, there are also distinct bands and peaks where the absolute or relative differences are more pronounced (representing approximately one percent of data points). The behavior of these differences varies depending on the magnitude of the values being compared. The relative difference is important when comparing two very small numbers, while the absolute difference is important when comparing two large numbers.

The tests presented in this study were run on a Leonardo Booster machine and compiled using the NVIDIA NVHPC compiler with the options `-i4 -r8 -O2 -Kieee -Ktrap=fp`, the code was run in a fully MPI setup, and on CPUs. The matrix method is approximately six times faster than the iterative (Fig. 5f). For performance comparison, the RT solver runtime is summed over two years. The matrix-based solver is faster because it calculates the interactions between layers using derived analytical equations, thus avoiding the repeated calculations required in iterative methods.

4. Conclusion

The presented results indicate that the matrix-based solver generally provides results in good agreement with other LSM RT solvers, thus validating its use for modeling under various vegetation covers. The substantial speedup provided by the matrix approach suggests it is advantageous for large-scale simulations, for which computational resources and execution times are critical factors. The results argue for the adoption of the matrix method in the standard version of the LSMs, which could lead to more efficient and faster calculations.

Acknowledgements

We acknowledge CINECA and EuroHPC JU for awarding access to the Leonardo supercomputing hosted at CINECA (project ID: EHPC-DEV-2023D10-042).

References

[1] Boucher, Olivier, et al. “Presentation and evaluation of the IPSL-CM6A-LR climate model.” *Journal of Advances in Modeling Earth Systems* **12**.7 (2020): e2019MS002010. DOI: 10.1029/2019MS002010.

[2] Sellers, P. J. (1985) “Canopy reflectance, photosynthesis and transpiration.” *International Journal of Remote Sensing* **6** (8): 1335–1372. DOI: 10.1080/01431168508948283.

[3] Dickinson, R. E. (1983) “Land surface processes and climate—Surface albedos and energy balance.” *Advances in Geophysics* **25**: 305–353. Elsevier. DOI: 10.1016/S0065-2687(08)60176-4.

[4] Thornton, E. (2010) “Technical Description of version 4.0 of the Community Land Model (CLM).” NCAR, Climate and Global. No DOI available.

[5] Pinty, B., et al. (2006) “Simplifying the interaction of land surfaces with radiation for relating remote sensing products to climate models.” *Journal of Geophysical Research: Atmospheres* **111** (D2). DOI: 10.1029/2005JD005952.

[6] Meador, W. E., and W. R. Weaver. (1980) “Two-stream approximations to radiative transfer in planetary atmospheres: A unified description of existing methods and a new improvement.” *Journal of Atmospheric Sciences* **37** (3): 630–643. DOI: 10.1175/1520-0469(1980)037¡0630¿2.0.CO;2.

[7] McGrath, M. J., Ryder, J., Pinty, B., Otto, J., Naudts, K., Valade, A., et al. (2016) “A multi-level canopy radiative transfer scheme for ORCHIDEE (SVN r2566), based on a domain-averaged structure factor.” *Geoscientific Model Development Discussions* **2016**: 1–22. DOI: 10.5194/gmd-9-3417-2016.

[8] Toon, O. B., et al. (1989) “Rapid calculation of radiative heating rates and photodissociation rates in inhomogeneous multiple scattering atmospheres.” *Journal of Geophysical Research: Atmospheres* **94** (D13): 16287–16301. DOI: 10.1029/JD094iD13p16287.

[9] Myneni, Ranga B., Juhan Ross, and Ghassem Asrar. “A review on the theory of photon transport in leaf canopies.” *Agricultural and Forest Meteorology* **45** (1-2): 1–153. DOI: 10.1016/0168-1923(89)90054-7.

[10] University of East Anglia Climatic Research Unit; Harris, I.C. “CRU JRA: Collection of CRU JRA forcing datasets of gridded land surface blend of Climatic Research Unit (CRU) and Japanese reanalysis (JRA) data.” *Centre for Environmental Data Analysis*, 2019.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 13–22

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

Towards effective continued pre-training of EU institutional LLMs on EuroHPC supercomputers

Ilja Rausch*, Bhavani Bhaskar**, Anna Safont-Andreu, Hans Ewetz, David Kolovratnik, Csaba Oravecz, Markus Runonen

European Commission, Directorate-General for Translation

Abstract

Large Language Models (LLMs) are a significant advancement in artificial intelligence (AI), capable of learning from vast textual datasets and excelling in tasks such as text generation and translation. However, the current general LLMs often do not meet the specific requirements of the public sector and other entities in Europe due to various limitations, including in particular language coverage gaps. In response, the European Commission's Directorate-General for Translation (DGT), in the context of its partnership with the Directorate-General for Communications Networks, Content and Technology (DG CONNECT) under the Digital Europe programme, aims to leverage its high-quality multilingual data coming from all the European Union (EU) institutions to contribute to the European ecosystem of LLMs through continued pre-training of open-source models. This paper presents these ongoing efforts on the supercomputers provided by the European High Performance Computing Joint Undertaking (EuroHPC JU), with a focus on adapting Meta's open-weight LLMs to European linguistic diversity. To this end we leverage the datasets of the European Advanced Multilingual Information System (EURAMIS), a unique and voluminous corpus of multilingual text from all EU institutions. Our approach utilizes state-of-the-art AI tools, including Hugging Face libraries and DeepSpeed's ZeRO-3 data parallelism. We report on the results of our experiments, including the human evaluation of our models and various automated benchmarks such as ARC and HellaSwag, and machine translation tasks. Our findings demonstrate the potential of continued pre-training for enhancing the multilingual capabilities of open source LLMs for Europe.

Keywords: generative artificial intelligence; large language models; distributed computing

1. Introduction

Large Language Models (LLMs) represent a considerable leap in generative artificial intelligence (AI), with their ability to encode, process and generate human language enabling novel applications across various sectors. Their deep

* Corresponding author. Tel.: +32 2 29 86108
** Corresponding author. Tel.: +352 4301 32181
E-mail address: Ilja.RAUSCH@ec.europa.eu, Bhavani.BHASKAR@ext.ec.europa.eu

10.1016/j.procs.2025.02.256

neural networks allow them to learn from extensive textual data, pushing the boundaries of machine capabilities in tasks like text generation, translation, and content creation. The research and development in LLMs is not only crucial for technological advancement but also for addressing ethical considerations and enabling positive societal impact.

While the advent of LLMs has marked a transformative phase in AI, these models, which are typically proprietary, often fall short of the specific needs of the public sector and other entities in Europe. These general LLMs are hampered by a number of limitations, including in particular incomplete language coverage. The European Commission's Directorate-General for Translation (DGT) aims to address that limitation by leveraging its high-quality multilingual data coming from all the European Union (EU) institutions. The initiative is in the context of DGT's partnership with DG CONNECT for AI-based multilingual services under the Digital Europe programme.

To realize this vision, DGT is utilizing continued pre-training of existing open-source LLMs, with the goal of improving their multilingual capabilities. This initiative leverages the European Institutions' high-quality, multilingual internal data. Trained on this data, the resulting models are expected to improve the coverage of all official languages of the EU. DGT's vision is to contribute to the European eco-system of LLMs that demonstrate improved adaptability to the multilingual landscape of the EU.

DGT has a notable track-record of implementing AI-powered multilingual services through its partnership with DG CONNECT under the Digital Europe programme. Such services include the flagship AI project eTranslation for machine translation with transformer models, eSummary for automated multilingual summarization, and eBriefing for drafting support [6]. These services are available to public administrations, SMEs, civil society, academia and other eligible users across the EU. They represent DGT and DG CONNECT's commitment to fostering AI-driven innovation and multilingualism, which was further evidenced by last year's training of specialized models for neural machine translation on MeluXina, one of the EuroHPC supercomputers [19].

This paper outlines DGT's most recent utilization of the EuroHPC supercomputers. It is a feasibility study, with the goals of assessing (1) whether LLMs can be trained with institutional data on the EuroHPC infrastructure, (2) what approaches are successful in improving the training efficiency on the EuroHPC JU hardware and (3) how continued pre-training with low-resource European languages impacts the LLM performance. To answer these questions, we leveraged a EuroHPC Development Access project **EHPC-DEV-2023D09-008** (henceforth, *EHPC1*), which concluded in April 2024. This project successfully implemented continued pre-training of a Llama2 13B model on the MeluXina system [14]. By leveraging the EURAMIS inter-institutional data [13] of Slovenian and Croatian languages, the project delivered improved performance on those languages, as validated during tests with human evaluators.

At the time of writing, we are using a new EuroHPC Development Access project (EHPC-DEV-2024D05-041) to further optimize the hardware utilization to prepare for future scaled-up training runs. In particular, this project will feed into future large-scale EuroHPC projects such as our current EuroHPC AI and Data Intensive Applications project (EHPC-AI-2024A02-026) that started in July 22, 2024. The successful outcome of these and subsequent large-scale EuroHPC projects is expected to lay the groundwork for the creation of LLMs that have better multilingual capabilities in all EU official languages in handling various tasks such as translation, summarization, and document drafting. Training LLMs with comparable capabilities across all its languages is still a difficult challenge, partially due to its significant computational cost. We hope that our work will provide useful insights to training multilingual LLMs on EuroHPC infrastructure and help the scientific and engineering community focus on efficient configurations and avoid costly roadblocks.

The remainder of this paper is organized as follows. The next section provides a brief overview of the related work, Section 3 will outline the approaches and technologies we used during the EHPC1 project. Subsequently, Section 4 will provide insights from our scaling efforts and the evaluation results after continued pre-training of Llama2 13B. The paper will conclude in Section 5.

2. Related work

LLMs represent a frontier in AI research. Our feasibility study aimed to harness the state-of-the-art in AI by employing an advanced LLM developed by Meta, which has been pre-trained on an expansive dataset comprising 2 trillion tokens. Our project distinguishes itself from existing work by integrating the EURAMIS dataset into the LLM. This corpus, largely inaccessible to the public, augments the LLM with a data source that is both unique and voluminous, representing a pioneering initiative to enhance a top-tier LLM with such a multilingual dataset.

In the context of European LLM development, several notable initiatives have emerged, including Tower [28], Viking [23], TrustLLM [27], and collaborations with High Performance Language Technologies (HPLT) [5]. While some of these projects resulted in pretrained LLMs like Tower [28] and Viking [23], they typically concentrate on a limited set of up to ten EU languages. Furthermore, the linguistic diversity within the HPLT dataset, although spanning 75 languages, presents a significant disparity in token counts for lesser-resourced official EU languages, which are several magnitudes lower compared to our EURAMIS dataset. Although EuroLingua-GPT [2] encompasses 45 languages, its potential impact remains uncertain until its training completes, presumably by mid-2025. Lastly, Mistral open-sourced several models, which are effective mainly in high-resource languages [17]. Recently, more independent teams have reported training small-scale language models that reinforce an emerging trend towards building generative AI with a focus on European languages [11, 15].

In juxtaposition, the landscape of US-based proprietary models is dominated by OpenAI's GPT versions, among the closed-source models, and Meta's Llama3 among the open-weights models. While these technologies are highly competitive overall, preliminary evaluations suggest that Llama3 exhibits suboptimal performance on key tasks such as summarization and translation [7]. Although larger iterations, like the Llama3.1 405B model, demonstrate enhanced capabilities, their practical application is limited due to their size, which poses significant challenges to local or on-premise deployment. Additionally, most proprietary models, including the leading GPT models from OpenAI, face difficulties due to copyright issues, suspected biases, proprietary restrictions, and opaque practices concerning the curation of their training datasets.

Our approach is unique in its comprehensive inclusion of all 24 official EU languages. Each low-resource language is represented with at least one billion tokens. The corpus is meticulously curated to adhere to stringent quality standards, while being devoid of any copyright infringements and aligned with core European values.

In contrast to the prevalent top-down methodology employed in related work, which involves processing publicly available large corpora through automated steps such as cleaning, deduplication, and filtering, this project adopts a bottom-up approach. The EURAMIS texts have been painstakingly constructed from the ground up, benefiting from the work of professional translators and editors who have significantly contributed to the corpus curation process for over two decades. This proximity to language professionals and their direct feedback is a notable advantage in terms of data preparation and model evaluation, setting our study apart within the domain of multilingual LLMs.

3. Methodology

Our study revolves around the continued pre-training of Meta's open-weights LLMs. In EHPC1, we continued the pre-training of Meta's Llama2 13B first on 3.2 billion tokens of Slovenian language and, subsequently, on 2.2 billion tokens of Croatian language. As the name suggests, continued pre-training resumes the model improvement on previously unseen data such that all model parameters can be updated during the backward pass. Thus, it requires more compute resources than fine-tuning where typically a significant proportion of the weights is frozen. Fine-tuning is usually applied to specialize the model to a specific task or domain on specific data. This approach often narrows the model's scope, while continued pre-training updates the entire model, addressing a wider range of purposes, which is the intention behind our project.

However, continued pre-training leverages the foundation established by prior training, often performed by another team. The datasets and methods utilized in this earlier phase are not always perfectly transparent, potentially leading to unforeseen effects in downstream results. Such effects may include the inadvertent incorporation of biases or other artifacts from the original training. While training from scratch might alleviate these concerns, it demands extensive datasets and computational resources that are typically beyond the reach of most AI teams and organizations.

3.1. Project overview and compute power consumption

While the EHPC1 project focused on the continued pre-training of the Llama3 13B model on the MeluXina supercomputer, the node hours were utilized for several additional purposes. The EHPC1 project was a Development Access project stretching from November 2023 to May 2024.

The first few node hours were used to set up the model training script, the data processing scripts, the `singularity` container and the launch script. We ensured that the training can be started and distributed over multiple GPUs and

multiple nodes, and that part of the computation can be offloaded to the CPU. However, most of the first-month quota was used to adjust the parallelization and distribution parameters, in order to increase training efficiency. This included launching dozens of test runs to profile the impact of the micro-batch size and the number of gradient accumulation steps, and to improve our understanding of the VRAM consumption. Many of our initial runs terminated prematurely with out-of-memory errors. These initial explorative efforts were done in a non-systematic trial-and-error manner. The hyperparameter tuning continued into the second month (December 2023), where the training efficiency was improved by an order of magnitude, from 0.2 samples per second (samples/s) to ca. 2.3 samples/s. These efforts were not as computationally demanding as a model training running in full swing. The floating point operations per second (FLOPS) could be estimated at half of what we achieved during computationally stable months (e.g., month 3 and 4).

Upon achieving a sufficient training efficiency, the largest portion of the second month's quota was spent on the continued pre-training of Llama2 13B on Slovenian data. This pre-training continued into the third and fourth months (January and February 2024). Throughout the entire training process, model checkpoints were regularly saved. Subsequently, we launched the second training phase, which was the continued pre-training of our latest model checkpoint on the Croatian language data. This second training concluded in the fifth month (March 2024).

The automated benchmarking of our latest checkpoints, executed outside of the supercomputer, indicated catastrophic forgetting as the model's performance on English texts has deteriorated compared to the original Llama2 13B model. Consequently, to mitigate catastrophic forgetting, we launched a third continued pre-training phase in March 2024, this time on English language data. This training stretched into the final month (April 2024). The rest of this month's quota was used to conduct additional training efficiency analysis, profiling and optimization. In this month, we ran sporadic scaling analysis and set up Graphics Processing Unit (GPU) memory profiling to obtain a clearer picture of our hardware utilization and improve our understanding of the impact of DeepSpeed's hyperparameters.

The monthly node hour and power consumption are shown in Tab. 1. The latter was estimated based on the required or measured FLOPS and MeluXina's energy efficiency [24].

Table 1: Node hours and power consumed by month during the EHPC1 project.

Month	Nov 2023	Dec 2023	Jan 2024	Feb 2024	Mar 2024	Apr 2024
Node hours	500	500	540	650	500	500
Power in kWh	723	1447	1562	1881	1447	723
Mean tera FLOPS per GPU	9.75	19.5	19.5	19.5	19.5	9.75

3.2. Implementation details

3.2.1. Parallelization approaches

A crucial part of our work is the tuning of software configuration to the underlying hardware. In EHPC1, we worked on the Accelerator modules of the MeluXina supercomputer, which were based on the Atos Bullsequana XH2000 architecture [1, 14]. This architecture contains A100 NVIDIA GPUs with 40 GB VRAM. The main challenge is the effective distribution of the computational load such that the batched LLM training is parallelized and all necessary components fit into memory, including the model weights, the optimizer states, the activations and the gradients.

To this end, we used the DeepSpeed Zero Redundancy Optimizer (ZeRO) and other tools from DeepSpeed's state-of-the-art open-source software suite [21]. In particular, we used the DeepSpeed plugin [9] that was part of the Hugging Face `transformers` module. ZeRO aims to eliminate memory redundancies in data- and model-parallel training, without drastically increasing communication volume. Crucially, this enables scaling the model size proportionally to the number of devices (GPUs) [21]. We experimented with various stages and configuration hyperparameters of ZeRO and opted for stage 3 with CPU offloading of the optimizer states. We also experimented with various sharding strategies with Hugging Face `transformers` plugin of PyTorch's FSDP [31] and decided not to pursue it further because it did not improve on DeepSpeed stage 3 performance. Moreover, the hyperparameter space of FSDP was smaller compared to DeepSpeed, offering less fine-grained customization. Utilizing FlashAttention-2 [4] provided the biggest increase in efficiency and doubled the micro-batch size that could be fit in the GPU memory.

3.2.2. Scaling and profiling

A gradual model scaling approach was applied to measure the hardware utilization and training efficiency. This stage did not involve the LLAMA2 weights but only the model architecture. This enabled experiments with untrained models of any size. For example, it was cost-effective to use a model with under 70 MB for rapid debugging. This approach was facilitated by HUGGING FACE's `transformers` library, which allowed easy control of the model size by setting architectural components such as the context length, the hidden embedding size, the number of hidden layers and the number of attention heads.

We ran several ablation training iterations, each with under ten training steps, to profile the GPU memory utilization and FLOPS per GPU. Admittedly, the first hundreds or even thousands of training steps are often used as a warm up, while the training performance measurements are more reliable during the later stable phase. Nonetheless, with additional preliminary tests we found that the mid-phase average computational efficiency did not differ critically from the initial stages of training.

3.2.3. Data preparation

The data for our study comes from the EURAMIS repository [13]. EURAMIS is an inter-institutional set of databases that contains multilingual text from translations produced by all EU institutions. The EURAMIS databases contain segments that are full sentences, phrases, or single words, all labelled based on their language. Most segments have at least one translation in another language. Many segments have translations in multiple languages, in many cases in all official EU languages.

Upon extraction from EURAMIS, the segments were combined into text documents. Segments were typically smaller than the LLAMA2 (4096 tokens) context windows. The data samples were created by iterative concatenation of segments. A sample already containing one or more segments could be unable to fit another new segment within the rest of its context window. In that case, the new segment was moved into a new sample.

3.2.4. Model training configuration

We ran the continued pre-training on eight MELUXINA Accelerator nodes, with four A100 40 GB NVIDIA GPUs each, and 32 CPU cores in total. We chose a gradient accumulation of 8, a micro-batch size of 9 and a warmup ratio of 0.1. Otherwise, we selected the same hyperparameters as the original LLAMA2 13B training [26]. We trained with bf16 precision as advised by the authors of the LLAMA2 13B paper. No instruction fine-tuning, preference optimization or alignment based on human feedback was used, other than what was already applied to LLAMA2 13B by META [26].

3.2.5. Evaluation methods

The training progress was monitored using the cross-entropy loss. However, due to the abstract nature of this metric, it was only useful for seeing the trend of the training progress. For a more rigorous assessment of the models performance, we applied more sophisticated benchmarking approaches, which were divided in three main phases. First, the models were tested using standardized benchmarks [8] that we previously translated to our target languages using the eTranslation translation engines [6]. Second, the models were tested on samples of EURAMIS data in a machine translation task. Third, evaluation was performed by human professionals (mainly translators and editors) within DGT and by other native speakers from other parts of the European Commission and other EU institutions, bodies and agencies, comparing the results between our models. The models that were evaluated included: (1) **LLAMA2 - *X*B**: versions of HUGGING FACE's LLAMA2 model with *X* billion parameters (7, 13, 70). (2) **SL-MODEL**: the model resulting from continued pre-training of the original LLAMA2 13B on Slovenian language data. (3) **HR-SL-MODEL**: the model resulting from continued pre-training of the latest SL-MODEL checkpoint on Croatian language data. (4) **BALANCE-MODEL**: the model resulting from continued pre-training of the latest HR-SL-MODEL checkpoint on English data.

4. Results and Discussion

4.1. Hardware utilization

The continued pre-training of LLAMA2 13B on Slovenian language ran on 3.2B tokens, 354 steps, within ca. 125 wall-time hours, or 4000 GPU hours. The continued pre-training on Croatian language ran on 2.2B tokens, 227 steps,

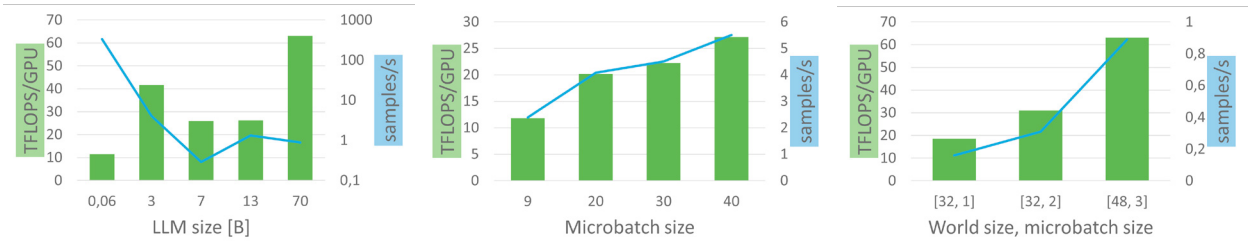


Fig. 1: Highest achieved TFLOPS per GPU (bars) and samples per second (lines), averaged over multiple training steps. (left): Best achieved results over LLM sizes (in billions of parameters) (with variable context lengths and world sizes); (center): Results for the 3B model (context length 1024 and world size 4); (right) Results for the 70B model (context length 4096).

within ca. 84 wall-time hours, or 2688 GPU hours. In both of these training phases, we achieved on average around 19.5 tera FLOPS (TFLOPS) per GPU and a training speed of ca. 225 tokens/second/GPU (or 1143 seconds per step) with 32 A100 40 GB NVIDIA GPUs. For comparison, Meta reported to have pre-trained Llama2 13B on 2T tokens within 368640 GPU hours on A100 80 GB NVIDIA GPUs [26]. The exact number of GPUs used by Meta to pre-train Llama2 13B is not reported but it could be two orders of magnitude higher than what was available to us, based on scattered information [25, 26, 12].

The GPU memory capacity (40 GB) and the count of available GPUs (maximum 32 per task) presented a challenge to the efficient load distribution. At mixed precision, the Llama2 13B training requires at least 208 GB for the weights, gradients and optimizer states [21]. Furthermore, additional memory is needed for the activations, temporary buffers, memory fragmentation and other transient computational components. To achieve batched training, which increases memory requirements, we used CPU offloading. However, the data transfer between the CPUs and the GPUs increases latency, and the CPU computations are significantly inferior to GPU in terms of speed for machine learning loads.

To obtain a clearer understanding of the model size impact on the training efficiency on the EuroHPC hardware, we ran scalability tests for different sizes of the Llama2 model architecture. As the EHPC1 development access provided a limited number of node hours, which were distributed across multiple questions of our feasibility study, the scope of the scalability tests was rather limited. Therefore we focused on few targeted tests, instead of applying a comprehensive grid-search approach over the numerous hyperparameters that can contribute to the model scalability. To study the GPU memory usage and training speed (in terms of FLOPS), we altered the model size, the context length and the micro-batch size. Fig. 1 displays the scalability test results. Note that Fig. 1 depicts what we were able to achieve and not what was achievable. Fig. 1 (center) and (right) show an expected trend: the higher the batch size, the more samples/s are processed during the training. However, a larger model not only implies a higher number of floating point operations (due to bigger tensors and heavier computation) but also a higher data transfer. Collective operations and communication (e.g., gather, reduce, scatter, broadcast) between processing units can significantly slow down training speed when the data transfer is inefficient or close to the bandwidth limit. The dip in 7B's samples/s is due to the relatively low macro-batch size. Only 4 GPUs and a micro-batch size of 1 were used in that case. For the 70B model, we achieved a computational efficiency of ca. 63 TFLOPS per GPU, i.e., ca. 20% of the A100 NVIDIA GPU's maximum. Due to the memory limitations, we ran these experiments with a rather small micro-batch size of 3. For this micro-batch size, our achieved computational efficiency is in line with other state-of-the-art studies [18]. Moreover, the 20% ratio is close to what Meta reported for its latest Llama3 training with 16k H100 NVIDIA GPUs [16].

Most parallelization techniques increase the data transfer between the processing units (GPUs) due to the load distribution, with the goal of fitting large models onto limited memory. As the details of such trade-offs are highly specific to the underlying High Performance Computing (HPC) system, we conducted a few additional experimental training runs and logged the communication patterns using DeepSpeed's built-in profiler. The training speed was measured in terms of samples/s on tiny models, as opposed to large models, because (1) small models are faster and more cost-efficient (in terms of GPU hours) and (2) large models do not fit onto single GPUs, making it impossible to execute control runs where no data is transferred between processing units. The goal was to gain qualitative and not quantitative insights. The results are shown in Tab. 2.

Based on the measurements, several patterns emerged: (1) increasing the model size decreases samples/s but not significantly (by a factor less than two); (2) increasing the batch size considerably increases samples/s; (3) increasing the GPU count from 1 to 4 decreases samples/s despite the higher macro-batch size (i.e., micro-batch size times the

Table 2: Training speed results for different distribution parameters.

Model size [MB]	62	62	62	62	62	62	204	204	204	204
GPU count	1	1	2	3	4	4	1	1	4	4
Micro-batch size	9	80	80	80	9	80	9	80	9	80
samples/s (on average)	24	175	32	36	4	39	16	104	3	38

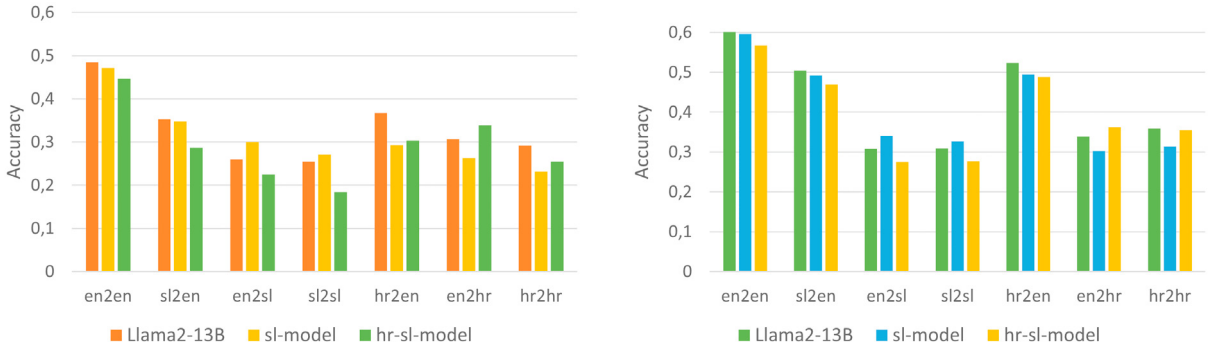


Fig. 2: Accuracy results from ARC (left) and HellaSwag (right) benchmarks for Slovenian and Croatian languages.

GPU count). Additional tests suggest that the DeepSpeed ZeRO stage [21] is an important hyperparameter. Changing from DeepSpeed ZeRO stage 3 to stage 2 drastically increased samples/s from 32 to approx. 241 (with a model size 62 MB, micro-batch size 80 and 2 GPUs). Changing to DeepSpeed ZeRO stage 1 further significantly increased samples/s to approx. 345 (with a model size 62 MB, micro-batch size 80 and 2 GPUs). DeepSpeed ZeRO stage 3 is useful when the GPU memory is scarce, but a lower DeepSpeed ZeRO stage is preferable when GPU memory is not a bottleneck.

4.2. Evaluation results

The evaluation loss reduced steadily during training and approached 0.9 and 0.8 for Slovenian and Croatian, respectively. Given the linguistic proximity of the two languages, it is possible that the Croatian model benefited from the previous pre-training on Slovenian. However, for a more in-depth understanding of the model performance, we discuss below the results from more sophisticated evaluation approaches.

4.2.1. Standardized benchmarks

Various LLM benchmarking tools are openly available through repositories such as Hugging Face and GitHub. For the evaluation of our models we selected two of these benchmarks, ARC [3], a multiple-choice dataset of scientific questions, and HellaSwag [30], a sentence completion dataset. Each of these benchmarks were used in four different ways, depending on the language of both the question and the answer: English for both the questions and the answers (en2en), English only for either the question or the answer while the other was in the target language (en2xx or xx2en) and finally an approach with both questions and answers in our target language (xx2xx).

Fig. 2 shows the ARC and HellaSwag benchmark results for the Slovenian and Croatian languages, measured in accuracy. The results are consistent in both benchmarks and for both languages. The models' performance showed a decrease in accuracy when the benchmarks were translated. However, both ARC and HellaSwag evaluations show that the sl-model slightly outperformed the rest in its respective target language, i.e. for xx2sl, while hr-sl-model performed best only for en2hr.

4.2.2. Machine Translation

The next evaluation task consisted of a segment-based translation task. Here, segments similar to the input segment (in English) and their respective translations (in Slovenian and Croatian) were extracted from the EURAMIS database and added as translation examples to the prompt context. This triggers in-context learning such that the model pro-

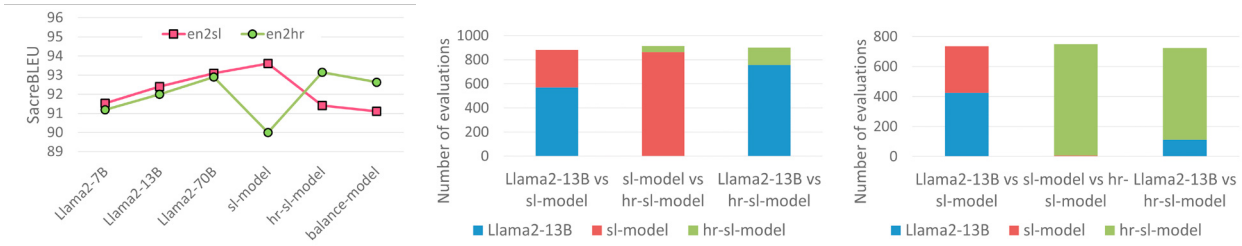


Fig. 3: Results of the different LLAMA2 models in a segment-based translation task (left). Human preferences for text completion output by LLMs in pair comparison, for Slovenian (centre) and Croatian (right) text input.

duced a translation in line with the provided examples. The results were measured using the SacreBLEU score [20], a score that measures the similarity of terms between the model output and the reference translation.

Fig. 3 (left) compares the translation performance across models. While the results increase steadily with the LLAMA2 model size, the best results are obtained by our SL-MODEL and HR-SL-MODEL for their respective target language. Note that they outperform even the bigger LLAMA2 70B on this task, showcasing the importance of continued pre-training in the target languages for translation. The BALANCE-MODEL performed worse as its training language (English) does not correspond to the target languages.

4.2.3. Human evaluation

For the human evaluation, outputs from two out of the three candidate models - LLAMA2 13B, SL-MODEL and HR-SL-MODEL - were presented to the evaluators. The output of each model was text completion from a prompt of about seven words. The evaluators did not know which two models were competing and were asked to choose their preferred output. Each sample was evaluated by two different evaluators. The language of the input text was always either Slovenian or Croatian and it always corresponded to the evaluator's native language. The volunteers were also asked to evaluate the text *relevance* (of the output in relation to the input) and *fluency* (of the output independently of the input) of each completion on a discrete scale from 0 to 5.

Table 3: Human evaluation of relevance and fluency of LLAMA2 13B, SL-MODEL and HR-SL-MODEL - averaged over the total number of trials.

	Input in Slovenian			Input in Croatian		
Model	LLAMA2 13B	SL-MODEL	HR-SL-MODEL	LLAMA2 13B	SL-MODEL	HR-SL-MODEL
Relevance	2.69	**2.79**	2.03	2.57	1.31	**4.26**
Fluency	2.40	**2.64**	**2.64**	2.41	1.00	**4.06**

Fig. 3 (centre) shows the preference count for Slovenian and Fig. 3 (right) for Croatian inputs. Both graphs reflect an underlying issue with LLMs known as catastrophic forgetting [22, 29], where the performance of any model drops as soon as said model is further trained for a different task - in our case, a different (even if related) language. Thus, Fig. 3 (centre), LLAMA2 13B and the SL-MODEL are heavily preferred to the HR-SL-MODEL when generating text in Slovenian, even when the latter has been trained in Slovenian before Croatian. Similarly, in Fig. 3 (right), although the HR-SL-MODEL appears as the preferred language, the second best performing model is still LLAMA2 13B.

These results are backed by their evaluation of the individual outputs in terms of *relevance* and *fluency*, see Tab. , where independently of the language, the LLAMA2 13B model scores always in second place after the model whose latest training corresponds to the input language.

The results of all three evaluations show consistent behaviour of both models. While the SL-MODEL performs best in tasks in Slovenian, the HR-SL-MODEL performs best in tasks in Croatian, as we expected from the training approach. It can be observed through all the experiments in Slovenian that better results are obtained from LLAMA2 13B, a model barely trained on that language, than from our HR-SL-MODEL that was extensively trained in Slovenian text before being later fine-tuned in Croatian. This presents a challenge to our current and future projects where we scale from Slovenian and Croatian to all 24 official EU languages.

5. Conclusion and Outlook

Our experiments demonstrate the feasibility of training LLMs on the EuroHPC infrastructure. We successfully continued pre-training the Llama2 13B model on Slovenian and Croatian languages, achieving competitive training speeds compared to industry reports. Our scalability tests provide valuable insights into the impact of model size, context length, micro-batch size and other hyperparameters on training efficiency. Furthermore, we identify the importance of collective operations and communication patterns in parallelizing large models, confirming the trade-off between load distribution and data transfer bandwidth.

The evaluation results of our Llama2 13B - Slovenian and Croatian - models on standardized benchmarks, machine translation tasks, and human evaluation demonstrate the potential of continued pre-training for improving LLM's capturing of linguistic nuances. While our models' performance was overall mixed, they outperformed the Llama2 13B base model in several tasks in their respective target languages. However, the performance improvements were rather modest in automated benchmarking tests and, in several cases, our models were inferior to the base model. We believe that the worsening performance can be attributed to catastrophic forgetting. The human evaluation results also highlight this behaviour, as the model performance in Slovenian dropped considerably when it was further trained on Croatian. To mitigate catastrophic forgetting, we consider implementing a more sophisticated data sampling for future trainings. One possible approach is chaining documents from different languages in a cyclical manner, where a document is drawn randomly from each language but with the constraint that every full training cycle must see each language at least once. Other promising techniques include replay, re-decaying and re-warming [10].

This study contributes to the development of LLMs better covering all EU official languages. Our findings are of particular importance for future LLM training on the EuroHPC infrastructure as it highlights several practical insights into the challenges and solutions. These insights will feed into our future goals to continue pre-training European open-sourced foundational models on all official EU languages and, potentially, deliver niche EU institutional LLMs using EuroHPC supercomputers. For instance, one of our ongoing projects includes the preparation for the continued pre-training of a Mixtral 8x22B model [17], available under the Apache 2.0 license. Moreover, we welcome future work on trustworthy AI, including further legal and ethical considerations of training LLMs within the EU and beyond.

Acknowledgements

The authors are grateful to all team members who helped realize this project and provided support from various angles. In particular, the authors acknowledge the support from Mihai Cristian Brasoveanu, Markus Foti, Andreas Eisele and Carolina Oliveira Costa. Furthermore, the authors greatly appreciate the 60 colleagues who volunteered to evaluate the output of our pre-trained models. The bulk of the model training was performed on the MeluXina supercomputer in Luxembourg, operated by LuxProvide and funded by the EuroHPC JU and by the national government and other entities in Luxembourg. The authors gratefully acknowledge the support received from both the EuroHPC team and the LuxProvide team.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used GPT-4 by Azure OpenAI in order to seek inspiration and ideas for the transformation of the authors' thoughts into text. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

[1] Atos, 2020. Bullsequana xh2000. URL: https://atos.net/wp-content/uploads/2020/07/BullSequanaXH2000_Features_Atos_supercomputers.pdf. Accessed: July, 2024.

[2] Berkler, K., Silke, L., Fraunhofer IAIS, 2024. Press release — breakthrough for generative ai research in germany and europe. URL: https://www.iais.fraunhofer.de/en/press/press-release-240516.html. Accessed: July, 2024.

[3] Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., Tafjord, O., 2018. Think you have solved question answering? try arc, the AI2 reasoning challenge. CoRR abs/1803.05457. URL: http://arxiv.org/abs/1803.05457, arXiv:1803.05457.

[4] Dao, T., 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. URL: https://arxiv.org/abs/2307.08691, arXiv:2307.08691.

[5] De Gibert, O., Nail, G., Arefyev, N., Bañón, M., Van Der Linde, J., Ji, S., Zaragoza-Bernabeu, J., Aulamo, M., Ramírez-Sánchez, G., Kutuzov, A., et al., 2024. A new massive multilingual dataset for high-performance language technologies. arXiv preprint arXiv:2403.14009 URL: https://arxiv.org/abs/2403.14009.

[6] Directorate-General for Translation, 2024. Digital europe — ai-based multilingual services. URL: https://language-tools.ec.europa.eu/. Accessed: July, 2024.

[7] Dunlap, L., Frick, E., Li, T., Ong, I., Gonzalez, J.E., Chiang, W.L., 2024. What's up with llama 3? arena data analysis. URL: https://lmsys.org/blog/2024-05-08-llama3/. Accessed: July, 2024.

[8] Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., Zou, A., 2023. A framework for few-shot language model evaluation. URL: https://zenodo.org/records/10256836, doi:10.5281/zenodo.10256836.

[9] Hugging Face, 2024. Deepspeed. URL: https://huggingface.co/docs/transformers/main_classes/deepspeed. Accessed: August, 2024.

[10] Ibrahim, A., Thérien, B., Gupta, K., Richter, M.L., Anthony, Q., Lesort, T., Belilovsky, E., Rish, I., 2024. Simple and scalable strategies to continually pre-train large language models. arXiv preprint arXiv:2403.08763 .

[11] Ji, S., Li, Z., Paul, I., Paavola, J., Lin, P., Chen, P., O'Brien, D., Luo, H., Schütze, H., Tiedemann, J., Haddow, B., 2024. EMMA-500: Enhancing massively multilingual adaptation of large language models. arXiv preprint 2409.17892 URL: https://arxiv.org/abs/2409.17892.

[12] Lee, K., Sengupta, S., 2022. Introducing the ai research supercluster — meta's cutting-edge ai supercomputer for ai research. URL: https://ai.facebook.com/blog/ai-rsc/. Accessed: July, 2024.

[13] Leick, J.M., 1995. Euramis: Integrated multilingual services for a large multilingual community, in: Proceedings of Machine Translation Summit V. URL: https://aclanthology.org/1995.mtsummit-1.15.pdf.

[14] LuxProvide, 2024. System overview - meluxina user documentation. URL: https://docs.lxp.lu/system/overview/. Accessed: July, 2024.

[15] Martins, P.H., Fernandes, P., Alves, J., Guerreiro, N.M., Rei, R., Alves, D.M., Pombal, J., Farajian, A., Faysse, M., Klimaszewski, M., et al., 2024. Eurollm: Multilingual language models for europe. arXiv preprint arXiv:2409.16235 .

[16] Meta, 2024. Introducing meta llama 3: The most capable openly available llm to date. URL: https://ai.meta.com/blog/meta-llama-3/. Accessed: July, 2024.

[17] Mistral AI, 2024. Models — mistral ai large language models. URL: https://docs.mistral.ai/getting-started/models/. Accessed: July, 2024.

[18] Narayanan, D., Shoeybi, M., Casper, J., LeGresley, P., Patwary, M., Korthikanti, V., Vainbrand, D., Kashinkunti, P., Bernauer, J., Catanzaro, B., et al., 2021. Efficient large-scale language model training on gpu clusters using megatron-lm, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–15. URL: https://dl.acm.org/doi/abs/10.1145/3458817.3476209.

[19] Oravecz, C., Bhaskar, B., Bontcheva, K., Kovachev, B., 2024. Building high capacity machine translation models for knowledge distillation and production workflows, in: Proceedings of the 20th Conference on Hungarian Computational Linguistics, pp. 97–114.

[20] Post, M., 2018. A call for clarity in reporting BLEU scores. CoRR abs/1804.08771. URL: http://arxiv.org/abs/1804.08771, arXiv:1804.08771.

[21] Rajbhandari, S., Rasley, J., Ruwase, O., He, Y., 2020. Zero: Memory optimizations toward training trillion parameter models, in: SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE. pp. 1–16. URL: https://doi.org/10.1109/SC41405.2020.00024.

[22] Ratcliff, R., 1990. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. Psychological review 97, 285. URL: https://psycnet.apa.org/buy/1990-18992-001.

[23] Silo AI, 2024. Viking 7b/13b/33b: Sailing the nordic seas of multilinguality. URL: https://www.silo.ai/blog/viking-7b-13b-33b-sailing-the-nordic-seas-of-multilinguality. Accessed: July, 2024.

[24] Strohmaier, E., Dongarra, J., Simon, H., Meuer, M., 2024. Green500 list - june 2024. URL: https://www.top500.org/lists/green500/list/2024/06/. Accessed: August, 2024.

[25] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al., 2023a. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 URL: https://arxiv.org/abs/2302.13971.

[26] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al., 2023b. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 URL: https://arxiv.org/abs/2307.09288.

[27] TrustLLM, 2024. Home — trustllm. URL: https://trustllm.eu/. Accessed: July, 2024.

[28] Unbabel Research Team, 2024. Announcing tower: an open multilingual llm for translation-related tasks. URL: https://unbabel.com/announcing-tower-an-open-multilingual-llm-for-translation-related-tasks/. Accessed: July, 2024.

[29] van de Ven, G.M., Soures, N., Kudithipudi, D., 2024. Continual learning and catastrophic forgetting. arXiv preprint arXiv:2403.05175 URL: https://arxiv.org/abs/2403.05175.

[30] Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., Choi, Y., 2019. Hellaswag: Can a machine really finish your sentence? CoRR abs/1905.07830. URL: http://arxiv.org/abs/1905.07830, arXiv:1905.07830.

[31] Zhao, Y., Gu, A., Varma, R., Luo, L., Huang, C.C., Xu, M., Wright, L., Shojanazeri, H., Ott, M., Shleifer, S., Desmaison, A., Balioglu, C., Damania, P., Nguyen, B., Chauhan, G., Hao, Y., Mathews, A., Li, S., 2023. Pytorch fsdp: Experiences on scaling fully sharded data parallel. URL: https://arxiv.org/abs/2304.11277, arXiv:2304.11277.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 23–32

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

Enhancing Performance of High-Speed Engineering Flow Computations: The URANOS Case Study

Francesco De Vanna[a,*]

[a]*Department of Industrial Engineering, Università degli Studi di Padova, Padua 35131, Via Venezia 1, Italy*

Abstract

This paper aims to discuss efforts to enhance the performance of the in-house developed Computational Fluid Dynamics (CFD) solver URANOS. In particular, URANOS-2.0 is presented, an evolution of the 2023 solver release [7], as optimized for pre-exascale architectures. As contemporary European HPC facilities within the current EuroHPC JU panorama utilize distinct GPU architectures—primarily AMD and NVIDIA—URANOS-2.0 adopts the OpenACC standard for portability. The latest release, resulting from several tuning and refactoring efforts, demonstrates excellent multi-GPU scalability, achieving strong scaling efficiency of over 80% across 64 compute nodes (256 GPUs) on both LUMI and Leonardo and weak scaling efficiency of about 95% on LUMI and 90% on Leonardo with up to 256 nodes (1024 GPUs). These improvements establish URANOS-2.0 as a leading supercomputing platform for compressible wall turbulence applications, making it ideal for aerospace and energy engineering tasks in the field of Direct Numerical Simulations (DNS), Wall-Resolved Large Eddy Simulations (WRLES), and the latest Wall-Modeled LES (WMLES). The open-source code is available at https://github.com/uranos-gpu/uranos-gpu.

Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day

Keywords: GPU; OpenACC; Compressible Flows; DNS; LES; WMLES; Open-source

1. Introduction

In recent years, CFD has become indispensable for analyzing and designing fluid devices across various engineering applications. CFD not only reduces the costs of experimental testing but also shortens the time-to-market for innovative solutions in fluid engineering. CFD methods, in fact, are crucial in two key engineering processes: the parametric approach, which optimizes engineering designs by evaluating CFD models across diverse design variables, and the physical-analytical approach, which involves detailed analysis of flow fields around specific devices. These approaches often intersect, both requiring substantial computational resources as the number of variables and accuracy demands increase. Thus, utilizing HPC infrastructures and advanced computational tools is vital for advancing fluid dynamics simulations in modern engineering design and prototyping.

* Corresponding author.
E-mail address: francesco.devanna@unipd.it

1877-0509
Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day
10.1016/j.procs.2025.02.257

The drive for improving computational models and utilizing more accurate computing tools extends beyond engineering fluid dynamics to all applied sciences. Over the past decades, there has been a global trend toward concentrating computing power in specialized centers. A recent milestone in this field is the development of supercomputers that surpass the exascale threshold, performing over 10^{18} floating-point operations per second. This achievement, highlighted by the Top500 ranking[1], includes the Frontier and Aurora clusters in the United States. While the United States currently leads the charge, Europe is well-represented with three pre-exascale facilities in the top ten: LUMI (Finland) ranks fifth, Leonardo (Italy) seventh, and Marenostrum 5 (Spain) eighth. This robust European presence demonstrates the Old Continent's strength in HPC and its commitment to further advancing supercomputing capabilities. However, even with advanced tools, the architectural heterogeneity of top HPC systems currently available in Europe presents challenges. LUMI, for example, uses AMD GPUs, Leonardo adopts NVIDIA A100 GPUs, while Marenostrum 5 uses NVIDIA H100 graphic cards. This diversity complicates software portability, as programs are often optimized for specific architectures preferred by research centers, making cross-platform usability complex.

Surely, significant progress has been made in developing state-of-the-art CFD solvers that leverage modern HPC facilities. However, these numerical platforms are often optimized for specific architectures, particularly NVIDIA-based clusters using CUDA-derived programming languages. Notable examples in this field include AFiD [39] and CaNS [5], both structured incompressible flow solvers for DNS of canonical flows, utilizing CUDA Fortran and OpenACC, respectively. OpenSBLI [28], a Python-based compressible CFD solver for DNS on CPUs and GPUs. PyFr [37], ZEFR [32], STREAmS [1], CharLES [3], STREAmS-2.0 [2], HORSES3D [19], and AMFlow [27]. All predominantly exploiting CUDA-based programming languages.

In this context, the URANOS project prioritizes vendor neutrality, aiming to create a portable CFD solver compatible with various HPC architectures. Recognizing the dynamic HPC landscape, URANOS-2.0 maintains this commitment by using the OpenACC standard and supporting both the latest NVIDIA and AMD GPU architectures through the `nvhpc` and `cray` compilers, respectively. This adaptability allows URANOS-2.0 to operate seamlessly across the diverse supercomputing facilities available within the EuroHPC JU framework. However, the portability enhancement through AMD support is not the only improvement in URANOS-2.0. The current solver release is also significantly refactored, achieving a 2× speedup over its predecessor on equivalent architectures; the solver remains capable of handling compressible flows using several turbulence modeling frameworks, including DNS, WRLES, and WMLES strategies. Thus, in this paper, performance achievements obtained by computational kernel refactoring are focused, as well as the solver single and multi-GPU performance on LUMI and Leonardo are highlighted. Further details are provided in [8].

The text is organized as follows: Section 2 provides a brief description of the URANOS structure, with peculiar emphasis on numerical schemes. Section 3 discusses the portability enhancements and makefile options. Section 4 presents GPU performance on cutting-edge architectures, comparing URANOS-2.0 with the previous release across top-ranked multi-GPU environments within the EuroHPC JU. Finally, Section 5 summarizes the conclusions.

2. Numerical model description

URANOS is a Fortran-based high-fidelity CFD solver developed for wall flow applications at the University of Padova's Industrial Engineering Department. It solves the filtered compressible Navier-Stokes equations in a conservative form, supporting DNS, WRLES, and WMLES models. The Navier-Stokes equations are discretized using a high-order finite-difference framework on structured Cartesian meshes, supporting both uniform and non-uniform grids, with various convective schemes. These include a central, nominally zero-dissipative Energy-Preserving (EP) method for smooth flows [31], three high-order Weighted Essentially Non-Oscillatory (WENO) methods [25] in -Z version [4], two Targeted Essentially Non-Oscillatory (TENO) schemes [20], and a fifth-order adaptive TENO-A version [21]. URANOS incorporates shock-capturing techniques to limit numerical viscosity at shock/shocklet sites, integrating WENO/TENO methods with the central EP method to create hybrid schemes like hybrid-WENO/EP and hybrid-TENO/EP [16, 17]. Shock locations are determined using density-gradient and density-jump methods, along

[1] The Top500 project ranks the world's 500 most powerful computer systems. Since 1993, it has been updated biannually https://www.top500.org.

with the enhanced Ducros sensor, effective in wall turbulence [18]. Notably, the pure WENO/TENO method or the pure EP scheme can be configured by adjusting the shock sensor threshold to 1 for pure WENO/TENO or to 0 for pure EP. This configuration simplifies the solver's maintainability and facilitates a clearer comparison of the performance of various computational kernels.

A distinct feature of URANOS is its management of viscous fluxes, separating incompressible from compressible viscous components using a high-order finite-difference method [12]. The solver -2.0 release also includes a standard Laplacian formulation for viscous terms, compatible with DNS frameworks but not recommended for WRLES or WMLES. Temporal integration is achieved with a third-order Total-Variation-Diminishing (TVD) low-storage Runge-Kutta method [23]. URANOS-2.0 offers four algebraic turbulence models: the classical Smagorinsky model [33], the Wall-Adaptive Large-Eddy (WALE) model [30], the Sigma model [35], and the Mixed Time Scale (MXTS) model [24]. Except for the Smagorinsky model, the other three are *wall-turbulence consistent*, ensuring a smooth transition to SubGrid-Scale (SGS) viscosity and diffusivity from the bulk flow to the wall, where turbulent parameters vanish. For WMLES, URANOS-2.0 uses the equilibrium-based wall model [26] to determine wall shear stress and heat flux across under-resolved wall regions, integrating this information as a boundary condition [14, 15, 9]. To provide a clear overview of the modeling features in URANOS-2.0, Table 1 summarizes the full range of numerical schemes and models integrated into the current software release. Each entry includes brief notes on its application scope, order of accuracy, and key references.

Table 1: Comprehensive overview of numerical schemes and modeling options in URANOS-2.0.

	Scheme	References	Application	Ord. of accuracy
Convective terms discretization	EP	[31, 38]	smooth flows	2/4/6
	WENO-Z	[25, 4]	shock-dominated flows	3/5/7
	TENO	[20]	shock-dominated flows	5/7
	TENO-A	[22]	shock-dominated flows	5
	Hybrid-WENO-Z/EP	[16]	shock-dominated flows	3/5/7 - 2/4/6
	Hybrid-TENO/EP	[17]	shock-dominated flows	5/6 - 2/4/6
	Hybrid-TENO-A/EP	[6, 13]	shock-dominated flows	5 - 2/4/6
Viscous terms discretization	Semi-conservative	[12]	Intensively varying diffusive properties	2/4/6
	Laplacian	[29]	Smoothly varying diffusive properties	2/4/6
Time Integration	TVD Runge-Kutta	[23]	Convective-dominated PDE	3
Shock-detection	Density Jump	-	A priori known flow patterns	2
	Density Gradient	-	A priori known flow patterns	2
	Ducros Sensor	[18]	Wall-turbulence	2/4/6
Laminar viscosity modeling	Sutherland's law	[34]	Wide temperature ranges	-
	Power law	[36]	Temperature clustered to reference	-
Turbulence modeling	Classical Smagorinsky	[33]	Free-shear turbulence	2/4/6
	WALE	[30]	wall turbulence	2/4/6
	Sigma	[35]	wall turbulence	2/4/6
	MXTS	[24]	wall turbulence	2/4/6
Wall modeling	Thin BL equations	[26]	-	2

3. Portability improvements

As outlined, the key objective of this research is to improve the portability of the URANOS solver across the latest multi-GPU systems. In particular, the URANOS-2.0 version marks a significant advancement by leveraging OpenACC to its full potential, distinguishing it from URANOS-1.0 [7], which, although based on OpenACC, was limited to testing on NVIDIA-based clusters. This enhancement boosts URANOS's compatibility with Europe's dominant supercomputing architectures, particularly the LUMI cluster, featuring AMD MI250X GPUs, and the Leonardo supercomputer in Italy, with NVIDIA A100 GPUs. Additionally, preparations are made for the NVIDIA H100 architecture now available on the Marenostrum 5 system. A notable distinction exists between LUMI-G and Leonardo: LUMI-G relies exclusively on OpenACC with the Cray compiler for Fortran codes, while Leonardo supports OpenACC via the NVHPC SDK. Thus, significant efforts are made to ensure URANOS's compatibility with the Cray compiler, an as-

pect not addressed in the previous release. The current makefile is restructured to enable seamless transitions between architectures, allowing compilation with:

```
make comp=<compiler> mode=<mode_option>
```

This approach lets users select the appropriate compiler based on the available architecture. The solver also supports the Cray and GNU compiler on CPU modes for users without GPUs. Details of the compilation options and intended architectures are outlined in Table 2.

Table 2: Overview of URANOS-2.0 compiling options and supported architectures.

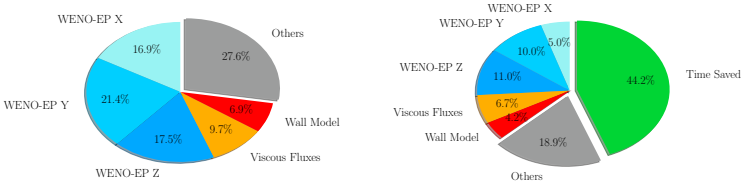
compiler	mode_option	intended supported architectures	tested GPUs
nvhpc	gpu gpu_debug gpu_profiling	NVIDIA architectures	V100, A100, H100
cray	gpu gpu_profiling	AMD architectures	MI250X
cray	cpu	Cray multi/many core architectures	-
gnu	cpu cpu_debug	Vendor neutral multi/many core architectures	-

Profiling options are also provided to differentiate between NVIDIA and AMD products and visualize kernel execution to identify the solver's most time-consuming segments. A dedicated profiling module, `src/profiling_module.f90`, serves as an interface for external profiling libraries, allowing any code segment to be monitored by wrapping it between `StartProfRange` and `EndProfRange` calls. The profiling process targets computationally intensive code sections using the NVIDIA NVHPC SDK 23.1 with the NVTX API for NVIDIA products and ROCTX 5.2.3 for AMD architectures. Profiling options are activated with user-defined compiler flags: `-DNVTX` and `-DROCTX`, along with `-DNVIDIA` and `-DAMD`, using the `mode=gpu_profiling` compilation setting.

Finally, when using URANOS with NVIDIA or AMD architectures, special attention is required for random number generation. The OpenACC framework does not natively support thread-safe random number generation, necessitating calls to specific libraries. Thus, for NVIDIA, the `curand` library, part of the NVHPC SDK, is utilized with the `-cudalib=curand` flag during compilation. For AMD, the `rocrand` library is used, accessible in LUMI by installing the `hipfort` interface via EasyBuild. The `-lhipfort-amdgcn -lrocrand` flags enable random number generation. The complexity behind this process remains transparent to the user, as the makefile manages it during compilation.

4. GPU Performance Optimization

4.1. Bottleneck Identification



(a) URANOS-1.0 (b) URANOS-2.0

Fig. 1: Time distribution per iteration: comparison between baseline and optimized versions of URANOS on a single NVIDIA A100 GPU.

The performance of the current solver version with the initial release is compared. Since URANOS-1.0 did not support AMD cards, direct comparisons are made using NVIDIA graphics cards, while URANOS-2.0 is evaluated on both NVIDIA and AMD cards to showcase improvements across GPU architectures. For realistic performance analysis, a turbulent boundary layer under hypersonic conditions as documented in Ref. [7] (case BL03) is simulated. Although nominally shock-free, this flow requires shock-capturing schemes in regions not identifiable *a priori*,

Table 3: Comparison of wall clock times between URANOS-1.0 and URANOS-2.0 for the most computationally intensive kernels. Data are collected on a single NVIDIA A100 GPU.

	URANOS-1.0		**URANOS-2.0**	
Kernel	time [ms]	time/iteration time [%]	time [ms]	time/iteration time [%]
WENO-EP X	139.8	16.9%	41.38	8.96%
WENO-EP Y	177.1	21.4%	82.77	17.9%
WENO-EP Z	144.8	17.5%	91.04	11.0%
Viscous Fluxes	80.28	9.70%	55.46	19.7%
Wall Model	57.11	6.90%	34.76	7.52%
Others	228.4	27.6%	156.4	33.9%
Tot.	**827.7**	**100%**	**461.7**	**100%**

adding complexity and reflecting realistic hybrid scheme operations (WENO/TENO + EP) over time and space. Figure 1 shows the bottleneck analysis for URANOS-1.0 and URANOS-2.0, illustrating the percentage of time per iteration dedicated to key numerical kernels. Data collected on a single NVIDIA A100 GPU at the Leonardo supercomputing facility, averaged over 100 iterations, ensures statistical reliability. To enhance clarity, Table 3 provides the same information, supplemented with the wall clock time for each computational kernel, detailed in milliseconds. As observed, in URANOS-1.0, time-intensive routines are mostly related to Navier-Stokes flux computations, including hybrid WENO/EP schemes, diffusive terms, and the wall model kernel. These five routines account for about 75% of iteration time, indicating a need for optimization. The remaining 25% involves tasks like time integration, updating variables, computing SGS terms, shock sensors, and MPI data movements, where optimization potential is limited without extensive refactoring. For conciseness, the present discussion focuses on advection fluxes optimization only, and further details are provided in Ref. [8].

4.2. Advection fluxes optimization

Investigations show a strong correlation between computation time and specific advection components, with decreased efficiency observed when calculating the y and z advection components compared to the x component, as illustrated in Figure 1. This discrepancy arises from the hybrid WENO(TENO)+EP algorithm, which requires extracting one-dimensional arrays from three-dimensional data structures to compute advection fluxes. The extraction aligns with Fortran data sorting for the x fluxes but suffers performance penalties with non-contiguous data for the y and z directions. Thus, although operations for computing convective flows are consistent across all directions, significant performance degradation occurs in streamwise-orthogonal terms. To address this, the pseudocode for the y flux components is reported. This algorithm mirrors the approach for the z direction and slightly differs from the x terms computation, even if the same principle is applied for the x direction.

Algorithm 1 shows the pseudocode. To clarify the notation, `sx` and `ex` denote the start and end indices of the inner domain pertaining to a single MPI block along the x Cartesian coordinate, while `sy` and `ey` and `sz` and `ez` serve the same purpose for the y and z directions, respectively. The variables `lbx`, `ubx`, `lby`, `uby`, `lbz`, and `ubz` represent the global domain indices that include the ghost nodes, defined as `sx-GN`, `ex+GN`, `sy-GN`, `ey+GN`, `sz-GN`, and `ez+GN`, respectively. GN denotes the number of ghost nodes, which are set to four. These nodes are essential for managing boundary conditions in high-order finite difference computations. A key optimization is consisted in chunking one-dimensional variables and storing them in shared memory. This approach reduces L1/L2 cache pressure, improving memory access and computational performance. The optimal chunk size was determined through testing, with `vec_size=64` striking a balance between performance gains and avoiding warp/wavefront stalling. Warp/wavefront stalling in GPUs refers to a condition where the execution of a group of threads (called a warp in NVIDIA GPUs and a wavefront in AMD GPUs) is paused or delayed due to factors that prevent further progress. One primary cause of stalling is memory latency, where a warp needs to access data from memory that is not immediately available, often due to cache misses or inherently slow memory access. This latency forces the warp to pause until the required data is retrieved, leading to idle computation time. Thus, variables for flux computations are defined statically, using the `!$acc cache(list_of_variables)` directive to allocate cached variables in shared memory. Variables segmented into chunks are three-dimensional, with an inner dimension of size $0 : 3$, a second dimension of size $1-GN : chunkSize+GN$, and a third packing various fields (e.g., density, velocity, pressure, etc.). This structure opti-

```
subroutine PseudoCodeOfAdvectionFluxesComputation
! ...
integer , parameter                                :: vec_size  = 64
integer , parameter                                :: chunkSize = vec_size - 2*GN
real(rp), dimension(0:3,1-GN:chunkSize+GN)         :: chunked_variable
! ...

!$acc parallel vector_length(vec_size) default(present) &
!$acc private(list_of_private_variables)
!$acc loop gang collapse(2)
do k    = sz,ez
  do i = sx,ex,4

     iimax = min(3,ex-i)

     do str_y = lby,uby,chunkSize
        end_y = min(str_y+ChunkSize, uby-2*GN+1) - str_y

        !$acc cache(list_of_lcl_chunked_variables)

        !$acc loop vector collapse(2)
        do j = 1-GN,end_y+GN
           do ii = 0,iimax
              jgbl = j + str_y-1+GN

              ! storing data in chunked and cached variables
              lcl_chunked_variable(ii,j,1) = gbl_variable(i+ii,jgbl,k,1)

           enddo
        enddo

        ! computing advection fluxes using lcl chunked variables
        do ii=0,iimax
           !$acc loop vector
           do j = 1-GN,end_y+GN

              if(weno_flag_chunk(j) == is_smooth) then
                ! Perform EP algorithm
              else
                ! Perform WENO/TENO algorithm
              endif

           enddo
       enddo

   enddo
enddo

return
end subroutine PseudoCodeOfAdvectionFluxesComputation
```

Algorithm 1: Pseudocode associated with advection fluxes computation.

mizes global data continuity, reducing cache misses. For general domains not constrained by multiples of `vec_size`, operations on indices are incorporated, as shown in lines 14 and 17 in Algorithm 1.

Beyond chunking, warp/wavefront stalling is addressed by spreading data structures containing shock information. This issue is particularly relevant in the context of hybrid WENO(TENO)-EP methods. In these schemes, in fact, local computations of advection fluxes are performed in a non-contiguous manner due to the integration of shock-capturing reconstructions (WENO/TENO) and the EP method, which are controlled via conditional statements based on the local value of the shock detector. In URANOS-1.0, dynamically spreading the `weno_flag` structure (`integer(kind = 4)`) during flux computation was found particularly detrimental to GPU operation with respect to warp stalling. However, `weno_flag` spreading is crucial, as shock detection methods may identify solitary nodes due to noise or fluctuations, threatening simulation stability. Thus, expanding shock coverage ensures stability, covering at least the stencil size of the WENO/TENO scheme. Thus, in URANOS-2.0, the spreading of shock positions is separated from convective flux computation into a dedicated kernel. This kernel solely executes a localized reduction, defining the minimum value of `weno_flag` within a stencil of size depending on WENO/TENO accuracy. A cell flagged as shocked has a zero value in `weno_flag`, while smooth flow is indicated by a value of one. `minval` Fortran functions are used for this purpose. Additionally, the `weno_flag` values are multiplied by 2 and 4 along the y and z axes, respectively, minimizing the field variables needed to store shock locations and implementing a bitwise flag selection procedure to define the chunked variable associated with `weno_flag`.

4.3. Single-GPU performance comparison

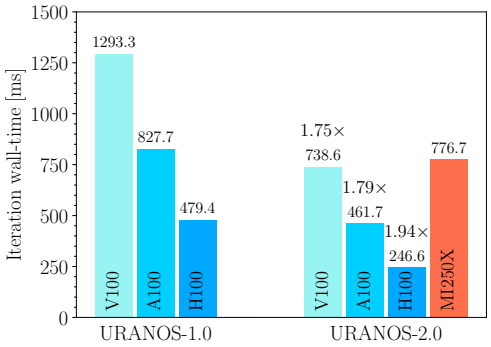


Fig. 2: Time per iteration for URANOS-1.0 and URANOS-2.0 across various GPU architectures. For the AMD MI250X, tests are conducted on a single Graphics Compute Die (GCD), and the reported times are halved to reflect the potential of utilizing the full GPU.

Figure 2 highlights the performance speedup achieved by URANOS-2.0 compared to its initial release. Data compare various GPU architectures, including NVIDIA V100, A100, and H100, as well as AMD MI250X. The evaluation is based on a realistic flow scenario involving a 50-million-point hypersonic boundary layer test case and exploiting the $z-$periodicity to include MPI calls since periodic boundary conditions are handled with MPI even on a single rank execution. In particular, in Figure 2, the left-hand columns represent URANOS-1.0 results, while the right-hand bars display URANOS-2.0 outcomes.

The rightmost columns show the iteration times for URANOS-2.0, revealing a 1.75-fold acceleration on V100, a 1.79-fold improvement on A100, and a 1.94-fold increase on H100 compared to URANOS-1.0. Data for AMD cards is not included for URANOS-1.0, as the earlier version did not support these architectures, preventing the establishment of a historical trend for AMD products. However, URANOS-2.0's performance on AMD cards is found to be comparable to that on an NVIDIA V100 GPU.

4.4. Multi-GPU scaling and performance

For multi-GPU performance comparison the computational power of the Leonardo-booster (Cineca) and LUMI-G (CSC) supercomputer partitions is used. Leonardo, a TIER-0 architecture based on the Atos BullSequana XH2000,

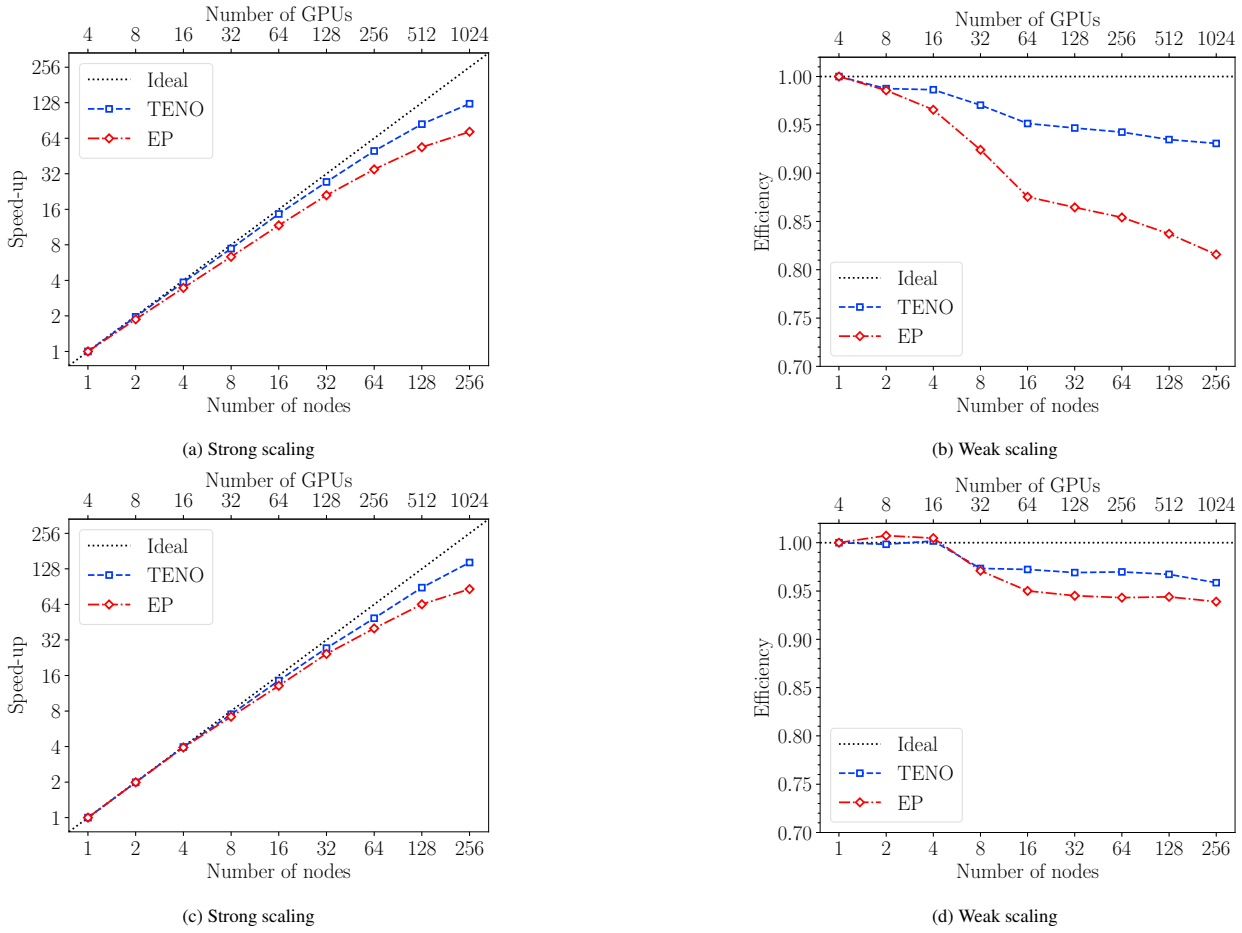


Fig. 3: First row: Leonardo strong (3a) and weak (3b) scaling. Strong scaling compares the elapsed time on one node to n nodes for a fixed grid of $1024 \times 1024 \times 512$ points. Weak scaling compares the elapsed time on one node to n nodes, keeping points-per-node fixed at $1024 \times 1024 \times 512$. Second row: LUMI strong (3c) and weak (3d) scaling. Strong scaling compares the elapsed time on one node to n nodes for a fixed grid of $1024 \times 1024 \times 512$ points. Weak scaling compares the elapsed time on one node to n nodes, keeping points-per-node fixed at $1024 \times 1024 \times 512$. Each MI250X has two GCDs, so MPI ranks are doubled for LUMI compared to Leonardo.

includes 3456 Intel Ice Lake nodes, each with 32 cores and 4 NVIDIA A100 SXM4 64GB GPUs. LUMI is an HPE Cray EX system with 2978 nodes, each featuring a 64-core AMD Trento CPU and four AMD MI250X GPUs. Leonardo and LUMI rank seventh and fifth in the world's fastest supercomputers as of the June 2024 Top500 ranking. Present tests are conducted using the NVIDIA HPC SDK 23.1 compiler and the AMD ROCm 5.2.3 suite with the Cray Programming Environment (CPE) release 23.09. A GPU-aware communication model is used to optimize data exchange efficiency, minimizing unnecessary CPU-GPU transfers.

Figure 3a and 3b display the parallel performance on the Leonardo architecture, highlighting strong scaling and weak scalings. For strong scaling, the total number of grid points is fixed at $N_x \times N_y \times N_z = 1024 \times 1024 \times 512 \simeq 536 \cdot 10^6$, while resources vary. For weak scaling, the computational unit size per node remains fixed at $536 \cdot 10^6$ grid points. Experiments involve a tri-periodic flow with zero velocity, unit temperature, and unit pressure using EP 6 and TENO 7 schemes. Figure 3a shows strong scaling efficiency nearing ideal levels, with TENO 7 at nearly 50% and EP 6 at 30% over 256 nodes (1024 GPUs). Efficiency remains high with 64 nodes, at approximately 80% for TENO and 54% for EP, due to TENO's handling of computational workload versus MPI communications. In weak scaling, the solver maintains good performance up to $1.4 \cdot 10^{11}$ grid points over 256 nodes. TENO 7 sustains around 93% efficiency, while EP achieves 80%.

Figure 3c and 3d show results for LUMI, which are similar to Leonardo's. Here, it is noteworthy that each MI250X GPU contains two GCDs, which effectively doubles the number of MPI ranks available on the LUMI system compared to Leonardo, enhancing its parallel processing capacity. In strong scaling (Fig. 3c), LUMI matches Leonardo with 57% efficiency for TENO on 256 nodes and 34% for EP. In weak scaling (Fig. 3d), LUMI maintains very high efficiency, reaching approximately 94% for EP and 96% for TENO at 256 nodes.

5. Conclusions

URANOS-2.0, an enhanced version of a GPU-accelerated Navier-Stokes solver designed for compressible wall flows, has been presented. The solver uses a high-order/high-resolution finite difference framework to address DNS, WRLES, and WMLES of compressible wall flows, and the current release is optimized for pre-exascale HPC architectures within the framework of EuroHPC JU. In particular, URANOS-2.0 has been evaluated on two premier multi-GPU architectures, LUMI and Leonardo, ranked fifth and seventh globally according to the June 2024 Top500 ranking. The solver demonstrates a strong scaling efficiency of over 80% across 64 compute nodes (256 GPUs) and a weak scaling efficiency of approximately 95% on LUMI and 90% on Leonardo when scaled up to 256 nodes (1024 GPUs). Thus, URANOS-2.0 establishes itself as a flexible, multi-platform solver capable of exploiting Europe's top supercomputing architectures and addressing limitations in other freely available CFD software. The open-source baseline code is available at https://github.com/uranos-gpu/uranos-gpu while, future advancements will include complex geometries as in [11, 10].

Acknowledgements

The author thanks Professor Ernesto Benini for leading the URANOS project and Dr. Matt Bettencourt for code optimization support. Appreciation is extended to NVIDIA Corporation for access to H100 GPU architectures and to EuroHPC JU for LUMI and Leonardo supercomputing access through URANOS-AMD, OptimURANOS-AMD, and CINECA projects, primarily IsB26_HERMES (HP10BJK91V) and IsB28_ARTEMIDE (HP10BEJ9YD).

References

[1] Bernardini, M., Modesti, D., Salvadore, F., Pirozzoli, S., 2021. STREAmS: a high-fidelity accelerated solver for direct numerical simulation of compressible turbulent flows. Comput. Phys. Commun. 263, 107906. doi:https://doi.org/10.1016/j.cpc.2021.107906.

[2] Bernardini, M., Modesti, D., Salvadore, F., Sathyanarayana, S., Della Posta, G., Pirozzoli, S., 2023. STREAmS-2.0: Supersonic turbulent accelerated Navier-Stokes solver version 2.0. Comput. Phys. Commun. 285, 108644. doi:https://doi.org/10.1016/j.cpc.2022.108644.

[3] Bres, G.A., Bose, S.T., Ivey, C.B., Emory, M., Ham, F., 2022. GPU-accelerated large-eddy simulations of supersonic jets from twin rectangular nozzle, in: 28th AIAA/CEAS aeroacoustics 2022 conference, p. 3001. doi:https://doi.org/10.2514/6.2022-3001.

[4] Castro, M., Costa, B., Don, W.S., 2011. High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws. J. Comput. Phys. 230, 1766–1792. doi:https://doi.org/10.1016/j.jcp.2010.11.028.

[5] Costa, P., Phillips, E., Brandt, L., Fatica, M., 2021. GPU acceleration of CaNS for massively-parallel direct numerical simulations of canonical fluid flows. Comput. Math. Appl. 81, 502–511. doi:https://doi.org/10.1016/j.camwa.2020.01.002.

[6] De Vanna, F., Avanzi, F., Cogo, M., Sandrin, S., Bettencourt, M., Picano, F., Benini, E., 2023a. GPU-acceleration of Navier-Stokes solvers for compressible wall-bounded flows: the case of URANOS, in: AIAA SCITECH 2023 Forum, p. 1129. doi:https://doi.org/10.2514/6.2023-1129.

[7] De Vanna, F., Avanzi, F., Cogo, M., Sandrin, S., Bettencourt, M., Picano, F., Benini, E., 2023b. URANOS: A GPU accelerated Navier-Stokes solver for compressible wall-bounded flows. Comput. Phys. Commun. 287, 108717. doi:https://doi.org/10.1016/j.cpc.2023.108717.

[8] De Vanna, F., Baldan, G., 2024. Uranos-2.0: Improved performance, enhanced portability, and model extension towards exascale computing of high-speed engineering flows. Comput. Phys. Commun. , 109285doi:https://doi.org/10.1016/j.cpc.2024.109285.

[9] De Vanna, F., Baldan, G., Picano, F., Benini, E., 2023c. Effect of convective schemes in wall-resolved and wall-modeled LES of compressible wall turbulence. Comput. Fluids 250, 105710. doi:https://doi.org/10.1016/j.compfluid.2022.105710.

[10] De Vanna, F., Baldan, G., Picano, F., Benini, E., 2023d. High-Reynolds Compressible Flows Simulation with Wall-Modeled LES and Immersed Boundary Method, in: ERCOFTAC Workshop Direct and Large Eddy Simulation, Springer. pp. 203–208. doi:https://doi.org/10.1007/978-3-031-47028-8_31.

[11] De Vanna, F., Baldan, G., Picano, F., Benini, E., 2023e. On the coupling between wall-modeled LES and immersed boundary method towards applicative compressible flow simulations. Comput. Fluids 266, 106058. doi:https://doi.org/10.1016/j.compfluid.2023.106058.

[12] De Vanna, F., Benato, A., Picano, F., Benini, E., 2021a. High-order conservative formulation of viscous terms for variable viscosity flows. Acta Mech. 232, 2115–2133. doi:https://doi.org/10.1007/s00707-021-02937-2.

[13] De Vanna, F., Benini, E., 2024. Towards new insights in gas turbine aerothermodynamics with wall-modeled les and immersed boundary method, in: Turbo Expo: Power for Land, Sea, and Air, American Society of Mechanical Engineers. p. V12BT30A001. doi:https://doi.org/10.1115/GT2024-121022.

[14] De Vanna, F., Cogo, M., Bernardini, M., Picano, F., Benini, E., 2021b. Unified wall-resolved and wall-modeled method for large-eddy simulations of compressible wall-bounded flows. Phys. Rev. Fluids 6, 034614. doi:https://doi.org/10.1103/PhysRevFluids.6.034614.

[15] De Vanna, F., Michele, C., Matteo, B., Picano, F., Benini, E., et al., 2021c. A wall-modeled/wall-resolved LES method for turbulent wall flows, in: ECCOMAS Congress 2020). doi:10.23967/wccm-eccomas.2020.045.

[16] De Vanna, F., Picano, F., Benini, E., 2020. A sharp-interface immersed boundary method for moving objects in compressible viscous flows. Comput. Fluids 201, 104415. doi:https://doi.org/10.1016/j.compfluid.2019.104415.

[17] De Vanna, F., Picano, F., Benini, E., Quinn, M.K., 2021d. Large-eddy simulations of the unsteady behavior of a hypersonic intake at mach 5. AIAA J. 59, 3859–3872. doi:https://doi.org/10.2514/1.J060160.

[18] Ducros, F., Ferrand, V., Nicoud, F., Weber, C., Darracq, D., Gacherieu, C., Poinsot, T., 1999. Large-eddy simulation of the shock/turbulence interaction. J. Comput. Phys. 152, 517–549. doi:https://doi.org/10.1006/jcph.1999.6238.

[19] Ferrer, E., Rubio, G., Ntoukas, G., Laskowski, W., Mariño, O., Colombo, S., Mateo-Gabín, A., Marbona, H., de Lara, F.M., Huergo, D., et al., 2023. HORSES3D: A high-order discontinuous Galerkin solver for flow simulations and multi-physics applications. Comput. Phys. Commun 287, 108700. doi:https://doi.org/10.1016/j.cpc.2023.108700.

[20] Fu, L., Hu, X.Y., Adams, N.A., 2017. Targeted ENO schemes with tailored resolution property for hyperbolic conservation laws. J. Comput. Phys. 349, 97–121. doi:https://doi.org/10.1016/j.jcp.2017.07.054.

[21] Fu, L., Hu, X.Y., Adams, N.A., 2018. A new class of adaptive high-order targeted ENO schemes for hyperbolic conservation laws. J. Comput. Phys. 374, 724–751. doi:https://doi.org/10.1016/j.jcp.2018.07.043.

[22] Fu, L., Liang, T., 2022. A new adaptation strategy for multi-resolution method. J. Sci. Comput. 93, 43. doi:https://doi.org/10.1007/s10915-022-02012-5.

[23] Gottlieb, S., Shu, C.W., 1998. Total variation diminishing runge-kutta schemes. Math. Comput. 67, 73–85. doi:https://doi.org/10.1090/S0025-5718-98-00913-2.

[24] Inagaki, M., Kondoh, T., Nagano, Y., 2005. A mixed-time-scale SGS model with fixed model-parameters for practical LES. J. Fluids Eng. 127, 1–13. doi:https://doi.org/10.1115/1.1852479.

[25] Jiang, G.S., Shu, C.W., 1996. Efficient implementation of weighted ENO schemes. J. Comput. Phys. 126, 202–228. doi:https://doi.org/10.1006/jcph.1996.0130.

[26] Larsson, J., Kawai, S., Bodart, J., Bermejo-Moreno, I., 2016. Large eddy simulation with modeled wall-stress: recent progress and future directions. Mech. Eng. Rev. 3, 15–00418. doi:https://doi.org/10.1299/mer.15-00418.

[27] Liu, X., Ma, Y.Q., 2023. AMFlow: A Mathematica package for Feynman integrals computation via auxiliary mass flow. Comput. Phys. Commun. 283, 108565. doi:https://doi.org/10.1016/j.cpc.2022.108565.

[28] Lusher, D.J., Jammy, S.P., Sandham, N.D., 2021. OpenSBLI: Automated code-generation for heterogeneous computing architectures applied to compressible fluid dynamics on structured grids. Comput. Phys. Commun. 267, 108063. doi:https://doi.org/10.1016/j.cpc.2021.108063.

[29] Modesti, D., Pirozzoli, S., 2016. Reynolds and Mach number effects in compressible turbulent channel flow. Int. J. Heat Fluid Flow. 59, 33–49. doi:https://doi.org/10.1016/j.ijheatfluidflow.2016.01.007.

[30] Nicoud, F., Ducros, F., 1999. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. Flow Turbul. Combust. 62, 183–200. doi:https://doi.org/10.1023/A:1009995426001.

[31] Pirozzoli, S., 2011. Numerical methods for high-speed flows. Annu. Rev. Fluid Mech. 43, 163–194. doi:https://doi.org/10.1146/annurev-fluid-122109-160718.

[32] Romero, J., Crabill, J., Watkins, J.E., Witherden, F.D., Jameson, A., 2020. ZEFR: A GPU-accelerated high-order solver for compressible viscous flows using the flux reconstruction method. Comput. Phys. Commun. 250, 107169. doi:https://doi.org/10.1016/j.cpc.2020.107169.

[33] Scotti, A., Meneveau, C., Lilly, D.K., 1993. Generalized Smagorinsky model for anisotropic grids. Phys. Fluids A: Fluid Dynamics 5, 2306–2308. doi:https://doi.org/10.1063/1.858537.

[34] Sutherland, W., 1893. LII. The viscosity of gases and molecular force. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 36, 507–531. doi:https://doi.org/10.1080/14786449308620508.

[35] Toda, H.B., Cabrit, O., Balarac, G., Bose, S., Lee, J., Choi, H., Nicoud, F., 2010. A subgrid-scale model based on singular values for LES in complex geometries , 193–202URL: https://web.stanford.edu/group/ctr/Summer/SP10/3_03_bayatoda.pdf.

[36] Williams, F.A., 1926. The effect of temperature on the viscosity of air. Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character 110, 141–167. doi:https://doi.org/10.1098/rspa.1926.0008.

[37] Witherden, F.D., Farrington, A.M., Vincent, P.E., 2014. PyFR: An open source framework for solving advection–diffusion type problems on streaming architectures using the flux reconstruction approach. Comput. Phys. Commun. 185, 3028–3040. doi:https://doi.org/10.1016/j.cpc.2014.07.011.

[38] Zhao, G., Sun, M., Memmolo, A., Pirozzoli, S., 2019. A general framework for the evaluation of shock-capturing schemes. J. Comput. Phys. 376, 924–936. doi:https://doi.org/10.1016/j.jcp.2018.10.013.

[39] Zhu, X., Phillips, E., Spandan, V., Donners, J., Ruetsch, G., Romero, J., Ostilla-Mónico, R., Yang, Y., Lohse, D., Verzicco, R., et al., 2018. AFiD-GPU: a versatile Navier–Stokes solver for wall-bounded turbulent flows on GPU clusters. Comput. Phys. Commun. 229, 199–210. doi:https://doi.org/10.1016/j.cpc.2018.03.026.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 33–42

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

Efficient and scalable atmospheric dynamics simulations using non-conforming meshes

Giuseppe Orlando[a,*], Tommaso Benacchio[b,*], Luca Bonaventura[c]

[a] *CMAP, CNRS, École polytechnique, Institute Polytechnique de Paris, Route de Saclay, 91120 Palaiseau, France*
[b] *Weather Research, Danish Meteorological Institute, Sankt Kjelds Plads 11, 2100 Copenhagen, Denmark*
[c] *Dipartimento di Matematica, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy*

Abstract

We study the massively parallel performance of a *h*-adaptive solver for atmosphere dynamics that allows for non-conforming mesh refinement. The numerical method is based on a Discontinuous Galerkin (DG) spatial discretization, highly scalable thanks to its data locality properties, and on a second order Implicit-Explicit Runge-Kutta (IMEX-RK) method for time discretization, particularly well suited for low Mach number flows. Simulations with non-conforming meshes for flows over orography can increase the accuracy of the local flow description without affecting the larger scales, which can be solved on coarser meshes. We show that the local refining procedure has no significant impact on the parallel performance and, therefore, both efficiency and scalability can be achieved in this framework.

Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day

Keywords: Numerical Weather Prediction; Non-conforming meshes; Flows over orography; Discontinuous Galerkin methods; IMEX schemes

1. Introduction

Efficient numerical simulations of atmospheric flows are crucial for weather and climate predictions and pose several computational challenges. Peculiar to this application are the timeliness constraints that dictate maximum admissible simulation runtimes in operational weather prediction. Medium range numerical weather prediction (NWP) forecasts up to ten days ahead are typically expected to complete within one hour in the forecast cycles of weather centres, thus imposing demanding modelling choices in order to guarantee computational efficiency. In a context of increasing need for computational resources due to higher spatial resolutions, massively parallel model scalability is therefore required.

From a physical standpoint, slow-moving atmospheric flows of meteorological interest are characterized by a speed much lower than the speed of sound, so that their Mach number is low and compressibility effects are usually deemed

* Corresponding author.
E-mail addresses: giuseppe.orlando@polytechnique.edu, tbo@dmi.dk

Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day
10.1016/j.procs.2025.02.258

not very relevant. Weakly compressible flows are an example of problem with multiple length and time scales [11]. Moreover, atmospheric flows display phenomena on a very wide range of spatial scales that interact with each other. Strongly localized features require a very high spatial resolution to be correctly resolved, while larger scale features, such as midlatitude pressure systems and stratospheric flows, can be adequately resolved on coarser meshes. Hence, the design of efficient and stable numerical schemes for such models is a challenging task.

Because of its multi-scale nature, NWP and, in particular, flows over orography are an apparently ideal framework to develop adaptive numerical approaches based on variable resolution meshes. However, mesh adaptation strategies have slowly found their way into the NWP literature, due to concerns about the accuracy of variable resolution meshes for the correct representation of atmospheric wave phenomena, and the greater complexity of an efficient parallel implementation for non-uniform or adaptive meshes. Moreover, numerical strategies with variable resolution meshes typically employ local mesh refinement only in the horizontal directions, while columns of cells with the same horizontal dimension are employed in the vertical direction [12, 17, 24]. A non-conforming mesh is characterized by neighbouring cells with different resolution on both the horizontal and the vertical direction [21]. In [9], a full 3D nesting approach for atmospheric flows is presented. However, the method is tested only on cases without orography. To the best of our knowledge, in [21] the authors firstly proposed a method able to decrease both horizontal and vertical resolution as height increases, filling a gap in the NWP literature and showing how fully 3D non-conforming meshes can be successfully employed for flows over orography. The solver is based on the IMEX-DG method proposed in [19] (see also [18]) and employed for atmospheric flows in [20, 21, 22]. Thanks to its data locality features, DG simulations are characterized by small communication-to-computation ratios and increasingly good scalability at higher orders of accuracy.

In this work, we analyze the parallel performance of the solver implementation, carried out in the framework of the `deal.II` library [1, 2] which natively allows for the use of non-conforming meshes. Using a well-established library with an active user community allows the investigation of advanced numerical choices without the need to code basic features of the numerical method or to implement parallel paradigms. Here we show that the local mesh refinement procedure does not adversely affect the parallel efficiency and scalability of the model compared to the simulations using uniform meshes, as measured in runs performed on the MeluXina EuroHPC high-performance computing facility.

The manuscript is structured as follows. In Section 2, we briefly review the model equations and the numerical framework. The application and the analysis of parallel performance for a relevant benchmark is presented in Section 3. Finally, some conclusions are reported in Section 4.

2. The model equations and the numerical framework

The compressible Euler equations of gas dynamics represent the most comprehensive mathematical model for atmosphere dynamics [7, 23]. Let $\Omega \subset \mathbb{R}^d, 2 \leq d \leq 3$ be a domain and denote by $\mathbf{x}$ and t the spatial coordinates and the temporal coordinate, respectively. The mathematical model reads as follows:

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \\ &\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = \rho \mathbf{g} \\ &\frac{\partial (\rho E)}{\partial t} + \nabla \cdot \left[(\rho E + p) \mathbf{u} \right] = \rho \mathbf{g} \cdot \mathbf{u}, \end{aligned} \tag{1}$$

for $\mathbf{x} \in \Omega, t \in (0, T_f]$, supplied with suitable initial and boundary conditions. Here T_f is the final time, ρ is the density, $\mathbf{u}$ is the fluid velocity, p is the pressure, and $\otimes$ denotes the tensor product. Moreover, $\mathbf{g} = -g\mathbf{k}$ is the acceleration of gravity, with $g = 9.81\,\mathrm{m\,s^{-2}}$ and $\mathbf{k}$ being the upward pointing unit vector in the standard Cartesian frame of reference. Finally, E denotes the total energy per unit of mass and we point out that one can rewrite $\rho E = \rho e + \rho k$, where e is the internal energy and $k = \frac{1}{2} |\mathbf{u}|^2$ is the kinetic energy. System (1) is complemented by the equation of state of ideal gases, given by $p = \rho R T$, where R is the specific gas constant and T denotes the temperature. We take $R = 287\,\mathrm{J\,kg^{-1}\,K^{-1}}$.

Atmospheric flows, as those considered in this work, are characterized by low values of the Mach number M. In the low Mach number limit, pressure gradient terms yield stiff components for the resulting semi-discretized ODE

system, since the pressure gradients in (1) are proportional to $\frac{1}{M^2}$. Hence, following [4, 5], the method proposed in [19] couples implicitly the energy equation to the momentum equation, while the continuity equation is treated in a fully explicit fashion. High-order accuracy in time is then achieved making use of Implicit-Explicit Runge-Kutta (IMEX-RK) time integrators [10], which are widely employed for ODE systems that include both stiff and non-stiff components. Finally, for the spatial discretization we employ the Discontinuous Galerkin (DG) method [6], which combines high-order accuracy and flexibility in a highly data-local framework. More specifically, we aim at employing non-conforming meshes, i.e. meshes for which the resolution between two neighbouring cells can be different along both horizontal and vertical direction. The DG method naturally allows the use of this kind of meshes [8] without any hanging node appearing. We refer to [21] for a short introduction to non-conforming meshes, and to [18, 19, 20] for a complete analysis and discussion of the numerical methodology.

3. Numerical results

In this Section, we consider an idealized three-dimensional test case of an atmospheric flow over orography [15, 16, 21]. Simulation parameters are related to two Courant numbers, the so-called acoustic Courant number C, which is based on the speed of sound c, and the advective Courant number C_u, which is based on the speed of the local flow velocity u:

$$C = rc\Delta t\sqrt{d}/\mathcal{H} \qquad C_u = ru\Delta t\sqrt{d}/\mathcal{H}.$$

Here, r is the polynomial degree used for the DG spatial discretization, $\mathcal{H}$ is the minimum cell diameter of the computational mesh, and Δt is the time step adopted for the time discretization. We consider polynomial degree $r = 4$. Recall that d represents the number of dimensions. In the following, we analyze the accuracy of the simulations with mesh refinement and then focus on the scalability of the numerical model. The 9.5.2 `deal.II` [1, 2] release has been used to produce the results in this Section. The simulations have been run using up to 1024 2x AMD EPYC Rome 7H12 64c 2.6GHz CPUs at MeluXina supercomputer [1] and OpenMPI 4.1.5 has been employed. The compiler is GCC version 12.3 and the Vectorization level is 256 bits.

3.1. 3D medium-steep bell-shaped hill

We consider a three-dimensional configuration of a flow over a bell-shaped hill, originally proposed in [15] and also employed in [16, 20, 21]. The computational domain is $\Omega = (0, 60) \times (0, 40) \times (0, 16)$ km. The mountain profile is defined as follows:

$$h(x, y) = \frac{h_c}{\left[1 + \left(\frac{x - x_c}{a_c}\right)^2 + \left(\frac{y - y_c}{a_c}\right)^2\right]^{\frac{3}{2}}}, \tag{2}$$

with $h_c = 400\,\mathrm{m}, a_c = 1\,\mathrm{km}, x_c = 30\,\mathrm{km}$, and $y_c = 20\,\mathrm{km}$. The buoyancy frequency is $N = 0.01\,\mathrm{s}^{-1}$, whereas the background velocity is $\overline{u} = 10\,\mathrm{m\,s}^{-1}$. The final time is $T_f = 1\,\mathrm{h}$. The initial conditions read as follows [3]:

$$p = p_{ref}\left\{1 - \frac{g}{N^2}\Gamma\frac{\rho_{ref}}{p_{ref}}\left[1 - \exp\left(-\frac{N^2 z}{g}\right)\right]\right\}^{1/\Gamma} \tag{3}$$

$$\rho = \rho_{ref}\left(\frac{p}{p_{ref}}\right)^{1/\gamma}\exp\left(-\frac{N^2 z}{g}\right), \tag{4}$$

[1] https://docs.lxp.lu/

where $p_{ref} = 10^5$ Pa and $\rho_{ref} = \frac{p_{ref}}{RT_{ref}}$, with $T_{ref} = 293.15$ K. Finally, we set $\Gamma = \frac{\gamma-1}{\gamma}$, with $\gamma = 1.4$ being the isentropic exponent. Wall boundary conditions are employed for the bottom boundary, whereas non-reflecting boundary conditions are required by the top boundary and the lateral boundaries. We refer to [21] for the implementation of non-reflecting boundary conditions.

We consider two different meshes: a uniform mesh composed by $N_{el} = 60 \times 40 \times 16 = 38400$ elements, i.e. a spatial resolution of 250 m along all the directions, and a non-conforming mesh composed by $N_{el} = 1958$, with its finest resolution corresponding to that of the uniform mesh (Figure 1). We refer to [21] for a detailed discussion about the resolution requirements. Notice that the resolution depends on the polynomial degree r employed for the spatial discretization. More specifically, the effective resolution is computed dividing the size of the element along each direction by the polynomial degree. We take $\Delta t = 2$ s, yielding a maximum acoustic Courant number $C \approx 2.75$ and a maximum advective Courant number $C_u \approx 0.13$ for the finest uniform mesh.

The contours plots of the vertical velocity on a $x-z$ slice placed at $y = 20$ km and on a $x-y$ slice at $z = 800$ m show the accuracy and the robustness of simulations employing non-conforming meshes, without significant differences compared to the simulation with uniform meshes (Figures 2 and 3). Specifically, no spurious wave reflections arise at the internal boundaries that separate regions with different mesh resolutions. Hence, it is sufficient to employ a higher resolution only around the orography, whereas larger scales along all the directions can be resolved at a much coarser resolution.

Besides producing results that are of comparable accuracy with those obtained with a uniform mesh, the non-conforming mesh provides sizeable computational savings. With reference to the results in Figures 2 and 3, the wall-clock time of the simulation with the non-conforming mesh is 2560 s, whereas the wall-clock time of the simulation with the uniform mesh is 36 500 s. Hence, the use of the non-conforming mesh yields a computational time saving of around 93% with respect to the uniform mesh (see also Table 6 in [21]).

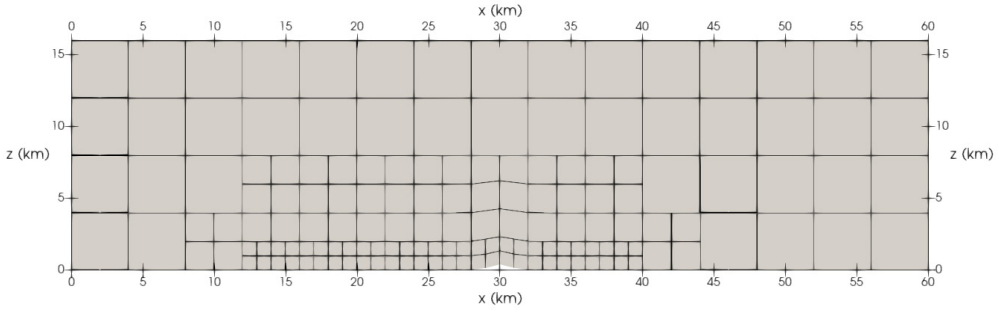


Fig. 1: 3D medium-steep bell-shaped hill test case, non conforming mesh. $x-z$ slice at $y = 20$ km.

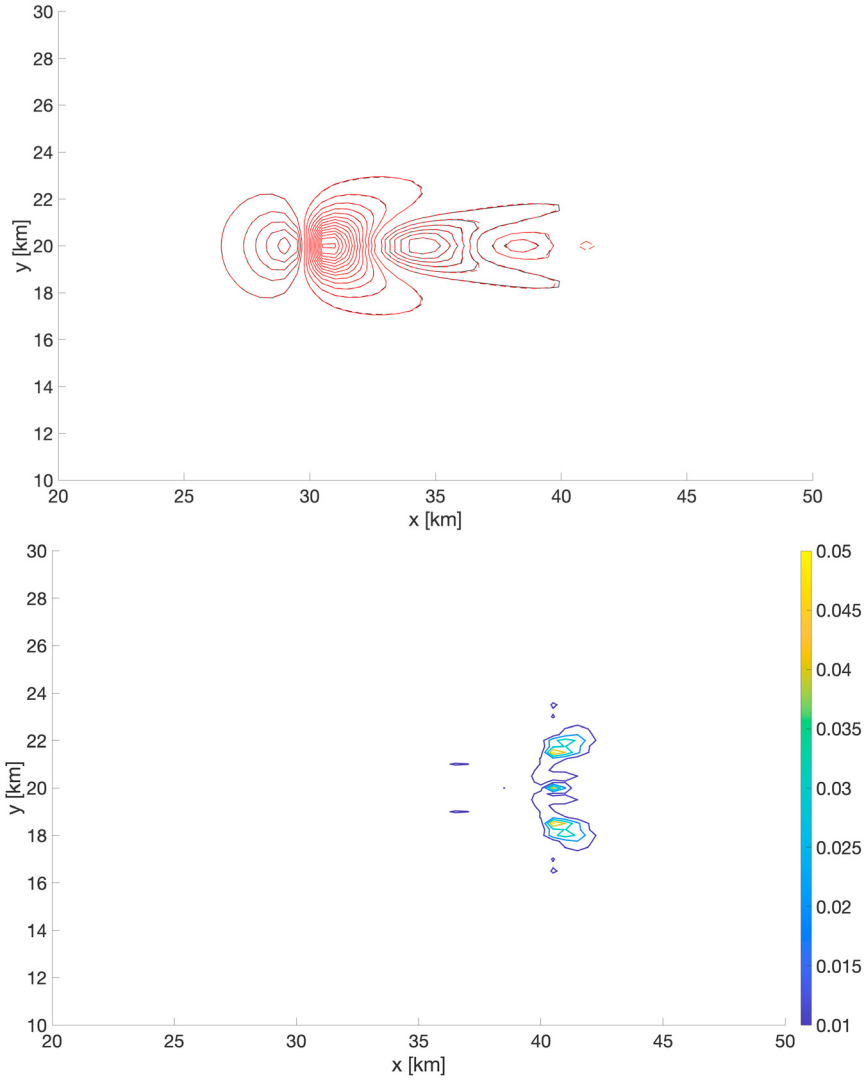


Fig. 2: Vertical velocity in the 3D medium-steep bell-shaped hill test case at $T_f = 1$ h, $x-y$ slice at $z = 800$ m. Contours in the range $[-1.5, 1.3]$ m s^{-1} with a 0.1 m s^{-1} interval. Top: comparison between the uniform mesh (black lines) and the non-conforming mesh (red lines). Negative contours are dashed. Bottom: absolute difference between the uniform mesh and the non-conforming mesh.

3.2. Scalability results

The size of the benchmark in Section 3.1 makes it a good candidate for a parallel scaling test. We consider a uniform mesh composed by $N_{el} = 120 \times 80 \times 32 = 307200$ elements with polynomial degree $r = 4$, leading to around 38.5 millions of unknowns for each scalar variable and a resolution of 125 m, and a non-conforming mesh composed by $N_{el} = 204816$ elements, yielding around 25.6 millions of unknowns for each scalar variable and a maximum resolution of 62.5 m. In order to put these figures into the appropriate context, it should be observed that global numerical weather prediction applications require $O(10^8 - 10^9)$ degrees of freedom at current operational resolutions. More specifically, simulations with a global horizontal resolution of around 9 km and 100 vertical levels employ almost a billion of degrees of freedom for each scalar variable. Meshes are being further refined to head towards the km-scale and the requirements necessary to reach the hectometric scale are a current research topic [14]. While the numbers might vary depending on the application, the resolutions and configurations in this Section where

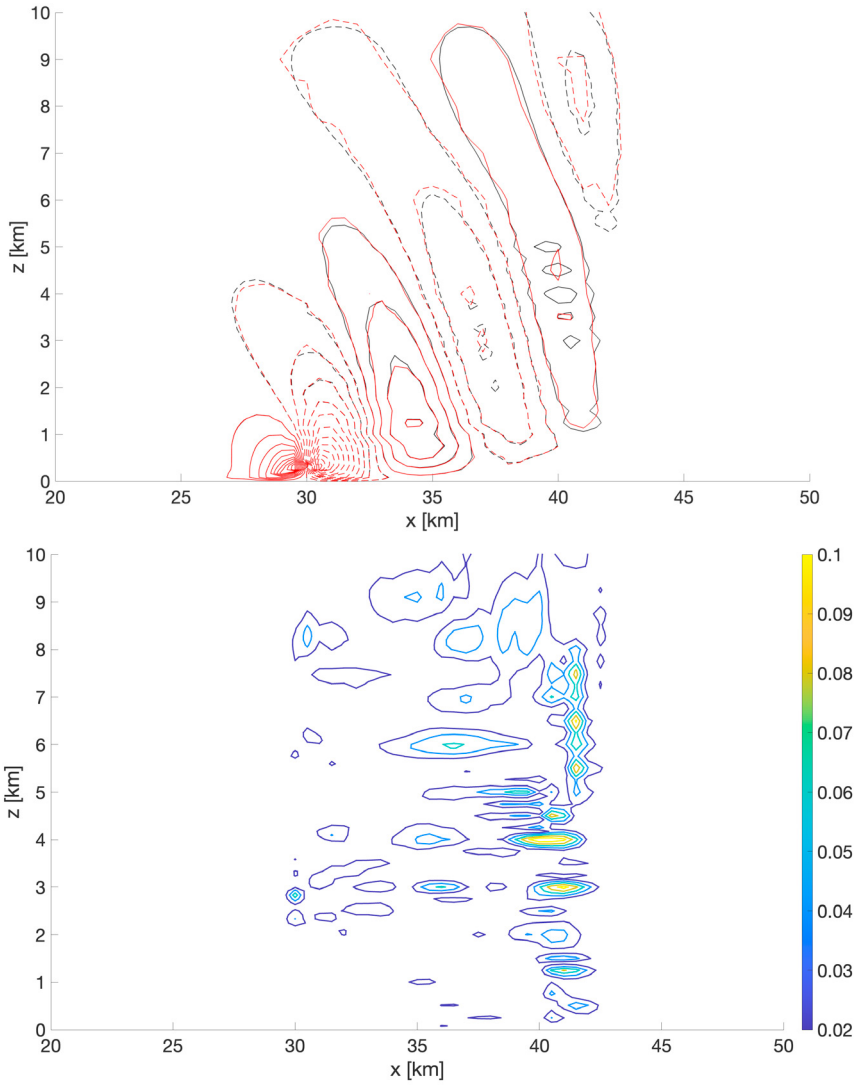


Fig. 3: As in Figure 2, but showing an $x - z$ slice at $y = 20$ km, and contours in the range $[-2.25, 2]$ m s^{-1} with a 0.2 m s^{-1} interval.

chosen striking a balance between feasibility with the available computational resources and degree of realism with a view to more realistic applications. For the runs whose results are presented here, the following optimization flags are employed throughout:

```
-O2 -funroll-loops -funroll-all-loops -fstrict-aliasing -Wno-unused-local-typedefs
```

which are the standard ones in the Release mode of `deal.II` (see https://www.dealii.org/developer/users/cmake_user.html). Unfortunately, because of the available computational resources, the scaling analyses reported in this Section could only be performed once for each of the different configurations.

The strong scaling test evaluates the wallclock time of the simulations at fixed computational load (resolution) and using an increasing amount of computational resources. More specifically, we use 1, 2, 4, 8 and 16 full MeluXina CPU nodes with 128 cores each. Hence, for the uniform mesh, we employ 300000, 150000, 75000, 37500, and 18750 degrees of freedom per core for each scalar variable, respectively, whereas for the non-conforming mesh, we employ around 200000, 100000, 50000, 25000, and 12500 degrees of freedom per core for each scalar variable, respectively.

In an ideal situation, the simulations speed up linearly with the amount of resources used. A good linear scaling is obtained up to 2048 cores, even with super-linear behaviour due to cache effects up to 1024 cores (Figure 4). In addition to the mentioned data locality of the discontinuous finite element approach, the favorable scaling is due to the matrix-free approach adopted in the solver, where no global sparse matrix is built and only the action of the linear operators is actually computed. These results represent a sensible improvement with respect to those previously presented in [20], which evaluated scalability up to approximately half the cores used in this paper. The apparent better behaviour of the uniform mesh for a larger number of cores is due to the fact that more degrees of freedoms are involved and, therefore, the role of communication costs is less evident.

In order to broaden the scope of the analysis, we perform an analogous scalability analysis with shared nodes, i.e. using computational nodes in which other jobs are simultaneously running. More specifically, we use 48, 96, 192, 384, 768 and 1536 MeluXina cores in shared nodes, and also perform runs at different polynomial degrees. One can easily notice that the use of shared nodes strongly degrades the parallel performance for more than about a thousand cores, and the effect is more marked at lower polynomial orders (Figure 4). Importantly, the speedup with the non-conforming mesh is comparable with that with the uniform mesh. Overall, the results confirm that the use of the Discontinuous Galerkin method, for which the stencil involves only the neighbours of each element, independently of the polynomial degree, provides an advantageous framework in terms of parallelization.

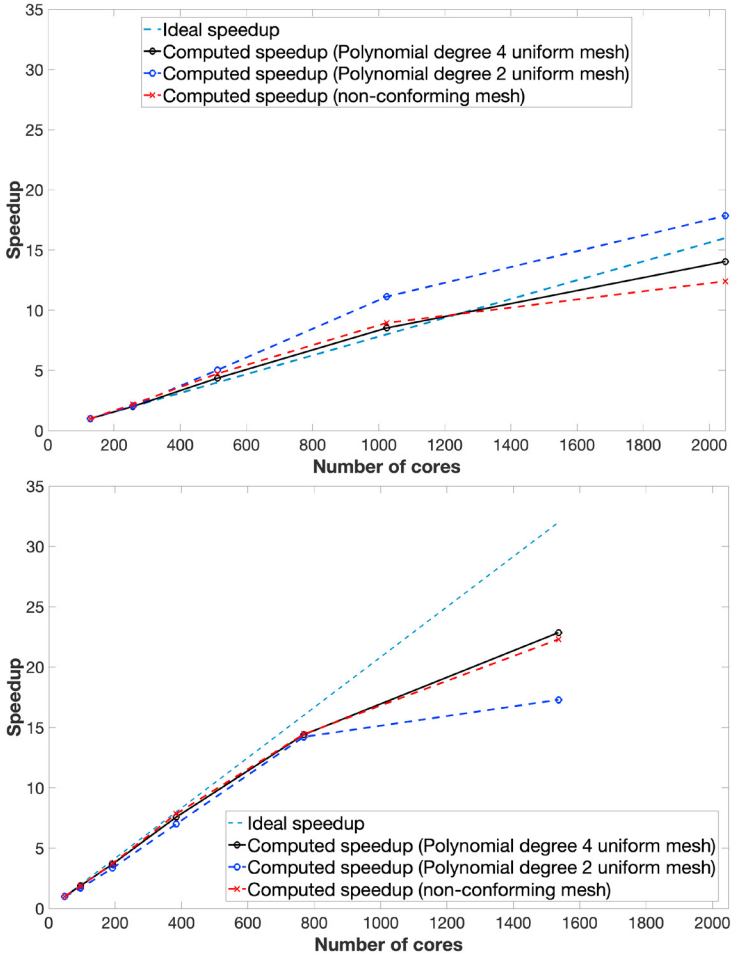


Fig. 4: Strong scaling analysis. Computed speedup as a function of the number of cores used in the simulations with the uniform mesh using polynomial degree 4 (solid black lines) and polynomial degree 2 (dashed blue lines), and with the non-conforming mesh (dashed red lines). Top: exclusive use of computational nodes. Bottom: shared use of computational nodes.

Moreover, a weak scaling analysis has been performed, using around 10^5 unknowns per core for each scalar variable and increasing the problem size for an increasing amount of resources. For ideal scaling, wallclock time should remain constant for increasing problem size. For the simulations both using the uniform mesh and using the non-conforming mesh, parallel performance is less than optimal in this case (Figure 5). However, the implementation actually outperforms the findings of previous `deal.II` studies [13], where the efficiency of the Navier-Stokes solver implemented using the same library as in this paper drops to 20%. In our simulations, a profiling study reveals that most of the time is spent in the fixed point loop to update the pressure variable, for which a non-symmetric linear system arises and a GMRES solver is therefore employed [19, 20, 21]. However, in terms of percentage time with respect to the total run time, the time spent in the linear solver is similar with increasing core counts (Figure 5). It is therefore expected that an improved solver strategy based on, e.g., advanced preconditioning techniques, will improve efficiency independently of the computational resources employed. A more detailed analysis of the loss of performance for the weak scaling will be the focus of future studies.

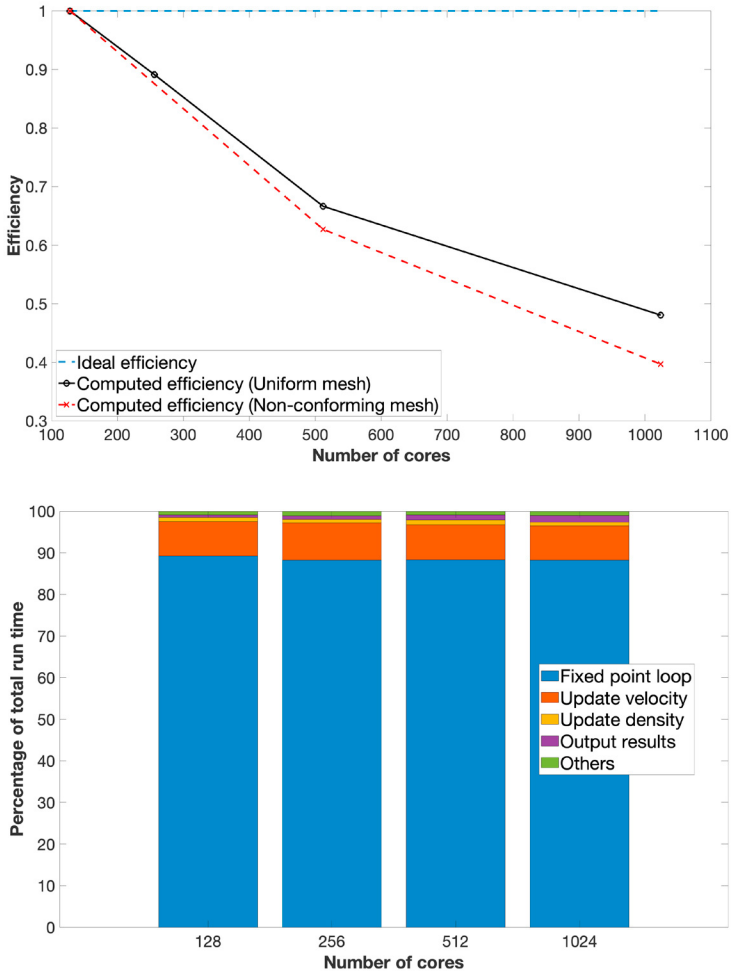


Fig. 5: Weak scaling analysis. Top: parallel efficiency as a function of number of cores used in the simulations with the uniform mesh (solid black line) and the non-conforming mesh (dashed red line). Bottom: distribution of the computational time spent in various blocks of the algorithm as a function of number of cores.

4. Conclusions

This paper has reported results of performance tests with a new high-order Discontinuous Galerkin model for atmospheric dynamics simulations using non-conforming mesh refinement. On a three-dimensional Cartesian benchmark of dry compressible fluid flow over orography with a stably stratified background atmosphere, the simulations using non-conforming meshes provide results that are equally accurate and more than 90% more efficient than simulations using uniform meshes that are standard in operational atmospheric modelling. In addition, the data locality features of the matrix-free, discontinuous finite element-based approach ensure good CPU scalability as measured in parallel runs with the state-of-the-art MeluXina EuroHPC facility.

These results open up a number of future avenues for investigations. First, enhanced realistic simulations should be performed, including more complex physical phenomena, in particular moist air, and the use of more general equations of state for real gases, for which the numerical method proposed in [19] has been already shown to be effective. Next, the development of a three-dimensional dynamical core in spherical geometry including rotation will enable the testing on more realistic atmospheric flows, for which more sizeable computational resources will be required. This more general and computationally heavier context will make it easier to fine-tune the performance and improve the findings in this paper, especially regarding weak scaling, and to gauge the viability of the proposed model towards full-fledged numerical weather prediction capability.

Acknowledgements

We would like to thank the two anonymous reviewers for their constructive suggestions and remarks, which have contributed to improving the manuscript. The simulations have been run thanks to the computational resources made available through the EuroHPC JU Benchmark And Development project EHPC-BEN-2024B03-045. This work has been partly supported by the ESCAPE-2 project, European Union's Horizon 2020 Research and Innovation Programme (Grant Agreement No. 800897).

References

[1] D. Arndt, W. Bangerth, B. M., M. Feder, M. Fehling, J. Heinz, T. Heister, L. Heltai, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, B. Turcksin, D. Wells, and S. Zampini. The deal.II library, Version 9.5. *Journal of Numerical Mathematics*, 31:231–246, 2023.

[2] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II: a general-purpose object-oriented finite element library. *ACM Transactions on Mathematical Software*, 33:24–51, 2007.

[3] T. Benacchio, W. O'Neill, and R. Klein. A blended soundproof-to-compressible numerical model for small-to mesoscale atmospheric dynamics. *Monthly Weather Review*, 142(12):4416–4438, 2014.

[4] V. Casulli and D. Greenspan. Pressure method for the numerical solution of transient, compressible fluid flows. *International Journal for Numerical Methods in Fluids*, 4:1001–1012, 1984.

[5] M. Dumbser and V. Casulli. A conservative, weakly nonlinear semi-implicit finite volume scheme for the compressible Navier-Stokes equations with general equation of state. *Applied Mathematics and Computation*, 272:479–497, 2016.

[6] F. Giraldo. *An Introduction to Element-Based Galerkin Methods on Tensor-Product Bases.* Springer Nature, 2020.

[7] F. Giraldo and M. Restelli. A study of spectral element and discontinuous Galerkin methods for the Navier-Stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases. *Journal of Computational Physics*, 227:3849–3877, 2008.

[8] J. Heinz, P. Munch, and M. Kaltenbacher. High-order non-conforming discontinuous Galerkin methods for the acoustic conservation equations. *International Journal for Numerical Methods in Engineering*, 124:2034–2049, 2023.

[9] A. Hellsten, K. Ketelsen, M. Sühring, M. Auvinen, B. Maronga, C. Knigge, F. Barmpas, G. Tsegas, N. Moussiopoulos, and S. Raasch. A nested multi-scale system implemented in the large-eddy simulation model PALM model system 6.0. *Geoscientific Model Development*, 14: 3185–3214, 2021.

[10] C. Kennedy and M. Carpenter. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. *Applied Numerical Mathematics*, pages 139–181, 2003.

[11] R. Klein. Semi-implicit extension of a Godunov-type scheme based on low Mach number asymptotics I: One-dimensional flow. *Journal of Computational Physics*, 121(2):213–237, 1995.

[12] M. Kopera and F. Giraldo. Analysis of adaptive mesh refinement for IMEX Discontinuous Galerkin solutions of the compressible Euler equations with application to atmospheric simulations. *Journal of Computational Physics*, 275:92–117, 2014.

[13] M. Kronbichler, A. Diagne, and H. Holmgren. A massively parallel two-phase flow solver for the simulation of microfluidic chips. *The International Journal of High Performance Computing Applications*, 2016.

[14] H. Lean, N. Theeuwes, M. Baldauf, J. Barkmeijer, G. Bessardon, L. Blunn, J. Bojarova, I. Boutle, P. A. Clark, M. Demuzere, et al. The hectometric modelling challenge: Gaps in the current state of the art and ways forward towards the implementation of 100-m scale weather and climate models. *Quarterly Journal of the Royal Meteorological Society*, in press, 2024. doi: 10.1002/qj.4858.

[15] S.-J. Lock, H.-W. Bitzer, A. Coals, A. Gadian, and S. Mobbs. Demonstration of a cut-cell representation of 3d orography for studies of atmospheric flows over very steep hills. *Monthly Weather Review*, 140(2):411–424, 2012.

[16] T. Melvin, T. Benacchio, B. Shipway, N. Wood, J. Thuburn, and C. Cotter. A mixed finite-element, finite-volume, semi-implicit discretization for atmospheric dynamics: Cartesian geometry. *Quarterly Journal of the Royal Meteorological Society*, 145:2835–2853, 2019.

[17] A. Müller, J. Behrens, F. Giraldo, and V. Wirth. Comparison between adaptive and uniform Discontinuous Galerkin simulations in dry 2D bubble experiments. *Journal of Computational Physics*, 235:371–393, 2013.

[18] G. Orlando and L. Bonaventura. An asymptotic-preserving scheme for Euler equations I: non-ideal gases, 2024. URL https://arxiv.org/abs/2402.09252.

[19] G. Orlando, P. Barbante, and L. Bonaventura. An efficient IMEX-DG solver for the compressible Navier-Stokes equations for non-ideal gases. *Journal of Computational Physics*, 471:111653, 2022.

[20] G. Orlando, T. Benacchio, and L. Bonaventura. An IMEX-DG solver for atmospheric dynamics simulations with adaptive mesh refinement. *Journal of Computational and Applied Mathematics*, 427:115124, 2023.

[21] G. Orlando, T. Benacchio, and L. Bonaventura. Robust and accurate simulations of flows over orography using non-conforming meshes. *Quarterly Journal of the Royal Meteorological Society*, in press, 2024. doi: 10.1002/qj.4839.

[22] G. Orlando, T. Benacchio, and L. Bonaventura. Impact of curved elements for flows over orography with a Discontinuous Galerkin scheme. *Journal of Computational Physics*, 519:113445, 2024.

[23] J. Steppeler, R. Hess, G. Doms, U. Schättler, and L. Bonaventura. Review of numerical methods for nonhydrostatic weather prediction models. *Meteorology and Atmospheric Physics*, 82:287–301, 2003.

[24] L. Yelash, A. Müller, M. Lukáčová-Medvid'ová, F. Giraldo, and V. Wirth. Adaptive discontinuous evolution Galerkin method for dry atmospheric flow. *Journal of Computational Physics*, 268:106–133, 2014.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 43–52

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

OpenWebSearch.eu - Building an Open Web Index on EuroHPC JU Infrastructures

Michael Granitzer[a,*], Mohamad Hayek[b], Sebastian Heineking[c], Gijs Hendriksen[d], Martin Golasowski[e], Michael Dinzinger[a], Saber Zerhoudi[a]

[a] *Chair of Data Science, University of Passau, Passau, Germany*
[b] *Leibniz-Rechenzentrum (LRZ), Munich, Germany*
[c] *University of Leipzig, Leipzig, Germany*
[d] *Radboud University, Nijmegen, The Netherlands*
[e] *IT4I, VSB – TU of Ostrava, Ostrava, Czech Republic*

Abstract

The OpenWebSearch.eu project aims to develop an Open Web Index (OWI), an openly accessible data structure that supports the creation of web search engines. Building such an index requires a data- and compute-intensive pipeline for cleaning, pre-processing, enriching and indexing large amounts of web data. Beyond search, the availability of clean and preprocessed web data is also crucial for fields like web analytics and generative AI. This paper presents our approach to constructing the OWI using High-Performance Computing (HPC) resources from both EuroHPC JU and non-EuroHPC JU data centers. We contribute in two main areas: first, by detailing the development of pre-processing and indexing pipelines embedded in HPC workflows; and second, by describing the iRODS-based federated storage infrastructure and the LEXIS[1] platform that will manage the cross-data centre workflows and facilitate the publication of the OWI as daily datasets. During the alpha phase, from October 2023 to April 2024, we processed approximately 76 TB of web data, encompassing over 2 billion URLs. By addressing the challenges of large-scale web data processing and retrieval, this work lays the foundation for an innovative, competitive, and transparent web search ecosystem, while also supporting the development of European generative AI solutions.

Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day

Keywords: Web Data and Web Search; Open Web Index; Information storage and retrieval

1. Introduction

The Web is a vital resource for various applications, including search engines, data analytics, and generative AI. However, harnessing web data presents significant challenges due to high demands on hardware, technical complexity,

[1] https://portal.lexis.tech

* Michael Granitzer Tel.: +49-851-509-3300.
E-mail address: michael.granitzer@uni-passau.de

1877-0509
Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day
10.1016/j.procs.2025.02.259

legal constraints, and data quality issues. These barriers particularly affect small innovators and researchers, making it difficult for them to compete with major industry players. The resulting lack of competition reinforces the dominance of a few large search engines, stifling innovation and reducing diversity in search services, analytics, and generative AI.

To address these challenges, we present the Open Web Index (OWI) [4, 6], an openly available index of the web created through a collaboration of High Performance Computing (HPC) centres in the OpenWebSearch.eu project. Creating an index of the Web usually requires large-scale crawling of web content, data cleaning, data preprocessing, deduplication, and indexing. The sheer scale of the data presents the biggest challenge when indexing the web, particularly in terms of computing and storage resources. Just for comparison: it is assumed that Google's index contained roughly 400 billion web pages in 2020, according to information from a lawsuit[2], which would equate to several 100 PB in storage size. Clearly small single organisations, research institutes, and even larger companies cannot provide the necessary resources for creating such an index.

By utilising Europe's HPC infrastructure, particularly resources provided by EuroHPC JU, the OpenWebSearch.eu project aims to create an openly available index of the Web, called the Open Web Index. While the index size will remain smaller than commercial indices, it is the first openly available index following open source and open data principles. Although the EuroHPC JU initiative provides significant HPC infrastructure for compute-intensive tasks, using web data for search, analytics, or AI also presents data-centric and IO-centric challenges. These include indexing web data for search, preprocessing data (including natural language processing for semantic enrichment), and computing AI-based embeddings for dense retrieval and Retrieval Augmented Generation (RAG).

This paper presents our current pipelines and achieved results for creating an Open Web Index on a collaborative network of HPC centres, both within EuroHPC JU - particularly IT4Innovations National Supercomputing Centre (IT4I) and CSC - IT Center for Science (CSC) - and outside of EuroHPC JU - particularly the Leibniz Supercomputing Center in Munich (LRZ), CERN and the German Aerospace Center (DLR). Specifically, we address the following contributions:

- Development of robust preprocessing and indexing pipelines using HPC resources for converting crawled data into a shareable and extensible index.
- Cross-data center execution of HPC workflows and dataset management based on the LEXIS Platform[3] and central authentication and access management via B2ACCESS.
- A federated, iRODS[3]-based storage system for storing and sharing workflow outputs across HPC centers.
- Tools for pulling, pushing and querying the index datasets computed via HPC resources, promoting collaborative management of web data and the Open Web Index.

These contributions collectively address the challenges of processing large-scale web data and set the foundation for an open, collaborative web search infrastructure. However, our work also goes beyond creating an open web index through the establishment of federated data storage across HPC data centres, a single point of execution for HPC jobs across data centres, and data set publishing and management tools for managing very large data sets.

The paper is structured as follows: In section 2 we start by describing the vision of an Open Web Index and describe the application scenario. Afterwards, we give an overview on the HPC workflows 3 including individual components, steps and the underlying storage concept. Section 4 outline the achieved results so far in terms of HPC-utilisation and data set size, while section 5 concludes the work.

2. Application Scenario: Vision of an AI-powered Open Web Index

We envision the Open Web Index (OWI) as a distributed information system built on a federated storage infrastructure. This system allows search engines and web data-centric applications to retrieve data from storage systems

[2] https://zyppy.com/seo/google-index-size/
[3] Integrated Rule-Oriented Data System https://irods.org/

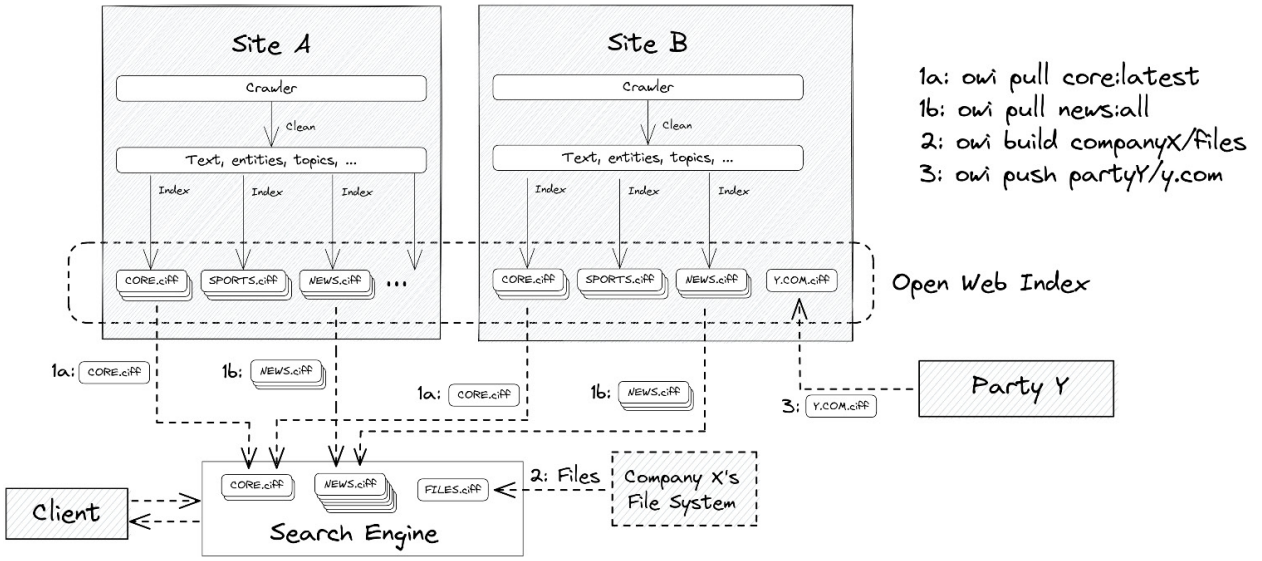


Fig. 1. General architecture of the OWI and its interaction with search engines for index retrieval.

through flexible selection methods. Users can choose horizontal slices based on date and language or vertical slices by selecting specific attributes of interest (e.g., title, plain text, and URL of a web page). This "slice & dice" concept enables search engines to obtain data at regular intervals according to their specific requirements.

Pre-computed indices are provided in the Common Index File Format (CIFF) [9], ensuring compatibility with existing open-source search engines such as Lucene[4], (Py)Terrier [10, 12], and PISA [11]. Beyond traditional inverted indices stored in CIFF format, modern AI-based retrieval applications typically rely on dense retrieval of sub-document units for use with Large Language Models, a technique known as Retrieval Augmented Generation (RAG) [8]. Dense retrieval, particularly in the RAG context, necessitates the computation of dense vector embeddings, which usually involves applying (Large) Language Models to chunked web text.

We further envision that specifications for search engines can be stored in a descriptive manner, detailing not only the required slices and index requirements but also search and configuration modalities (e.g., ranking mechanisms, search and database backends). Storing search engine declarations in this way could conceptually yield a system similar to Docker Hub, but specifically for web search and web analytics applications.

Beyond data retrieval, the federated storage would also allow users to push data, such as embeddings for dense retrieval or annotations of web content. This push-pull-slice paradigm for web data would form the basis for collaborative management of web data on top of HPC-backed federated data storage. Figure 1 illustrates the general architecture of the OWI and its interaction with search engines.

3. High-Performance-Computing Pipelines for creating an Open Web Index

High-Performance Computing (HPC) centers play a crucial role in creating an Open Web Index by providing the necessary computational power for big data processing. Web data usually consists of terabytes to petabytes, necessitating meticulous planning of storage strategies. Furthermore, the compute and storage resource demands for building an OWI can exceed the capacity of a single HPC center, especially when considering potential future extensions to generative AI. In order to pool the necessary resource for building the OWI, we therefore aim to coordinate workflows across multiple data centers and have a federated storage across HPC centers.

Figure 2 provides an overview of the workflows in one data center, consisting of the following stages:

[4] https://lucene.apache.org/

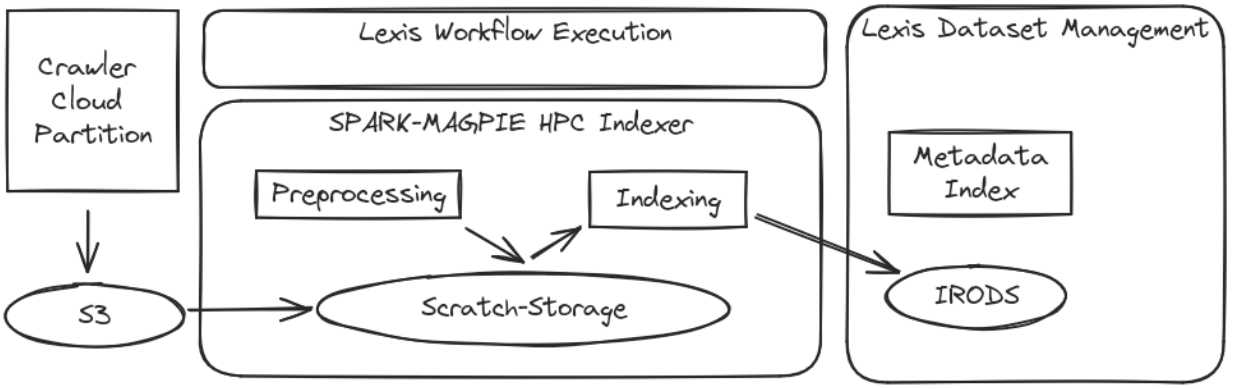


Fig. 2. HPC Workflows for a single data center

1. **Data Ingestion**: Data is ingested via a separate crawling system [2] delivering approximately 2 TiB per day. Data is staged in S3, with crawlers currently running at all participating data centers, coordinated via a central frontier at CERN. This approach allows for the distribution of crawling and HPC load among different HPC centers.
2. **Workflow Execution**: Workflows are managed and executed using the LEXIS Platform and run across three HPC centers: IT4I, LRZ, and CSC. Workflows begin by moving S3 data to cluster scratch space. The processing is run on an Apache Spark cluster that we create on the HPC infrastructure using the script collection Magpie[5].
3. **Workflow Components**: The workflows consist of two main components that (1) preprocess[6] the data to create Parquet files, and (2) index and partition[7] the Parquet files. The result is a set of CIFF index files [9, 6]. Intermediate results between indexing and preprocessing are stored in scratch space and only moved to the federated iRODS storage after indexing is completed. At this stage, metadata such as size and crawl information is added to the files.
4. **Federated iRODS Storage**: iRODS serves as the backend for storing and distributing datasets. Per day and data center, we store a set of Parquet files containing preprocessing results and CIFF index files, partitioned according to year, month, day, and language of the data. The data can be consumed via the LEXIS Platform.
5. **Authentication and Access Management**: LEXIS integrates authentication via B2ACCESS, ensuring proper control over access to workflow execution and data.
6. **Tooling for Data Access**: On top of the LEXIS platform, we developed the OWI Python client *owilix*[8] which provides OWI-specific dataset management, including means for pushing and pulling datasets as well as the ability to conduct SQL queries remotely over datasets. The client uses the Python interface to the LEXIS Platform - py4lexis[9] which wraps the LEXIS Platform API to a convenient Python library.

In the following subsections, we provide more details on the individual steps.

3.1. Preprocessing

Preprocessing is the first step in the Open Web Index pipeline. It is primarily handled by two software components, Resiliparse and Resilipipe, that work in tandem to efficiently process and analyze web archive data at scale.

[5] https://github.com/LLNL/magpie
[6] https://opencode.it4i.eu/openwebsearcheu-public/preprocessing-pipeline
[7] https://opencode.it4i.eu/openwebsearcheu-public/spark-indexer
[8] https://opencode.it4i.eu/openwebsearcheu-public/owi-cli
[9] https://opencode.it4i.eu/lexis-platform/clients/py4lexis

3.1.1. Resiliparse: Core Parsing Library

Resiliparse[10] is an open-source web archive processing library and as such the foundational component for reading and parsing the web crawls. The native Python module is designed to be both efficient and robust to be able to process large amounts of web documents while handling the diversity that comes with that data. This library facilitates the rapid and safe processing of potentially malformed or malicious web content, emphasizing minimal assumptions about data well-formedness. The two main modules of Resiliparse are:

- **FastWARC**: The FastWARC library is a faster and more efficient alternative to existing WARC parsing libraries. It supports uncompressed WARCs as well as gzip- und lz4-compressed archives.
- **Resiliparse Core**: The core library offers a collection of tools to process web data. These include (1) efficient HTML parsing and DOM processing, (2) reliable character encoding detection and conversion to Unicode, (3) fast detection of common MIME types, (4) fast heuristic-based main content extraction [1], and (5) basic but fast language detection.

Resiliparse is written primarily in native C and C++ using Cython, with some parts written in Python, to offer significant memory and CPU efficiency. The tools offered by Resiliparse collectively support the extraction of plaintext and main content from web pages with high reliability and speed.

3.1.2. Resilipipe: Scalable Content Analysis Framework

Building upon Resiliparse, Resilipipe is a scalable framework implemented for cluster-based web content analysis. It is built to handle large amounts of web archive data and extracting valuable metadata that enriches the Open Web Index. Core features of Resilipipe include:

- **Cluster Deployment**: Resilipipe uses Apache Spark for parallel processing of WARC files, effectively distributing the workload across cluster nodes. In combination with Magpie and our collection of Spark deployment scripts[11] it can be easily deployed on HPC clusters with common resource managers such as Slurm or Moab.
- **Content Analysis**: The framework uses Resiliparse to efficiently read and parse WARC files. It then extracts metadata from the parsing output such as MIME types, languages, and web page categories. Advanced metadata related to geolocation, microdata, and JSON Linked Data are also extracted to enhance search engine functionalities.
- **Pass-through Metadata:** Resilipipe also allows to pass-through metadata from the crawler to indexing and storage. This becomes relevant as the crawler can provide information relevant to further processing steps, such as, for example, the fetch speed of a web page or the allowed usage pattern. Particularly important is *gen-AI flag*, which indicates whether the web page is allowed to be used with generative AI or not.
- **Modular Design**: Resilipipe supports the integration of user-created content analysis modules via a standardized interface. This allows for the extension of its capabilities based on project needs and third-party contributions. With the help of TIREx[12], The Information Retrieval Experiment platform, we can evaluate content analysis modules in a scalable and reproducible way.

3.2. Indexing

The indexing phase is a crucial step in the Open Web Index pipeline, transforming the preprocessed content into a usable, efficient format for search and retrieval. This process is implemented as part of our multi-tiered architecture, focusing on creating inverted files that can be easily consumed by various search engine implementations. Our indexing process takes the cleaned and enriched content from the preprocessing stage and converts it into an inverted file structure, fundamental to efficient information retrieval. Implemented as a Spark batch job, the indexing process

[10] https://resiliparse.chatnoir.eu
[11] https://opencode.it4i.eu/openwebsearcheu-public/spark-deployment
[12] https://www.tira.io/tirex

runs across our HPC infrastructure, allowing for parallel processing of large volumes of data. We partition the index into daily "index-shards" based on metadata derived during preprocessing, such as topic and language, enabling the creation of semantically coherent subsets of the full web index.

This approach to index creation offers significant flexibility. By leveraging various metadata types, we can create specialized index shards. For instance, language-based shards can support country-specific search engines, while topic-based shards can focus on specific areas like news or sports. The indexer produces Common Index File Format (CIFF) files, a standardized format that ensures compatibility with a wide range of open-source search engines.

CIFF[13], a Protobuf schema, describes inverted files in a structured, consistent, and minimal format. A CIFF file contains a header with basic collection statistics, term records including document/collection frequencies and "postings" (i.e. in which document each term occurs), and document records with identifiers and lengths. This standard provides the essential information needed to build a successful search engine, facilitating easy import and transformation of data into various search engine architectures.

3.3. Cluster Deployment Strategy

Our cluster deployment strategy is critical for managing the vast scale of data processed. The indexing process is deployed as an Apache Spark job within our multi-tier cluster setup, optimized for the specific demands of index creation. We utilize tools like Magpie to automate the deployment process, aligning with the scheduling systems of various clusters (e.g., SLURM) to ensure efficient resource allocation and job management.

The data flow in our system is designed for efficiency and scalability. Post-indexing, the CIFF files are stored in our federated iRODS storage system, ready for distribution and use by various search engine implementations. Our current deployment spans multiple HPC centers, including LRZ, IT4I and CSC, with ongoing efforts to scale up the indexer to handle the continuously growing volume of crawled content. Data access requires authentication via the European EUDat / B2ACCESS service [16] which has been integrated with the data center's access and authentication systems.

This comprehensive indexing and deployment strategy enables us to efficiently process and index vast amounts of web data, creating a flexible and powerful foundation for the Open Web Index. By leveraging the power of distributed computing and standardized formats, we're able to create a resource that can support a wide range of search and analysis applications, fostering innovation in web search technology.

3.4. IRODS-Storage and the LEXIS Platform

The Integrated Rule-Oriented Data System (iRODS) is an open-source data management middleware that allows the creation of a unified view over different geographically distributed storage systems[14]. It also maintains a rich database that allows the assignment of metadata to single files or directories and corresponding fast querying capabilities. The LEXIS Platform, initiated through a Horizon 2020 project (GA #825532), adopted iRODS as the main component of its Distributed Data Infrastructure (DDI). In addition, LEXIS also indices iRODS metadata in an ElasticSearch engine to ensure fast full-text queries. On top of iRODS, an asynchronous Staging API is deployed at each computing centre to stage data between storage systems and HPC clusters. This mechanism is leveraged in the indexing and pre-processing workflows executed on HPC resources through the LEXIS Platform.

3.5. Dataset-based Publishing of Daily-index Shard

While the crawlers run continuously within the cloud partitions of the involved HPC centers, HPC workflows are executed once per day to process the crawled data from the previous day at each specific HPC center. Consequently, we publish a daily index slice in the form of a dataset, which is made available under a unique UUID-based identifier via iRODS. The metadata for these datasets partially follows the DataCite Metadata Standard 4.5 [5], containing information about creators, publishers, titles, and license information.

[13] https://github.com/osirrc/ciff

The data is partitioned using a HIVE-like access path scheme[15]: `year=<year>/month=<month>/day=<day>/language=<lang>`. We chose HIVE partitioning because it allows the use of standard tools and simplifies file-based merging of datasets. Additionally, every dataset includes a `changelog.json` file at the root, which indicates any changes made. This feature is necessary to log dataset modifications, such as those required for legal reasons when an external entity requests the removal of certain data items for privacy concerns.

Publishing index shards as daily datasets, instead of updating a single web index daily, offers several advantages:

1. The index is broken down into smaller, self-contained units, typically ranging around 20 GB.
2. Access to individual increments is based on time and metadata, making management easier.
3. Metadata is associated with each dataset, facilitating organization and querying.
4. Complex workloads can be partitioned on a per-dataset basis.
5. Data usage can be tracked per dataset, which is particularly useful when monitoring training data for use in generative AI systems.

However, managing numerous datasets can become complex. To address this, we developed an open-source command-line dataset management tool, *owilix*[14], built on top of LEXIS and B2ACCESS[15]. This tool offers several key features:

1. **Pulling Data:** The primary function of *owilix* is to pull data from remote systems using a specifier concept—essentially, a short query string that describes the data center, time range, and metadata. For example, the command `all:latest/license=OWIV1` pulls the latest datasets from all data centers that have the OWIV1 license. Data is transferred file-based, allowing for additional file filters to limit the data amount.
2. **Pushing Data:** To encourage collaboration in data cleaning, enrichment, and provision, *owilix* also allows users to modify pulled data and subsequently push those modifications back to the server. This pushing process is file-based, enabling, for example, the addition of annotations to web content or the integration of additional indices, such as dense embeddings for Retrieval Augmented Generation or probabilistic indices like Bloom filters.
3. **SQL Queries:** While push and pull operations are file-based, allowing selection based on pre-partitioned data, *owilix* also supports running SQL queries against datasets. This capability is powered by DuckDB[16], which efficiently queries Parquet files on local and remote filesystems. With DuckDB's predicate pushdown and query optimization strategy for Parquet files[13], combined with iRODS-mounted datasets, we can efficiently select rows and columns, reducing data transmission between server and client. The use of the parquet format follows best-practices in big data setup, which have shown to deliver good query performance[7].

Since *owilix* allows the creation of new datasets from existing ones, such as through SQL queries, we also implemented a mechanism to track the provenance of datasets. This is done using a URI-based provenance schema, which references the original dataset and specifies any transformations applied. For example, the URI `owi://UUID/select=*&where="url like de"` links to a dataset with the specified UUID, filtered using the indicated `SELECT` and `WHERE` statements.

Overall, our dataset-oriented approach reduces the complexity of managing large web datasets and provides powerful tools for users to slice and dice the data as needed, making web data manageable even in low-resource settings. An initial URI-based provenance schema allows to track dataset deviates and allows to establish collaborative data curation workflows.

[14] https://opencode.it4i.eu/openwebsearcheu-public/owi-cli
[15] https://b2access.eudat.eu/
[16] https://duckdb.org/

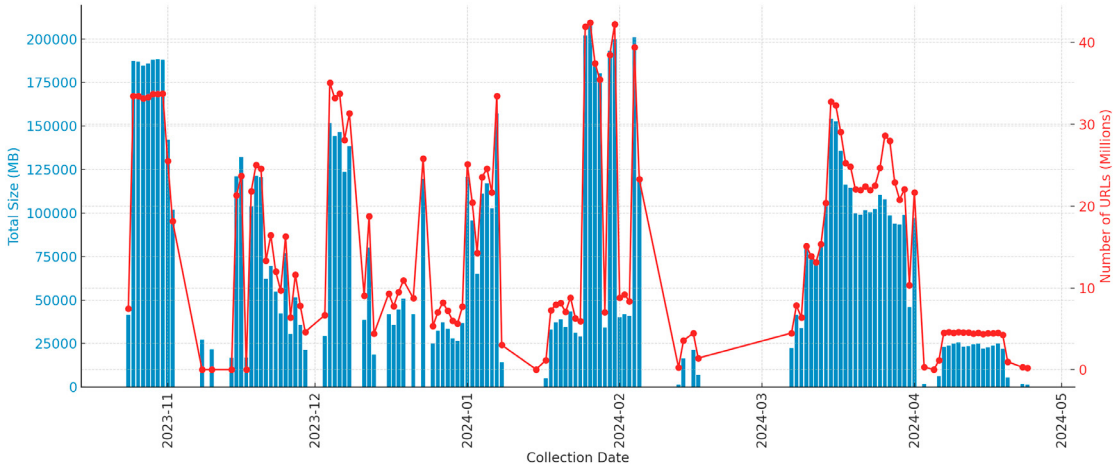


Fig. 3. Dataset size (in MB) and number of million URLs per dataset between October 2023 and April 2024.

4. Results

In this section, we present the results of the first development phase (alpha phase) in terms of data volume and HPC resource utilization. The alpha phase, which began on October 23, 2023, and continued until April 30, 2024, provided valuable insights into the processing and storage capacities required, even though datasets were not generated daily due to ongoing pipeline development. More recent data can be accessed through our online dashboard at https://openwebindex.eu.

4.1. Development and Deployment Status

During the first phase, we had the system running at two data centers: The EuroHPC JU partner IT4Innovations National Supercomputing Center (IT4I) in Ostrava and the Leibniz Supercomputing Center (LRZ) in Munich. Both sites had crawlers running ingesting data to S3 and running the above mentioned preprocessing and indexing pipeline.

After the successful first proof-of-concept, we have extended the setup to five data centers running different components.

The iRODS-based federated data storage is currently deployed at IT4I, LRZ, the German Aerospace Center (DLR) and CSC – IT Center for Science. The iRODS federation has been partially established between the different centers allowing users to access the available data from all the sites available. The access to the data is continually improved based on feedback from both project partners and external users.

The HEAppE middleware that allows the LEXIS orchestrator to access the different HPC resources is deployed at LRZ (Linux Cluster), IT4I (Karolina), and CSC (Puhti cluster). The deployment of HEAppE at DLR is in progress and will allow the LEXIS orchestrator to execute workflows on the newly established Terrabyte infrastructure for geo-spatial data analysis.

4.2. Dataset Size and Index Partitions

For the alpha phase we have crawled approximately 76 TB of raw data on in total 127 different days, which is approximately 500 GB per day. After running the preprocessing and index pipelines, we ended up with in total roughly 2 billion URLs (2,013,843,377) - around 15.9 Million URLs per day - with a cleaned dataset size of ca. 9.2 TB distributed over 172 datasets. Figure 3 shows the dataset distribution in terms of MB and Millions of URLs per Dataset. From the numbers we can derive a raw HTML size of roughly 38 kB and a plain text size of around 5 kB per web page (excluding multimedia elements in both cases). Note that the plain text also contains microformats and hence has some redundancy.

The output datasets are partitioned into daily language-based shards according to ISO-639 Part 3[17]. On average, each daily index slice contains between 300 and 450 shards. The language distribution is skewed significantly, where roughly 40% of the index consists of English documents, and there is a long tail of languages for which we only crawled a few documents. Note that it is as of yet unclear how accurate our language detection module is and how this influences the shard distribution of our datasets. The size of the resulting output datasets (i.e. Parquet files containing cleaned text and extracted metadata combined with the inverted files in CIFF format) is roughly 10-15% of the size of the original (gzipped) WARC files.

4.3. HPC Utilization

For preprocessing and indexing, we usually run our workflows on 4-6 HPC nodes, on which we allocate 12-24 cores depending on the node's available memory. Most of the time (70-90%) is spent on preprocessing and metadata extraction/enrichment, which takes roughly 1-2 minutes per WARC file. For one of the larger datasets (~14k WARC files of 100 MB each), which resulted in an output dataset of ~150 GB the processing times were distributed as follows: preprocessing and enrichment took ~3.5 hours; indexing finished in ~1.5 hours; and copying the data to the iRODS-based DDI took an additional ~1.5 hours.

When datasets grow larger, we can increase the horizontal scaling factor by assigning more nodes to each HPC job in order to ensure index shards can still be produced daily. In case of insufficient capacity within a HPC center, we would have to adjust the amount of data to be processed, either by reshuffling crawled data or adjusting crawling capacities for this particular data center. Nevertheless, our distributed approach allows to scale vertically per HPC data center as well as horizontally accross data centers.

5. Conclusion

In this paper we have presented the utilisation of EuroHPC JU resources for creating an open index for web search, the so-called Open Web Index. Scaling up the necessary storage and compute requires a collaborative effort between HPC centers. To execute workflows across data-centers and for managing the related dataset we have presented the LEXIS platform and our iRODS-based federated data storage. We also developed `owilix` as dataset management tool on top of the federated storage, which allow pushing, pulling and querying of data. In the first development phase lasting from October 23 to April 24 we processed 76 TB of web data which equates to approximately 2 billion URLs. HPC workflows take ~6.5 hours per day and data-center, utilizing an modes amount of 4-6 HPC nodes.

After this successful alpha phase, we significantly increased the data ingestion from 0.5 TB / day up to 3 TB /day with an aim to reach 5 TB / day. This would increase the number of URLs per day by a factor of 6-10, reaching around 95 million to 159 million URLs per day. Furthermore, we increased the up time of the crawler significantly so that we can deliver these numbers on a daily basis. We also assume that compute resources will increase linearly by a factor of 10 yielding to significant compute load for HPC centers. This would again increase when utilizing generative AI based dense indexing techniques, which utilize large language models. However, we hope that in the next 6 months we can provide stable services, delivering daily index patches of 100 million to 200 million URLs per day and thus support innovators and research in need for web data.

Acknowledgements

This work is part of the OpenWebSearch.eu project. The OpenWebSearch.eu Project is funded by the EU under the GA 101070014 and we thank the EU for their support. This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254). We acknowledge EuroHPC Joint Undertaking for awarding us access to Karolina at IT4Innovations, Czech Republic and LUMI at CSC, Finland.

[17] https://iso639-3.sil.org/

References

[1] Bevendorff, J., Gupta, S., Kiesel, J., Stein, B., 2023. An empirical comparison of web content extraction algorithms, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2594–2603.

[2] Dinzinger, M., Al-Maamari, M., Zerhoudi, S., Istaiti, M., Mitrović, J., Granitzer, M., . OWler: Preliminary results for building a collaborative open web crawler, in: Open Search Symposium 2023, 4-6 October 2023, CERN, Geneva, Switzerland. URL: https://zenodo.org/records/10581841, doi:10.5281/zenodo.10581841. publisher: Zenodo.

[3] Golasowski, M., Martinovič, J., Křenek, J., Slaninová, K., Levrier, M., Harsh, P., Derquennes, M., Donnat, F., Terzo, O., 2022. The lexis platform for distributed workflow execution and data management, in: HPC, Big Data, and AI Convergence Towards Exascale. Taylor & Francis.

[4] Granitzer, M., Voigt, S., Fathima, N.A., Golasowski, M., Guetl, C., Hecking, T., Hendriksen, G., Hiemstra, D., Martinovi'c, J., Mitrovi'c, J., et al., 2023. Impact and development of an open web index for open web search. Journal of the Association for Information Science and Technology doi:10.1002/asi.24818.

[5] Group, D.M.W., 2024. Datacite metadata schema for the publication and citation of research data and other research outputs. URL: https://doi.org/10.14454/g8e5-6293.

[6] Hendriksen, G., Dinzinger, M., Farzana, S.M., Fathima, N.A., Fr"obe, M., Schmidt, S., Zerhoudi, S., Granitzer, M., Hagen, M., Hiemstra, D., et al., 2024. The open web index: Crawling and indexing the web for public use, in: European Conference on Information Retrieval, Springer. pp. 130–143.

[7] Ivanov, T., Pergolesi, M., 2020. The impact of columnar file formats on sql-on-hadoop engine performance: A study on orc and parquet. Concurrency and Computation: Practice and Experience 32, e5523.

[8] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., et al., 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems 33, 9459–9474.

[9] Lin, J., Mackenzie, J., Kamphuis, C., Macdonald, C., Mallia, A., Siedlaczek, M., Trotman, A., de Vries, A., 2020. Supporting interoperability between open-source search engines with the common index file format, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2149–2152.

[10] Macdonald, C., Tonellotto, N., 2020. Declarative experimentation in information retrieval using pyterrier, in: Proceedings of ICTIR 2020.

[11] Mallia, A., Siedlaczek, M., Mackenzie, J., Suel, T., 2019. Pisa: Performant indexes and search for academia. Proceedings of the Open-Source IR Replicability Challenge .

[12] Qunis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Lioma, C., 2006. A high performance and scalable information retrieval plateform, in: SIGR workshop on open source information retrieval.

[13] Raasveldt, M., Mühleisen, H., 2019. Duckdb: an embeddable analytical database, in: Proceedings of the 2019 International Conference on Management of Data, pp. 1981–1984.

[14] Rajasekar, A., Moore, R., Hou, C.Y., Lee, C.A., 2010. iRODS primer: integrated rule-oriented data system. Morgan & Claypool Publishers.

[15] Thusoo, A., Sarma, J.S., Jain, N., Shao, Z., Chakka, P., Zhang, N., Antony, S., Liu, H., Murthy, R., 2010. Hive-a petabyte scale data warehouse using hadoop, in: 2010 IEEE 26th international conference on data engineering (ICDE 2010), IEEE. pp. 996–1005.

[16] de Witt, S., Lecarpentier, D., van de Sanden, M., Reetz, J., 2017. Eudat-a pan-european perspective on data management, in: 2017 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), IEEE. pp. 1–5.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 53–62

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

EuroLLM: Multilingual Language Models for Europe

Pedro Henrique Martins[a], Patrick Fernandes[b,c], João Alves[a], Nuno M. Guerreiro[a,b,d], Ricardo Rei[a], Duarte M. Alves[b], José Pombal[a,b], Amin Farajian[a], Manuel Faysse[d,e], Mateusz Klimaszewski[f], Pierre Colombo[d,g], Barry Haddow[f,h], José G. C. de Souza[a], Alexandra Birch[f,h], André F. T. Martins[a,b]

[a] *Unbabel, Portugal*
[b] *Instituto de Telecomunicações, Instituto Superior Técnico, Portugal*
[c] *Carnegie Mellon University, United States of America*
[d] *MICS, CentraleSupelec, Université Paris-Saclay, France*
[e] *Illuin Technology, France*
[f] *University of Edinburgh, Scotland*
[g] *Equall, France*
[h] *Aveni, Scotland*

Abstract

The quality of open-weight LLMs has seen significant improvement, yet they remain predominantly focused on English. In this paper, we introduce the *EuroLLM* project, aimed at developing a suite of open-weight multilingual LLMs capable of understanding and generating text in all official European Union languages, as well as several additional relevant languages. We outline the progress made to date, detailing our data collection and filtering process, the development of scaling laws, the creation of our multilingual tokenizer, and the data mix and modeling configurations. Additionally, we release our initial models: EuroLLM-1.7B and EuroLLM-1.7B-Instruct[1] and report their performance on multilingual general benchmarks and machine translation.

Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day

Keywords: LLMs; Multilinguality; European Languages.

1. Introduction

Large language models (LLMs) are driving significant advancements in natural language processing and AI, as demonstrated by OpenAI's GPT series and Anthropic's Claude. LLMs are pre-trained on vast amounts of unlabelled data to perform a self-supervised task (*e.g.,* next word prediction or missing word prediction), enabling them to develop

[1] The EuroLLM models are available here.

* Corresponding author: pedro.martins@unbabel.com

Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day
10.1016/j.procs.2025.02.260

a deep understanding of language. These pre-trained LLMs can already perform various downstream tasks, often leveraging in-context learning techniques, but are typically fine-tuned to better follow natural language instructions, improve performance on specific tasks, and adhere to safety protocols.

However, the most advanced models are owned by large corporations with a piecemeal commitment to open science. Moreover, despite the growing availability of open-weight LLMs (*e.g.,* LLaMA, Mistral, or Gemma [44, 17, 41]), these are predominantly limited to English and a few high-resource languages, leaving out many European languages. To address this gap, we have started the EuroLLM project with the goal of creating a suite of LLMs capable of understanding and generating text in all European Union languages (Bulgarian, Croatian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Greek, Hungarian, Irish, Italian, Latvian, Lithuanian, Maltese, Polish, Portuguese, Romanian, Slovak, Slovenian, Spanish, and Swedish) as well as some additional relevant languages (Arabic, Catalan, Chinese, Galician, Hindi, Japanese, Korean, Norwegian, Russian, Turkish, and Ukrainian).

Thus far, we have explored the methodologies for training multilingual LLMs and have developed our initial models: EuroLLM-1.7B and EuroLLM-1.7B-Instruct. This process involved:

- Collecting and filtering a large volume of text data for all the targeted languages from various sources, as detailed in §2.
- Defining the mixture of data that composes the training corpus used to train the model. We describe the decisions we took in §2. These decisions were based on scaling laws and on the data availability for each language.
- Developing a multilingual tokenizer, which we depict in §3.
- Setting the models' hyperparameters and performing pre-training, as described in §4.
- Fine-tuning the LLMs to follow natural language instructions, which we describe in §5.
- Evaluating the models' performance. Results are reported in §6.

2. Data

To train the EuroLLM models, we collect and filter data from various sources for all supported languages. The data included in the final corpus can be divided into four categories: web data, parallel data, code / math data, and high-quality data. Figure 1 shows the percentage attributed to each data category.

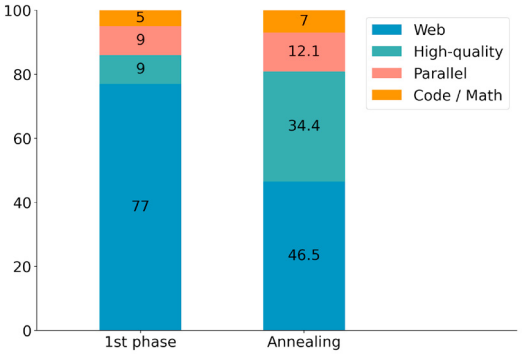


Fig. 1. Percentage attributed to each data category in the first training phase (left) and annealing phase (right).

2.1. Data Collection and Filtering

Web Data. Regarding web data, for English, we use the FineWeb-edu dataset [30] which went through individual dump deduplication, heuristic filtering, and model filtering according to the educational level of the documents (we select documents with scores above 2). For other high-resource languages (German, Spanish, French, and Italian), we collect data from the RedPajama-Data-v2 [9], which has been pre-deduplicated. Additionally, we employ a perplexity filter along with a variety of heuristic filters. For the remaining languages, we collect data from several datasets: HPLT [10], MADLAD-400 [27], CulturaX [31], and mC4 [51], which we concatenate. We then perform deduplication, language identification, perplexity filtering, and apply a set of heuristic filters, using a preprocessing pipeline based on CCNet [47].

Parallel Data. Regarding parallel data, we collect to-English (xx→en) and from-English (en→xx) data from various public sources. We ensure translation quality by removing sentence pairs below quality thresholds for Bicleaner [37, 34] and CometKiwi-22 [36].

Code / Math Data. Regarding code and mathematical data, we collect data from the Stack [21], the Algebraic-stack [3], and the Open-web-math [33] datasets.

High-quality Data. Regarding higher quality data, we use the Wikipedia [14] for all languages and the Arxiv [7], Books [55], and Apollo [46] for English.

Annealing Data. In the last 10% of the pre-training we increase the predominance of high-quality data in the data mix. To do so, we filter the monolingual data using a binary classifier, inspired by FineWeb-edu [30], which was trained to predict whether a document has some educational value, and collect additional high-quality datasets for this phase: Cosmopedia-v2 [5] which is a synthetic dataset composed of textbooks, blog posts, and stories generated by Mixtral-8x7B-Instruct-v0.1 [18]; Python-Edu [5] which is a subset of Python data from the Stack that was filtered by its educational value; and the training sets of the Grade School Math 8K (GSM8K) [8] and of the Mathematics Aptitude Test of Heuristics (MATH) [15]. We also collect document-level parallel data from Europarl [25] and ParaDocs [48].

2.2. Data Mixture

Before starting the training of multilingual LLMs, it is crucial to carefully define the data mixture to be used. This involves deciding how much parallel data to include (§2.2.1), determining whether to repeat high-quality data (§2.2.2), and deciding how to allocate the total number of tokens among the different languages (§2.2.3).

2.2.1. Parallel Data

Parallel data (sentences / documents with their translations in another language) can benefit multilingual LLMs in two aspects: improving the alignment between languages and enhancing the model's machine translation capabilities. However, determining the optimal proportion of parallel data can be challenging.

Joint Scaling Laws. Recent research suggests that the performance of large LLMs can be predicted by a function of the number of non-embedding parameters N, using a power-law [19]. In particular, [13] found that, for *multilingual* models, by training smaller models with varying weights for each language in the data mix, one can fit a *multilingual, joint* scaling law that predicts performance for a model trained with p weight for a language: $\mathcal{L}(N, p) = f(p)\beta N^{-\alpha} + L_\infty$, with a *ratio* function $f(p) = p + c_1 p^{c_2}(1-p)^{c_3}$, and where α, β, L_∞ and $c_{\{1,2,3\}}$ are empirically estimated parameters of the scaling law. This law can predict the language performance trade-off of larger models, even for novel language weightings not encountered during the fitting of the scaling law.

Thus, to decide on the appropriate amount of parallel data, we re-purpose this law to predict the impact on performance as we change its *weighting* in training: we train models with varying numbers of non-embedding parameters (100M, 203M, and 341M) on a 100B token corpus, for which parallel data constitutes different percentages (0%, 25%, and 37.5%) of the total data for each language, excluding English. Figure 2 reports the scaling laws for test sets from three domains: web data, Wikipedia data, and parallel data. Results indicate that adding parallel data does not negatively impact performance on web and Wikipedia domains, while significantly enhancing the performance on parallel data. Moreover, increasing the percentage of parallel data from 25% to 37.5% yields diminishing returns. Therefore, in the final corpus, we include 20% parallel data for each language.

2.2.2. Repeating High Quality Data

To determine whether it is beneficial to repeat datasets considered to be of higher quality, we analyze scaling laws using a method similar to that described in §2.2.1. To do so, we train models on two 100B token corpora: one where Wikipedia data is repeated for all languages and one where it is not.

Figure 3 shows the scaling laws for test sets from web and Wikipedia domains. The results clearly indicate that repeating Wikipedia data improves performance on the Wikipedia test sets without degrading performance on the web test sets. Therefore, we choose to repeat data from high-quality datasets.

2.2.3. Division between Languages

Regarding the allocation of the corpus to each language, we designate 50% for English, as both high-quality data and web data are predominantly in English, and include 5% of code / math data. The remaining 45% of the tokens are distributed among the other languages based on the amount of data obtained after the collection and filtering processes. In order to increase EuroLLM's multilinguality, in the annealing phase, we decrease the English allocation to 32.5% and distribute the surplus across the other languages. We also increase the code / math allocation to 7%. Figure 4 shows the exact percentage attributed to each language.

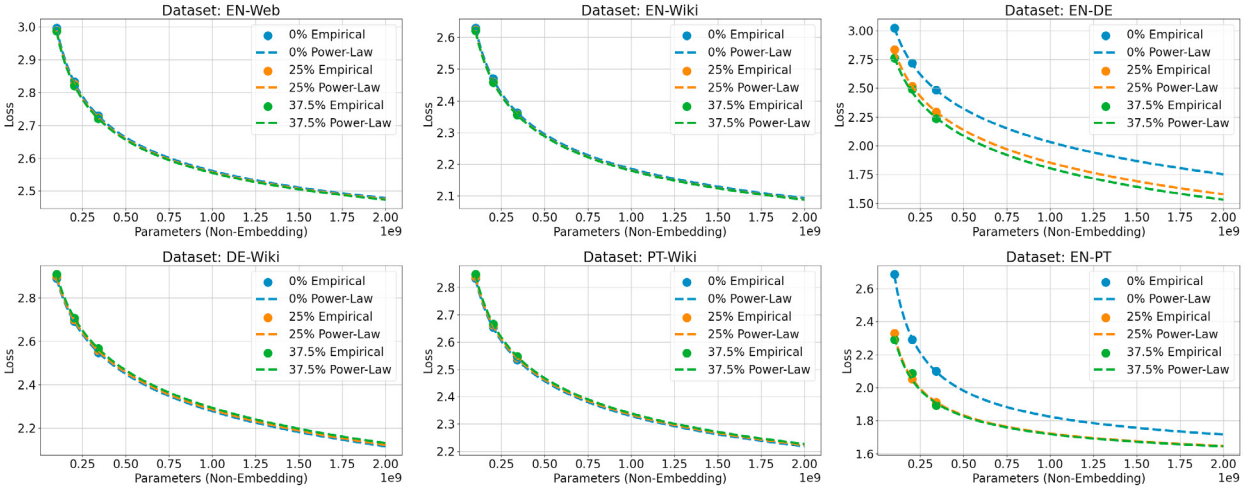


Fig. 2. Joint Scaling laws obtained when varying the percentage of parallel data.

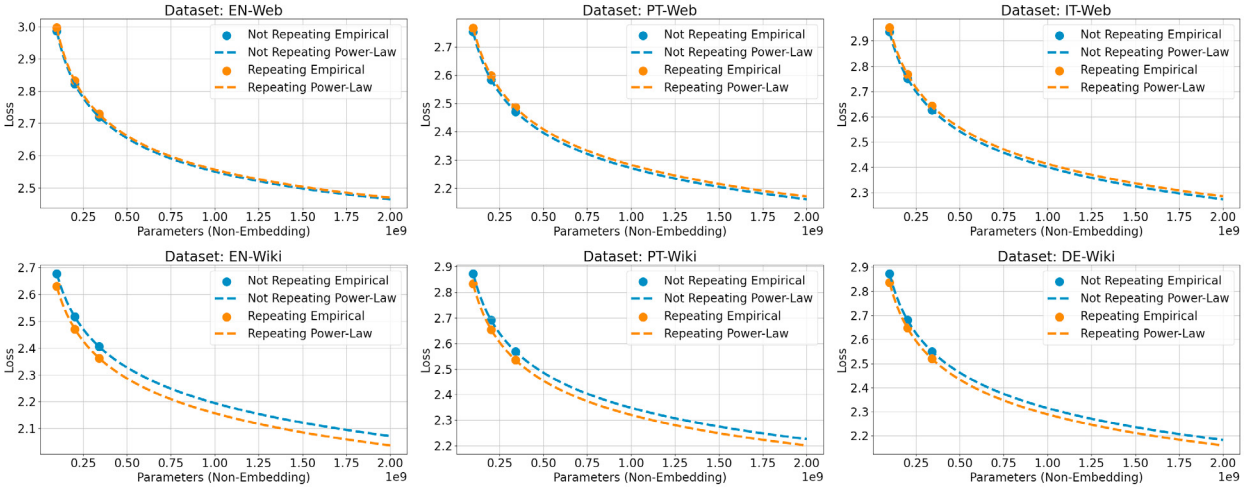


Fig. 3. Joint Scaling laws obtained when repeating vs not-repeating Wikipedia.

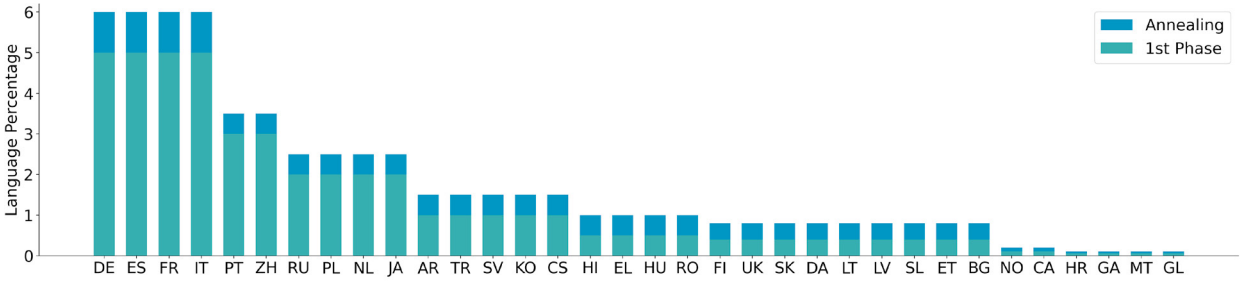


Fig. 4. Percentage of the training corpus attributed to each language, excluding English which accounts to 50% in the first phase and 32.5% during annealing. 5% of the corpus is left for datasets composed of code and math in the first phase and 7% during annealing.

3. Tokenizer

To train the tokenizer, we adopt the approach used by the LLaMa-2 and Mistral models [44, 17], training a BPE tokenizer with byte-fallback. To do so, we use the SentencePiece framework [26]. For an LLM to be efficient across a large number of languages, it is crucial to have a tokenizer with a large vocabulary. However, this comes with the drawback of having a high number of embedding parameters. Through experimentation, we reach the conclusion that a vocabulary of 128,000 pieces provides the best trade-off.

We compare the fertility achieved by the EuroLLM tokenizer with those of Mistral, LLaMa-3, and Gemma tokenizers [17, 1, 41] which have vocabularies of 32,000, 128,256, and 256,000 pieces, respectively. Figure 5 presents the fertilities for a subset of the languages included in EuroLLM. Compared to the Mistral tokenizer, the larger vocabulary of EuroLLM results in significantly lower fertilities. In comparison with the LLaMa-3 and Gemma tokenizers, the LLaMa-3 tokenizer shows the lowest fertility in English but higher fertility for most other languages, while the Gemma tokenizer seems to be better for Asian languages but very similar to EuroLLM for the European ones.

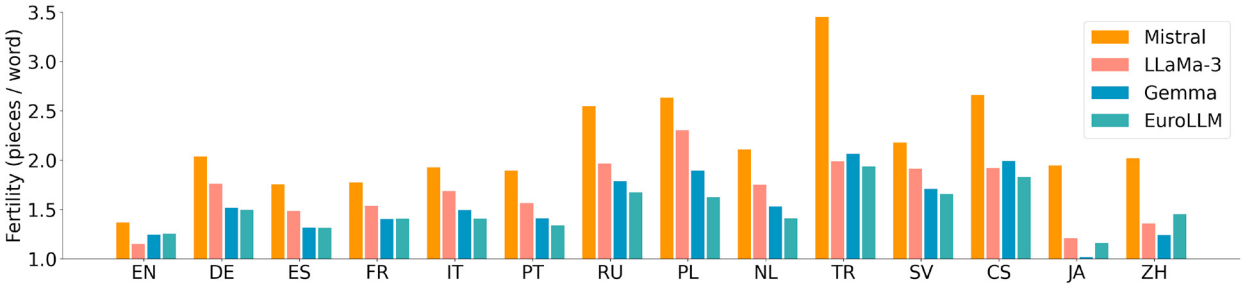


Fig. 5. Fertility (pieces / word) obtained with the Mistral, LLaMa-3, Gemma, and EuroLLM tokenizers for a subset of the EuroLLM languages.

4. Modeling

EuroLLM uses a standard, dense Transformer architecture [45]:

- We use grouped query attention (GQA; [2]) with 8 key-value heads since it has been shown to increase speed at inference time while maintaining downstream performance [42].
- We use pre-layer normalization [50], since it improves training stability, and use the RMSNorm [53], which is faster than LayerNorm [4].
- We use the SwiGLU activation function [38] since it has been shown to lead to good results on downstream tasks [38, 29].
- We use rotary positional embeddings (RoPE) [40] in every layer since these have been shown to lead to good performances while allowing the extension of the context length.

	1.7B
Sequence Length	4,096
Number of Layers	24
Embedding Size	2,048
FFN Hidden Size	5,632
Number of Heads	16
Number of KV Heads (GQA)	8
Activation Function	SwiGLU
Position Encodings	RoPE (Θ=10,000)
Layer Norm	RMSNorm
Tied Embeddings	No
Max Learning Rate	3×10^{-4}
Min Learning Rate	3×10^{-5}
Embedding Parameters	0.262B
LM Head Parameters	0.262B
Non-embedding Parameters	1.133B
Total Parameters	1.657B

Table 1. Overview of EuroLLM hyperparameters.

Training. We pre-train EuroLLM-1.7B on 4 trillion tokens, increasing the predominance of high-quality data on the final 10% of the pre-training process. We use 256 Nvidia H100 GPUs of the MareNostrum 5 supercomputer, training the model with a constant batch size of 3,072 sequences (approximately 12 million tokens), using the Adam optimizer [20], and `bfloat16` mixed precision. All relevant model and training hyperparameters are shown in Table 1.

4.1. Learning Rate Scheduler

Regarding the learning rate scheduler, we experiment with two options. In the first option, we use a cosine scheduler with a warm-up phase corresponding to 10% of the steps. The second option consists of using a trapezoid scheduler [49] (also named Warmup-Stable-Decay [16]). This scheduler has three phases: warm-up for 10% of the steps; constant learning rate; linear decay of the learning rate to the minimum learning rate in the final 10% of the pre-training process, a phase in which we use the higher quality annealing data. To decide which option to use in future models we compare the two options on two multilingual general benchmarks: Hellaswag [52, 28] and Arc Challenge [6, 28] and on machine translation on the FLORES-200 [32], WMT-23 [22], and WMT-24 [23] datasets. The machine translation scores are obtained using COMET-22 [35]. The average results, reported on Table 2, show that using the trapezoid scheduler leads to scores consistently better on the multilingual benchmarks and on machine translation.

	GENERAL		TRANSLATION		
MODEL	Hellaswag	Arc Challenge	FLORES-200	WMT-23	WMT-24
EuroLLM-1.7B with cosine scheduler	0.4646	0.3206	86.48	82.88	78.87
EuroLLM-1.7B with trapezoid scheduler	**0.4744**	**0.3268**	**86.75**	**83.13**	**79.35**

Table 2. Comparison between models trained using the two learning rate scheduler options: cosine scheduler and trapezoid scheduler.

5. Post Training

EuroBlocks. In order for EuroLLM-1.7B to be able to follow natural language instructions, we create a multilingual dataset — EuroBlocks — which encompasses publicly available human-written and synthetic data. We use instruction-following conversations collected from OpenHermes-2.5 [43] and Aya [39] datasets, as well as high-quality machine translation examples from NTREX-128 [12], FLORES-200-DEV [32], WMT-21 [11], and WMT-22 [24]. Overall the dataset is composed by 1M samples covering all supported languages and a variety of tasks.

Supervised fine-tuning (SFT). We fine-tune EuroLLM-1.7B on EuroBlocks to turn it into an instruction-following conversational model: EuroLLM-1.7B-Instruct. We use the standard cross-entropy loss, enabling `bfloat16` mixed precision and packing. We only calculate the loss on target tokens (thus masking loss on prompt tokens). We train for 4 epochs using a learning rate of $7 \cdot 10^{-6}$ over the course of around 6K steps.

Formatting. EuroLLM is finetuned with chat format control tokens (`<|im_start|>` and `<|im_end|>` tokens) that are included in the original tokenizer described in Section 3. As such, no tokens are added during SFT; however, we overwrite the end-of-sequence token with the `<|im_end|>` token. These control tokens create a clear separation between user/model turns. The chat template may also include a "system prompt" that can be used for added functionality and steerability of the model's responses. We show an example in Table 3.

6. Results

In order to evaluate our models we recur to: general benchmarks (§6.1) to understand if EuroLLM-1.7B acquired world knowledge during training; and machine translation (§6.2) to understand if the model is able to understand and generate text in all the supported languages.

6.1. General Benchmarks

Regarding general benchmarks, we consider a commonsense natural language inference test set: Hellaswag [52] and a test set of science exam questions: Arc Challenge [6]. These benchmarks are originally English-only, so we recur to translations [28]. As baselines, we use Gemma-2b [41] and TinyLlama [54]. Figure 6 reports the results. On Hellaswag, EuroLLM-1.7B matches or outperforms Gemma-2b and TinyLlama on all languages besides English,

System	<s><|im_start|>system
	Translate all user texts to English.<|im_end|>
User	<|im_start|>user
	A Comissão Europeia é uma instituição politicamente independente e que representa e defende os interesses da União Europeia na sua globalidade. <|im_end|>
	<|im_start|>assistant
Model	The European Commission is a politically independent institution that represents and defends the interests of the European Union as a whole. <|im_end|>
User	<|im_start|>user
	La Comisión Europea no consta únicamente de los 27 miembros que integran la institución constitucional, el llamado Colegio de Comisarios. Está integrada por todo un cuerpo administrativo de algo más de 38 000 funcionarios cuya función es asistirlo y actuar como brazo ejecutivo del Colegio. <|im_end|>
	<|im_start|>assistant
Model	The European Commission is not only composed of the 27 members who form the constitutional institution, the so-called College of Commissioners. It is composed of a whole administrative body of more than 38,000 officials whose function is to assist it and act as the executive arm of the College. <|im_end|>

Table 3. Example of a dialogue with EuroLLM-1.7B-Instruct. We mark the system, user and model control tokens.

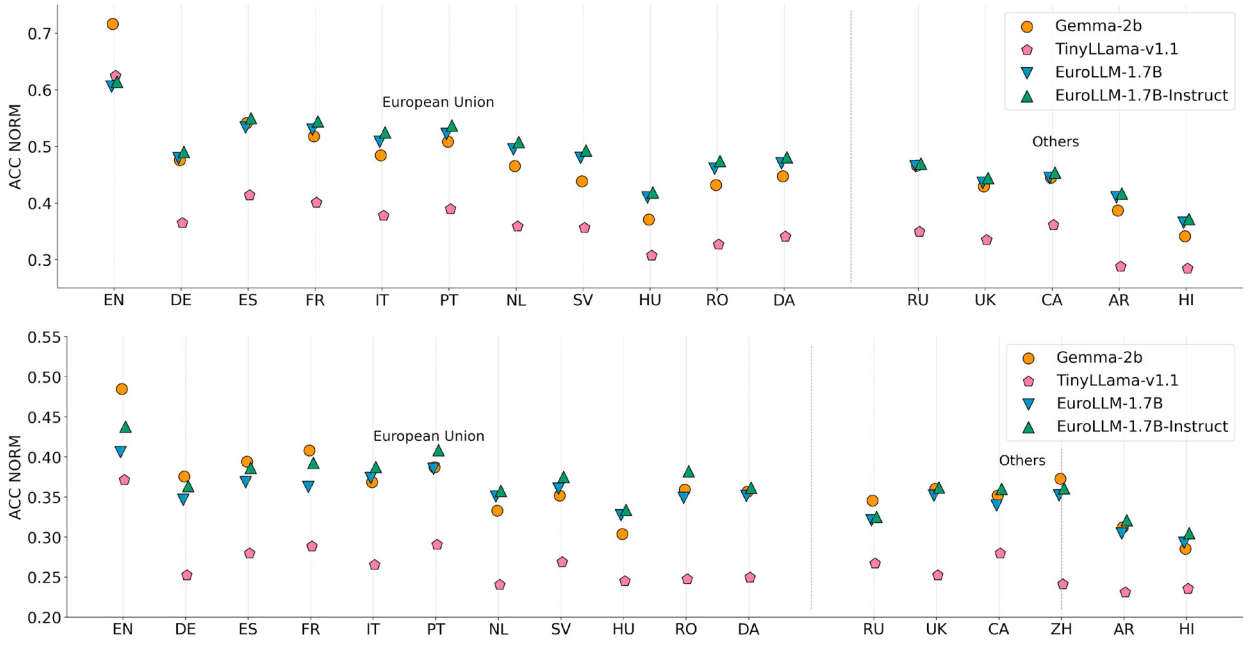


Fig. 6. Results on the Hellaswag (top) and Arc Challenge (bottom) benchmarks. The results were obtained using 10-shot and 25-shot prompts for Hellaswag and Arc Challenge, respectively.

which showcases its increased multilinguality. On Arc Challenge, EuroLLM-1.7B outperforms TinyLlama on all languages but is worse than Gemma-2b. This can be caused by the lower number of parameters (EuroLLM-1.7B has 1.133B non-embedding parameters while Gemma-2B has 1.981B). Interestingly, EuroLLM-1.7B-Instruct leads to slightly better results than EuroLLM-1.7B for both benchmarks.

6.2. Machine Translation

Regarding machine translation, we compare EuroLLM-1.7B-Instruct with Gemma-2b and Gemma-7b [41] on three datasets: Flores-200-test [32], WMT-23 [22], and WMT-24 [23] and evaluate the translations using Comet-22. To have a fair comparison, we also fine-tune Gemma-2b and Gemma-7b on the EuroBlocks dataset. Figures 7 and 8

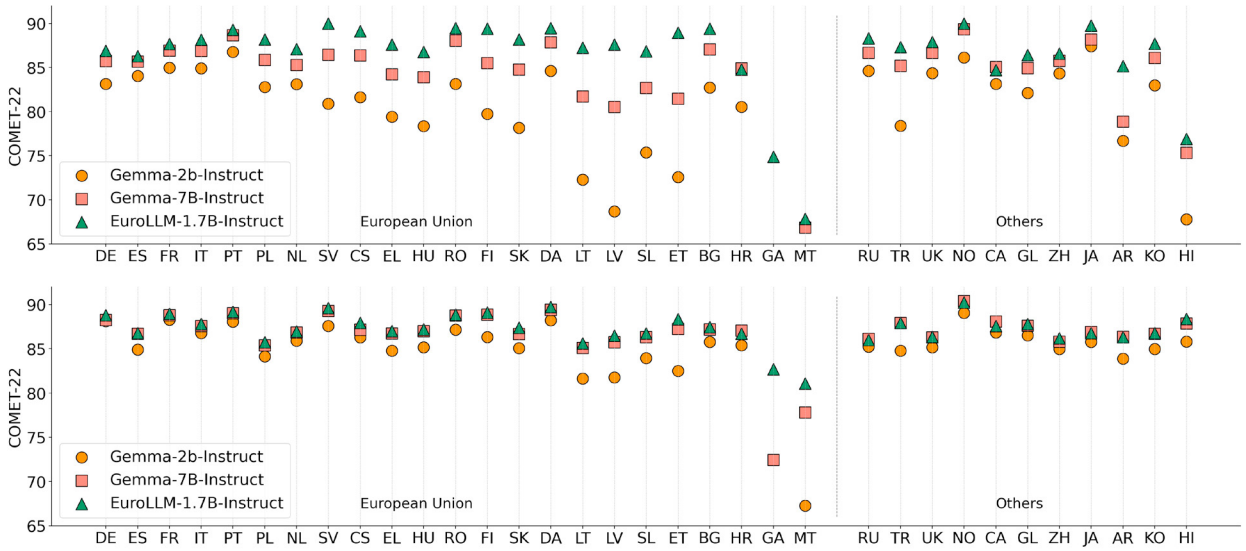


Fig. 7. COMET-22 scores on the FLORES-200 dataset on EN-XX (top) and XX-EN (bottom) language pairs. All models were fine-tuned with the EuroBlocks dataset and the translations were obtained using 0-shot prompts and greedy search.

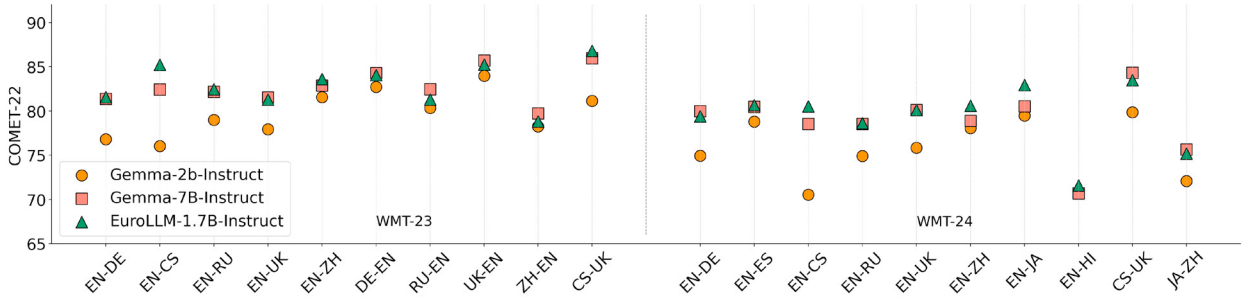


Fig. 8. COMET-22 scores on the WMT-23 and WMT-24 datasets. All models were fine-tuned with the EuroBlocks dataset and the translations were obtained using 0-shot prompts and greedy search.

report the results. We can see that EuroLLM-1.7B-Instruct clearly outperforms Gemma-2b-Instruct on all languages pairs and datasets, and is competitive with Gemma-7b-Instruct despite the much lower number of parameters.

7. Conclusions and Future Work

In this paper, we present the work done so far in the EuroLLM project. We describe our data collection and filtering process, how we build a multilingual tokenizer, and the data mixture and modeling configurations. We also release our initial models: EuroLLM-1.7B and EuroLLM-1.7B-Instruct and report their performance on multilingual general benchmarks and machine translation. In future work, we will continue training multilingual LLMs for Europe, focusing on scaling up the number of model parameters and improving further the quality of our data.

Acknowledgments

Part of this work was supported by the EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631), by the project DECOLLAGE (ERC-2022-CoG 101088763), and by the Portuguese Recovery and Resilience Plan through project C645008882-00000055 (Center for Responsible AI). We thank EuroHPC for the HPC resources used to support this work through grant EHPC-EXT-2023E01-04.

References

[1] AI@Meta. Llama 3 model card. 2024.

[2] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

[3] Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics, 2023.

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[5] Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. Smollm-corpus, 2024.

[6] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

[7] Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. On the use of ArXiv as a dataset, 2019.

[8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[9] Together Computer. RedPajama: an open dataset for training large language models, 2023.

[10] Ona de Gibert, Graeme Nail, Nikolay Arefyev, Marta Bañón, Jelmer van der Linde, Shaoxiong Ji, Jaume Zaragoza-Bernabeu, Mikko Aulamo, Gema Ramírez-Sánchez, Andrey Kutuzov, Sampo Pyysalo, Stephan Oepen, and Jörg Tiedemann. A new massive multilingual dataset for high-performance language technologies, 2024.

[11] Akhbardeh Farhad, Arkhangorodsky Arkady, Biesialska Magdalena, Bojar Ondřej, Chatterjee Rajen, Chaudhary Vishrav, Marta R Costa-jussa, España-Bonet Cristina, Fan Angela, Federmann Christian, et al. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, 2021.

[12] Christian Federmann, Tom Kocmi, and Ying Xin. NTREX-128 – news test references for MT evaluation of 128 languages. In *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, pages 21–24, Online, nov 2022. Association for Computational Linguistics.

[13] Patrick Fernandes, Behrooz Ghorbani, Xavier Garcia, Markus Freitag, and Orhan Firat. Scaling laws for multilingual neural machine translation. In *International Conference on Machine Learning*, pages 10053–10071. PMLR, 2023.

[14] Wikimedia Foundation. Wikimedia downloads.

[15] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[16] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

[17] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[18] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[19] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[21] Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro von Werra, and Harm de Vries. The Stack: 3 TB of permissively licensed source code. *Preprint*, 2022.

[22] Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, et al. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, 2023.

[23] Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondrej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, et al. Preliminary WMT24 Ranking of General MT Systems and LLMs. *arXiv preprint arXiv:2407.19884*, 2024.

[24] Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, et al. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, 2022.

[25] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand, September 13-15 2005.

[26] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. *EMNLP 2018*, page 66, 2018.

[27] Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Christopher A. Choquette-Choo, Katherine Lee, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. MADLAD-400: A multilingual and document-level large audited dataset, 2023.

[28] Viet Lai, Chien Nguyen, Nghia Ngo, Thut Nguyn, Franck Dernoncourt, Ryan Rossi, and Thien Nguyen. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 318–327, 2023.

[29] Teven Le Scao, Thomas Wang, Daniel Hesslow, Stas Bekman, M Saiful Bari, Stella Biderman, Hady Elsahar, Niklas Muennighoff, Jason Phang, Ofir Press, et al. What language model to train if you have one million GPU hours? In *Findings of the Association for Computational*

Linguistics: EMNLP 2022, pages 765–782, 2022.
[30] Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. Fineweb-edu, 2024.
[31] Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. CulturaX: A cleaned, enormous, and multilingual dataset for large language models in 167 languages, 2023.
[32] James Cross Onur Çelebi Maha Elbayad Kenneth Heafield Kevin Heffernan Elahe Kalbassi Janice Lam Daniel Licht Jean Maillard Anna Sun Skyler Wang Guillaume Wenzek Al Youngblood Bapi Akula Loic Barrault Gabriel Mejia Gonzalez Prangthip Hansanti John Hoffman Semarley Jarrett Kaushik Ram Sadagopan Dirk Rowe Shannon Spruit Chau Tran Pierre Andrews Necip Fazil Ayan Shruti Bhosale Sergey Edunov Angela Fan Cynthia Gao Vedanuj Goswami Francisco Guzmán Philipp Koehn Alexandre Mourachko Christophe Ropers Safiyyah Saleem Holger Schwenk Jeff Wang NLLB Team, Marta R. Costa-jussà. No language left behind: Scaling human-centered machine translation. 2022.
[33] Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. OpenWebMath: An open dataset of high-quality mathematical web text, 2023.
[34] Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz-Rojas. Bifixer and Bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal, November 2020. European Association for Machine Translation.
[35] Ricardo Rei, José GC De Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André FT Martins. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Seventh Conference on Machine Translation*, 2022.
[36] Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In Philipp Koehn, Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Tom Kocmi, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Marco Turchi, and Marcos Zampieri, editors, *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics.
[37] Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez-Sánchez. Prompsit's submission to WMT 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation*.
[38] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.
[39] Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Souza Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergün, Ifeoma Okoh, Aisha Alaagib, Oshan Mudannayake, Zaid Alyafeai, Vu Minh Chien, Sebastian Ruder, Surya Guthikonda, Emad A. Alghamdi, Sebastian Gehrmann, Niklas Muennighoff, Max Bartolo, Julia Kreutzer, Ahmet Üstün, Marzieh Fadaee, and Sara Hooker. Aya dataset: An open-access collection for multilingual instruction tuning, 2024.
[40] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
[41] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.
[42] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.
[43] Teknium. OpenHermes 2.5: An Open Dataset of Synthetic Data for Generalist LLM Assistants, 2023.
[44] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
[46] Xidong Wang, Nuo Chen, Junyin Chen, Yan Hu, Yidong Wang, Xiangbo Wu, Anningzhe Gao, Xiang Wan, Haizhou Li, and Benyou Wang. Apollo: Lightweight Multilingual Medical LLMs towards Democratizing Medical AI to 6B People, 2024.
[47] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. CCNet: Extracting High-Quality Monolingual Datasets from Web Crawl Data, 2019.
[48] Rachel Wicks, Matt Post, and Philipp Koehn. Recovering document annotations for sentence-level bitext, 2024.
[49] Chen Xing, Devansh Arpit, Christos Tsirigotis, and Yoshua Bengio. A walk with sgd. *arXiv preprint arXiv:1802.08770*, 2018.
[50] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020.
[51] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer, 2021.
[52] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.
[53] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.
[54] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.
[55] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *International Conference on Computer Vision*, 2015.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 63–72

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

Voice Liveness Detection KYC Project: Distinguishing Genuine and Spoofed Voices Using Deep Learning

Buğra Eyidoğan[a,*], Gökberk Özsoy[a], Pedram Khatamino[b], Bilal Avvad[a], Enis Şen[c], Deniz Kumlu[b]

[a]*WesterOps, Etiler District Toprakkale Street No:2/3 34330 Besiktas/Istanbul, Turkey*

[b]*KOBIL Technology Inc. Teknopark Istanbul Sanayi Mahallesi Teknopark Boulevard No: 1/7C Yeditepe University Technology Base Floor: 3 Interior Door No: 303 34906 Pendik, Istanbul*

[c]*AI4SEC, Sepapaja 6 Tallinn,Estonia*

Abstract

Voice-based authentication systems are increasingly integral to secure user identification in various applications, from banking to smart devices. As these systems become more prevalent, their vulnerability to spoofing attacks, such as replayed or synthetic voices, poses significant security concerns. This paper addresses these challenges by focusing on speech recognition and voice liveness detection using advanced deep learning models. We explore the effectiveness of detecting pop noise—a low-frequency artifact characteristic of live speech—using the Constant-Q Transform (CQT) for feature extraction. Through a comprehensive analysis of datasets, including ASVspoof2015, ASVspoof2017, and POCO, we developed and trained a Convolutional Neural Network (CNN) model on the MeluXina supercomputer, achieving a test accuracy of 96.95% on the ASVspoof2017 dataset. Our findings demonstrate the potential of using pop noise detection and CNNs for robust voice liveness detection. Future work will expand on these results by exploring additional feature extraction methods and alternative machine learning algorithms to further enhance system reliability across various spoofing scenarios.

Keywords: Voice-based authentication; Liveness detection; Speech recognition; Pop noise detection; Convolutional Neural Network (CNN); Constant-Q Transform (CQT); ASVspoof2017; MeluXina supercomputer

1. Introduction

Voice-based authentication systems have become increasingly vital in various applications, ranging from secure user identification in banking systems to access control in smart devices. However, with the growing reliance on voice

* Corresponding author. Tel.: +90-535-4597604
E-mail address: bugra.eyidogan@westerops.com

1877-0509
10.1016/j.procs.2025.02.261

as a biometric identifier, the security of these systems against spoofing attacks has become a critical concern. Speech recognition and liveness detection play crucial roles in ensuring the reliability and security of these systems.

Speech recognition involves the automatic identification and understanding of spoken language by machines, allowing for accurate transcription and interaction with users. Recent advancements in deep learning and machine learning have significantly improved the accuracy and efficiency of speech recognition systems, making them more adaptable to diverse languages and dialects.

Liveness detection, on the other hand, is a security feature that distinguishes between genuine, live input and spoofed or manipulated input, such as recorded or synthetic voices. This is essential to protect against attacks that aim to deceive voice-based systems using pre-recorded or computer-generated speech.

Several key studies have advanced the fields of speech recognition and liveness detection. This literature review explores these developments, focusing on the methodologies, innovations, and outcomes presented in recent research. In our work, we aim to leverage these advancements to develop a comprehensive KYC (Know Your Customer) solution that integrates both speech recognition and liveness detection. This project focuses on training models, analyzing results, and ultimately applying these technologies in real-world scenarios to enhance security in voice-based authentication systems.

1.1. Speech Recognition

One of the foundational models in modern speech recognition is the work by Desplanques et al. on ”ECAPA-TDNN,” which emphasizes channel attention and aggregation, allowing for more robust speaker verification in challenging acoustic environments. The architecture enhances the capability to capture essential features from speech data, leading to improved accuracy in speaker identification and verification tasks, achieving Equal Error Rates (EER) as low as 0.808% on the VoxCeleb1 dataset [1].

Another significant model is developed by Koluguri et al., who introduced ”TitaNet,” a neural model for speaker representation using 1D depth-wise separable convolutions and global context. TitaNet outperforms previous models in speaker verification tasks, achieving EERs as low as 0.85% on the VoxCeleb1 test set [2].

Baevski et al. introduced ”Wav2Vec 2.0,” a breakthrough model that leverages a self-supervised learning approach to significantly improve automatic speech recognition (ASR) performance. Wav2Vec 2.0 achieves Word Error Rates (WER) of 2.9% on the LibriSpeech test-clean set and 5.1% on the test-other set [3].

Chen et al. extended the Wav2Vec framework with ”UniSpeech-SAT,” incorporating speaker-aware pre-training to further refine ASR and speaker recognition tasks. This model achieves a WER of 2.4% on the LibriSpeech test-other set and maintains robust performance across various speaker verification benchmarks [4].

Finally, Chen et al. also contributed ”WavLM,” which expands on the capabilities of Wav2Vec and HuBERT by incorporating masked speech denoising and prediction in pre-training. WavLM achieves state-of-the-art performance on multiple benchmarks, including a WER of 2.9% on the LibriSpeech test-clean set and 4.6% on the test-other set, while also excelling in speaker verification with an EER of 0.383% on VoxCeleb1 [5].

1.2. Liveness Detection

Recent advancements in voice liveness detection have focused on developing efficient and accurate methods to distinguish between live and spoofed voices, which is crucial for maintaining the security of voice-based authentication systems.

Ahmed et al. introduced ”Void,” a fast and lightweight voice liveness detection system that uses spectral power differences between live and replayed voices to identify spoofing attempts. The model achieves an impressive Equal Error Rate (EER) of 0.3% on a proprietary dataset and 11.6% on the ASVspoof 2017 public dataset, positioning it as the second-best performer in the ASVspoof 2017 competition. Void is also remarkably efficient, being approximately eight times faster and using 153 times less memory than the top model in the competition [6].

Gupta et al. proposed a novel approach using bump wavelet-based Continuous Wavelet Transform (CWT) features combined with a Convolutional Neural Network (CNN) architecture. Their system outperforms previous methods, achieving an overall accuracy of 80.19%, significantly higher than the 62.08% accuracy of the STFT-based baseline. The model performs particularly well on plosive, whisper, and fricative sounds, with accuracy rates of 81.58%, 81.09%, and 80.77%, respectively [7].

In another study, Gupta et al. developed a deep learning-based approach for voice liveness detection using a customized CNN architecture. This model, tested on the POCO dataset, achieved an accuracy of 80.51%, with a further improvement to 82.15% in 10-fold cross-validation. The model showed superior performance in detecting pop noise in specific words, such as 'division,' 'fat,' 'funny,' and 'thong,' with an average accuracy of 85% [8].

Khoria et al. explored the use of the Constant-Q Transform (CQT) for enhancing the detection of pop noise, a low-frequency indicator of live speech. Their CQT-based method showed a 4.2% absolute improvement in accuracy over the STFT-based baseline, achieving a 10-fold cross-validation accuracy of 66.49% on the POCO dataset [9].

Lastly, Gupta et al. investigated the use of Morse wavelet transforms as a feature extraction method for voice liveness detection. This approach leverages the unique time-frequency localization properties of Morse wavelets to enhance the detection of subtle differences between live and spoofed voices. The Morse wavelet-based system achieved an accuracy of 83.25% on the POCO dataset, demonstrating its effectiveness in distinguishing live voices from replayed or synthetic ones [10].

2. Datasets

In this project, we utilize several datasets that are essential for developing and evaluating robust voice liveness detection and speaker verification systems. These datasets provide a comprehensive collection of genuine and spoofed speech samples, enabling the development of models that can effectively distinguish between live and synthetic voices. The selected datasets are particularly valuable for their diversity in recording conditions, speaker variations, and spoofing techniques, making them ideal for training and testing our models.

2.1. ASVspoof 2015 Dataset

The ASVspoof 2015 dataset was created for the First Automatic Speaker Verification Spoofing and Countermeasures Challenge. This dataset is specifically designed to benchmark the performance of speaker verification systems against various types of spoofing attacks, including speech synthesis and voice conversion [11].

Data Composition:

- **Training Set:** Includes 3,750 genuine and 12,625 spoofed utterances from 25 speakers (10 male, 15 female). Spoofed data is generated using five known speech synthesis and voice conversion algorithms.
- **Development Set:** Comprises 3,497 genuine and 49,875 spoofed utterances from 35 speakers (15 male, 20 female). This set is used for tuning and optimizing anti-spoofing systems.
- **Evaluation Set:** Contains 9,404 genuine and 184,000 spoofed utterances from 46 speakers (20 male, 26 female), including both known and unknown spoofing algorithms to simulate real-world conditions.

2.2. ASVspoof 2017 Dataset

The ASVspoof 2017 dataset is designed to address the vulnerabilities of Automatic Speaker Verification (ASV) systems to replay spoofing attacks. It consists of both genuine and spoofed speech data, aiming to benchmark the performance of anti-spoofing countermeasures [12].

Data Composition:

- **Training Set:** The training set comprises 1,507 genuine and 1,507 spoofed utterances, generated from 25 speakers (10 male, 15 female). Spoofed data was created using five known speech synthesis (SS) and voice conversion (VC) algorithms.
- **Development Set:** This set includes 760 genuine and 950 spoofed utterances from 35 speakers (15 male, 20 female), using the same spoofing algorithms as in the training set. It is used for system tuning and optimization.
- **Evaluation Set:** The evaluation set includes 1,298 genuine and 12,008 spoofed utterances from 46 speakers (20 male, 26 female). Spoofing algorithms for this set include the five known attacks from the training and development sets, along with five additional unknown attacks, making it more representative of real-world scenarios.

2.3. POCO Dataset

The POCO (Pop Noise Corpus) dataset is specifically designed for studying the liveness feature of voice recordings, focusing on pop noise as an indicator of genuine speech. The dataset was created to promote research in protecting automatic speaker verification (ASV) systems from voice spoofing attacks, such as replay attacks [13].

Data Composition:

- **RC-A:** High-quality recordings using the audio-technica AT4040 microphone, capturing genuine speaker recordings with pop noise at a fixed speaker-microphone distance of 10 cm.
- **RC-B:** Recordings with a microphone array (audio-technica AT9903) with 15 microphones positioned at different distances (5 cm, 10 cm, and 20 cm) from the speaker's mouth, allowing the study of how pop noise varies with position and angle.
- **RP-A:** Simulated eavesdropping scenario where the speaker's voice is recorded at a distance without pop noise using a pop filter, recorded with the audio-technica AT4040 microphone at a 10 cm distance.

The POCO dataset includes recordings from 66 subjects (34 female and 32 male), covering all 44 phonemes in English, and consists of 402,391 utterances after post-processing.

2.4. Addressing the Nature of Spoofing and Recording Differences

This work represents the initial phase of our project, laying the groundwork for voice liveness detection using pop noise as a distinguishing feature. We acknowledge that certain assumptions and challenges must be considered as we advance our research. One of the primary challenges in voice liveness detection is differentiating between genuine live speech and spoofed audio, such as replayed recordings or synthetic voices. We emphasize that even though all audio input is technically a recording, genuine live speech exhibits distinct acoustic characteristics, particularly in the form of pop noise. This low-frequency artifact, produced by bursts of air hitting the microphone during the articulation of certain consonants, is typically present in direct speech and diminishes significantly in replayed audio captured from a distance. For our current study, we assume that attackers would record the target's voice at a distance that fails to capture these low-frequency artifacts adequately. Morever, Our system's performance may vary depending on several factors, including the language spoken, the speaker's sex, and the environmental context. For instance, the harmonic and spectral characteristics of different languages may affect the model's ability to detect liveness. Similarly, variations in pitch, timbre, and the way pop noise is produced across male and female voices could influence detection accuracy. Environmental factors, such as background noise and room acoustics, can also impact the reliability of our approach. We plan to conduct extensive testing under diverse conditions to better understand and mitigate these variables.

3. Results

In this study, we trained our voice liveness detection model using the datasets mentioned previously. The training process was significantly accelerated by leveraging the computational power of the MeluXina supercomputer, which we accessed through the EuroHPC benchmark project. Given our initial allocation of 400 node hours, we strategically optimized our experiments to maximize the use of these resources. The supercomputing power provided by MeluXina was crucial in enabling us to handle the computational demands of processing high-resolution CQT spectrograms and training our deep learning models efficiently.

Our initial focus was on detecting voice liveness through the identification of pop noise in speech signals. Pop noise is characterized by low-frequency sound artifacts that occur when air pressure from speech hits the microphone directly, a phenomenon typically found in genuine live speech recordings. To analyze this feature, we employed the Constant-Q Transform (CQT) function, following the methodology outlined by Gupta et al. The use of CQT allowed us to emphasize the 0-40 Hz frequency range, where pop noise is most prominent, and extract meaningful spectrogram representations for both genuine and spoofed audio samples.

Despite the limited node hours in our benchmark project, the supercomputer's efficiency allowed us to perform extensive experiments that would be infeasible on standard hardware. This initial phase of our research has yielded

promising results, but further optimization and refinements are necessary to fully leverage the capabilities of our proposed methodology.

3.1. Impact of Language, Speaker Characteristics, and Environmental Conditions

Our system's performance may vary depending on several factors, including the language spoken, the speaker's sex, and the surrounding sound environment. Preliminary observations suggest that the harmonic and spectral content of different languages can affect the model's ability to distinguish between genuine and spoofed voices. Similarly, male and female voices may produce different patterns of pop noise, and variations in pitch and timbre could influence detection accuracy.

Environmental factors, such as background noise or room acoustics, also play a critical role. Future experiments will involve systematically varying these parameters to better understand their impact and develop techniques to mitigate their effects. Additionally, data augmentation strategies and environment-invariant feature extraction methods will be explored to enhance model robustness.

3.2. CQT Feature Extraction and Pop Noise Analysis

In our implementation, we utilized Librosa's Constant-Q Transform (CQT) function to extract meaningful features from the audio signals, focusing particularly on the identification of pop noise, a low-frequency artifact characteristic of genuine live speech. The CQT is a spectral transform where the frequency bins are geometrically spaced, providing a logarithmic frequency resolution that aligns well with the human auditory system. This property makes the CQT especially suitable for analyzing the spectral content of speech signals.

Mathematically, the CQT of an audio signal $x(t)$ is defined as:

$$X(k,t) = \sum_{n=0}^{N_k-1} x(n)\, h_k(n)\, e^{-2\pi i f_k n/f_s} \tag{1}$$

where:

- k is the frequency bin index,
- f_k is the center frequency of the k-th bin,
- N_k is the window length for the k-th bin,
- $h_k(n)$ is a window function centered around f_k,
- f_s is the sampling rate of the audio signal.

The bandwidth of each bin Δf_k is related to the center frequency f_k, allowing for higher frequency resolution at lower frequencies and broader bandwidth at higher frequencies. This design ensures a more detailed analysis of lower frequencies, which is crucial for detecting features like pop noise.

In our implementation, we set the minimum frequency f_{min} to C1 (approximately 32.7 Hz) and used 96 bins per octave over 7 octaves, covering a frequency range up to approximately 8 kHz. This setup ensures fine granularity across the spectrum, making it ideal for capturing subtle variations in the low-frequency range where pop noise is most prominent.

The output of the CQT, represented as a spectrogram, was converted into decibel (dB) scale to enhance feature visibility:

$$C_{\text{dB}}(k,t) = 20 \log_{10}\left(\frac{|X(k,t)|}{\max(|X(k,t)|)}\right) \tag{2}$$

where $C_{\mathrm{dB}}(k,t)$ denotes the amplitude of the CQT spectrogram in decibels, and $|X(k,t)|$ is the magnitude of the CQT at bin k and time t.

By focusing on the 0-40 Hz frequency range, we aimed to isolate and highlight pop noise features. The CQT spectrograms were generated with a color map that made these features visually distinguishable, allowing for an effective comparison between genuine and spoofed audio samples. Specifically, we observed that pop noise was more pronounced in genuine recordings, providing a reliable indicator for liveness detection. This decibel scaling highlights subtle differences between genuine and spoofed speech. We visualized the CQT spectrograms using a high-contrast 'turbo' colormap and adjusted the display to focus on essential time-frequency patterns, removing axes and labels to prevent distractions.

This approach, inspired by Gupta et al.'s methodology, provided a robust way to differentiate between genuine and spoofed audio based on the presence of pop noise. Our comparison revealed that genuine audio samples often contained distinctive low-frequency features that were absent or diminished in spoofed samples, validating the effectiveness of our feature extraction technique.

Pop noise, a low-frequency artifact that occurs when bursts of air from speech hit the microphone, is particularly prominent in the 0-40 Hz frequency range. By analyzing this band within the CQT spectrograms, we aimed to discern differences between genuine and spoofed audio samples. Pop noise is often associated with the pronunciation of hard consonants and is a critical indicator of live speech. The CQT-based approach allowed us to systematically compare the time-frequency patterns of genuine and spoofed samples, demonstrating the effectiveness of pop noise as a feature for liveness detection.

3.3. Comparison of Genuine and Spoofed Audio Samples

First, We started to conduct an in-depth analysis using the POCO dataset, which contains various words recorded under controlled conditions. By extracting CQT features from both genuine and spoofed recordings of specific words, we observed that some words presented a significant difference in their CQT representations, while others did not.

For example, the word "quick" exhibited a clear distinction between genuine and spoofed samples within the 0-40 Hz range, with the genuine sample showing pronounced pop noise, as depicted in Figure 1. Conversely, the word "bird" presented almost identical CQT images for both genuine and spoofed samples, indicating that pop noise was less discernible in this case, as shown in Figure 2.

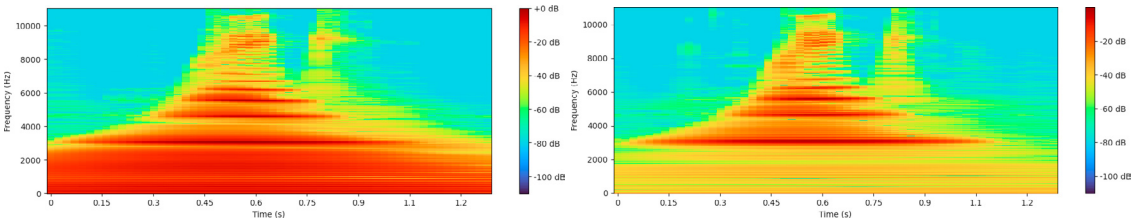


Fig. 1. CQT Features for the word "quick": (Left) Genuine, (Right) Spoofed

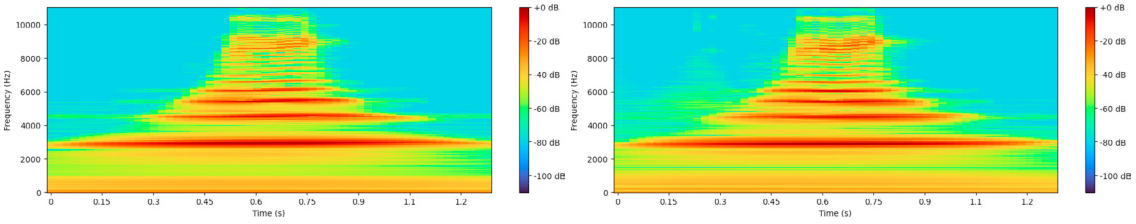


Fig. 2. CQT Features for the word "bird": (Left) Genuine, (Right) Spoofed

After analyzing the CQT images from the POCO dataset, we determined that while the dataset was useful for preliminary analysis, it had limitations due to its focus on individual words. In real-world applications, particularly in

our planned use case for voice KYC (Know Your Customer) systems, audio recordings are likely to be captured over the phone and consist of full sentences rather than isolated words.

Given this context, we decided to implement the CQT feature extraction on the ASVspoof2017 dataset. The ASVspoof2017 dataset is more suitable for our case as it contains sentences rather than individual words, allowing the CQT feature images to capture more extensive pop noise segments. This shift in focus provides a richer and more accurate representation of the audio's liveness characteristics, better aligning with the scenarios we expect in voice KYC applications.

3.4. Dataset Preparation and Splitting

The ASVSpoof2017 dataset, containing a mix of genuine and spoofed audio samples, was used to train and evaluate our model. The dataset was preprocessed into Constant-Q Transform (CQT) spectrograms to highlight critical low-frequency components. We split the dataset into training, validation, and test sets using an 80-10-10 ratio to ensure a balanced distribution of classes across all splits. The training set was used to optimize the model parameters, while the validation set guided hyperparameter tuning and early stopping to prevent overfitting. The test set, kept entirely separate during training, was utilized for the final evaluation of model performance.

However, it's important to note that these results are preliminary, and further experiments are necessary to confirm the model's robustness across different words and varying recording conditions. Future work will focus on optimizing the model's architecture, refining the CQT feature extraction process, and expanding the dataset to include a broader range of speech samples.

3.5. Model Selection and CNN Architecture

Given the importance of accurately identifying these low-frequency components, we selected a Convolutional Neural Network (CNN) model known for its proficiency in feature extraction and pattern recognition in time-series data. The architecture was specifically designed to emphasize the detection of pop noise within the broader context of voice liveness detection. Our architecture begins with three convolutional layers (32, 64, and 128 filters respectively) that extract increasingly complex features from the input CQT spectrograms. Each convolutional layer is followed by max pooling to reduce spatial dimensions while retaining critical information. The extracted features are then passed through fully connected layers (128 and 64 neurons), with dropout applied to prevent overfitting. Finally, a softmax output layer classifies the input as genuine or spoofed. This architecture balances complexity with efficiency, making it well-suited for real-time voice-based applications.

In our work, we did not perform extensive hyperparameter optimization to find the network configuration and training parameters. Instead, our initial architecture and settings were chosen based on prior research and domain knowledge. Once we achieved high accuracy and satisfactory model performance, we retained the current architecture without further adjustments. However, future work will consider hyperparameter tuning and scaling laws to better understand the relationship between the model size, data quantity, and computational resources, especially for scaling up the system for larger datasets or more complex scenarios

3.6. Training and Validation Loss

The training process showed a consistent decrease in loss, demonstrating that the model was effectively learning the required features. As illustrated in Figure 3 the training loss started at approximately 0.32 in the first epoch and reduced to 0.064 by the ninth epoch. Similarly, the validation loss decreased from 0.17 to 0.097 over the same period. Early stopping was triggered at epoch 10, indicating optimal performance.

3.7. Training and Validation Accuracy

The accuracy metrics reflect the model's ability to generalize well. As shown in Figure 4 the training accuracy improved from 85.25% in the first epoch to 97.52% by epoch 10. The validation accuracy also increased from 94% to 96.89%, confirming that the model was not overfitting and maintained strong performance on unseen data.

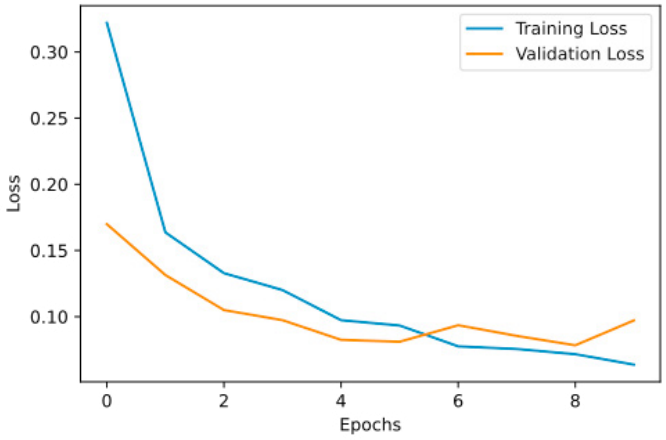


Fig. 3. Training and Validation Loss over Epochs

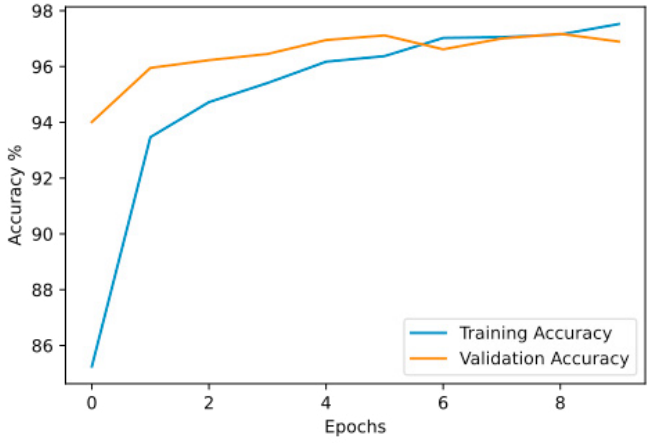


Fig. 4. Training and Validation Accuracy over Epochs

3.8. Confusion Matrix and Test Accuracy

The final test on unseen data provided a test accuracy of 96.95%. The confusion matrix in Figure 5 shows that out of 1803 test samples, the model correctly identified 1427 spoofed voices and 321 real voices, with only 29 false positives and 26 false negatives.

4. Conclusion and Future Work

The results demonstrate the robustness of the CNN model in detecting liveness in voices, with promising initial outcomes showing an accuracy of 96.95% on the ASVSpoof2017 dataset. While these results highlight the potential of using pop noise detection and CNNs for voice liveness detection, additional work is required to validate these findings across more varied scenarios. Future efforts will focus on expanding the dataset, experimenting with alternative feature extraction methods such as Short-Time Fourier Transform (STFT) and Morse wavelet, and training other algorithms from the literature, including Gaussian Mixture Models (GMM) and Support Vector Machines (SVM). These efforts

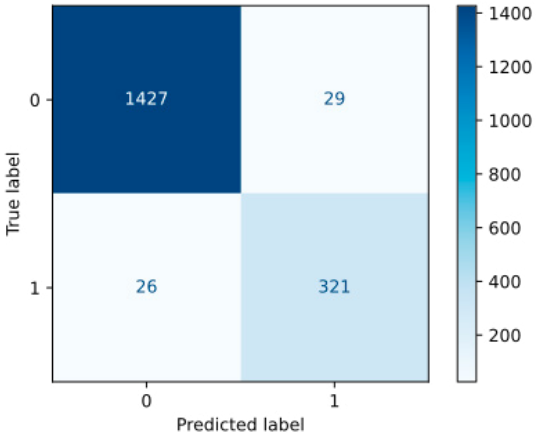


Fig. 5. Confusion Matrix on Test Data

aim to further optimize the CNN architecture and enhance performance and generalization across a broader range of spoofing scenarios.

While our approach effectively distinguishes genuine from spoofed audio using the presence of pop noise, we recognize the risk that sophisticated attackers might attempt to improve their spoofing techniques. For example, post-processing could be employed to enhance recordings, making them sound more like genuine speech captured directly into a microphone. We are actively exploring methods to enhance model robustness against such advanced spoofing techniques, including the use of adversarial training and features that are less susceptible to post-processing. Additionally, future work will investigate how to detect subtle inconsistencies that remain in post-processed audio.

We acknowledge that certain details of our methodology have been intentionally withheld. This decision is driven by ongoing work on two company projects where we plan to further develop and potentially productize this technology. The core techniques—such as the use of Constant-Q Transform (CQT) for feature extraction and the overall CNN-based structure—are fully detailed within this paper to enable replication and scientific evaluation. However, specific optimizations and proprietary algorithms that enhance efficiency and robustness have been withheld to maintain a competitive advantage. We believe this balance between transparency and commercial protection will allow the scientific community to benefit from our findings while we continue refining the model for practical applications.

Acknowledgements

The authors would like to thank the EuroHPC Joint Undertaking for providing access to the Meluxina supercomputer, which offered substantial computational power necessary for the efficient training of our deep learning models. The allocation of 400 node hours through the EuroHPC benchmark project was instrumental in achieving the high accuracy and reliability demonstrated in our results.

This work is supported by TUBITAK-TEYDEB under Project No. 3230629, whose financial support made this research possible.

References

[1] Desplanques, Brecht, Jenthe Thienpondt, and Kris Demuynck. (2020). “Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification.” *arXiv preprint arXiv:2005.07143*.

[2] Koluguri, N. R., Park, T., Ginsburg, B. (2022). “TitaNet: Neural Model for speaker representation with 1D depth-wise separable convolutions and global context.” *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. https://doi.org/10.1109/icassp43922.2022.9746806.

[3] Baevski, Alexei, et al. (2020). "wav2vec 2.0: A framework for self-supervised learning of speech representations." *Advances in neural information processing systems* **33**: 12449-12460.
[4] Chen, Sanyuan, et al. (2022). "Unispeech-sat: Universal speech representation learning with speaker aware pre-training." *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.
[5] Chen, Sanyuan, et al. (2022). "Wavlm: Large-scale self-supervised pre-training for full stack speech processing." *IEEE Journal of Selected Topics in Signal Processing* **16.6**: 1505-1518.
[6] Ahmed, Muhammad Ejaz, et al. (2020). "Void: A fast and light voice liveness detection system." *29th USENIX Security Symposium (USENIX Security 20)*.
[7] Gupta, Priyanka, Siddhant Gupta, and Hemant Patil. (2021). "Voice liveness detection using bump wavelet with CNN." *9th International Conference on Pattern Recognition and Machine Intelligence*.
[8] Gupta, Siddhant, et al. (2021). "Deep convolutional neural network for voice liveness detection." *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE.
[9] Khoria, Kuldeep, Ankur T. Patil, and Hemant A. Patil. (2021). "Significance of Constant-Q transform for voice liveness detection." *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE.
[10] Gupta, Priyanka, and Hemant A. Patil. (2024). "Morse wavelet transform-based features for voice liveness detection." *Computer Speech and Language* **84**: 101571.
[11] Wu, Z., Kinnunen, T., Evans, N., Yamagishi, J., Hanilçi, C., Sahidullah, Md., Sizov, A. (2015). "ASVSPOOF 2015: The first automatic speaker verification spoofing and countermeasures challenge." *Interspeech 2015*. https://doi.org/10.21437/interspeech.2015-462.
[12] Delgado, H., Todisco, M., Sahidullah, M., Evans, N., Kinnunen, T., Lee, K. A., Yamagishi, J. (2018). "ASVSPOOF 2017 version 2.0: Meta-data analysis and baseline enhancements." *The Speaker and Language Recognition Workshop (Odyssey 2018)*. https://doi.org/10.21437/odyssey.2018-42.
[13] Akimoto, K., Liew, S. P., Mishima, S., Mizushima, R., Lee, K. A. (2020). "Poco: A voice spoofing and liveness detection corpus based on pop noise." *Interspeech 2020*. https://doi.org/10.21437/interspeech.2020-1243.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 73–82

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

MASFENON: implementing a multi-agent simulation framework for interconnected networks with distributed programming

Giorgio Locicero[a,*], Antonio Di Maria[b], Salvatore Alaimo[b], Alfredo Pulvirenti[b]

[a]*Dept. of Physics and Astronomy, Via Santa Sofia 64, Catania 95123, Italy*

[b]*Dept. of Clinical and Experimental Medicine, c/o Dept. of Math and Comp. Science, Via Santa Sofia 64, Catania 95123, Italy*

Abstract

The complexity of networked systems, particularly interconnected networks, necessitates advanced simulation frameworks to accurately emulate real-world dynamics, especially in the context of big data and high-performance computing. Most software used for simulation and temporal inference usually falls short in large data and optimization, since it is generally used in particular contexts, like simulating the dynamics of a specific group of entities, such as cellular and community interactions. We present "Multi-Agent Adaptive Simulation Framework for Evolution in Networks of Networks" (MASFENON). MASFENON employs a temporal multi-layered approach to simulate and analyze dynamic processes in interconnected networks. The framework leverages parallel programming techniques for matrix and linear algebra operations and distributed and reactive programming for agent and environment communication, all implemented in C++ using the Message Passing Interface (MPI) standard. MASFENON has been validated against several common network models and could simulate the behavior of real systems in the context of epidemic simulations (See [4]). The framework demonstrates sublinear speedup and scalability with network size. The implementation is open source and available in a regularly updated GitHub repository[1] . MASFENON's integration of MPI and distributed programming techniques provides a powerful and versatile tool for modeling complex network interactions and dynamics. Its capabilities extend beyond traditional models, offering new insights and applications in network science.

Keywords: Adaptive systems; Multi-agent; Temporal evolution; distributed programming

1. Introduction

Studying interconnected dynamic systems is a prevalent paradigm across various scientific and real-world domains. Networks have proven to be powerful tools for understanding their complexity, and the emergence of interconnected networks has added a new dimension to the study of systems comprising multiple interacting entities. In this context,

[1] https://github.com/josura/c2c-sepia

* Corresponding author.
E-mail address: giorgio.locicero@phd.unict.it

1877-0509
10.1016/j.procs.2025.02.262

developing simulation frameworks capable of handling large amounts of data and accurately capturing real-world dynamics is essential.

We present MASFENON, a Multi-Agent Adaptive Simulation Framework for Evolution in Networks of Networks. MASFENON employs a temporal multi-layered approach to simulate and analyze dynamic processes in interconnected networks. The framework integrates propagation, dissipation, and conservation principles to emulate intra-network and inter-network dynamics over time. The framework can be applied in any domain requiring analyzing an attributed network.

MASFENON's implementation leverages high-performance computing techniques to handle the computational demands of simulating large and complex networks. The core of MASFENON is written in C++ and utilizes the Message Passing Interface (MPI) standard for parallel programming. This allows for efficient distribution of computational tasks across multiple processors and nodes. Parallel programming techniques are employed for matrix and linear algebra operations, while distributed and reactive programming handles the implementation of the multi-agent system.

2. Materials and methods

The MASFENON framework is a novel methodology for modeling complex systems, using a multi-agent approach and principles of chaos theory to simulate and analyze dynamic processes in interconnected networks. MASFENON relies on an algorithm that integrates a two-fold communication channel - within individual agents and between them. Every agent has an underlying network topology. In our context, networks have a hierarchical structure in which node connections can happen at different resolution levels. Additionally, the MASFENON framework is conceived as a flexible and adaptable framework that could be tuned to model the behavior of a real system, this is done by estimating the best parameters that will emulate the real data observations.

The algorithm also incorporates some ideas of deterministic chaos and chaos theory, where concepts like sensitivity to initial conditions, unstable behavior, unpredictability under certain conditions, and attractors are directly integrated and considered during the algorithm's development. The algorithm's components resemble the logistic map and other topics (i.e., periodic orbits, topological mixing). Indeed, central to MASFENON's operation are the principles of **decay** and **conservation**. These dual mechanisms are pivotal in mimicking real-world system dynamics, where some components diminish over time or through interactions while others are preserved or passed on within the network.

Since the framework was implemented with MPI and in a distributed setting, it can handle large amounts of data and throughput.

As a use-case workflow for the framework, we present a setting where biological data is modeled, in particular, we model Cell-to-Cell interactions via the use of scRNA-seq, signaling networks, and metabolic networks seen in Fig. 1. The use case is in the context of bioinformatics, and the simulation size could range from 100 GB (for 10-15 cell types) to some TB(100-200 cell types).

2.1. Theoretical methods

MASFENON integrates multi-agent modeling with a two-fold communication channel, enabling intricate interactions within and across networks in two separate timeframes: within individual agents and between them. Each agent is associated with an underlying network topology, which can be unique or shared among agents, facilitating diverse interaction patterns.

In MASFENON, communication between agents occurs temporally, with agents interacting at different contact times, forming a temporal multi-graph. This structure allows for detailed simulation of dynamic processes while maintaining static network topologies for individual agents throughout each step.

Within our model, the state vector $\vec{v}^{(n)}$ is influenced by dissipation and conservation functions, which together shape the most important framework's capabilities.

Throughout the paper, we use the following notation:

- $G_{Ag} = (V(G_{Ag}), E(G_{Ag}))$: A graph representing the network of agent $Ag \in V(\text{Agents})$, where $V(G_{Ag})$ is the set of nodes, and $E(G_{Ag})$ is the set of edges.

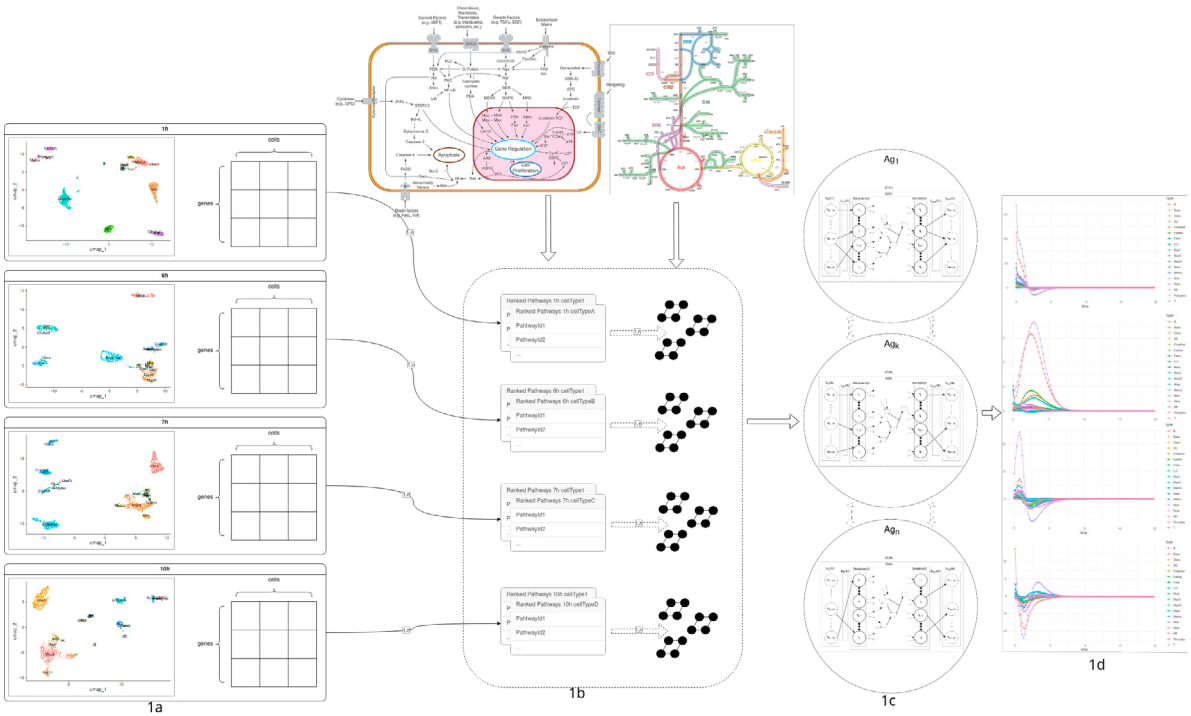


Fig. 1: MASFENON workflow for scRNA-seq data to model C2C communication. The toy example starts from longitudinal scRNA-seq data coupled with metabolic networks and signaling pathways. We create expression matrices for every timepoint (Fig. 1a). Then, the interactions between cell types are inferred from differential expression values between cell types. Since we have temporal scRNA-seq data, we have a dynamic cell-cell interaction. According to the expression values, for each cell, and for each time point we can select the most activated pathways and metabolic networks (Fig. 1b). Such generated data is then passed to MASFENON which allows the simulation of the temporal evolution of cells (Fig. 1c). The results will be a time series, showing the dynamic evolution of the system of cells (Fig. 1d). This kind of simulation could be helpful in many life science applications.

- $V_{in}(Ag) = \{v_{in_1}, v_{in_2}, ..., v_{in_q}\}$: Set of virtual nodes representing input information to the network associated with agent Ag.
- $V_{out}(Ag) = \{v_{out_1}, v_{out_2}, ..., v_{out_z}\}$: Set of virtual nodes representing output information from the network associated with agent Ag.
- $E_{in}(Ag) = \{(v_{in_i}, v_j) | v_{in_i} \in V_{in}(Ag), v_j \in V(G_{Ag})\}$: Set of edges with virtual inputs as sources and graph nodes as targets.
- $E_{out}(Ag) = \{(v_j, v_{out_i}) | v_{out_i} \in V_{out}(Ag), v_j \in V(G_{Ag})\}$: Set of edges with graph nodes as sources and virtual outputs as targets.
- $\mathcal{G}_{Ag} = (V(\mathcal{G}_{Ag}), E(\mathcal{G}_{Ag}))$: The augmented graph with $V(\mathcal{G}_{Ag}) = V(G_{Ag}) \cup V_{in}(Ag) \cup V_{out}(Ag)$ and $E(\mathcal{G}_{Ag}) = E(G_{Ag}) \cup E_{in}(Ag) \cup E_{out}(Ag)$. Nodes in $\mathcal{G}_{Ag}$ include both normal and virtual nodes, and edges represent connections between them.
- $\vec{v}^{(n)}(\mathcal{G}_{Ag}) \in \mathbb{R}^{\dim(Ag)}$: Vector of node values at iteration n, representing perturbation values.
- $U(v, \mathcal{G})$: Function returning the set of predecessors (upstream nodes) of v in $\mathcal{G}$.
- $D(u, \mathcal{G})$: Function returning the set of successors (downstream nodes) of u in $\mathcal{G}$.
- $w_{\mathcal{G}_{Ag}}$: Weight function for each edge $(v_u, v_d) \in E(\mathcal{G}_{Ag})$.
- $\overline{w}_{\mathcal{G}_{Ag}}$: Normalized weight function for each edge $(v_u, v_i) \in E(\mathcal{G}_{Ag})$.
- $\overline{W}(\mathcal{G}_{Ag}) \in \mathcal{M}_{\dim(Ag)\times\dim(Ag)}(\mathbb{R})$: Normalized adjacency matrix of $\mathcal{G}_{Ag}$.
- Agents $= (V(\text{Agents}), E(\text{Agents}))$: Temporal multi-graph of agent interactions, where $V(\text{Agents}) = \{Ag_0, Ag_1, ..., Ag_s\}$ and $E(\text{Agents}) = \{(v_i, v_j, d) | v_i, v_j \in V(\text{Agents}), d \in D\}$, with D as the set of edge attributes including contact times.

The dissipation function adjusts the node states vector to reflect the diminishing influence over time. The function is defined as follows:

$$d(\vec{v}, t_n) = \vec{v} - \lambda(t_n)\vec{v} \tag{1}$$

Where $\lambda : \mathbb{R} \longrightarrow \mathbb{R}$ is a scaling function that modulates the magnitude of dissipation.

The conservation function models the constraints over specific conditions within the network. It is defined as follows:

$$c(\mathcal{G}, \vec{v}, t_n) = \theta(t_n)((\overline{W}(\mathcal{G})\vec{q}(\mathcal{G})) \odot \vec{v}) \tag{2}$$

Here, $\theta : \mathbb{R} \longrightarrow \mathbb{R}$ is a scaling function that adjusts the conservation parameter, and $\overline{W}(\mathcal{G})$ is the normalized adjacency matrix associated with the graph $\mathcal{G}$, and $\vec{q} \in \mathbb{R}^{dim(Ag)}$ is the vector representing the user-defined parameters used to weight the conservation function. The operation $\odot$ represents element-wise multiplication.

To propagate the values in the network, we define the following propagation function:

$$f(\mathcal{G}, \vec{v}, t_n) = \vec{v} + \omega(t_n)\overline{W}(\mathcal{G})^T\vec{v} \tag{3}$$

Where $\omega : \mathbb{R} \longrightarrow \mathbb{R}$ is the scaling function for the propagation. The iterative dynamics of the model, which includes the dissipation and conservation mechanisms alongside propagation, is defined as follows:

$$\vec{v}^{(n)}(\mathcal{G}) = \begin{cases} f(\mathcal{G}, \vec{v}^{(0)}(\mathcal{G}), t_n) & \text{if } n = 1 \\ f(\mathcal{G}, d(\vec{v}^{(n-1)}(\mathcal{G}), t_n), t_n) - c(\mathcal{G}, d(\vec{v}^{(n-1)}(\mathcal{G}), t_n), t_n) & \text{if } n > 1 \end{cases}, \tag{4}$$

where, the functions f, d, and c correspond to Eq 3, Eq 1, and Eq 2, respectively, and t_n is the time at iteration n. At the initial step ($n = 0$), the system starts with the input state vector $\vec{v}^{(0)}(\mathcal{G})$, where each value denotes the initial state of the nodes in the augmented graph $\mathcal{G}$.

Inter-propagation mechanism is implemented with the use of virtual nodes associated with the interaction between different networks. The inter-propagation mechanism follows the intra-propagation after N_{intra} iterations. After that, values computed for virtual outputs in the source augmented networks are transferred to their counterpart virtual input nodes in target networks, effectively mimicking inter-community member interactions.

Formally, let n be an iteration where the inter-propagation is done. Given a virtual input node $v_x \in V(\mathcal{G}_{target})$ and its source virtual output node $v_y \in V(\mathcal{G}_{source})$. We update its value to:

$$\vec{v}_x^{(n)}\left(\mathcal{G}_{target}\right) = \vec{v}_y^{(n)}(\mathcal{G}_{source}) \tag{5}$$

The process of intra-propagation followed by inter-propagation is repeated for a fixed amount of iterations (N_{inter}).

2.2. Computational methods

MASFENON is implemented in C++ using the MPI standard for distributed programming to achieve high performance. Matrix and linear algebra operations are parallelized to efficiently distribute computational tasks across multiple processors. Distributed and reactive programming techniques manage agent communication and environment interactions, ensuring scalable and efficient simulation of large and complex networks. This HPC-oriented implementation allows MASFENON to handle extensive computational demands, achieving sublinear speedup and enabling detailed and scalable simulations of network dynamics.

The MASFENON framework integrates into a Multi-agent System (MAS) where autonomous agents interact within a shared environment.

2.2.1. Agent Types

MASFENON models agents as a temporal multi-graph

$$Agents = (V(Agents), E(Agents))$$

Here, $V(Agents) = \{Ag_0, Ag_1, \ldots, Ag_s\}$ are the agents, and $E(Agents) = \{e_i | e_i \in V(Agents) \times V(Agents) \times D\}$ are the interactions, with D representing attributes like contact times.

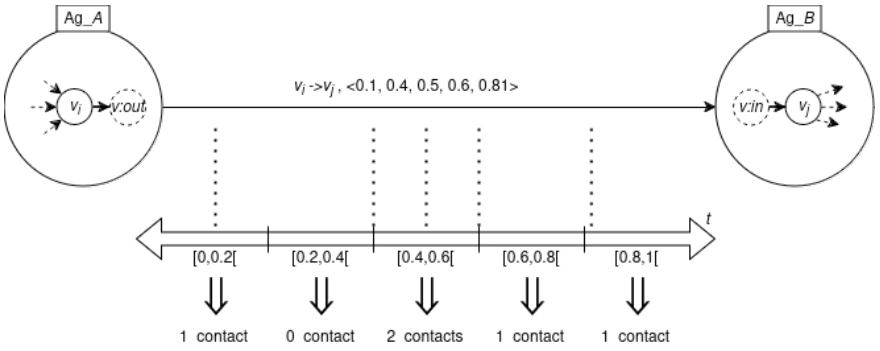


Fig. 2: **Contact quantization** for an edge in the *Agents* temporal network.

Agent types can be defined based on application needs, such as individuals, communities, or organizations. Agents of the same type share behaviors and communication patterns, facilitating scalable and modular simulations.

2.2.2. Temporal Communication

Each edge in the *Agents* network is linked to real contact times. Virtual outputs in the source agent are linked to virtual inputs in the target agent, enabling inter-agent communication via temporal edges.

The iterative propagation algorithm that was described uses the following time parameters:

- $currentN_{inter}$: the number of the current inter-iteration;
- $currentN_{intra}$: the number of the current intra-iteration;
- N_{intra}: the maximum number of intra-iterations for each inter-iteration;
- N_{inter}: the maximum number of inter-iterations;
- $timestep$: the time between two inter-iterations.

The iteration $n \in \mathbb{N}$ is defined as: $n = currentN_{inter} * N_{intra} + currentN_{intra}$

In MASFENON, time is divided into uniform intervals, each spanning the same length ($timestep$). Therefore, the current time (t_n) at iteration n is defined as $t_n = \frac{timestep}{N_{intra}} \times n$.

2.2.3. Iterative Propagation Algorithm

Communication between agents occurs at intervals

$$\{[t_0, t_{N_{intra}}[, [t_{N_{intra}}, t_{N_{intra}*2}[, \ldots, [t_{N_{intra}*(N_{inter}-1)}, t_{N_{intra}*N_{inter}}[\}$$

each corresponding to N_{intra} iterations. Contact times can be treated in two ways :

- single events within each interval, if at least one event is recorded in the interval.
- multiple events within each interval, meaning that each interaction inside the intervals counts toward the inter-propagation.

Since contact times for the interactions between agents are defined with single units and not intervals, we quantize the contact times inside each interval, meaning that more contact times inside of an interval will be seen as a single propagation with more than one contact. An example of this time quantization of the contact times can be seen in Fig. 2

Since there could be more contact times in an interval, we established two different alternatives that some parameter in the software can specify, these two cases can be seen in Eq. 6 and 7:

$$\#contacts(t_i, t_{i+1}) \begin{cases} = 0 & \text{edge is not considered} \\ > 0 & \text{edge is considered} \end{cases} \tag{6}$$

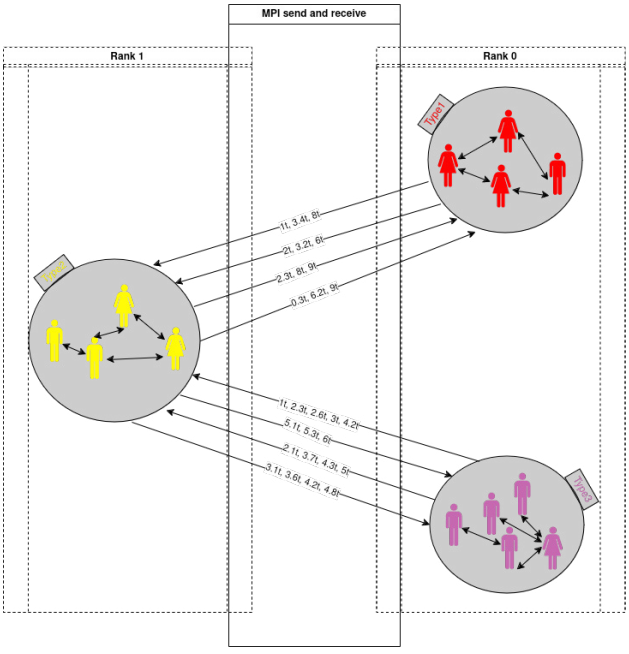


Fig. 3: MAS implementation of MASFENON in MPI. Groups of agents are organized in workgroups where the maximum number of agents is the workload. Every group is associated with a rank that uniquely identifies the group and dictates how the groups communicate. Details are available in the documentation and the repository .

The Eq. 6 considers the multiple contact times in the interval as a single inter-propagation between two agents.

$$\#contacts(t_i, t_{i+1}) \begin{cases} = 0 & \text{edge is not considered} \\ = n, n >= 1 & \text{edge is considered, value is multiplied by } n \end{cases} \tag{7}$$

The Eq. 7 considers the multiple contact times in the interval as the force of interaction of the inter-propagation.

The implementation for the multi-agent system uses the MPI standard, specifically the OpenMPI implementation. Each agent operates as a single unit within a workgroup, with each workgroup represented as a single MPI processor. This setup aligns processors with real nodes in a cluster, ensuring generality with available resources. Figure 3 illustrates this implementation in the context of communities, where a single community(a sub-network of the whole population) is treated as as single agent.

The number of agents per workgroup defines the workload for each processor, calculated as:

$$workload = \left\lceil \frac{|V(Agents)|}{n_{processors}} \right\rceil \tag{8}$$

The number of processors cannot exceed the number of agents on available MPI system. If we have an even distribution of workload, the last processor may have fewer agents. This possible under-load of the processor do not negatively impact the performance of the framework. However, additional improvements could use an optimized intra-processor communication.

The intra-propagation mechanism is implemented using the Armadillo library (See [1] and [2]) for efficient linear algebra operations. This ensures high-performance computation within each agent. For inter-propagation, OpenMPI (See [3]) manages communication between different agents and workgroups. Inter-agent communication across workgroups uses MPI to transfer information, ensuring robust and scalable interactions.

2.2.4. Computational complexity

The computational complexity of the MASFENON framework primarily hinges on the matrix multiplication operations involved in simulating propagation dynamics across interconnected networks. The complexity is most pro-

nounced during the intra-propagation mechanism, where subgraph adjacency matrices undergo multiplication to capture the influence of neighboring nodes.

The critical operation driving computational complexity is the multiplication of subgraph adjacency matrices. Considering a graph with n nodes, the size of the adjacency matrices is $n \times n$. Therefore, the complexity of matrix multiplication is traditionally $O(n^3)$.

As the size of the graph (n) increases, the cubic nature of n^3 complexity becomes a significant factor in determining the computational load. This complexity arises during the propagation steps, where the influence of each node on its neighbors is computed.

Efforts have been made to optimize the computational performance, particularly focusing on the matrix multiplication steps. These leverage on parallel computing library (i.e. Armadillo) together with parallel programming standards OpenMP and MPI. This allows us to exploit multicore architectures, mitigating the impact of $O(n^3)$ complexity and consequently enhancing the overall computational efficiency.

Given the computational demands, scalability becomes a key consideration. While $O(n^3)$ complexity poses challenges, MASFENON is designed to harness parallel processing capabilities, enabling the framework to scale efficiently with increasing graph sizes.

As we acknowledge the inherent computational complexity, ongoing research explores avenues for further optimization and potential algorithmic enhancements, especially considering the use of Armadillo for linear algebra (using LAPACK[2] and an implementation of blas, for example, cuBLAS[3]) that limits our freedom in additional optimization on memory and the use of GPUs. Profiling of the CPUs-GPUs suggests that the hardware is used but seems underloaded during transfers between memory and when sending data with MPI. Future iterations of MASFENON may introduce refined algorithms or leverage advancements in parallel computing technologies to address scalability concerns, especially in the face of the continuous extension of the framework capabilities and adaptability.

The space complexity of the problem is mainly based on the order of the graphs passed as the input, the number of graphs, and the interaction between the different networks (the size of the inter-communication network between the agents). Formally, given the maximum order among the graphs passed as input $max|V|$, the number of graphs $|V(Agents)|$ and the size of the inter-communication network $|E(Agents))|$, the space complexity is:

$$O\left(\left(max|V| + \frac{|E(Agents))|}{|V(Agents)|}\right)^2 * |V(Agents)|\right) \tag{9}$$

This space complexity is composed of the following parts:

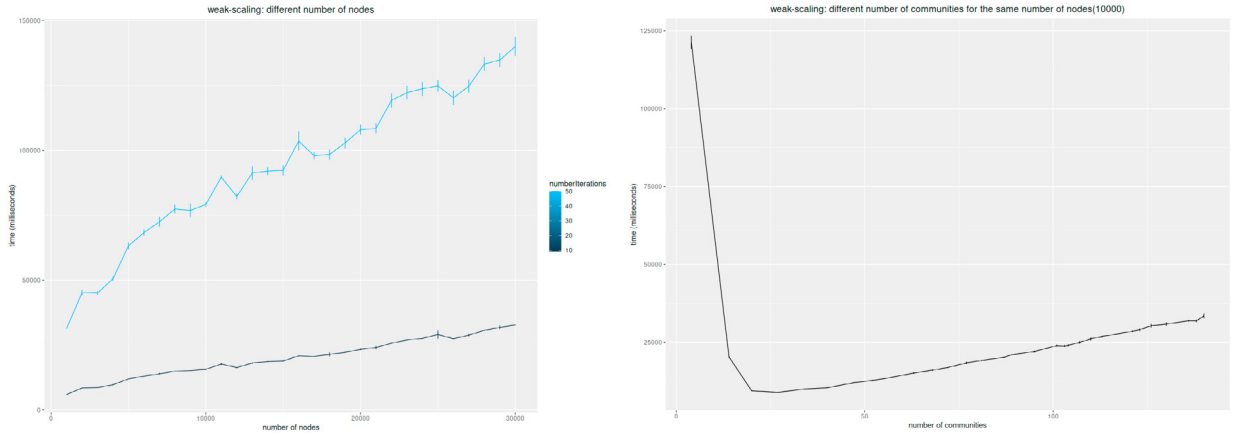
- $\left(max|V| + \frac{|E(Agents))|}{|V(Agents)|}\right)$ is the maximum amount of nodes that can be contained in an augmented graph;
- $\left(max|V| + \frac{|E(Agents))|}{|V(Agents)|}\right)^2$ is the amount of memory needed for an adjacency matrix;
- $O\left(\left(max|V| + \frac{|E(Agents))|}{|V(Agents)|}\right)^2 * |V(Agents)|\right)$ is the maximum amount of memory needed for all the adjacency matrices.

3. Results

In this section, we present the performance results of the MASFENON framework, evaluating its computational efficiency, scalability, and effectiveness in various multi-agent system scenarios.

The performance of the framework has been rigorously evaluated through a series of experiments focusing on both weak and strong scaling scenarios. These experiments provide valuable insights into the framework's behavior concerning graph size, community structure, and the utilization of parallel processing resources.

[2] https://netlib.org/lapack/
[3] https://docs.nvidia.com/cuda/cublas/index.html



(a) Variable number of nodes in the graph, runtime for 5 total experiments and 50 iterations, processor count fixed to 16

(b) Variable number of communities for the same graph of 10000 nodes, runtime for 5 experiments and 50 iterations, processor count fixed to 16

Fig. 4: Weak scaling performance testing, plotting runtime (milliseconds) for the number of nodes for the graphs used, and for the number of communities for a graph of 10000 nodes, line plot is done with the averages of the runtimes for every variable, 5 runs are done for every variable, error bars are shown

3.1. Weak Scaling: Graph Size and Community Structure

Experiments were conducted to assess the weak scaling behavior of SC-PHENSIM concerning graph size. The execution time was measured while varying the number of nodes in the graph, ranging from 1000 to 30000. The results, presented in Figure 4a, exhibit a linear growth in execution time with the graph size. This observation aligns with expectations given the $O(n^3)$ complexity associated with matrix multiplication. Larger graphs necessitate more extensive matrix operations, resulting in a proportional increase in computational time.

Another facet of weak scaling exploration involved manipulating the community structure within a fixed-size graph of 10000 nodes. Varying the number of communities while keeping the total graph size constant, experiments revealed nuanced dependencies. The plot in Figure 4b indicates that very low or high community counts lead to longer execution times. A notable observation is the presence of a lower bound on execution time, demonstrating that an optimal balance in community count exists for efficient processing.

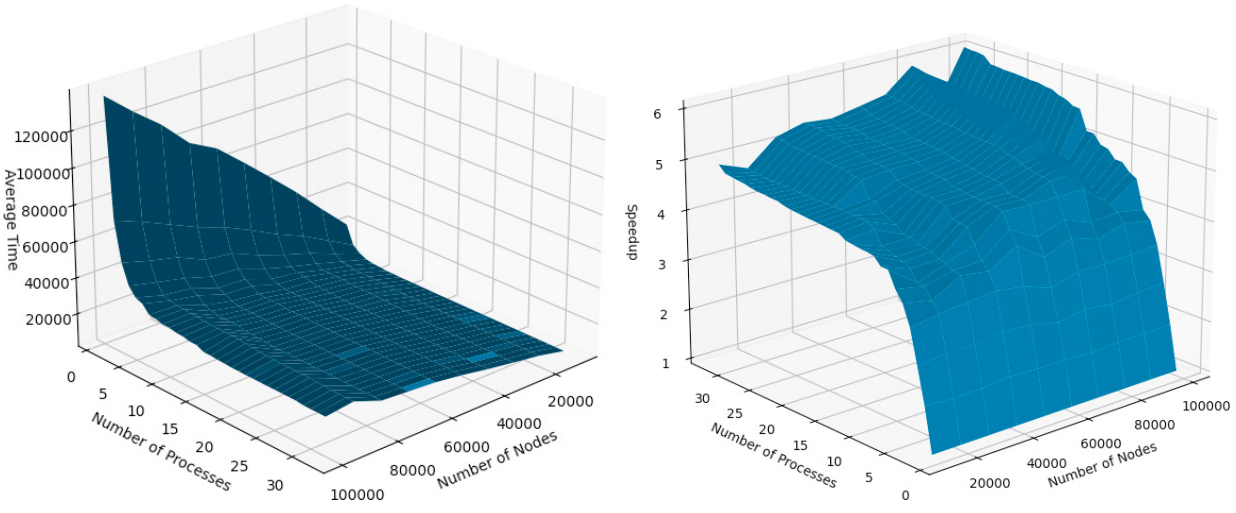
3.2. Strong Scaling: MPI Processor Variation

The strong scaling behavior of MASFENON was scrutinized by varying the number of MPI processors from 1 to 32 on random graphs generated with the Barabasi-Albert[5] model with nodes from 10000 to 100000, by incrementing the number of nodes by 10000 at every step.

The configuration of the cluster used for the experiments is:

- 1 to 4 number of nodes with 32 GB;
- 1 to 8 number of tasks per node;
- 1 to 4 number of CPUs Ice Lake at 2.60 GHz per task;
- 1 GPU NVIDIA A100 per node;
- Internal Network: 200G HDR Infiniband Dragonfly+ ;
- SLURM 22.05

The results, depicted in Figure 5a, exhibit a hyperbolic growth pattern. As the processor count increases, average execution times between all experiments decrease significantly, showcasing the framework's ability to leverage parallel processing resources effectively.



(a) Runtime for variable number of processors

(b) Average speedup variation

Fig. 5: Strong scaling performance testing, plotting average runtime (milliseconds) for the number of processors for the graphs used, and speedup graph. 5 runs are done for every variable, average time is shown and used for speedup computation

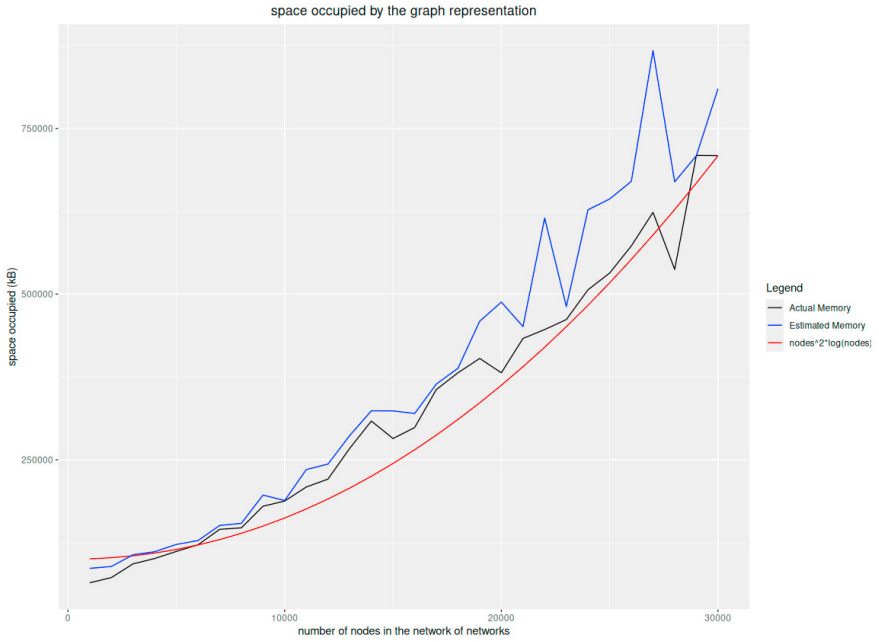


Fig. 6: Memory used for the total number of nodes for every problem, the blue line represents the estimated memory using the formula previously seen in 2.2.4, the red line represents the function $x^2 log(x)$

The average speedup, presented in Figure 5b, illustrates the efficiency gained by employing additional processors and is obtained with the computation of the speedup value as $\frac{\text{runtime}(1P)}{\text{runtime}(nP)}$, that is the average runtime with *1* processor divided by the average runtime with n processors. The logarithmic growth observed signifies the substantial impact of processor count on achieving higher speedups, especially when augmenting the dimension of the network.

The memory occupied by changing the number of nodes can be seen in Fig. 6

4. Conclusions

The MASFENON framework has been designed with a strong focus on handling big data and leveraging high-performance computing (HPC) environments. The framework can process large interconnected networks by utilizing OpenMPI for scalable operations across multiple processors.

The limitations of the framework

Moving forward, the framework will be expanded and enhanced to support even more complex analyses, such as optimizing the framework for emerging HPC architectures, including integrating GPU acceleration and hybrid computing strategies. Additionally, the framework will be adapted to incorporate more functionalities based on user feedback, such as vector states for the single nodes in every agent, enabling deeper insights into dynamic systems and their interactions in different contexts.

Acknowledgments

This research has been supported by the NextGenerationEU funding program, which provided essential support for Giorgio Locicero's Ph.D. Studies. The research has been also supported by the ICSC Project: "Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing - Spoke 8: Insilico Medicine and Omics Data" (CN_00000013 – Avviso n. 3138 del 16 dicembre 2021). We also acknowledge CINECA for awarding us access to the Leonardo supercomputer (EUROHPC project code: CNHPC_1452526)

References

[1] Conrad Sanderson and Ryan Curtin. Armadillo: a template-based C++ library for linear algebra. *Journal of Open Source Software*, Vol. 1, pp. 26, 2016. http://dx.doi.org/10.21105/joss.00026

[2] Conrad Sanderson and Ryan Curtin. Practical Sparse Matrices in C++ with Hybrid Storage and Template-Based Expression Optimisation. *Mathematical and Computational Applications*, Vol. 24, No. 3, 2019. https://doi.org/10.3390/mca24030070

[3] Hursey J., Mallove E., Squyres J.M., Lumsdaine A An Extensible Framework for Distributed Testing of MPI Implementations. In Recent Advances in Parallel Virtual Machine and Message Passing Interface. EuroPVM/MPI 2007. *Lecture Notes in Computer Science*, vol 4757. Springer, Berlin, Heidelberg.

[4] Locicero, Giorgio, et al. "MASFENON: Multi-Agent Adaptive Simulation Framework for Evolution in Networks of Networks." *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2023.

[5] Barabási, Albert-László, and Réka Albert. "Emergence of scaling in random networks." science 286.5439 (1999): 509-512.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 83–92

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

High-Performance Computing for Distributed Route Computation in Traffic Flow Models

Paulo Silva[a,*], Pavlína Smolková[a], Sofia Michailidu[a], Jakub Beránek[a], Roman Macháček[a], Kateřina Slaninová[a], Jan Martinovič[a], Radim Cmar[b]

[a]*IT4Innovations, VSB - Technical University of Ostrava, Ostrava, Czech Republic*
[b]*Sygic a.s., Twin City C, Mlynské Nivy 16, Bratislava, Slovakia*

Abstract

In the dynamic landscape of smart cities and traffic management, it is necessary to further explore the synergistic potential of real-time traffic data and high-performance computing to optimise traffic flow through dynamic re-routing strategies. High-performance computing plays an essential role in achieving effective traffic flow optimisation. Our research builds upon existing studies highlighting the positive correlation between the integration of live traffic updates and route optimisation. The methodology involves simulations with our Ruth traffic simulator, where vehicles dynamically adjust routes based on up to date traffic information available to them at different levels. Scalability tests are conducted with varying numbers of CPUs and nodes to assess the simulator's capacity to scale. The results showcase the impact of live traffic data on both driving time and average speed, emphasising the adaptability of our approach for broader applications. In conclusion, our work not only advances the understanding of real-time traffic optimisation but also underscores the critical role of high-performance computing in achieving scalable solutions. The findings present practical implications for the implementation of dynamic re-routing strategies in transportation systems, paving the way for future research and real-world applications on smart cities.

Keywords: Traffic Modelling; Alternative Route Computation; Distributed Computation; Scalability.

1. Introduction

Traffic modelling simulates traffic flow, congestion, and vehicle behaviour within transportation networks using computer-based representations. This approach enables urban planners, engineers, and policymakers to make informed decisions aimed at improving urban mobility and reducing congestion.

As smart cities evolve, leveraging real-time traffic data is crucial for optimising traffic flow through dynamic re-routing strategies. HPC plays an essential role in achieving effective traffic flow optimisation via traffic modelling and simulation.

* Corresponding author. Tel.: +420 597 329 500.
E-mail address: paulo.silva@vsb.cz

1877-0509
10.1016/j.procs.2025.02.263

In this work, Sygic provides a mobility platform (Figure 1) for supporting cities with advanced traffic modelling. The platform absorbs big data such as Floating Car Data (FCD) and deploys traffic services. FCD may be historical data from sensors or synthetically generated by a traffic simulator. For this work, we have generated realistic FCD with the traffic simulator (Ruth) [1] developed by the team of Advanced Data Analysis and Simulations lab at IT4Innovations, National Supercomputing Centre of the Czech Republic.

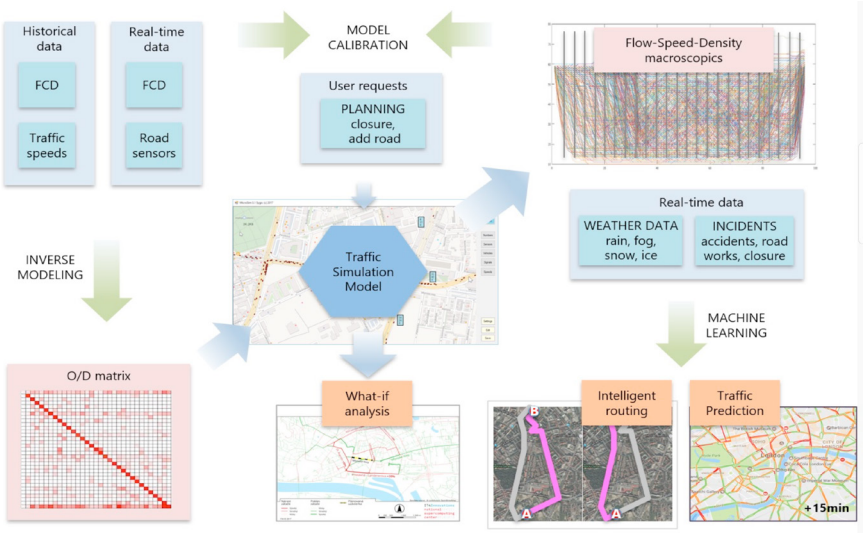


Fig. 1: Mobility Platform to Support Cities with Advanced Traffic Modelling.

The EVEREST [2] project provided the foundational tools and frameworks, such as evkit [3], that are crucial for distributed traffic simulation, while Sygic's platform integrates these tools with real-time data for enhanced urban traffic management. We have leveraged EVEREST project outputs and aimed at the removal of bottlenecks - conducting extensive simulations, and fine-tuning the system for enhanced benefits.

This paper presents our work on refining and optimizing the Ruth traffic simulator to improve urban traffic flow, reduce congestion, and enable scalable execution on HPC clusters. Our aim is to enhance traffic modeling for efficient urban planning and to reduce driving times while managing computational costs effectively. We focus on the following objectives:

1. Determining the optimal proportion of vehicles needing real-time traffic updates and rerouting to achieve faster driving times.
2. Identifying strategies to optimize the performance and efficiency of traffic simulations.
3. Evaluating the scalability of the simulator in distributed setups by comparing two urban areas (Prague and Boston) across two HPC clusters (Barbora and Karolina at IT4Innovations, Czech Republic).

Our research contributes to the assessment of how real-time traffic optimisation and alternative route computation influences the overall improvement of traffic flow within cities. The refined system is poised to significantly benefit urban mobility planning.

The remainder of the paper is organised as follows. Section 2 introduces the main concepts of traffic modelling and covers related work in the field. Section 3 presents the traffic simulator employed in this work. Section 4 presents the methodology applied, the simulation settings, data, and deployment parameters. Sections 5 presents the results and analysis of the results achieved. Section 6 concludes the document by highlighting the main considerations and comments on future research directions.

2. Background

This work focuses on traffic modelling for analysing traffic flow, congestion, and vehicle behaviour using computational representations of transportation networks. The following subsections introduce the main concepts, tools and approaches surrounding this work.

2.1. Modelling Approaches and Tools

There are multiple modelling approaches, such as Cellular Automata Models [4], Agent-based Models [5] or Probabilistic Models [6]. Along with the different modelling approaches, there are different kinds of simulations: macroscopic, mesoscopic, microscopic or nanoscopic. Macroscopic models focus on large-scale traffic flow, using principles similar to fluid dynamics. Microscopic models simulate individual vehicle movements, providing detailed insights into driver behaviour. Mesoscopic models strike a balance, offering a middle ground in terms of detail and computational demands.

Employing and implementing the aforementioned approaches and simulation types are various traffic modelling tools. Some of the most popular tools SUMO [7], VISSIM [8] or CORSIM [9] support different approaches of traffic modelling and simulation. In this work, we employ Ruth [1].

Ruth is a deterministic traffic simulator developed by the team of Advanced Data Analysis and Simulations lab at IT4Innovations, the National Supercomputing Centre of the Czech Republic. It supports traffic simulation on the mesoscopic level and it is the tool that supports the results presented on this paper. Additional details of the tool are presented in Section 3.

SUMO (Simulation of Urban Mobility) is an open source microscopic traffic simulation package. It provides a range of tools to generate traffic networks, simulate traffic flow and analyse results. It is designed to handle a wide variety of traffic scenarios, including urban and interurban traffic, public transport, and more. It is widely used in research and development for traffic management, urban planning, and vehicular communication systems.

VISSIM is commercial software package that digitally reproduces the traffic patterns on a microscopic scale. The software supports performance evaluation and optimisation of the transport infrastructure, enables data-based planning decisions, and can address challenges such as congestion, emissions, and the fair distribution of road space for different modes.

2.2. Routing Algorithms

In our work, routing maps are represented by directed graphs obtained via the OSMnx python library [10] connected to the OpenStreetMap API [11]. Edges of this graph represent road segments at a selected area. Among other data, the edges include information such as the maximum speed limit on the segment, its length and GPS coordinates defining the geometry of the road. During the visualisation, this geometry is used to maintain the original curvature of the road.

With respect to routing algorithms, in the scope of traffic modelling, the primary goal of alternative routing algorithm is not only to minimise the total travel time of the vehicles or distance in a journey, but also to decrease congestion in the selected area. For that end, various routing algorithms use specific approaches to find routes within a map (i.e., graph in computational terms).

Dijkstra's algorithm [12] is a well known and commonly used algorithm for finding a path between two nodes of a weighted graph; different approaches can apply the shortest or the fastest path based on the weights of the graph. For instance, K-shortest paths is a generalisation of the routing problem that can use Dijkstra or Bellman-Ford [13] algorithms. It computes the shortest route and subsequently iterates to generate alternative routes by making slight variations to the original path. Although this method is relatively straightforward, the resulting paths can be very similar to each other, often differing by only one or two segments.

Plateau [14] identifies potential alternative routes by identifying intersections of paths in both forward and backward directions on a map. These intersections, called "plateaus," are ranked based on how much they deviate from the overall path length. Plateau prefers routes that minimise detours, but it does not necessarily focus on finding the fastest or shortest routes - it looks for simple alternative paths even though these may still include detours. These two algorithms (Dijkstra and Plateau) are used in the experimental work presented in this paper.

3. Ruth - Traffic Simulator

Ruth is a deterministic, mesoscopic-level traffic simulator that serves as the foundational tool for the results presented in this paper. In this section, we provide an overview of Ruth's high-level architecture and describe the key functions and modules that were not only utilised but also optimised in this work.

3.1. Architecture

Figure 2 illustrates Ruth's high-level architecture. The "Simulator" block contains the core modules that drive the simulation. The "Distributed Workloads" block highlights some of the most computationally intensive modules, specifically the computation of alternative routes (*Plateau Speed*) and the Probabilistic Time-Dependent Routing (*PTDR*) kernels, which facilitate alternative route selection. These modules have been enhanced (in the scope of EVEREST project) to support parallel and distributed computation, leveraging High Performance Computing (HPC) resources for improved efficiency. Further optimisation of other modules is planned for future research and is beyond the scope of this work.

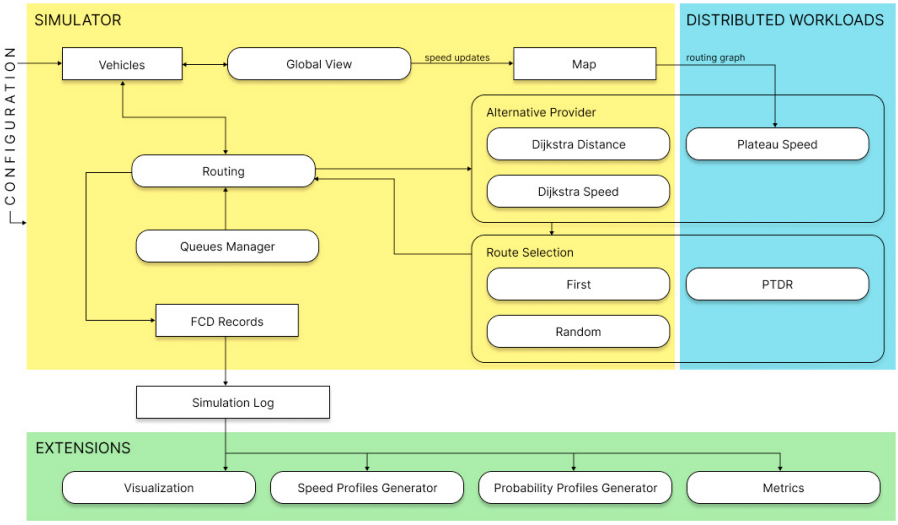


Fig. 2: High-level Architecture of Ruth Traffic Simulator.

3.2. Global View and Routing Mechanisms

The *Global View* mechanism logs vehicle instances at each road segment in the selected area over time. This allows the system to calculate the speed of a vehicle based on the number of vehicles ahead in the same segment. With this method, new vehicles entering a segment do not affect the speed of vehicles at the segment's end. The speed is determined using a set of thresholds, where a higher number of vehicles corresponds to a slower speed. Current segment speeds can be periodically updated in the routing map, enhancing the efficiency of alternative route calculation algorithms.

By determining vehicle speeds, the average speed on a segment can be calculated and used to inform routing algorithms. For example, if a segment's speed drops to 0.0 m/s, it is classified as a traffic jam, preventing further vehicles from entering the segment until conditions improve.

Vehicle movement is driven by the planned route and speed provided by the Global View. Routes can be dynamically adjusted based on updated traffic conditions and alternative route suggestions. During each simulation step, a vehicle updates its position on the current segment or moves to the next segment along its route. At the end of each

step, a Floating Car Data (FCD) record is generated, capturing the vehicle's timestamp, *ID*, and position on the segment. These records serve as the output of the simulation, forming the basis for data analysis and visualisation. The simulation concludes when all vehicles have reached their final destination.

Alternative routing becomes available when vehicles are at the end of their current segment. This ensures that alternative routes are suggested only when a vehicle can actually change its direction. Alternative routes are calculated using either the Dijkstra or Plateau algorithm, with the cost of each route determined by distance or updated traffic speed and road closures. The distributed runtime of the Plateau algorithm has been enabled through the integration of evkit [3].

To prevent continuous route switching in scenarios with similar travel times, a travel time threshold was also implemented (as shown in Section 4). When the time savings from switching routes do not exceed this threshold, the vehicle remains on its original route.

The simulator offers several route selection mechanisms, including basic random selection and an advanced Probability Time-Dependent Route Selection (PTDR) [15], which accounts for the probability distribution of speed within specific time windows for individual graph edges, estimating the total travel time distribution.

Queuing mechanisms prevent overtaking at crossroads and segment ends. A vehicle is added to a queue when it intends to cross to the next segment and is removed during the next movement step if no other vehicles block its path. Vehicles moving in the same step do not hinder those behind them, ensuring smooth transitions between the segments.

3.3. Interfaces

The *Simulator's Interface* serves as the user's gateway to configure, control, and monitor the simulation. It relies on an input *parquet*[1] file, which contains data on the road network, journey characteristics, and predefined routes, ensuring accurate initialisation.

A configuration file allows users to set all necessary simulation parameters, such as the percentage of vehicles using different algorithms for alternative route computation, the frequency of segment speed updates, the number of alternative routes computed, and several other parameters. The configuration file also enables the execution of planned events, such as altering segment speed limits or scheduling road closures to better simulate real-life scenarios.

The *Visualisation Interface* provides an immediate assessment of simulation results. After the simulation completes, a video showing changing segment speeds and vehicle counts can be generated. By logging vehicle positions during the simulation, the system can accurately represent traffic density not only between segments but also within individual segment parts.

The *Data Analysis Interface* supports in-depth examination of aggregated simulation results. Outputs include speed profiles, probability profiles, and CSV files that track simulation progress. These metrics capture essential details like average vehicle speed, the total number of active vehicles, and their behaviour based on different algorithms. The data also includes total distance driven, total driving time, and the number of segments crossed, all of which are tracked over the course of the simulation.

4. Methodology

The methodology followed in this work is divided in three main blocks that are summarised in Sections 4.1, 4.2, 4.3. The data and simulation parameters are presented in Section 4.4.

4.1. Determine How Many Vehicles Require Alternative Routing

The first part was to determine the optimal percentage of vehicles that require up to date traffic information and alternative routes computed to achieve faster driving times and smoother traffic flow.

A traffic simulation study was conducted with the road network of Prague, Czech Republic, and involved 10,000 vehicles. Two alternative routes were computed for the vehicles, with three different route change settings: 0%, 5%,

[1] An open source, column-oriented data file format designed for efficient data storage and retrieval.

and 10%. Vehicles would change their route if the alternative was at least as fast as their current route (0%) or 5% to 10% faster.

For each route change setting, 20 different simulations were run, varying the proportion of vehicles that could change routes from 0% to 100%, in 5pp increments (i.e., 0%, 5%, 10%, ..., 95%, 100%). This resulted in a total of 60 traffic simulations. This study measured the vehicles' driving time (simulation time) and execution time (computation time).

Real-time traffic information was represented by a global traffic view, with maps updated every 20 seconds. The simulations were executed on a HPC cluster, utilising one node on the Karolina system at IT4Innovations in the Czech Republic.

4.2. Distribution and Parallelisation

The second part was to enable distribution and parallelisation of the computationally expensive parts of the traffic simulation (i.e., alternative route computation).

This study involved integrating evkit (a distributed runtime enabler created within the EVEREST project) with Ruth traffic simulator to enable distributed computation of alternative routes. Evkit transfers inputs from the traffic simulator through the network to an evkit worker, which runs on another CPU or computational node. The alternative route (kernel) is computed by the worker and then sent back to the simulator.

Simulations were assessed for up to 3,600 seconds of computation time or up to 512 simulation steps. The performance of a sequential version (using Python and NetworkX Dijkstra) was compared with a distributed version (using C++ and Plateau). The computations were performed on an HPC cluster, utilising up to 16 nodes on the Barbora HPC system with 60,000 vehicles in the area of Prague.

4.3. Assess Traffic Simulator Overall Gains and Scalability

The third part was to launch a large scale simulation based on historical data on a city with dense traffic flow (Boston, USA) to collect information about the performance of the traffic simulator on a distributed setup with a large number vehicles.

This study was conducted with a road network of Boston, USA[2], and involved 300,000 vehicles (which resembles the real traffic load in the city. Two alternative routes were computed for each vehicle, with a route change setting that required the alternative to be 10% faster than the current route. We measured the number of simulation steps performed under various deployment setups, specifically focusing on the number of computational nodes used. The simulations were executed on an HPC cluster, using up to 50 nodes on Karolina at IT4Innovations, in the Czech Republic.

4.4. Data and Simulation Parameters

The input data necessary for this work considered an Origin and Destination (OD) matrix that was developed based on traffic flow descriptions and served as a key input for the traffic simulator. As mentioned before, this study focused on the routing networks of Prague and Boston, using previously mentioned algorithms (e.g., Plateau) to determine the fastest paths at every crossroad. The simulations were executed on different HPC clusters (Barbora and Karolina), with varying computational resources to assess the behaviour of the software under different conditions.

- Origin and Destination (OD) matrix: created based on traffic flow description.
- Prague routing network (graph): 21,817 nodes / 49,540 edges.
- Boston routing network (graph): 51,181 nodes / 125,888 edges.
- Routing algorithm: Dijkstra (Python) / Plateau (C++).
- Fastest path selection: 1, 2 or 3 alternatives available - selected if faster than (or as fast as) current path.

[2] https://zenodo.org/records/13285293

Attribute	Value(s)	Attribute	Value(s)
round-frequency-s	5	los-vehicles-tolerance-s	5
k-alternatives	[1,2,3]	travel-time-limit-perc	[0, 0.05, 0.1]
map-update-freq-s	[15,60]	saving-interval-s	[0, 100]

Table 1: Range of Simulations Settings Applied in the Various Simulations.

Each of the jobs considered the simulation setting described in Table 1 and explained hereafter:

- *round-frequency-s* is the interval for vehicle selection to be moved in one simulation step (in seconds, of simulation time).
- *k-alternatives* is the number of alternative routes calculated for each vehicle.
- *map-update-freq-s* is the frequency of updating the map with current speeds (in seconds, in simulation time).
- *los-vehicles-tolerance-s* is the time tolerance to count which vehicles (i.e., their timestamps) are considered for the calculation of Level of Service (LoS) in a segment (in seconds, of simulation time).
- *travel-time-limit-perc* represents the time (%) differential of the time between the current route and the fastest alternative route - meaning that for example vehicles only change to another path if the alternative path is 10% faster (in seconds, of simulation time).
- *saving-interval-s* represents the time interval in which the state of simulation is periodically saved.

Further details, datasets and configuration files can be accessed both on Ruth's GitHub repository [1] and Zenodo repository [16] of this experimental work.

5. Experimental Results

Figures 3 and 4 depict visualisation snapshots of the some simulations that took place throughout this work. On the left, the simulation in Prague has a more fluid traffic flow (represented with green colours). On the other hand, the simulations in Boston with significantly more vehicles, have increased traffic congestion and density (represented with yellow and red colours). The following subsections present the results of the previously mentioned experimental scenarios.

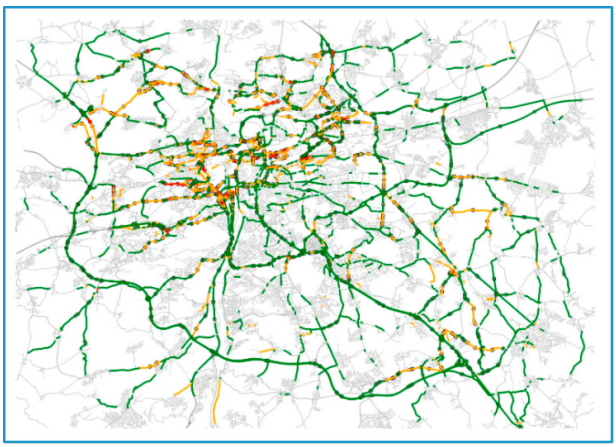


Fig. 3: Snapshot of Simulation with 10,000 Vehicles in Prague (Czech Republic).

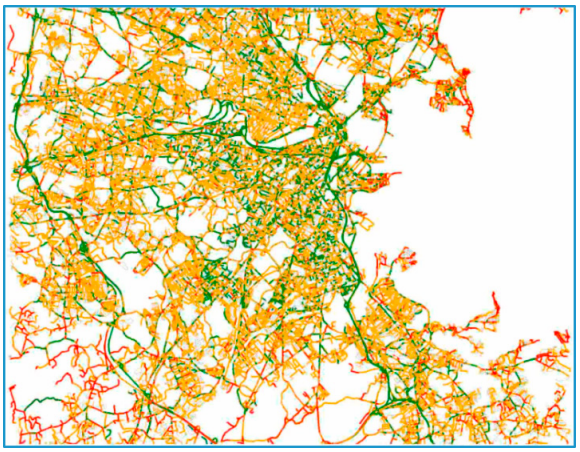


Fig. 4: Snapshot of Simulation with 300,000 Vehicles in Boston (U.S.A.).

5.1. Determining Portion of Vehicles Requiring Alternative Routing

In the first scenario, we have considered the assessment of how many vehicles would require live traffic updates and alternative route computation. As shown in Figure 5, the results indicate for the scenario of Prague with 10,000 vehicles the traffic flow actually improves when 45% to 50% of vehicles have live updates and compute their alternative routes. At the same time this leads to savings in computational time and indicates that traffic modelling might not require as many resources. These simulations were executed on Karolina HPC cluster.

As shown in Figure 5, the improvement in traffic flow with 45-50% of vehicles receiving live updates suggests a threshold beyond which additional updates yield diminishing returns.

		0% alternative change limit		5% alternative change limit		10% alternative change limit	
Cars	Alt.	simulation time	execution time	simulation time	execution time	simulation time	execution time
10K	0.0	2:35:24	0:49:56	2:35:24	0:49:56	2:35:24	0:49:56
10K	0.05	**2:33:39**	0:59:00	**2:35:28**	1:01:21	**2:35:56**	0:59:14
10K	0.1	2:31:38	1:03:18	2:34:30	1:05:53	2:27:57	1:03:44
10K	0.15	2:24:15	1:10:42	2:28:12	1:08:40	2:30:09	1:09:29
10K	0.2	2:22:10	1:19:10	2:19:31	1:14:15	2:27:04	1:15:00
10K	0.25	2:17:59	1:20:53	2:13:25	1:20:06	2:19:32	1:19:58
10K	0.3	2:16:19	1:27:05	2:15:56	1:31:34	2:12:01	1:28:44
10K	0.35	2:09:33	1:32:38	2:08:51	1:41:11	2:07:24	1:32:54
10K	0.4	1:59:44	1:47:25	2:03:32	1:39:02	2:01:45	1:41:38
10K	0.45	**1:56:45**	1:50:34	**1:58:15**	1:56:21	1:59:49	1:49:07
10K	0.5	2:02:30	2:08:29	2:03:28	1:53:58	**1:58:21**	1:55:52
10K	0.55	2:08:24	2:10:11	2:06:16	2:07:20	2:10:20	2:06:49
10K	0.6	2:07:19	2:15:11	2:08:35	2:12:20	2:06:03	2:12:27
10K	0.65	2:04:34	2:33:35	2:04:31	2:14:45	2:03:36	2:13:56
10K	0.7	2:11:02	2:27:59	2:02:27	2:18:00	2:04:55	2:36:20
10K	0.75	2:14:27	2:44:57	2:13:06	2:30:12	2:10:26	2:26:45
10K	0.8	2:22:48	2:55:11	2:08:46	2:37:19	2:15:01	2:31:24
10K	0.85	2:22:39	2:52:05	2:20:20	2:41:40	2:12:01	2:53:29
10K	0.9	2:14:48	2:47:14	2:13:04	2:58:49	2:15:22	2:45:09
10K	0.95	2:22:46	3:02:50	2:23:11	2:57:22	2:25:24	3:11:23
10K	1.0	2:27:34	3:28:34	2:24:53	3:11:06	2:21:32	2:55:17

Fig. 5: Comparison of Simulation and Execution Times Across 60 Simulations in Prague, with Varying Alternative Route Settings (0%, 5%, and 10% Faster Routes).

5.2. Enabling Distributed Computation

After enabling distribution of alternative route computation via evkit integration, in this scenario where we launch simulations with 60,000 vehicles in the area of Prague, it was possible to speed up the execution time of each simulation step by more than 7 times (as shown in Table 2). Due to the unavailability of Karolina (for maintenance) at the time of this experimental work, this batch of experiments was executed on Barbora - which also allowed us to validate that our simulation software runs seamlessly on a different hardware.

Number of Nodes	Workers per Node	Computation Time (seconds)	Steps Completed	Time per Step (seconds)	Time per Step Speed Up	Algorithm
1	1	3,600	16	225.00	0.07×	Dijkstra (Python)
1	1	3,600	230	15.65	Baseline	Plateau (C++)
1	114	1,736	512	3.39	4.62×	Plateau (C++)
2	114	1,272	512	2.48	6.31×	Plateau (C++)
4	114	1,170	512	2.28	6.86×	Plateau (C++)
8	114	1,118	512	2.18	7.18×	Plateau (C++)
16	114	1,097	512	2.14	7.31×	Plateau (C++)

Table 2: Distribution Improvements with Multi-CPU and Multi-node Settings: Sequential vs. Parallel Version.

The first row of the table shows the sequential version of the simulator, with Dijkstra's routing algorithm in Python. The second row shows a version with Plateau algorithm in C++ running with only one CPU (evkit worker) - the baseline of our comparison to avoid a comparison between the different algorithms. The differences between the two algorithms indicate that while 3,600 seconds are equally elapsed, many more simulation steps are completed: 230 (Plateau) instead of 16 (Dijkstra). This shows that an algorithm switch immediately offers performance gains as the time of step is reduced from 225 seconds to approximately 15 seconds.

When comparing only Plateau enabled versions in C++, from the second to the last rows, we observe that each simulation step is computed 7.31 times faster. Nevertheless, when considering a full utilisation of the computational nodes, we observe that using 16 nodes achieves a 1.58 times speed up against a fully utilised node - one of the reasons for this is that there are parts of the simulator that are not yet distributed and will be addressed as part of future work.

5.3. Scalability Assessment

In this part of our experimental work, we have executed a batch of simulations with 300,000 vehicles in the area of Boston. The simulations were launched on Karolina cluster with the optimised version of the simulator (with the C++ Plateau algorithm). The computational load of the alternative route computation on one node with 114 workers is approximately 73%. It was possible to determine these numbers - shown in Figure 6 - by profiling the executions launched to assess the scalability capabilities as presented in Figure 7.

Figure 6 shows that running a 1-node simulation with 300,000 vehicles in the area map of Boston (i.e., graph with 51,181 nodes and 125,888 edges) requires approximately 73% of execution time of alternative routes computation, 18% for route selection and 5% for advancing vehicles. When reaching 12 nodes, Figure 6 also shows that the other functions like "Select Routes" already require more execution time than Alternative routes. This means that there is no significant gain from scaling further than 12 nodes in this scenario.

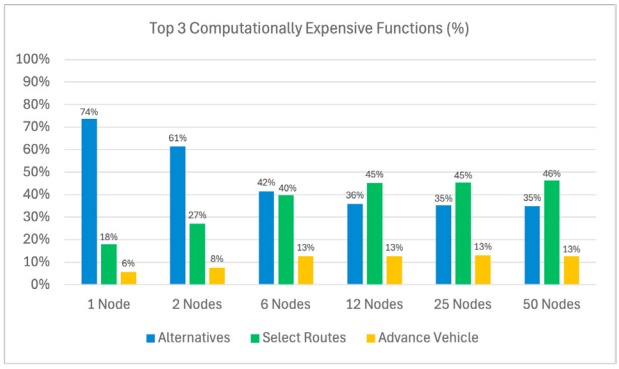


Fig. 6: Top 3 Computationally Expensive Functions.

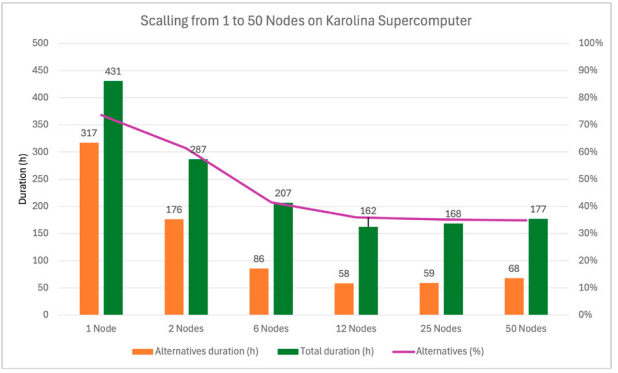


Fig. 7: 300,000 Vehicles Simulation from 1 to 50 Computational Nodes.

Figure 7 complements the previously mentioned findings as it shows an improvement up 12 nodes, stabilising at approximately 35% of demand for computation of alternative routes with 25 and 50 nodes - meaning that further optimisation (i.e., enabling distribution) is required on functions such as "Select Route" and "Advance Vehicle" (as seen in Figure 6).

6. Conclusion

One of the main findings of this work indicates that an informed management of live traffic data and selective alternative route computation may have a significant impact on the overall driving time and traffic flow within a city. While the percentage of vehicles updating their routes was a simulated parameter in our study, its broader applicability lies in its potential as a strategic tool for real-world implementations. Service providers of navigation and routing systems could deliberately enforce selective updates to reduce redundancy and prioritize critical traffic areas, thereby enhancing traffic flow while optimizing computational resources. This approach does not merely reduce computation but

translates into tangible operational cost reductions for service providers and urban planners, making large-scale traffic management more feasible. Based on the scalability assessment and profiling of the executions on both Barbora and Karolina, our future work will focus on further optimizing and distributing other computationally intensive functions such as route selection and advanced vehicle management strategies.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957269. This work was also supported by the Ministry of Education, Youth and Sports of the Czech Republic (ID: MC2102) and through e-INFRA CZ (ID:90140).

References

[1] IT4Innovations, Ruth - traffic simulator, https://github.com/It4innovations/ruth (2024).
[2] C. Pilato, S. Banik, J. Beránek, F. Brocheton, J. Castrillon, R. Cevasco, R. Cmar, S. Curzel, F. Ferrandi, K. F. A. Friebel, A. Galizia, M. Grasso, P. Silva, J. Martinovic, G. Palermo, M. Paolino, A. Parodi, A. Parodi, F. Pintus, R. Polig, D. Poulet, F. Regazzoni, B. Ringlein, R. Rocco, K. Slaninova, T. Slooff, S. Soldavini, F. Suchert, M. Tibaldi, B. Weiss, C. Hagleitner, A system development kit for big data applications on fpga-based clusters: The everest approach, in: 2024 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2024, pp. 1–6.
[3] E. Project, Everest sdk, https://github.com/everest-h2020/everest-sdk (2024).
[4] S. Maerivoet, B. De Moor, Cellular automata models of road traffic, Physics Reports 419 (1) (2005) 1–64. doi:https://doi.org/10.1016/j.physrep.2005.08.005.
URL https://www.sciencedirect.com/science/article/pii/S0370157305003315
[5] J. Nguyen, S. T. Powers, N. Urquhart, T. Farrenkopf, M. Guckert, An overview of agent-based traffic simulators, Transportation Research Interdisciplinary Perspectives 12 (2021) 100486. doi:https://doi.org/10.1016/j.trip.2021.100486.
URL https://www.sciencedirect.com/science/article/pii/S2590198221001913
[6] R. Mahnke, J. Kaupužs, I. Lubashevsky, Probabilistic description of traffic flow, Physics Reports 408 (1) (2005) 1–130. doi:https://doi.org/10.1016/j.physrep.2004.12.001.
URL https://www.sciencedirect.com/science/article/pii/S0370157304005095
[7] P. Alvarez Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, E. Wießner, Microscopic traffic simulation using sumo, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), IEEE, 2018, pp. 2575–2582.
URL https://elib.dlr.de/127994/
[8] M. Fellendorf, Vissim: A microscopic simulation tool to evaluate actuated signal control including bus priority, in: 64th Institute of transportation engineers annual meeting, Vol. 32, Springer Berlin/Heidelberg, Germany, 1994, pp. 1–9.
[9] A. Halati, H. Lieu, S. Walker, Corsim-corridor traffic simulation model, in: Traffic Congestion and Traffic Safety in the 21st Century: Challenges, Innovations, and OpportunitiesUrban Transportation Division, ASCE; Highway Division, ASCE; Federal Highway Administration, USDOT; and National Highway Traffic Safety Administration, USDOT., 1997.
[10] G. Boeing, Modeling and analyzing urban networks and amenities with osmnx (2024).
[11] T. Hughes, A. Allan, A. Khorev, openstreetmap-website, https://github.com/openstreetmap/openstreetmap-website (2024).
[12] E. W. Dijkstra, A Note on Two Problems in Connexion with Graphs, 1st Edition, Association for Computing Machinery, New York, NY, USA, 2022, p. 287–290.
URL https://doi.org/10.1145/3544585.3544600
[13] R. Bellman, On a routing problem, Quarterly of applied mathematics 16 (1) (1958) 87–90.
[14] J. Faltýnek, M. Golasowski, K. Slaninová, J. Martinovič, Shortest n-paths Algorithm for Traffic Optimization, Springer Singapore, Singapore, 2022, pp. 169–180. doi:10.1007/978-981-16-4287-6_12.
URL https://doi.org/10.1007/978-981-16-4287-6_12
[15] E. Vitali, D. Gadioli, G. Palermo, M. Golasowski, J. Bispo, P. Pinto, J. MartinoviČ, K. SlaninovÁ, J. M. P. Cardoso, C. Silvano, An efficient monte carlo-based probabilistic time-dependent routing calculation targeting a server-side car navigation system, IEEE Transactions on Emerging Topics in Computing 9 (2) (2021) 1006–1019. doi:10.1109/TETC.2019.2919801.
[16] P. Silva, P. Smolková, S. Michailidu, J. Beránek, R. Macháček, K. Slaninová, J. Martinovič, Ruth traffic simulator - input for prague, boston (Aug. 2024). doi:10.5281/zenodo.13285293.
URL https://doi.org/10.5281/zenodo.13285293

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 93–102

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

Towards full AI model lifecycle management on EuroHPC systems, experiences with AIFS for DestinE

Thomas Geenen[a*], Even Marius Nordhagen[b], Victor Sanchez[c], Cathal O'Brien[a], Simon Lang[a], Mihai Alexe[a], Ana Prieto Nemesio[a], Gert Mertes[a], Rakesh Prithiviraj[a], Jesper Dramsch[a], Baudouin Raoult[a], Florian Pinault[a], Helen Theissen[a], Sara Hahner[a], Mario Santa Cruz[a], Matthew Chantry[a], Nils Wedi[a]

[a] *European Centre for Medium-Range Weather Forecasts (ECMWF), Shinfield Park, Reading RG2 9AX, United Kingdom*
[b] *Norwegian Meteorological Institute, Oslo, Norway*
[c] *National Centre for Meteorological Research (CNRM) - UMR 3589, Toulouse, France*

Abstract

On October 13 2023 ECMWF released the first alpha version of its artificial intelligence forecasting system, AIFS, ECMWFs data-driven forecasts model. This first release came just a few months after ECMWF started the development of this new model that highlights the increased efforts in the field of machine learning (ML) that ECMWF has been building over the last few years. This paper describes the use of AIFS on EuroHPC systems in the context of DestinE. The main focus is on performance benchmarks on the different EuroHPC systems available to DestinE but also very much on the deployment and use of the tools to support the model lifecycle management. EuroHPC systems have already proven to be of great value for DestinE and in this paper, we describe how we leverage these systems for artificial intelligence (AI) and ML models in DestinE. We are closely working with EuroHPC and EuroHPC hosting sites through co-design and the optimization of existing solutions to optimize the usage of these systems in every step of the lifecycle management for AI and ML models. The performance benchmarks of our models on several EuroHPC systems showed that the speedup is close to linear up to several thousand GPUs, but that for each EuroHPC system a different optimization strategy must be used to achieve that. For model lifecycle management we found that we can use our in-house developed, domain specific, framework on EuroHPC systems and highlight some specific modifications and future improvements for EuroHPC systems. We also provide implementation details and share our experiences on how to retrieve and collect provenance data and information from models running on EuroHPC systems using (external to the EuroHPC system deployed) cloud native frameworks. Although we describe solutions in this paper that are designed to support our specific requirements and context, we believe that proposed solutions, developments and implementation details can also bring value beyond the broader NWP community.

Keywords: Type your keywords here, separated by semicolons;

1. Introduction

10.1016/j.procs.2025.02.264

DestinE is a flagship initiative of the European Commission to develop a highly accurate digital representation of the Earth (digital twins of the Earth) to model, monitor and simulate natural phenomena, hazards, and the related human activities. Through a strategic allocation from EuroHPC JU, DestinE has been granted access to its pre-exascale systems, which now form the computational backbone of the DestinE system. DestinE is implemented by ECMWF, ESA and EUMETSAT together with many partners across Europe. ECMWF is delivering the first two digital twins on Weather-induced Extremes, and Climate Change Adaptation together with over 90 partners throughout Europe. ECMWF is also delivering the Digital Twin Engine, i.e. a modular software infrastructure that allows to run the digital twins and handle and interact with their vast volumes of data.

AIFS is part of the ECMWF ML project that started in the summer of 2023 to increase the use of machine learning in Earth system modelling [1]. AIFS will be used in DestinE for complementing the physics-based models for enhancing uncertainty quantification for the global and regional component of the weather-induced extremes digital twin. After the ML model has been trained, it can be used to predict the state of the atmosphere from a given reference state or initial state, running the inference model. The output of the inference model is used for model evaluation and weather forecasting.

One focus is Limited-area modelling (LAM) with a goal to provide forecasting on weather-induced extreme events [2]. LAM or regional models are initialized, and their boundary conditions provided by physics-based, high resolution, global weather data assimilation and forecasts, with the latter provided by the global Extremes digital twin of DestinE. An alternative approach explores an AIFS high-resolution regional domain nested inside the global AIFS model. The model

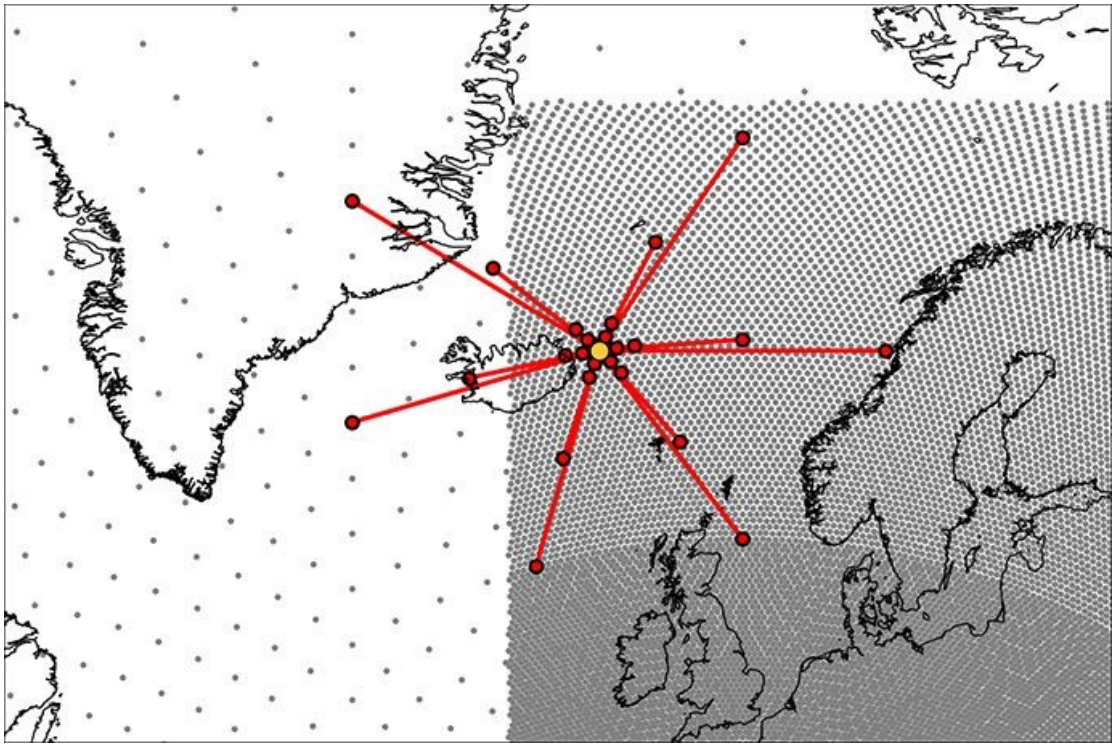


Fig. 1 Regional model with a global stretched-grid approach. The flexibility of GNNs used in AIFS allows for various grid resolution over different regions. Similarly, the processor mesh might have various refinements over different regions. Here, we demonstrate this by a nested regional model with a higher mesh resolution over the Nordics than globally. The message passing strategy allows for both short-range and long-range communication between mesh nodes, illustrated by the red line.

architecture of AIFS is based on graph neural networks (GNNs) [3,4]. This model architecture facilitates an approach where the mesh is locally refined and is agnostic to mesh characteristics and topologies. An advantage of this approach is that by combining a global domain and regional high-resolution domains, one does not need to deal with the boundary problems often associated with regional models. This approach allows in principle for kilometer-scale resolution in regions of interest without the expense of a global kilometer-scale model. *Fig. 1* illustrates this nested

approach and the associated mesh refinement over the area of interest. Fig. 2 shows initial results illustrating that the transition from the global mesh resolution to the regional higher resolution mesh is smooth and does not provoke model artefacts.

To develop, train, deploy and operate AI/ML models, development teams can benefit from an environment where tools and frameworks are available that support them in each of these tasks. In addition, data and model provenance and lineage tools are required to make sure that the full lifecycle of the data and the models is captured. ECMWF is developing a domain specific framework for this purpose but also leverages tools and frameworks from the machine learning community to support these tasks. The framework ECMWF is developing with the meteorological and climate communities is called AnemoI [5]. It is composed of a variety of modular components, each addressing a specific aspect of the ML workflow. This modular architecture allows for better concept abstraction, ensuring that AnemoI can efficiently handle all stages of machine learning. Anemoi aims to build on top of existing tools were deemed suitable, e.g. PyTorch [6]. AnemoI supports data preparation and model training by providing tools and model components that can be integrated to develop data processing pipeline and ML models. In addition, it provides a model and data registry to use as a base for new models or reproduce results. It also provides tools to operate inference and can interface to other verification software for operational purposes.

The model lifecycle of AIFS and AIFS regional models are tracked with MLflow [7]. This framework allows for tracking of model runs and visualizing and exploring model runs. The model execution records metadata and model artefacts and sends them to a MLflow server for visual inspection and model exploration. Models are currently manually scheduled for most AIFS and AIFS regional experiments. MLflow supports such a setup since it allows for offline model runs and does not expect, or require, models to be run from the framework itself but will listen for incoming requests to store model tracking data from models that run on EuroHPC systems. On some EuroHPC systems this can be a challenge since the compute nodes where the model is run do not typically allow for connections to external services. Therefore, we developed procedures to collect data also from these systems. For these EuroHPC systems data is stored locally and periodically collected and sent in batched to the MLflow server.

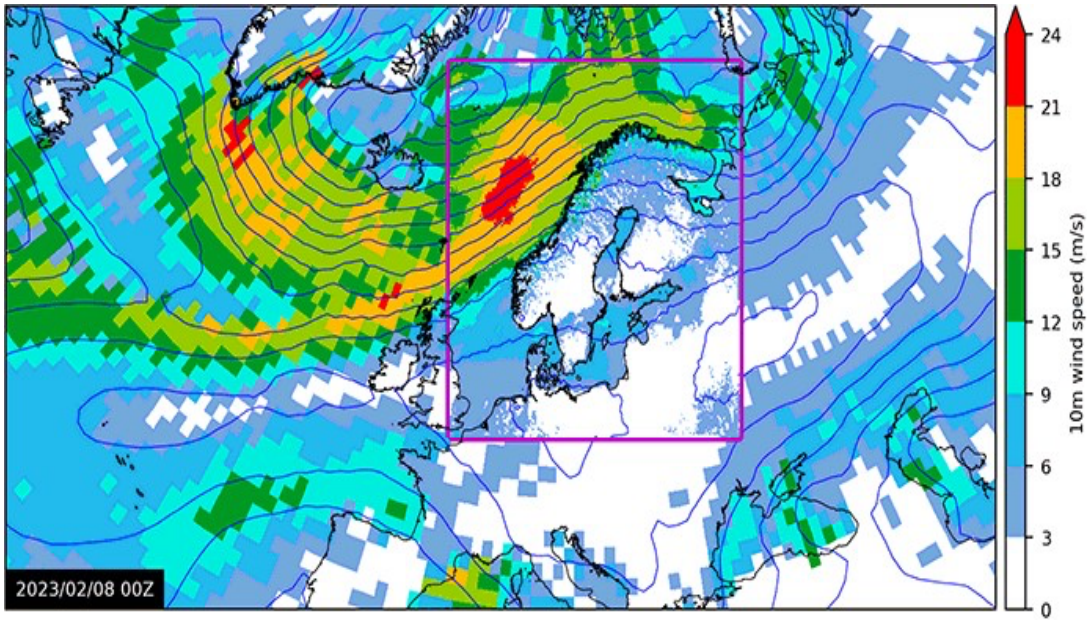


Fig. 2 10m wind speed over Northern Europe predicted by the AIFS regional model with higher spatial resolution over the Nordics. The spatial resolution increases when entering the local region of interest.

DestinE pushes the computational boundaries of current EuroHPC systems and requires more computational resources than can be made available on a single EuroHPC system to a single initiative. Therefore, DestinE has been granted computation time on all the available pre-exascale EuroHPC systems, and this strategic partnership with EuroHPC is crucial for delivering DestinE digital twins [8]. This however requires the porting of the model training and inference models to each of them. On some of the EuroHPC systems the ML models are run in containers,

leveraging system provided, optimized, implementation of the PyTorch [6] framework that is used by AIFS for its implementation. Although PyTorch supports several optimized implementations for different compute architectures, the user must make code modifications to exploit these and relies on the system providing frameworks and system libraries for efficient execution. This prevents model performance portability between EuroHPC systems and requires a bespoke model deployment and execution solution for each of them. In addition, the supported container runtimes also differ between different EuroHPC systems requiring different model run commands to be implemented by the developers. This has resulted in most AIFS developers still using Conda [9] to install PyTorch and build a private version of AIFS for model runs.

The training of AIFS and AIFS regional models have shown to scale to significant numbers (thousands) of GPUs on EuroHPC systems. (LUMI and Leonardo). A hybrid model and data parallelization strategy is used to achieve this scaling [3]. Model parallelization is used for intra node optimization where each node contains multiple GPUs. Model parallelization does not only speed up the training, but it also reduces the GPU memory usage and allows for larger models. Data parallelization is exploited for inter node optimization to distribute the training batches over multiple nodes and speeding up the training. In addition, the data is prepared for the model to minimize data read operations during model training, a potentially costly operation on parallel or distributed filesystems currently used in EuroHPC systems.

2. Model Lifecycle management

2.1. Data preparation

AIFS is trained on meteorological data from the ERA5 reanalysis dataset [10] and runs from initial conditions from the operational ECMWF system. These datasets are stored in formats that adhere to World Meteorological Organization (WMO) standards (GRIB [11]) but are generally not optimized for machine learning. To improve data handling functionality, ECMWF has developed a functionality in AnemoI that allows for the transformation of these datasets to Zarr [12] (a cloud native file format that allows efficient handling of large multidimensional arrays and is widely adopted in the machine learning community). In addition, AnemoI provides a thin wrapper around these Zarr stores for additional improvements for efficient data access where IO operations are minimized. AnemoI also provides a rich collection of datasets that can be transferred from ECMWF to EuroHPC systems for training. For the AIFS regional models, the ERA5 reanalysis dataset is used for the global domain and regional reanalysis datasets are used for regions of interest. More specifically, the MetCoOp ensemble prediction system (MEPS) [13] reanalysis dataset has been used for the regional weather prediction illustrated in Fig. 2.

When users have the requirement to upload large datasets to EuroHPC systems we find that this can be very time consuming. The size of training datasets has grown over time and is expected to continue to grow. The transfer of a typical training dataset for today's regional model is on the order of few tens of terabytes (TB) and can take multiple weeks. There is a need for tooling to speed up this process, understanding that AI/ML training datasets can typically consist of millions of small files, a transfer pattern not easily supported by available transfer tools for HPC, like rsync or scp. We understand that EuroHPC is looking into solutions that would alleviate this problem in the future.

2.2. Model development

For AIFS, ECMWF provides a set of models and a collection of building blocks that facilitates the rapid development of new models or the improvement of existing models. Typically, developers will download these models and building blocks to their laptop for initial ideation small-scale development and testing. Code is kept under revision control and models can be stored in the AnemoI model-registry. During development, typically, the code is modified, run, and tested in a Python virtual environment. After the small-scale testing has been concluded and the results are promising, the developer builds a docker container locally and uploads that to the EuroHPC system for deployment and model training. Alternatively, when supported by the EuroHPC system, a container image is built locally on the HPC. On the LUMI EuroHPC system a custom tool can be used, cotainr [14]. This tool allows building of some Python-based singularity containers without the need for root, sudo privileges or fakeroot. Although this is an elegant solution for LUMI, this tool is not available on other EuroHPC systems and relies on preexisting and optimized container images for the platform architecture and toolchains. On Meluxina Apptainer [15] is used instead of Singularity and for this particular tool Luxprovide has enabled support on the compute nodes to build Apptainer

container images without sudo rights for certain python applications. For Leonardo ECMWF, and its DestinE project partners, have not yet found a solution to create container images easily and users run and train models from Python Conda environments also after the model has been deployed. We expect that in the future we will be able to deploy a singularity-based solution on Leonardo also.

2.3. Model training

After an ML model has been developed it needs to be optimized to run at scale on the EuroHPC system. Potentially this requires additional development work but normally the PyTorch framework can exploit hardware optimizations for the EuroHPC architectures available. The usage of the correct container image or Conda packages is still the responsibility of the developer, and a good understanding of the underlying system architecture is required to make the right decisions. In addition, on some platforms additional runtime optimizations like CPU pinning and CPU-GPU bindings must be specified, requiring from the developers a good understanding of the underlying system architecture characteristics. The EuroHPC hosting site support teams can help with making these changes.

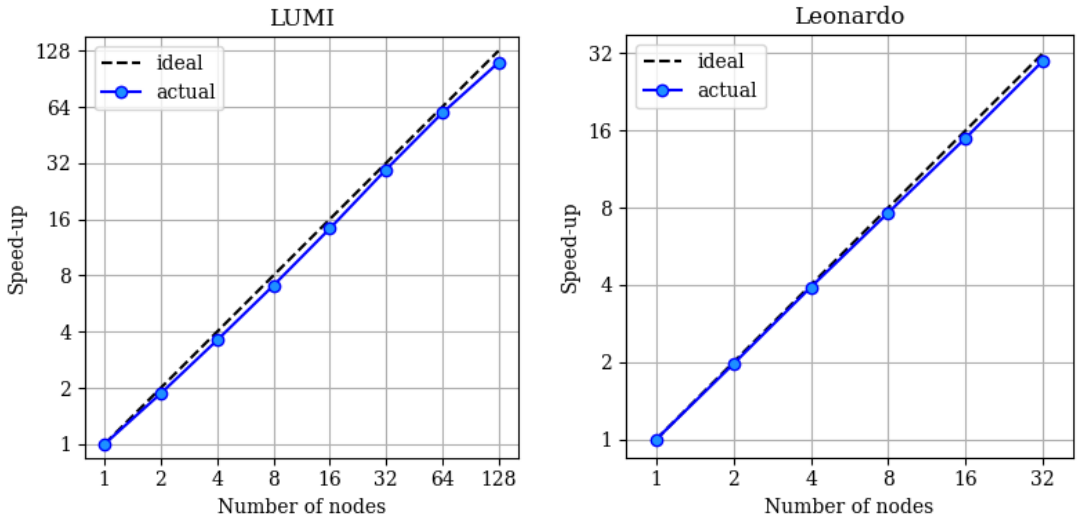


Fig. 3 Speedup curve for a AIFS training model run for 3 Epochs on Lumi and Leonard.

On LUMI, base containers come with hardware-specific packages, including GPU and libfabric drivers, and PyTorch. The usual practice is to extend these containers to meet individual requirements. Ideally, the plugins and libraries necessary for fast communications should be automatically enabled on HPC systems. The LUMI documentation offers recommendations for various CPU bindings, based on the parallelization method used. We discovered that hybrid MPI+OpenMP bindings provided the highest AIFS training speed, approximately 30%-40% faster than when CPU bindings were not explicitly specified.

To ensure fast training across multiple nodes on LUMI-G and Leonardo booster partition, we have performed strong scaling benchmark experiments focusing on a typical AIFS model with a global spatial resolution of 0.25°. The model is split across 4 GPUs. The training batches are split across nodes, resulting in reduced inter-node communication. The scaling results are shown in *Fig. 3*.

On Leonardo we ran a weak scaling experiment with a similar AIFS model **Error! Reference source not found.**. AIFS has demonstrated quasi-linear weak scaling up to 2048 GPUs on Leonardo. Data parallelism was predominantly

used to achieve this scaling, with each node getting a copy of the model. The dataset was also replicated to each node, resulting in a constant amount of work per node.

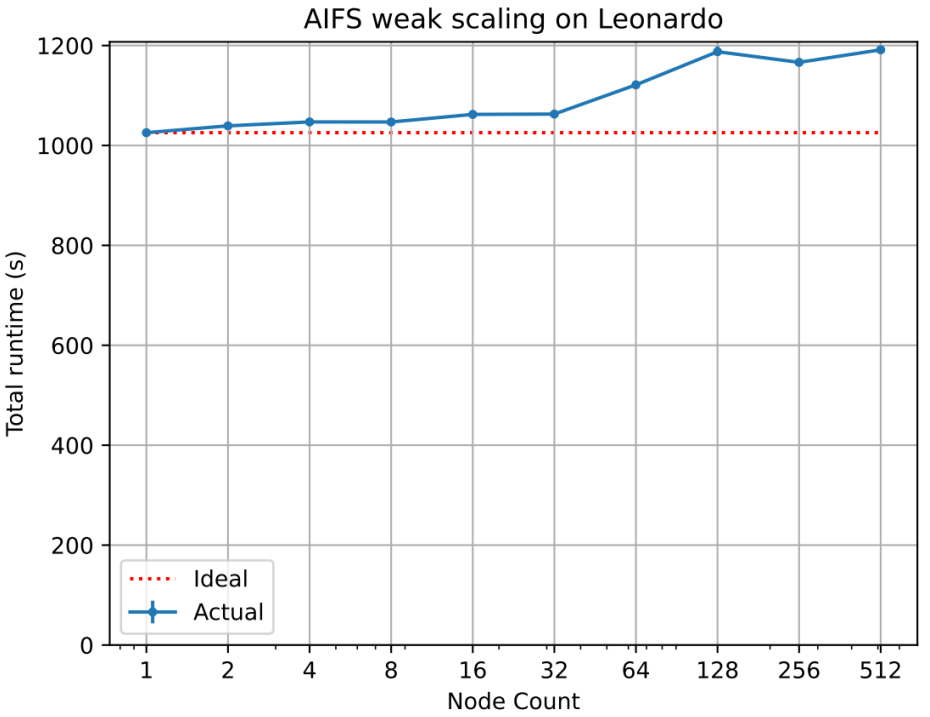


Fig. 4 Weak scaling results for 200 AIFS training steps on Leonardo (solid) line and optimal or ideal scaling (dotted line). The model scales close to optimal up to 128 GPUs. For larger number of nodes, we see a degradation in scalability with a loss of

On Meluxina we ran smaller model training experiments, using a single GPU node. This illustrates that also scalability on a single node for a pure data parallel approach scales optimal, *Fig. 5*.

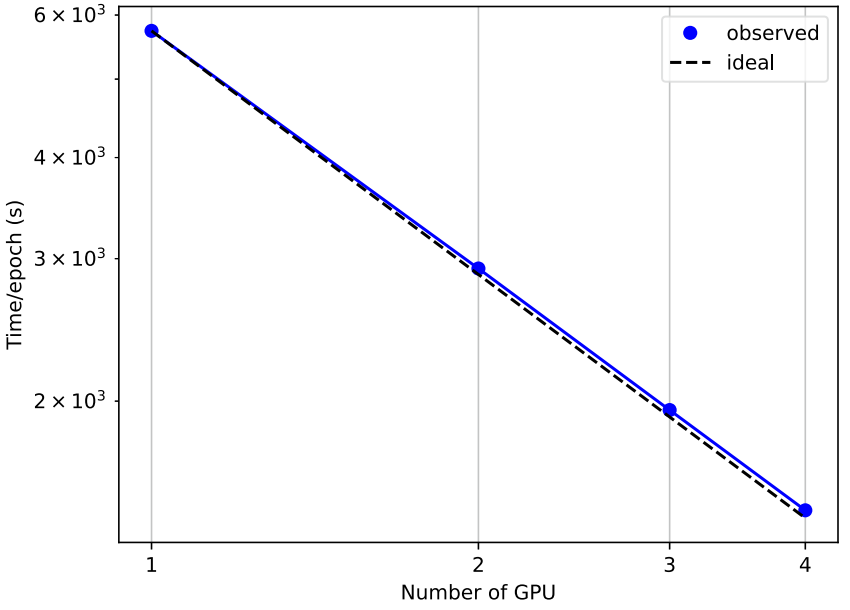


Fig. 5 Strong scaling of AIFS on a single GPU node from 1 to 4 GPUs on Meluxina

2.4. Model tracking and serving

In the training stage, when testing various model configurations (including architectures, datasets, and internal parameters—is essential), ECMWF uses MLflow [5] for model tracking during model training. MLflow has been selected for its wide adoption in the AI/ML community and for its rich set of features and open-source license model. During model execution, log files and model metrics such as the training loss, validation loss, learning rate and number of steps used in multi-step training is send to the MLflow server, as well as system metrics like CPU and GPU utilization, disk, RAM and GPU memory usage and network transition statistics. A central MLflow server is deployed in the ECMWF datacenter to collect the data from model runs on EuroHPC systems and ECMWF own HPC system. Having model tracking data available from all systems involved in DestinE allows developers to gain insight in model performance efficiently and effectively from a single location and prevent valuable information to remain in development silos with a single person or team on a specific system. Access from compute nodes, where the training model is run, to external servers, makes live model tracking using MLflow possible. To track GPU statistics on LUMI, we have extended the MLflow library to collect GPU statistics on AMD GPUs as well through the Python bindings for ROCm SMI.

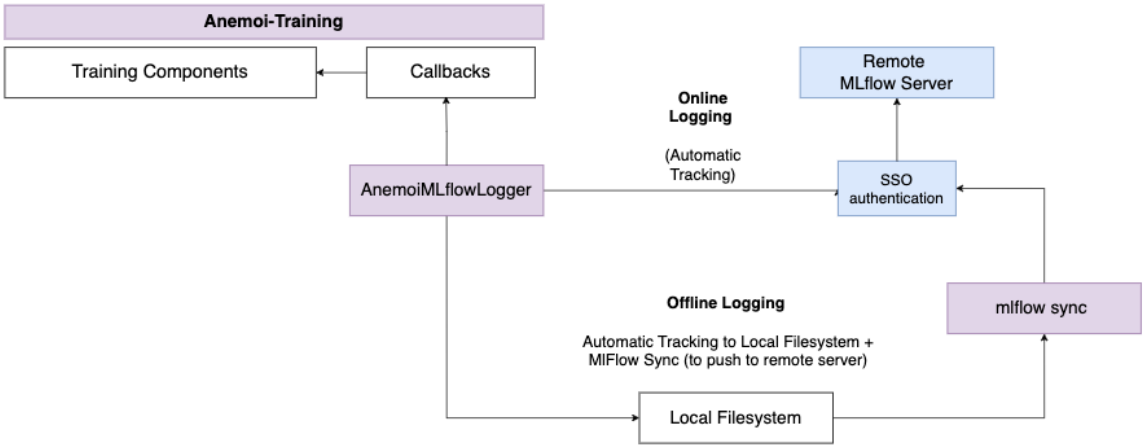


Fig. 6 High-level Architecture of Anemoi-training illustrating the connection between the AnemoiMLflowLogger and the Remote MLflow Server

Within the Anemoi ecosystem, Anemoi-Training handles the development and training of machine learning models for weather forecasting, using PyTorch Lightning [16] as the main deep learning framework. PyTorch Lightning simplifies training workflows by automating tasks like device management and model checkpointing, making it ideal for large-scale projects like weather forecasting. PyTorch Lightning uses callbacks to perform specific tasks during training, such as saving checkpoints or logging metrics.

As part of Anemoi-Training, we have developed a custom *AnemoiMlflowLogger*, extending the PyTorch Lightning MLflow logger by adding an authentication layer that meets ECMWF's security requirements. This logging mechanism facilitates secure tracking, ensuring visibility, control, and reproducibility across model experiments. The *AnemoiMlflowLogger* supports both Online and Offline Logging modes. In Online Logging, metrics are automatically tracked and sent directly to the remote MLflow server in real-time, where they can be immediately reviewed. In Offline Logging, tracking occurs locally on the filesystem, with metrics and logs stored for later syncing. The "mlflow sync" feature, built on top of existing open-source functionality, ensure that locally logged data can eventually be pushed and synchronized to the remote MLflow server, enabling centralized tracking and analysis even when a direct connection isn't available. A high-level architecture diagram to illustrate this is shown in *Fig. 6*. This setup allows to run Anemoi-Training in different EuroHPC systems, where compute nodes do not allow for connection to external services by pushing tracking data and information periodically to the central server after logfiles have been produced locally. This however prevents developers to track models while they are executing or take advantage of other MLflow features during model execution. It also creates a risk that information remains on the system and is not shared,

potentially preventing valuable information from reaching the wider AIFS community. For EuroHPC systems where no connection to external services is allowed from login nodes, we have yet to design a solution.

MLflow's experiment tracking functionality is currently in use within Anemoi-Training, while model registry and some additional features are handled by Anemoi-Inference and Anemoi-Registry to support operational needs specific to weather forecasting.

After the model has been trained, it can be used to predict the state of the atmosphere from a given reference state or initial state, running the inference model. The Inference model serving implies that the model is deployed in a system and exposed to the developer or users via, for instance, an API that accepts input data to produce model output. The inference model execution is typically fast compared to training the model but is becoming more expensive when models are getting larger and more complex. This implies that inference model executions require significant compute resources, up to several GPUs.

3. Data handling and management

Data handling and management concerns the input data for training and the output data of the inference models. We already discussed the data preparation steps for the model training and here want to describe the data handling and management of the data that is produced by the model. Like for the physics-based models used in DestinEs DTs, output data of the ML model is stored in a domain specific object store (FDB, [17]). From there the data can be moved onto a DestinE provided cloud stack that is collocated with each EuroHPC system used for DestinE, a data bridge (referenced as data bridge in the DestinE documentation, which is part of the data lake implemented by EUMETSAT). FDB has a client-server architecture that allows to handle data both on the EuroHPC system as well as the data bridge and move data between these instances through client server API calls. FDB makes this data available to clients inside the EuroHPC systems and to Polytope [18] for serving data inside DestinE.

Data provenance and lineage are an important aspect in data management where data provenance is more concerned about the data history and how it reached the model training phase where data lineage is more focused on tracking the data while it flows through the full model lifecycle pipeline. For both processes, AnemoI provides tools that complement the provenance features of MLflow and provide additional information about Python version, EuroHPC module environment loaded and GIT repo and branch information.

The fact that model output data is stored in the same FDB instance where the physics based DestinE digital twins store their output, highlights the synergies that can be exploited by this in situ concept, having the physical models that produce the training data and input data for AIFS model training and inference close to these AI/ML models. This is a clear example of the AI factory concept where data producers, data consumers and data products and services are collocated.

4. Deploying and Operating AI models

Since in DestinE we are running ML models on several of the (pre-)exascale EuroHPC systems, there is a desire for a consistent way to deploy both the models used for model training and the models for inference. Ideally, we could converge on a consistent containerized deployment mechanism where a single container and container format can be deployed. During runtime the optimized execution path for the underlying architecture and toolchains would be selected to ensure efficient model execution and optimal use of the shared computational resources on the EuroHPC systems. As described in the previous sections, a close collaboration between EuroHPC centers and AI/ML developers towards such a consistent setup, would be highly beneficial.

To be able to use ML models in a more operational setup in the future, one needs not only deep insight in the model performance and observability aspects related to model performance but also observability on the system itself and the sanity of the training models and the availability of inference services deployed. To be able to observe the EuroHPC system, access to system monitoring and accounting information is essential. Observability platforms would greatly benefit from API access to this information from an external location where this platform is hosted. In addition, running applications and services would also benefit from the capability to send observability data to this instance from the location they run. We can foresee additional services that play a role in collecting this information and relaying this to external observability platforms but a consistent approach between EuroHPC sites would be beneficial to keep the development and code maintainability efforts reasonable and manageable.

5. Conclusion

EuroHPC systems have a great potential to run large scale AI/ML training models, and we showed that indeed for AIFS we can exploit these systems very efficiently and scale model runs to a few thousand GPUs in parallel. However, we also find that exploitation of EuroHPC systems for developing and running AI/ML applications could be further improved by co-designing several key capabilities. DestinE could be a very suitable framework for such efforts. Improving the EuroHPC setup to allow for automation in the execution of several steps in the model life cycle management would allow for an increase in development speed and efficiency, a decrease in the risk of (human) error and a reduction in double work and information loss. Increased consistency between EuroHPC systems in programming models, deployment models, (data) access models and security models, would allow for an additional level of increased efficiency. We expect that the introduction of AI-factories and AI-optimized supercomputers will further improve the environment for AI developments. As highlighted in this paper, DestinE provides a stimulus from a developer's perspective and a framework towards the usage of EuroHPC for AI applications in NWP domain.

Acknowledgements

We acknowledge the EuroHPC Joint Undertaking for awarding this project access to the EuroHPC supercomputer LUMI, hosted by CSC in Kajaani, Meluxina, hosted by LuxProvide and Leonardo hosted by Cineca through a EuroHPC JU Special Access call. We would like to thank the EuroHPC hosting sites for their commitment in getting the models up and running and our (contract) partners for the implementation of the digital twins. We are looking forward to continuing our fruitful collaboration in the coming phases of DestinE. We also want to acknowledge the AIFS team at ECMWF for fruitful discussions during this document's drafting. We also like to acknowledge Computing and storage resources on LUMI provided by Sigma2 – the National Infrastructure for High-Performance Computing and Data Storage in Norway.

References

[1] ECMWF unveils alpha version of new ML model | ECMWF n.d. https://www.ecmwf.int/en/about/media-centre/aifs-blog/2023/ECMWF-unveils-alpha-version-of-new-ML-model (accessed July 12, 2024).

[2] Data-driven regional modelling | ECMWF n.d. https://www.ecmwf.int/en/about/media-centre/aifs-blog/2024/data-driven-regional-modelling (accessed July 12, 2024).

[3] Lang S, Alexe M, Chantry M, Dramsch J, Pinault F, Raoult B, et al. AIFS-ECMWF'S DATA-DRIVEN FORECASTING SYSTEM A PREPRINT 2024.

[4] Battaglia PW, Hamrick JB, Bapst V, Sanchez-Gonzalez A, Zambaldi V, Malinowski M, et al. Relational inductive biases, deep learning, and graph networks 2018.

[5] Zaharia M, Chen A, Davidson A, Ghodsi A, Hong SA, Konwinski A, et al. Accelerating the machine learning lifecycle with MLflow. PeopleEecsBerkeleyEduM Zaharia, A Chen, A Davidson, A Ghodsi, SA Hong, A Konwinski, S Murching, T NykodymIEEE Data Eng Bull, 2018•peopleEecsBerkeleyEdu 2018.

[6] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Adv Neural Inf Process Syst 2019;32.

[7] Welcome to anemoi-utils documentation! — Anemoi Utils 0.3.11.dev6+g4c39cff documentation n.d. https://anemoi-utils.readthedocs.io/en/latest/index.html (accessed July 14, 2024).

[8] Geenen T, Wedi N, Milinski S, Hadade I, Reuter B, Smart S, et al. Digital twins, the journey of an operational weather prediction system into the heart of Destination Earth. Procedia Comput Sci 2024;240:99–109. https://doi.org/10.1016/J.PROCS.2024.07.013.

[9] Conda Documentation — conda 24.5.1.dev67 documentation n.d. https://conda.io/projects/conda/en/latest/index.html (accessed July 14, 2024).

[10] Hersbach H, Bell B, Berrisford P, Hirahara S, Horányi A, Muñoz-Sabater J, et al. The ERA5 global reanalysis. Quarterly Journal of the Royal Meteorological Society 2020;146:1999–2049. https://doi.org/10.1002/QJ.3803.

[11] Abstract: GRIB2, The WMO standard for transmission of gridded data□Current status and NWS plans (2002 Annual) n.d. https://ams.confex.com/ams/annual2002/webprogram/Paper26175.html (accessed July 16, 2024).
[12] Zarr-Python — zarr 0.1.dev50 documentation n.d. https://zarr.readthedocs.io/en/stable/ (accessed July 16, 2024).
[13] Müller M, Homleid M, Ivarsson KI, Køltzow MAØ, Lindskog M, Midtbø KH, et al. AROME-MetCoOp: A nordic convective-scale operational weather prediction model. Weather Forecast 2017;32:609–27. https://doi.org/10.1175/WAF-D-16-0099.1.
[14] cotainr documentation — cotainr n.d. https://cotainr.readthedocs.io/en/stable/index.html (accessed July 16, 2024).
[15] Apptainer User Guide — Apptainer User Guide 1.3 documentation n.d. https://apptainer.org/docs/user/latest/ (accessed July 16, 2024).
[16] PT Lightning | Read the Docs n.d. https://readthedocs.org/projects/pytorch-lightning/ (accessed November 13, 2024).
[17] Smart SD, Quintino T, Raoult B. A Scalable Object Store for Meteorological and Climate Data 2017;8. https://doi.org/10.1145/3093172.3093238.
[18] Hawkes J, Manubens N, Danovaro E, Hanley J, Siemen S, Raoult B, et al. Polytope: Serving ECMWFs Big Weather Data 2020. https://doi.org/10.5194/EGUSPHERE-EGU2020-15048.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 103–111

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

HPC-Driven oceanographic predictions with Graph Neural Networks (GNNs) and Gated Recurrent Units (GRUs)

Paraskevi Vourlioti[a,*], Theano Mamouka[a], Maria Banti[a], Charalampos Paraskevas[a], Stylianos Kotsopoulos[a], Vasileios Alexandridis[b], Georgia Kalantzi[b]

[a]*Neuralio A.I. P.C., 12th km Thessalonikis—N. Moudanion, 57001 Thermi, Greece*
[b]*AlongRoute, 15 km N.R. Thessaloniki-Moudania, 57001 Thermi, Greece*

Abstract

In this work, we utilized the high-performance computing (HPC) capabilities of the Vienna Scientific Cluster (VSC5) to develop and validate advanced AI models for oceanographic forecasting, with a focus on predicting Significant Wave Height (SWH). Using the computational power of VSC5, particularly its NVIDIA A100 GPUs, allowed us to process and analyze over 500 million data points. The most promising model, a Graph Neural Networks - Gated Recurrent Unit GNN-GRU hybrid, was trained to generate six hourly forecasts for SWH and achieved a Mean Absolute Error (MAE) of 0.0071 and an R-squared (R^2) value of 0.98 against test data, demonstrating high accuracy and efficiency. The first validation results outside the training period are promising and efforts to refine the model are on-going. This work highlights the importance of HPC to the advancement of oceanographic forecasting, by enabling the processing of extremely large datasets and the creation of models that are up to the demanding standards of the marine sector. Subsequent efforts will center around enhancing these models, mitigating detected biases, and creating an operational deployment framework that can facilitate prompt decision-making in maritime operations.

Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day

Keywords: Significant Wave Height; HPC; GNN-GRU; CMEMS; CDS; EuroCC

1. Introduction

Almost 3% of greenhouse gas emissions worldwide are attributable to maritime transportation, which is expected to quadruple in less than 30 years. By 2050, the International Maritime Organization (IMO) [1] wants to cut these emissions by 70%. Due to inaccurate weather forecasts, weather routing —a method of reducing fuel use and

* Corresponding author.
E-mail address: vvourlioti@neuralio.ai

Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day
10.1016/j.procs.2025.02.265

emissions— has only succeeded in achieving 3–10% savings. The chaotic nature of physical processes in weather and climate systems, combined with the immense computational demands of traditional numerical forecast models, has made it challenging to improve the accuracy and resolution of these models. Despite significant research efforts over the past 50 years, these models still struggle to provide highly precise forecasts due to the inherent unpredictability of the systems they are trying to model and the limitations of current computing power. Existing weather ship routing systems provided by maritime software companies often struggle with inaccurate marine weather forecasts, hampering their efforts to reduce emissions in practice. By providing highly accurate forecasts of critical oceanographic parameters, the effort presented herein aims to help existing and new weather ship routing systems improve their performance and achieve the expected by the industry goals of reducing fuel consumption and greenhouse gas emissions by 15%, 20% or even more.

With the advancements in Artificial Intelligence (AI) methods and the services of Copernicus Marine Service [2] and Copernicus Data Store (CDS) [3], the time has come to develop new research and development processes, close to the private sector, which requires reliable products. The Copernicus Marine Environment Monitoring Service (CMEMS) is a crucial component of the EU Copernicus program, providing regular, systematic information on the physical, biogeochemical, and sea-ice conditions of the global ocean and European regional seas. CMEMS serves over 15,000 users and supports various applications, ranging from maritime safety to climate forecasting. The service's mission includes delivering short-term marine forecasts, analyzing past and present marine conditions, and providing data for climate change monitoring [4].

It is apparent that there are on-going efforts of the research community to develop models to predict oceanographic parameters and an effort on how to combine different architectures and the wealth of data. The target variables, vital for ship routing, include the Significant Wave Height (SWH). Ocean waves with a high SWH may submerge ships and destroy ocean or coastal infrastructure [5]. It endangers human life, agriculture output, and the viability of aquaculture goods. As a result, precise forecasting of SWH is critical since it can assist in avoiding social and economic losses. Furthermore, SWH prediction can provide various benefits. For example, improving ship routes based on SWH predictions might help avoid rough seas, decreasing sailing time and fuel costs. Furthermore, SWH prediction can help organize military and amphibious operations [5]. As climate change is expected to influence oceanographic conditions, leading to more frequent and severe storms, the ability to accurately predict SWH will become increasingly important, and further emphasizing the need for ongoing research and innovation in the field.

Due to the challenges in data collection and the limitations of computing power, Significant Wave Height (SWH) predictions were primarily based on empirical or numerical models ([5],[6],[7]). These methodologies do not have leaning ability that leads to a low prediction accuracy [5]. With the advent of machine learning, Bayesian Networks, XGBoost, Support Vector Machines (SVM) and Artificial Neural Networks (ANN) have been utilized for SWH prediction [8]. A common setback of these models was the negligence of the temporal dependencies of the data that made the models sensitive to noise that consequently lowered their reliability [5]. To address this issue, many researchers have investigated with promising results the Recurrent Neural Networks (RNN) and Long Short Term Memory (LSTM) methodologies ([9],[10]).

Following these efforts, a trend to use deep leaning approaches in SWH prediction emerged, taking advantage of the stronger feature extraction capacity compared to the machine learning methodologies. Convolutional Neural Networks (CNN) were found to outperform other deep-leaning approaches with works like [11] that extracted information from Synthetic Aperture Radar (SAR) images to predict SWH. Although CNNs are designed for visual data, promising results with CNNs resulted from a CNN-LSTM framework that used raw ocean images but with setbacks in night and foggy conditions [12]. Another promising and emerging application of CNNs is by correcting the predictions of numerical models [13].

Despite the ongoing efforts, two key challenges are identified that remain unresolved for accurate SWH prediction: effectively capturing the relationships between various input types while learning the complex non-linear mapping and temporal dependencies with SWH data and distinguishing between occasional extreme sea conditions and seasonal variations in SWH, which involves understanding both short-term and long-term patterns [4]. To this end in the works of [4], that utilized a Wavelet Graph Neural Network (WGNN) showed in their experimental results better performance than other models, including some numerical models and other deep-learning methods.

Understanding the computational limitations and the promising results of CNNs our first model was a CNN-based multi-stage model that predicts deltas instead of absolute values with noise in the input data to enhance the robustness. The validation of this model was done by comparing the model's forecasts with satellite measurements of Significant Wave Height (SWH), against the CMEMS operational forecasts compared to the same satellite measurements. A sample size of 1.914 collocated points within the Mediterranean was collected and the comparison demonstrated a better performance than the operational CMEMS forecasts, especially in the case of larger waves (>1 m) and a better fitting of the regression line at about 7% for AlongRoute's model.

After the promising results of the first model, the team has sought ways to improve the in-house modelling framework to evaluate the product (i.e., operational oceanographic forecasts) with maritime software companies. For the business model to work, it was crucial to develop more sophisticated models with:

1. Extensive training on larger amount of input data and crucial variables, and
2. Spatio-temporal characteristics that allow to extend the forecast horizon.

It became apparent that the first requirement could be satisfied with computational power. To this end, starts-up that need access to GPUs and computational resources are benefited by EuroCC (European Competence Center), that is the European network of 33 NCCs (National Competence Centers) national HPC competence centers with the aim of bridging the existing HPC skills gaps and infrastructure while promoting cooperation across Europe [14]. To gain access to the resources, a short project proposal was submitted outlining the amount of computational power needed and the scope of the project to EuroCC Austria. This granted the team with 100.000 CPU-hours or ~4.500 GPU-hours in the Vienna Scientific Cluster (VSC5) system. Utilizing supercomputer capabilities and offering technical support to its users, the Vienna Scientific Cluster (VSC) is an alliance of multiple Austrian universities. VSC-4 and VSC-5, the fastest supercomputers in Austria, are the flagship systems of the VSC family at the moment. Both systems are powering science and research at the leading academic institutions in Austria and fulfill the demand for high computing power in the areas such as physics, chemistry, meteorology, life sciences, and many others [15].

The second requirement was fulfilled by utilizing the most recent successful methodologies utilizing the computational power to process large data sets in HPC and perform training on extended data. To do so, we opted for a deep leaning methodology that is based in GNNs that have shown to be outperforming other deep learning methodologies while the temporal scale being resolved with Recurrent Gated Units, (GNU) a framework utilized in SWH predictions but in combination with a multivariate multi step time series forecasting [16]. This paper presents the usage of the VSC5, to train and validate models that can predict oceanographic parameters in extended temporal resolutions, the challenges faced in deploying the models, first validation results and future work both research-wise and business-wise.

2. Materials and Methods

To prepare a data driven model, our work was based on finding data that are significant and relate to the physics behind wave generation. This was achieved by utilizing reanalysis oceanographic data from the Copernicus Marine store (wave and physics related) as well as reanalysis atmospheric data from the Copernicus Data Store (CDS) [3]. In climate reanalysis, consistent time series of numerous climate variables are produced through the integration of past observations and models. Reanalysis are among the most frequently employed datasets in the geophysical sciences. They offer an extensive record of the observed climate as it has changed over the past few decades, with data collected on 3D grids at sub-daily intervals. This makes them suitable in our objective of training an Artificial Intelligence (AI) model that learns the relationship between the variables of interest and is capable to produce operationally forecasts of the SWH and wave direction given the previous state of the sea. The plethora of data availability and the access to VSC5 allowed as to investigate before the development of our model the most significant variables that did affect SWH. Feature importance algorithms (regression, Random Forest) combined with guidance from the experts, concluded into the set of the variables to train our model. The access to HPC facilitated the processing of all the variables included in the different data sets as well as the selection of the most important ones. This was extremely crucial for our model, considering that in an operational framework, we opt to retrain our model with up-to-date data that matter and enhancing the model accuracy.

2.1. Variable Selection

The data collection mechanisms to train the model were based on the offered mechanisms of CMEMS and CDS. Simple bash scripts were created to download the data for a full year from all three data sources and all their variables. The user must be registered in both CMEMS and CDS to obtain credentials. The data collection begun for the Mediterranean Sea (Fig.1) for two reasons. Firstly, when compared to available data sets, higher spatial resolution is available for the Mediterranean, compared to the global data sets. Secondly, the Mediterranean Sea

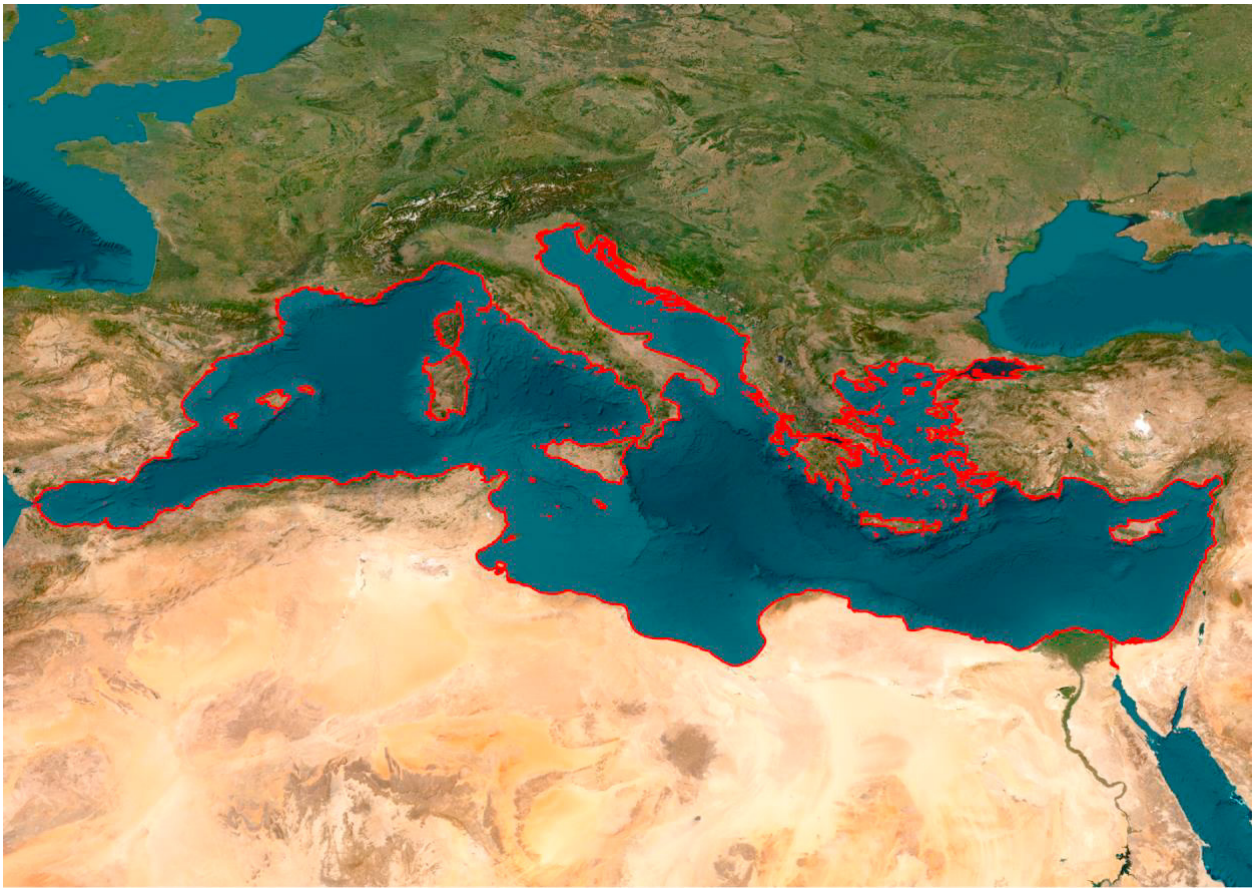


Fig. 1. Area of the Mediterranean Sea, used for training.

offers a closed boundary condition system that is easier to simulate compared to other seas. The year of 2014 was selected as was it was marked by several intense storms that generated high waves, especially in the western Mediterranean, complicating naval and commercial navigation.

In Table 1, the three data sources considered to train our model are presented. The datasets were first carefully selected in terms of providing operational forecasts counterparts, so as to later set-up the model in operational mode, but also to have equivalent information at global scale. Nonetheless, it must be noted that moving outside the Mediterranean Sea comes with losing the higher spatial resolution data sets. As can be seen in Table 1, the data from CMEMS have the same spatial grid and temporal resolution, meaning that the UERRA data had to be re-mapped to the MEDSEA data grid spatially. On the other hand, the MEDSEA data had to also match the 6-hourly temporal resolution of UERRA.

One year of data occupied almost 80G storage in VSC5 and the regression/RF model to finalize the selection of data (see Table 1) was sent in the CPU nodes as it was not computationally intensive to run. This is because

temporal and spatial aggregations were applied to the data to check temporal and spatial feature importance. These aggregations reduced the memory burden to handle this amount of data. Finally, 10 variables were selected to train our model.

Table 1. Reanalysis data, their characteristics, and the number of selected variables to train the model.

Reanalysis Data	Spatial Resolution	Temporal Resolution	Number of total variables	Number of selected variables
MEDSEA_MULTIYEAR_WAV_006_012 (CMEMS)	0.042° × 0.042°	Hourly	18	5
MEDSEA_MULTIYEAR_PHY_006_004 (CMEMS)	0.042° × 0.042°	Hourly	10	2
UERRA (CDS)	5.5km x 5.5km	6-hourly	19	3

2.2. Model Development

In this work, we aimed to introduce an extended temporal forecast framework that meets both our scientific interests and the business perspective that demands forecasts for more than a few hours ahead in the future. Graph Neural Networks (GNNs) combined with Gated Recurrent Units (GRUs) offer a powerful approach for significant wave height forecasting. GNNs excel in capturing spatial relationships within complex oceanographic data, while GRUs effectively model temporal dependencies. By integrating these two models, the GNN-GRU framework can simultaneously analyze spatial and temporal patterns in wave data, leading to more accurate and reliable wave height predictions. This hybrid approach is particularly well-suited for handling the dynamic and interconnected nature of ocean environments.

The dataset was split into 60% for training, 20% for validation, and 20% for testing. To fine-tune the model's hyperparameters, including the hidden dimension size, learning rate and batch size, we utilized Optuna [18], an optimization framework that allowed the detection of the best settings for these parameters after 20 trials. Following the training process, we evaluated the model's performance on the test set, by calculating metrics such as mean absolute error, root mean squared error, and R-squared to assess accuracy.

The model showed strong performance and efficiency as seen in Table 2. The model was trained on a large dataset of about 559 million data points, completing the process in approximately 13h50min, which reflects the capability of the hardware used. It must be noted that in order to process this amount of data, 512GB RAM was necessary, otherwise memory issues rose during run-time. Moreover, the usage of data frames for each data source (three different reanalysis datasets) in python and their merging into one data frame, led to large amounts of memory allocation that in turn crushed our runs. This was overcome not only by using the GPUs with the largest amount of RAM memory but also cleaning from memory all data frames the minute they were no longer necessary in the processing chain. The model achieved a Mean Absolute Error (MAE) of 0.0071, suggesting that its predictions are quite accurate with only a small average error. The R-squared (R^2) value of 0.98 shows that the model effectively captures most of the variability in the data. Additionally, the Root Mean Squared Error (RMSE) of 0.015 further confirms the accuracy of the model's predictions. Overall, the model demonstrates good predictive accuracy and efficiency in training. We trained three separate models, the first one being able to predict 6 hours ahead, the second one 12 hours ahead and the last one 24 hours ahead. The most stable and promising was the model built to forecast 6 hours ahead and our first validation results for this model are shown in the next section.

Table 2. Reanalysis data, their characteristics, and the number of selected variables to train the model.

VSC Partition	Total data points	Training Time	MAE	R^2	RMSE
zen3_0512_a100x2	559.238.400	13h50min	0.0071	0.98	0.015

3. Results

The validation strategy we followed is running our models and making predictions outside of the training period focusing on challenging dates with high waves. The models are initialized with forecasts and analysis datasets to mimic an operational set-up. Then we opted to understand if the seasonality is captured by our models by running our model for a month's period and comparing them to both reanalysis data and satellite SWH data from CMEMS. The reanalysis data, being gridded fields, can help us understand if the model is capturing the general circulation and the satellite data, as external validation, require also an extended period to collect a big enough size of overpassing satellite measurements.

A selection of days with SWH above 1.5 m was first located through filtering the data from the CMEMS with a focus on dates outside the training period. The first validation results are shown here for our model predictions, outside the training period and for 14/01/2019, 00 UTC and compared to reanalysis data. As can be seen in Fig. 2(a), our models can forecast the general wave circulation quite well when compared to the reanalysis data, with an average MAE of 0.34 m. The high waves seen near the coasts of France and cascading southwards are well captured in general by the GNN-GRU (Fig. 2(a)), although some local misalignments are seen west and east of the main event. Although this is a good result, we do have some overestimations by the model that we are seeking to understand. In Fig. 3(b), a scatter plot of model prediction versus reanalysis is given for the examined date, in which a lump is seen in low observed SWH that is overestimated by the model in the area of the Aegean Sea. Our first impression was that our model was not understanding well the coastlines and that it lets the SWH built up as if no land was there. This came apparent when creating a buffer zone around coastlines of around 4km (one grid box with regards to our spatial resolution), and the same plot was created as can be seen in Fig. 3(a) for all the Mediterranean. Although the bulk is gone, some points persist, and it is under investigation in our next steps. Here an extended validation that is undergoing will help us understand the model biases and its shortcomings.

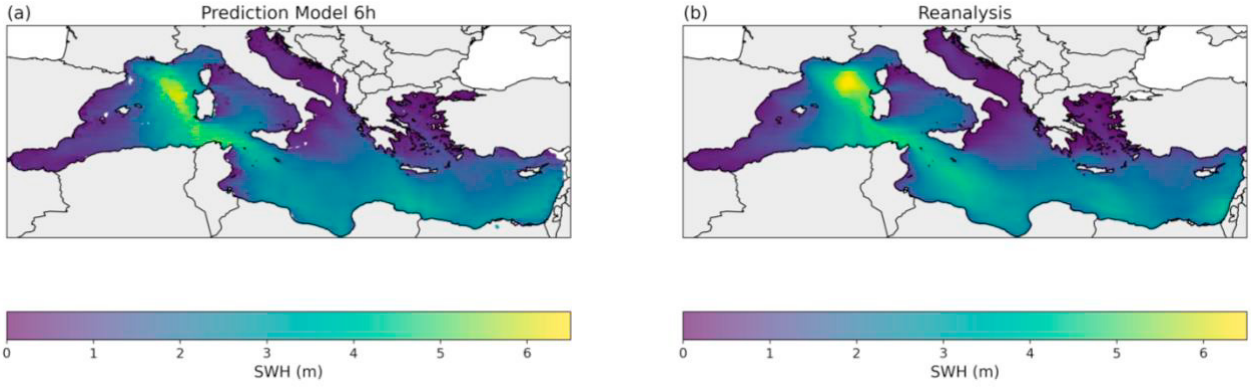


Fig. 2. (a) Model prediction on 14/01/2014; (b) Reanalysis data of SWH for the same time.

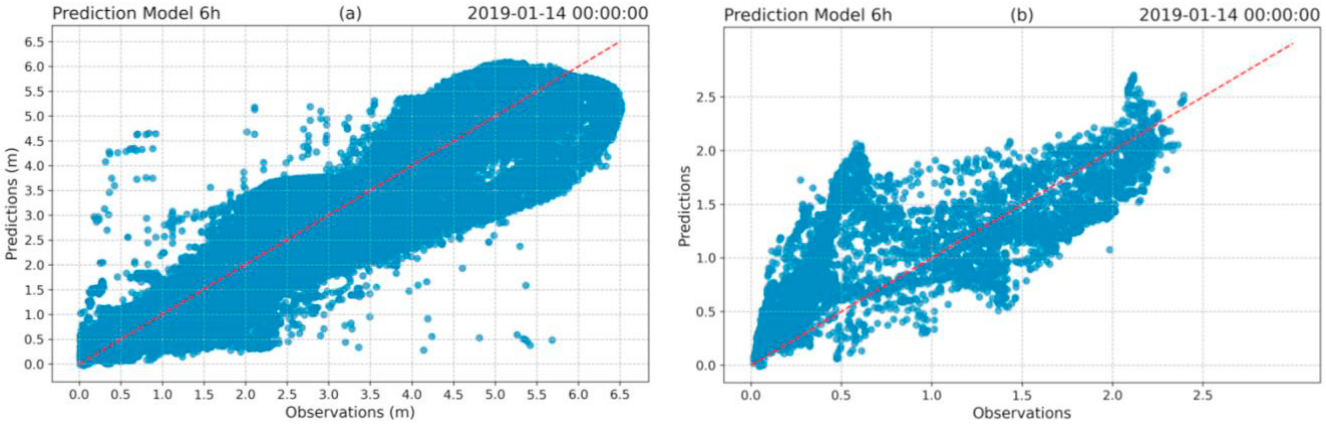


Fig. 3. (a) Scatter plot of predicted versus reanalysis data for Mediterranean (with buffer zone round coastlines); (b) Scatter plot of predicted versus reanalysis data for Aegean (without buffer zone around coastlines).

4. Discussion

Utilizing HCP resources granted though EuroCC, we were able to train and work on validating AI models for oceanographic parameters prediction and built a set up for future operational workflows that will benefit the shipping operations value chain. We have worked on the VSC5 cluster in which we developed and deployed a GNN-GRU that can handle spatiotemporal data and showed good performance metrics against the test dataset. Given the enormous amount of data processed, the most challenging part was not the usage of the HPC itself but rather handling the pre-processing step of the data in terms of memory. This was achieved by utilizing the GPU with the 512 GB RAM but also by cleaning up the cache and freeing the memory from redundant data frames. The training of each created model took around 14 hours on two GPUs, covering the entirety of the Mediterranean Sea and for one year of data (2014) accumulating to more than 500 million data points.

A significant advantage of having access to these HPC resources was the ability to perform hyperparameter tuning for the network with 20 trials, which consumed the majority of the total training time (about 90%). Without access to such computational power, we would have been forced to skip this critical step, likely resulting in the use of estimated configurations rather than the optimal ones. This would have negatively impacted the accuracy and reliability of our model's predictions. Thus, the HPC resources were instrumental in ensuring that our model was as well-tuned as possible, ultimately leading to better performance in predicting oceanographic parameters.

If we had more compute time this would allow us to increase the amount of data to train our model, given that the main bottleneck for this AI application is not the creation of the model itself and the training but importing a large amount of data. In our first model, with CNN we used 6 months of hourly data -reanalysis- and this gave sufficient low error against satellite observations, but it could not deliver beyond one hour forecast. Maritime software companies and their customers, i.e., shipping companies are in need of forecasts several hours ahead to be delivered to them. Subsequently, we introduced another architecture but also increased the training data sets from six months to one year. We have trained and validated with one year of data, AI models that can predict 6, 12 and 24 hours ahead. The new architecture with these data delivers small enough errors for the 6-hour ahead forecasts but declines in performance after 6 hours. This is where we believe that given more compute power, we could first test with more than one year of data and if this would not deliver, we would fall back into altering components of the AI architecture. The time in the day to produce a forecast is tied to the time the CMEMS releases the analysis data to initiate our model. Already the trained model can deliver in a matter of minutes a new forecast. We are currently looking into data assimilation methodologies to improve the initial states with local observations and a roll-out methodology to use the 6-hour model to cover one day ahead forecasts. In addition, this study has focused on

significant wave height (*Hs*), which is only one of the critical oceanographic parameters needed by maritime software companies to develop and perform accurate ship routing; others include wave direction, wave period, current speed and direction, and surface wind and direction. Therefore, more computing power would allow us to investigate the predictive performance of the full set of parameters required for the operational maturity of the intended product.

The validation of the model begun with checking the ability of the model to understand the general circulation in days with challenging sea conditions, with our model scoring good average MAE but showing large overestimations of small observed SWH as well as underestimations of observed SWH between 2.5m and 4.5m. This was attributed to the challenges of small islands and coastlines that our model seemed to miss and was overcome by creating a buffer zone around the coastlines. Some points did persist even after the buffer zone and are under investigation. Future work includes validating our model against satellite and reanalysis data for extended period to understand the strength and weakliness and calibrate the model towards optimal performance.

Although the model was trained on data from the Mediterranean, its performance in other regions, such as the Atlantic, would depend significantly on the quality and granularity of available data. Regions with less spatial and temporal resolution could pose challenges for generalizability, particularly where data is sparse. However, transfer learning could potentially mitigate some of these challenges. We intend to further explore this in future research, with a focus on generalizing the model through adaptive re-training on additional regional datasets. The main bottleneck for improving accuracy further is not only access to more computational resources but also the availability of high-quality, diverse datasets. More data would improve the model's ability to generalize across different oceanographic conditions. Another aspect is refining the model architecture to better capture complex spatiotemporal relationships, which could also lead to performance gains. Therefore, a combination of more compute, better data, and continual model improvements would collectively be needed to enhance predictive accuracy. It should be noted that our research has shown that despite the fact that the Copernicus Marine Data Service does not provide global data with similar resolutions, there are other sources of similar data for areas outside the Mediterranean, such as ECMWF. Given that this research effort has a business orientation, paid data is indeed an option when we enter the "paying customers" phase.

The access to HPC allows us to train the model with even more data and optimize the successful AI architectures in a continuous research and development loop. Once we secure the best model configuration, the next step is to provide these data in inference mode, in our premises, for the interested shipping companies.

Acknowledgements

This work got support from the Austrian National Competence Centre for High-Performance Computing, High-Performance Data Analytics and Artificial Intelligence (EuroCC Austria). It is legally represented by Advanced Computing Austria ACA GmbH and funded by the EuroCC 2 project that has received funding from the European High-Performance Computing Joint Undertaking (EuroHPC JU) and participating countries under grant agreement No. 101101903. The computational results presented have been achieved [in part] using the Vienna Scientific Cluster (VSC).

References

[1] International Maritime Organization (2023) Strategy on reduction of GHG emissions from ships. Report

[2] Von Schuckmann, K., Le Traon, P. Y., Smith, N., Pascual, A., Brasseur, P., Fennel, K. S., et al. (2018). Copernicus marine service ocean state report. J. Oper. Oceanograp. 11, 1–142. doi: 10.1080/1755876X.2018.1489208

[3] *CDS*. Climate Copernicus. Retrieved August 28, 2024, from https://cds.climate.copernicus.eu/cdsapp#!/home.

[4] Le Traon, P. Y., Reppucci, A., Alvarez Fanjul, E., Aouf, L., Behrens, A., Belmonte, M., ... & Zacharioudaki, A. (2019). From observation to information and users: The Copernicus Marine Service perspective. Frontiers in Marine Science, 6, 234.

[5] Chen, D., Liu, F., Zhang, Z., Lu, X., & Li, Z. (2021, July). Significant wave height prediction based on wavelet graph neural network. In 2021 IEEE 4th international conference on big data and artificial intelligence (BDAI) (pp. 80-85). IEEE.

[6] T. W. Group, "The wam model—a third generation ocean wave pre- diction model," Journal of Physical Oceanography, vol. 18, no.12, pp. 1775 – 1810, 1988.

[7] L. Mentaschi, G. Besio, F. Cassola, and A. Mazzino, "Performance evaluation of wavewatch iii in the mediterranean sea," Ocean Modelling, 06 2015.
[8] J. Mahjoobi and E. Adeli Mosabbeb, "Prediction of significant wave height using regressive support vector machines," Ocean Engineering, vol. 36, no. 5, pp. 339–347, 2009.
[9] S. Mandal and N. Prabaharan, "Ocean wave forecasting using recurrent neural networks," Ocean Engineering, vol. 33, no. 10, pp. 1401–1410, 2006.
[10] K. Osawa, H. Yamaguchi, M. Umair, M. A. Hashmani, and K. Horio, "Wave height and peak wave period prediction using recurrent neural networks," in 2020 International Conference on Computational Intelli- gence (ICCI), 2020, pp. 1–4.
[11] B. Quach, Y. Glaser, J. E. Stopa, A. A. Mouche, and P. Sadowski, "Deep learning for predicting significant wave height from synthetic aperture radar," IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 3, pp. 1859–1867, 2021.
[12] H. Choi, M. Park, G. Son, J. Jeong, J. Park, K. Mo, and P. Kang, "Real- time significant wave height estimation from raw oceanimages based on 2d and 3d deep neural networks," Ocean Engineering, vol. 201, p. 107129, 2020.
[13] J. Mooneyham, S. C. Crosby, N. Kumar, and B. Hutchinson, "Swrl net: A spectral, residual deep learning model for improving short-term wave forecasts," Weather and Forecasting, vol. 35, no. 6, pp. 2445 – 2460, 2020.
[14] Marchant, D. G., & Jensen, T. L. EuroCC WP 8.2/8.3/8.5 DK Survey.
[15] Austrian initiative on high performance computing. *Vienna Scientific Center. Retrieved August 25, 2024, from https://www.vsc.ac.at/home/*
[16] Lawal, Z. K., Yassin, H., Teck Ching Lai, D., & Che Idris, A. (2024). Understanding the Dynamics of Ocean Wave-Current Interactions Through Multivariate Multi-Step Time Series Forecasting. Applied Artificial Intelligence, 38(1), 2393978.
[17] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32
[18] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019, July). Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 2623-2631).

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 112–121

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

Scaling and Performance Analysis of Smilei in Hemispherical Foil Target Simulations for Inertial Fusion Energy

Valeria Ospina-Bohórquez*[a], Xavier Vaisseau[a]

[a]*Focused Energy GmbH, Im Tiefen See 45, 64293, Darmstadt, Germany*

Abstract

We conducted a performance evaluation of the particle-in-cell code Smilei, focusing on the interaction between a picosecond (ps) laser and free-standing hemispherical multi-layer targets with densities of $1.11 \times 10^{22}\,\mathrm{cm}^{-3}$. The simulations are performed the context of proton fast ignition, a promising approach for inertial fusion energy (IFE) that offers high gain potential and robust performance. Our study employed two-dimensional simulation geometries featuring hemispheres with diameters of 240 μm and 600 μm, representative of the values needed for IFE. Our final goal is to optimize proton focusing dynamics across various hemisphere sizes. The present objective is then to enhance simulation performance to facilitate the study of a large parameter space. We investigated parallelization strategies by varying the domain decomposition and employing or omitting the dynamic load balancing (DLB) algorithm of Smilei. We conducted simulations using between 192 and 48000 cores on the Vega supercalculator (Slovenia). Our results show that configurations with a large number of patches ($2^8 \times 2^7$) and cores (19200 to 48000) achieve optimal performance for both simulation setups. The extensive patch division acts as a local load-balancing mechanism, enhancing parallel execution efficiency. Although the DLB algorithm is beneficial for load balancing in smaller simulation boxes, it introduces communication and synchronization overhead in large hemispheres, resulting in longer simulation times compared to configurations without DLB.

Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day

Keywords: Fusion; ignition; clean energy; fast-ignition; laser-driven proton acceleration

1. Introduction

The successful demonstration on the National Ignition Facility (NIF) of achieving, for the first time, *ignition* and target gain $Q > 1$ [1, 2] has renewed interest in inertial fusion energy (IFE) as a possible source of clean and inexhaustible energy for humanity. While this historic result has validated the scientific basis for laser-driven inertial fusion, many scientific and technical challenges remain on the path to developing a commercially viable IFE scheme.

* Corresponding author. Tel.: +1 512 777 2663.
E-mail address: valeria.ospina@focused-energy.world

Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day
10.1016/j.procs.2025.02.266

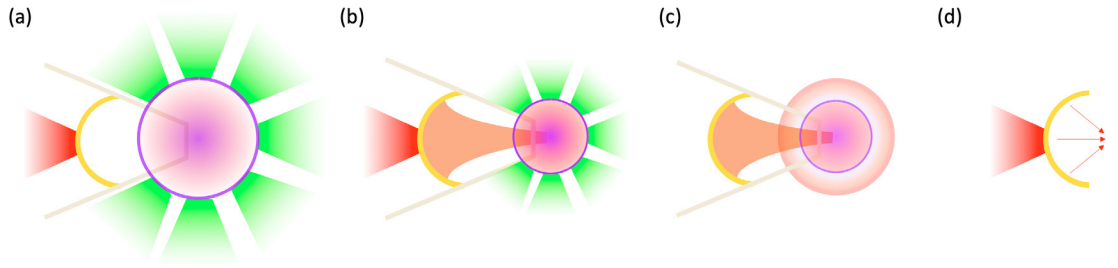


Fig. 1. **Proton fast ignition:** (a) A spherical capsule filled with DT fuel is compressed via rocket action after laser irradiation (green beams) onto an ablator shell. (b) At maximum compression, a proton beam generated by a high-intensity short pulse laser (red beam) interacting with a curved foil (a hemisphere) is sent through a protective cone to heat and (c) ignite the DT fuel. (d) The numerical study described here simplifies the problem by only considering the laser driven proton acceleration and focusing from free-standing hemispherical foil targets.

Focused Energy [3] is a startup pursuing proton fast ignition (pFI) [4], an advanced ignition scheme separating the stages of deuterium-tritium (DT - hydrogen) fuel compression and heating, potentially capable of achieving higher target gains and robust performance needed for an inertial fusion energy power plant. The success of this approach relies on the ability to generate a proton beam with the right characteristics to heat and ignite the isochoric DT fuel assembly. At the same time, a quasi-spherical DT fuel compression around a re-entrant cone needs to be achieved.

The approach followed to optimize the physics of proton beam generation, focusing and transport is based on large-scale numerical studies performed in conditions of interest for pFI, which demand a large number of computational hours. The numerical results will be benchmarked against experimental results obtained in down-scaled conditions, since the lasers needed to perform full-scale experiments are not yet available. Hence, numerical simulations are our primary tool for scientifically de-risking different IFE concepts, optimizing our target geometry and planning future experimental facilities. HPC access through EuroHPC to the Vega (Slovenia) and Karolina (Czech Republic) super-calculators is helping Focusing Energy to tackle these computational challenges. The performance analysis described hereinafter was done on the Vega supercalculator and will be extended to Karolina in the near future.

Figures 1a-1c illustrate the principal phases of pFI. Firstly, the DT fuel undergoes compression, driven inward by long-pulse (ns) laser irradiation (green beams in Fig. 1a) of the fuel capsule. At maximum compression, a ps laser (red beam in Fig. 1b) is used to generate a proton beam from a hemispherical curved foil. Subsequently, the proton beam is funneled down through a protective cone, traverses its tip (Fig. 1b) and ignites the compressed fuel (Fig. 1c). Focusing the ignitor proton beam down to a diameter of approximately 30 - 40 μm before entering the compressed fuel assembly is both crucial and challenging [5, 6].

2. Proton focusing from hemispherical foil targets

The most robust proton beam divergence mitigation technique takes advantage of the physics of Target Normal Sheath Acceleration TNSA [7] and uses a curved hemispherical foil to ballistically focus the proton beam. In TNSA, laser-driven electrons accelerated in the front side (laser side) of a dense and relatively thin ($\sim 10\,\mu$m) target are accelerated to relativistic speeds, cross the bulk of the target and create an intense electrostatic field in its rear side. This charge separation field can reach a strength of TV/m over several μm. The ions (mainly protons) present as organic impurities in the target rear surface are accelerated by this field to tens of MeV/amu energies. The acceleration occurs in the direction perpendicular to the target surface, hence curved foil targets naturally give rise to a focused proton beam. This was demonstrated by the pioneering work of Patel and collaborators [8], followed by subsequent campaigns on intermediate [9] and high-energy [10] scale laser systems. Hybrid particle-in-cell simulations predicted that a key parameter in the focusing abilities of curved foil targets is the uniformity of the laser irradiation [11]. When increasing the diameter of the hemisphere with respect to the laser focal spot size, the expansion of the plasma from the inner side becomes gradually non-uniform, with a maximum expansion velocity at the center and a slower value at the edge. The trajectory of lower energy protons generated far from the center consequently deviates from radial trajectories, resulting in a degraded focus.

Bartal *et al.* [12] later conducted an extensive study to compare partial (where the curved foil half angle $\theta_h < 90°$) and full hemispherical foils (where $\theta_h = 90°$), either freestanding or located inside a cone, revealing that proton trajectories from the former were in fact hyperbolic, converging to a finite focal spot before diverging from the increased electron pressure. The same authors demonstrated that an attached cone structure, primarily arising from the need to shield the proton source from the intense radiation generated during capsule compression in the fast ignition scheme, funnels the protons down to the tip and substantially improves the beam focusing. The latter is enhanced due to transient radial focusing fields generated by fast electron flowing along the cone walls. Subsequent theoretical and computational investigations concentrated mostly on partial hemispherical foils, either freestanding or inserted inside a cone, particularly on the dynamic field development and its effect on particle transport [13, 14]. In a recent study, King *et al.* explored the physics of partial hemispherical foils either freestanding or inserted in ~mm-size cone targets on the Orion laser facility (AWE, UK). The non-optimum focal spot size (5 µm FWHM) compared to the curved foil diameter (1.2 mm) yielded a flat foil-like behavior for the proton beam accelerated from the curved foil, with an initially diverging beam [15].

The complex physics at play when introducing the cone structure necessitates an examination of the individual roles played specifically by the hemispherical foil and the cone geometry on the focusing of protons. Yet only a handful of studies [12, 10, 16] follow such a modular approach. In addition, most studies have only investigated the effect of freestanding partial hemispherical targets, or inserted in a cone structure. Very little effort has been dedicated to the physics of full hemispherical foils, which may exhibit enhanced focusing capabilities. In this study, we opt for simplifying the target and consider only the laser-driven proton beam generation and focusing from free-standing full hemispherical foils, as shown in Fig. 1d, to optimize the proton focusing dynamics. The physics of short-pulse lasers of durations between tens of fs and some ps interacting with solid foil targets can be quantitatively described by 2/3D kinetic particle-in-cell (PIC) codes. For this purpose, we use the Smilei PIC code [17] which is open-source and has been designed for high performances on supercalculators.

3. Initial simulations

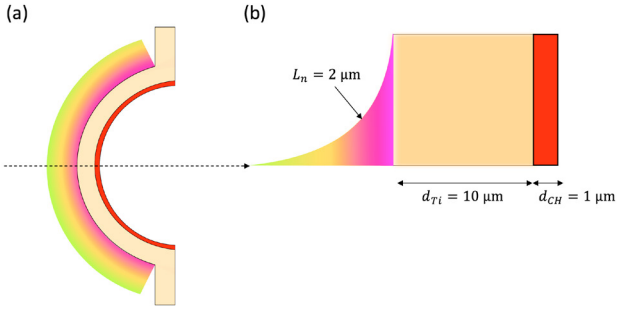


Fig. 2. **Target geometry used in the simulations:** (a) 2D geometry of a curved foil. (b) Lineout along the symmetry horizontal axis in (a). The hemisphere is composed of a preplasma layer (multicolor layer to the left), a tamper layer made of titanium and a proton-rich layer made of CH.

We performed an initial set of down-scaled simulations (with respect to pFI scales) considering three different hemisphere diameters ranging from 100 µm to 540 µm, as seen in Figs. 3a-3c. For these initial simulations we used the fully relativistic and electromagnetic PIC EPOCH code [18] in a 2D3V geometry (two-dimensional in configuration space and three-dimensional in momentum space). Note that the optimization study presented in the following sections was performed with the Smilei PIC code, a modern and massively parallelized code conceived to be run in the latest supercalculators. All simulations considered a constant laser focal spot of 40 µm full-width at half-maximum (FWHM) and a laser pulse duration of 40 fs. The idealized curved foil target consisted of three layers: a CH proton-rich layer, a titanium tamper layer and a preplasma with a density scale length $L_n = 2$ µm, as depicted in Figs. 2a and 2b.

Protons and electrons as well as all the ions present in the simulations have their actual mass and the speed of light is not altered, as done in some simulations known as speed-limited PIC [19, 20]. Following this technique, the speed of particles (often electrons) is artificially limited to a chosen value below the speed of light. This approach

allows to consider larger time steps in the simulations, making it useful for studies requiring long timescales, such as some high-energy-density applications where kinetic effects are not critical, like the study of cosmic rays transport on interstellar medium or star formation. However, this approach is not ideal for modeling laser-driven ion acceleration through TNSA. As previously stated, the latter mechanism relies on relativistic electron speeds to create a strong charge-separation field for ion acceleration at the rear side of the target. TNSA requires accurate spatial resolution and high temporal precision to capture rapid sheath formation of intense electric fields, both of which would be distorted by the speed cap.

In our down-scaled simulations, we observed that matching the hemisphere diameter with the laser focal spot seems essential for optimizing proton focusing, as recently stated by Kemp *et al.* [21]. We concluded that optimally illuminated hemispheres are characterized by a Ψ parameter, equivalent to the ratio between the hemisphere diameter and the laser focal spot ($\Psi = D_{hemi}/D_L$), between 6 and 8.5. The latter correspond to hemisphere diameters of $240\,\mu m \leq D_{hemi} \leq 340\,\mu m$ in our down-scaled conditions. In these cases, strong proton focusing of all energies with a small divergence half-angle $\theta_p \pm 10°$ is observed. Over-illuminated hemispheres characterized by an illumination parameter $\Psi = 2.5$ exhibit a strong focusing followed by a strong defocusing of the proton beam, as observed in Fig. 3a. Fig. 3c corresponds to a poorly illuminated hemisphere ($\Psi = 13.5$) characterized by degraded focusing capabilities.

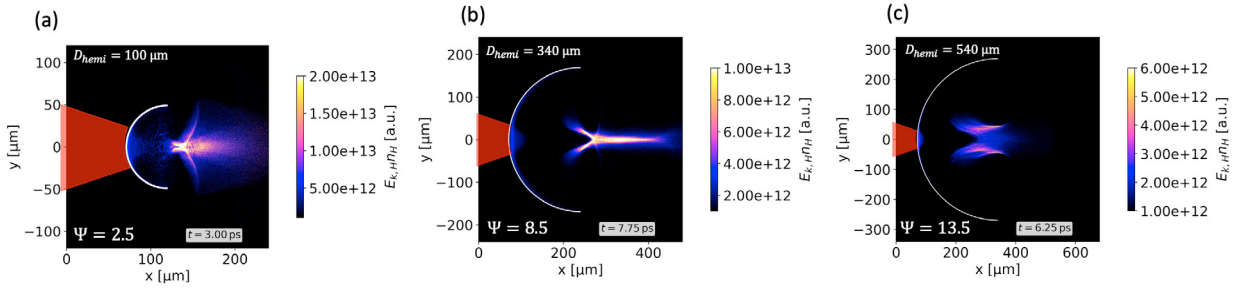


Fig. 3. **Proton energy density maps extracted at different time steps for three different hemisphere diameters compared with a flat foil.** a) $D_{hemi} = 100\,\mu m$, b) $D_{hemi} = 340\,\mu m$, c) $D_{hemi} = 540\,\mu m$. The white curved lines indicate the original location of the proton-rich layer. The red laser cartoon in each figure is drawn at scale. Notice that in all cases the laser focal spot is equal to 40 µm (FWHM).

4. Modeling approach

We intend to extrapolate the optimum proton focusing dynamics seen in Fig. 3b to pFI conditions i.e., larger hemispheres and ps laser pulses, ensuring that the illumination parameter remains in the optimal range $6 \leq \Psi \leq 8.5$. Larger hemispheres ($240\,\mu m \leq D_{hemi} \leq 1$ mm) will be needed in laser facilities with larger laser focal spots or multiple focal spots that can illuminate a large portion of the hemisphere's surface. The latter will be especially important when testing these targets in future pFI-scale facilities.

The optimization of proton focusing in large hemispheres will be done through a large parametric study involving several 2D simulations performed with the Smilei PIC code. As a start point, we focus here on the two simulation configurations summarized in Table 1. We wil simulate i) a 240 µm hemisphere interacting with a $D_L = 30$ µm, 430 J, 1 ps laser pulse (Prototype) and ii) a 600 µm hemisphere interacting with a $D_L = 100$ µm, 4300 J, 1 ps laser pulse (FullScale). The spatial resolution (0.035 µm) is chosen as to resolve the plasma Debye length $\lambda_D = \sqrt{\epsilon_0 k_B T_e/(n_e e^2)}$ (where ϵ_0 is the permittivity of free space, k_B is the Boltzmann constant, T_e is the electron temperature, n_e the electron density and e the elementary charge) and the time step (0.058 fs) is obtained from the Courant-Friedrichs-Lewy (CFL) condition considering a CFL constant of 0.7. The simulations were run for 180000 time steps equivalent to 10 ps physical time, a sufficiently long time to observe the evolution of the proton beam focusing dynamics. Absorbing boundary conditions are applied for both fields and particles in all directions and the ion and electron species are initialized with $T_{i,e} = 0$.

Figures 4a and 4b show the electron density at $t = 0$ for both simulation configurations. The initial $x = 300$ µm are left empty in both cases to let the laser pulse enter the simulation box before starting to interact with the hemisphere.

Name	Sim. geometry Hemi diam.	Laser params. $\lambda_L, I_L, \tau_L, D_L, E_L$	Target config. Preplasma, tamper, p^+-rich	ppc** /species Total particles	$\Delta x, y$ $L_x \times L_y$ $n_x \times n_y$	Δt t_{max}
Prototype	2D3V 240 μm	LP*, $\lambda_L = 1.054\,\mu$m, $I_L = 3\times10^{19}$ W cm^{-2}, $\tau_L = 1$ ps (200 fs fast rise), $D_L = 30\,\mu$m $E_L = 430$ J	$L_n = 2\,\mu$m, $n_{pp} = 6\,n_c^{***}$ Ti^{10+}, $x_{Ti} = 10\,\mu$m, $n_{Ti} = 10\,n_c$ $C^{6+}H^+$, $x_{CH} = 1\,\mu$m, $n_{CH} = 10\,n_c$	30 224 M	0.035 μm 660 μm × 270 μm 18688 × 7680	0.058 fs 10 ps
FullScale	2D3V 600 μm	LP, $\lambda_L = 1.054\,\mu$m, $I_L = 3\times10^{19}$ W cm^{-2}, $\tau_L = 1$ ps (200 fs fast rise), $D_L = 100\,\mu$m $E_L = 4300$ J	$L_n = 2\,\mu$m, $n_{pp} = 6\,n_c$ Ti^{10+}, $x_{Ti} = 10\,\mu$m, $n_{Ti} = 10\,n_c$ $C^{6+}H^+$, $x_{CH} = 1\,\mu$m, $n_{CH} = 10\,n_c$	30 575 M	0.035 μm 1200 μm × 630 μm 34048 × 17920	0.058 fs 10 ps

Table 1. **Summary of simulation parameters considering free-standing hemispherical foils of 240** μm **(top row) and 600** μm **(bottom row) diameters.** *Linearly polarized along the y-axis. **Number of particles per cell. ***Critical density for a 1 μm normalized laser pulse equal to $n_c = 1.11\times10^{21}$ cm^{-3}.

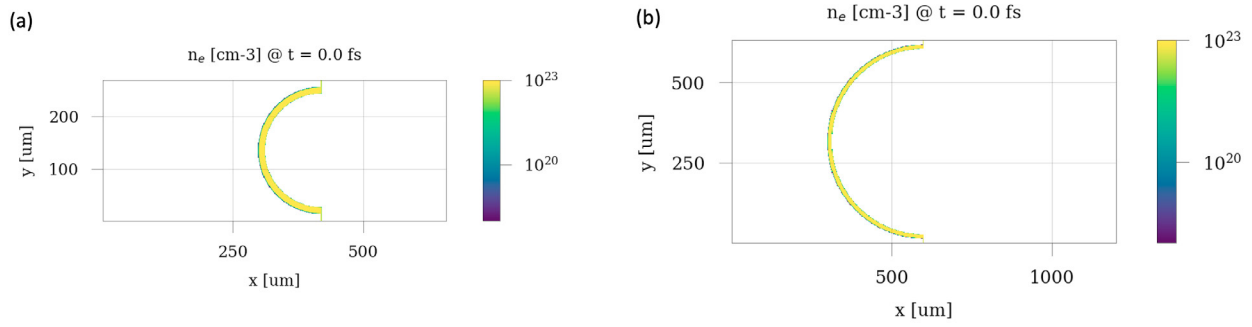


Fig. 4. **Electron density maps** extracted at $t = 0$ for a) the Prototype and b) FullScale simulation configurations (see Table 1).

The laser pulse has a Gaussian spatial distribution (transverse) and a trapezoidal temporal distribution mimicking a fast rise temporal laser pulse ($t_{rise} = t_{decay} = 200$ fs, $t_{plateu} = 1$ ps). For each simulation, we evaluate the accceleration of electrons and ions (especially protons) from the curved foil. It is then necessary to leave space ($\sim 1.5D_{hemi}$) for particle propagation at the right hand side of the hemisphere, resulting in relatively large simulation boxes. Figs. 5a - 5d correspond to electron (top row) and proton (bottom row) density maps of the Prototype simulation extracted 2.3 ps (left column) and 3.2 ps (right column) after the laser entered through the left simulation box boundary (the laser propagates from left to right). As observed, the laser interaction with the curved foil gives rise to a focused proton beam.

The final goal of the study is to optimize the characteristics of the focused proton beam e.g., its diameter, collimation length, flux and energy spectrum and to understand how do this characteristics evolve when increasing the hemisphere diameter. To achieve this one must unravel the role played by parameters such as the electron temperature T_e, the hemisphere illumination Ψ or the laser pulse duration τ_L, among others. It is then vital to perform a parametric study that spans across a large parameter space and therefore, to optimize the performance of the Smilei PIC code for these specific simulations.

5. Performance analysis

To optimize Smilei's parallel algorithm performance we have addressed both its distributed and shared memory as well as attempted to apply different domain decompositions. As any parallel algorithm, Smilei handles its communication between nodes through MPI protocol. The shared memory inside each MPI process is handled through OpenMP protocol.

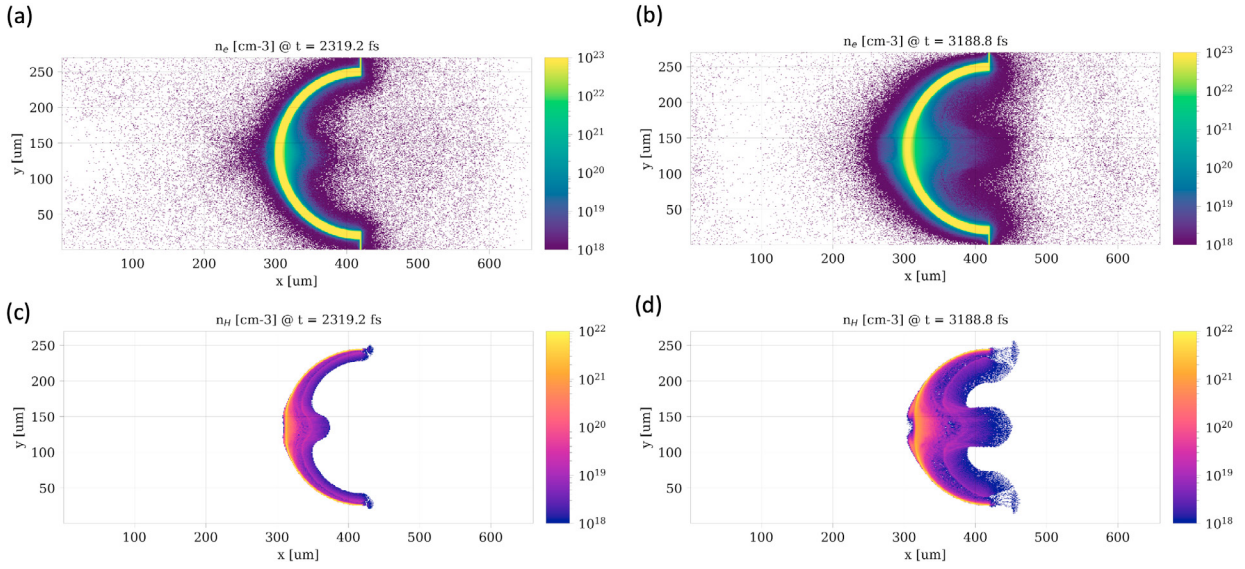


Fig. 5. **Free-standing hemispherical foil simulation results of the Prototype configuration**. a) Electron and c) proton density maps extracted 2.3 ps after the laser arrival (the laser enters through the left simulation boundary and propagates from left to right). b) Electron and d) proton density maps extracted 3.2 ps after the laser arrival.

Smilei's domain is divided into cells and patches. A patch is a subdivision of the domain and it is composed of a given number of cells per dimension (e.g., $8 \times 8, 100 \times 100$ in 2D). Several patches are then joined together in an MPI patch collections. Each MPI collection is assigned to a single process (and to all the threads inside this process). In this way, all threads assigned to the same MPI patch collection work in parallel, avoiding waiting times caused by synchronization.

Since particles are distributed unevenly in the domain some patches have naturally more particles than others. This can be observed in Figs. 5a-5d where particles are mostly present in the region $300\ \mu\text{m} \leq x \leq 450\ \mu\text{m}$ and absent elsewhere. In these cases, some MPI processes will take much longer to compute than the rest, creating long delays in the computation time. To address this issue, Smilei's dynamic load balancing (DLB) algorithm switches patches from one MPI process to another each n iterations, where n is a user-defined input parameter. The goal is to balance the computational load across all MPI processes, ensuring that none remain idle while others are still computing. We have also varied the n DLB parameter in order to obtain an optimum configuration. Table 2 summarizes the different parallelization configurations that were tested.

Simulation configuration	ID	Number of patches	DLB
Prototype	A	$2^8 \times 2^7$	100
	B	$2^8 \times 2^7$	0
	C	$2^7 \times 2^6$	100
	D	$2^7 \times 2^6$	0
FullScale	A	$2^8 \times 2^7$	100
	B	$2^8 \times 2^7$	0
	C	$2^7 \times 2^6$	100
	D	$2^7 \times 2^6$	0

Table 2. **Summary of parallelization configuration.** Configurations A and B were tested with 1, 10,100 and 250 nodes together with 48 tasks per node and 4 CPUs per tasks. Configurations C and D were tested with 1, 10, 100 and 150 nodes and the same number of tasks and CPUs per task, see details in the main text.

Configurations A and B were tested with 1, 10 ,100 and 250 nodes together with 48 tasks per node and 4 CPUs per tasks. Configuration D was tested with 1, 10, 100 and 150 nodes and the same number of tasks per node and CPUs per task. In this case, the maximum number of nodes could not reach the 250 value used in A and B since the number of

MPI processes (Nodes × ntasks per node = 250×48 = 12000) value must be lower than the number of patches ($2^7 \times 2^6$ = 8192). Finally, configuration C was only tested with 1 and 10 nodes since the dynamic load balancing algorithm requires to use at least 2 patches per MPI process. All simulation times were extrapolated up to 10 ps physical time for data acquisition purposes.

6. Results

Figures 6a and 6c and Figs. 6b and 6d correspond to the performance analysis results for the Prototype and FullScale simulation configurations, respectively. Table 3 summarizes the results for both simulation configurations, including the calculated speed-up and efficiency metrics for each run.

For the Prototype simulation (Fig. 6a), configuration A (blue) exhibits the best scalability up to 19200 cores. Configuration C (green) is not compatible with more than 1920 cores given its limited number of patches. Configurations B and D, where no dynamic load balancing was used, show a slower behavior independently of the number of cores, up to 19200 cores. In the case of 48000 cores, the large number of patches used in A most likely produce an overhead due to communication and synchronizations. As a result, configurations A and B with and without DLB, respectively, have a very similar wall clock time at the end of the simulation. As seen in Fig. 6c, each simulation run of configuration B can cost between 10^4 - 5×10^5 core-hours (c-h).

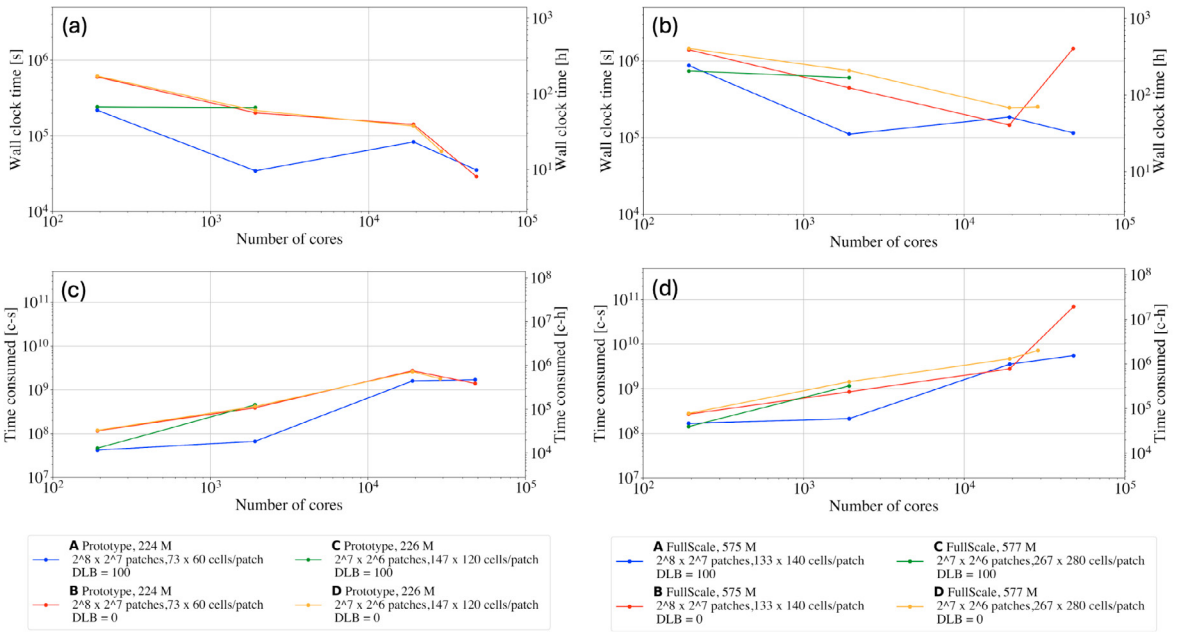


Fig. 6. **Performance analysis results for the Prototype and FullScale simulation configurations**. a) Wall clock time and c) consumed resources with respect to the number of cores employed for the Prototype simulation. b) and d) same as a) and b) for the FullScale simulation.

In the case of the FullScale simulation (Fig. 6b), configuration A (blue) also exhibits the best scalability up to 1920 cores. Configurations A and B (red) behave very similarly in the 19200 cores case where configuration B performs slightly better, suggesting again an overhead in communications due to the large patch number used in A. As in the Prototype case, Configuration A does not exhibit positive scaling beyond 1920 cores. As seen in Fig. 6d, each simulation run of configuration B can consume between 4×10^4 - 2×10^6 c-h. For the intermediate case of 1920 cores, the FullScale simulation is four times more costly than the Prototype simulation. For the fullScale simulation, enabling the DLB algorithm in the 48000 cores simulation (A vs B) seems essential to obtain a good performance.

As observed in Table 3, in the Prototype simulation the most efficient configurations are A with 1920 and 48000 cores which exhibit simulation times (efficiencies with respect to 192 cores) of 9.5 h (84.18%) and 9.8 h (83.77%). Configuration B with 48000 cores also exhibits a high efficiency of 95.17% and the lowest simulation time of 8.1 h.

In the case of the FullScale simulation, the most efficient configuration is also A with 1920 cores, 30.8 h of simulation time and an efficiency of 87.27%.

Sim. Config, ID	Nodes	Ntasks x node	CPUs per task	Cores	Time [h]	Speed-up wrt 192 cores	Efficiency wrt 192 cores
Prototype A	1	48	4	192	60.664	1.00	0.00%
	10	48	4	1920	9.595	6.32	84.18%
	100	48	4	19200	23.082	2.63	61.95%
	250	48	4	48000	9.844	6.16	83.77%
Prototype B	1	48	4	192	167.114	1.00	0.00%
	10	48	4	1920	55.973	2.99	66.51%
	100	48	4	19200	39.438	4.24	76.40%
	250	48	4	48000	8.072	20.70	95.17%
Prototype C	1	48	4	192	66.979	1.00	0.00%
	10	48	4	1920	65.737	1.02	1.86%
Prototype D	1	48	4	192	171.852	1.00	0.00%
	10	48	4	1920	60.151	2.86	65.00%
	100	48	4	19200	37.488	4.58	78.19%
	150	48	4	28800	17.389	9.88	89.88%

Sim. Config, ID	Nodes	Ntasks x node	CPUs per task	Cores	Time [h]	Speed-up wrt 192 cores	Efficiency wrt 192 cores
FullScale A	1	48	4	192	242.536	1.00	0.00%
	10	48	4	1920	30.871	7.86	87.27%
	100	48	4	19200	51.394	4.72	78.81%
	250	48	4	48000	32.013	7.58	86.80%
FullScale B	1	48	4	192	389.293	1.00	0.00%
	10	48	4	1920	123.993	3.14	68.15%
	100	48	4	19200	40.523	9.61	89.59%
	250				403.849	0.96	-3.74%
FullScale C	1	48	4	192	205.296	1.00	0.00%
	10	48	4	1920	167.940	1.22	18.20%
FullScale D	1	48	4	192	404.438	1.00	0.00%
	10	48	4	1920	208.304	1.94	48.50%
	100	48	4	19200	67.980	5.95	83.19%
	150	48	4	28800	69.906	5.79	82.72%

Table 3. **Summary of the performance analysis results for the Prototype and FullScale simulation configurations**. Optimum configurations are highlighted in green and correspond to low simulation times and high speed-up and efficiency coefficients.

7. Conclusion

We have conducted a performance analysis of the particle-in-cell code Smilei. We considered the interaction of a ps laser with free-standing hemispherical multi-layer targets with densities of $1.11 \times 10^{22}\,\mathrm{cm}^{-3}$ in the context of proton fast ignition, a promising IFE scheme which can potentially lead to high gains and robust performances. The 2D simulation geometries named Protoype and FullScale consisted of hemispheres of 240 μm and 600 μm diameters. The ultimate objective of this study is to optimize proton focusing dynamics for all hemisphere sizes. The intermediate goal of this paper is to enhance simulation performance for both configurations, maximizing the number of simulations that can be executed and allowing for the exploration of a broad parameter space.

The parallelization parameters considered for the optimization included the domain decomposition in a given number of patches in each dimension and the use or not of the dynamic load balancing algorithm of Smilei. We executed runs with 192 up to 48000 cores changing the number of nodes and setting constant task per node and CPU per task values equal to 48 and 4, respectively.

The configurations with a large number of patches ($2^8 \times 2^7$) and an intermediate number of cores (1920) exhibit the optimum performance for both simulation configurations. The subdivision of the computational domain in a large number patches acts as a local dynamic load balance strategy improving the parallel execution of the code.

The dynamic load balancing algorithm of Smilei switches patches from one MPI process to another to try to equalize the computational load in all the simulation domain. Our simulation geometry benefits from this algorithm since our particles are located in a reduced area of the simulation box and much of it is empty. This is specially true in the large hemisphere case with 48000 cores, where a good performance is only enabled when using the DLB

and a very poor performance is seen without it. While the overall strong scaling is acceptable, it is not optimal. Running simulations with an even larger number of patches might help identify a better performance peak. However, a synchronization overhead could hinder effective reduction of execution time. The DLB algorithm could also be stopped at a specific time, once the plasma has expanded through a substantial portion of the simulation box, as its operation may no longer be as critical at that stage.

Considering a 5 million core-hours allocation, one could perform about 160 simulations of rougly 10 h each using 1920 cores in the Prototype case. With the same 5 million core-hours one could perform 16 simulations of the FullScale case lasting about 20 h each and using, as well, 1920 cores. This number of simulations should be enough to optimize proton focusing in both cases. A factor 3 increase in computational time was considered to take into account I/O operations.

Acknowledgements

The authors gratefully acknowledge the HPC RIVR consortium (www.hpc-rivr.si) and EuroHPC JU (eurohpc-ju.europa.eu) for funding this research by providing computing resources of the HPC system Vega at the Institute of Information Science (www.izum.si).

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254).

References

[1] H. Abu-Shawareb, R. Acree, P. Adams, J. Adams, B. Addis, R. Aden, P. Adrian, B. Afeyan, M. Aggleton, L. Aghaian, et al., Lawson criterion for ignition exceeded in an inertial fusion experiment, Physical Review Letters 129 (7) (2022) 075001.

[2] H. Abu-Shawareb, R. Acree, P. Adams, J. Adams, B. Addis, R. Aden, P. Adrian, B. Afeyan, M. Aggleton, L. Aghaian, et al., Achievement of target gain larger than unity in an inertial fusion experiment, Physical Review Letters 132 (6) (2024) 065102.

[3] T. Ditmire, M. Roth, P. Patel, D. Callahan, G. Cheriaux, P. Gibbon, D. Hammond, A. Hannasch, L. Jarrott, G. Schaumann, et al., Focused energy, a new approach towards inertial fusion energy, Journal of Fusion Energy 42 (2) (2023) 27.

[4] M. Roth, T. Cowan, M. Key, S. Hatchett, C. Brown, W. Fountain, J. Johnson, D. Pennington, R. Snavely, S. Wilks, et al., Fast ignition by intense laser-accelerated proton beams, Physical review letters 86 (3) (2001) 436.

[5] S. Atzeni, Inertial fusion fast ignitor: Igniting pulse parameter window vs the penetration depth of the heating particles and the density of the precompressed fuel, Physics of Plasmas 6 (8) (1999) 3316–3326.

[6] S. Atzeni, M. Temporal, J. J. Honrubia, A first analysis of fast ignition of precompressed icf fuel by laser-accelerated protons, Nuclear Fusion 42 (1) (2002) L1–L4.

[7] A. Macchi, M. Borghesi, M. Passoni, Ion acceleration by superintense laser-plasma interaction, Rev. Mod. Phys. 85 (2013) 751–793.

[8] P. Patel, A. Mackinnon, M. Key, T. Cowan, M. Foord, M. Allen, D. Price, H. Ruhl, P. Springer, R. Stephens, Isochoric heating of solid-density matter with an ultrafast proton beam, Physical review letters 91 (12) (2003) 125004.

[9] R. A. Snavely, B. Zhang, K. Akli, Z. Chen, R. R. Freeman, P. Gu, S. P. Hatchett, D. Hey, J. Hill, M. H. Key, Y. Izawa, J. King, Y. Kitagawa, R. Kodama, A. B. Langdon, B. F. Lasinski, A. Lei, A. J. MacKinnon, P. Patel, R. Stephens, M. Tampo, K. A. Tanaka, R. Town, Y. Toyama, T. Tsutsumi, S. C. Wilks, T. Yabuuchi, J. Zheng, Laser generated proton beam focusing and high temperature isochoric heating of solid matter, Physics of Plasmas 14 (9) (2007) 092703.

[10] C. McGuffey, J. Kim, M. Wei, P. Nilson, S. Chen, J. Fuchs, P. Fitzsimmons, M. Foord, D. Mariscal, H. McLean, et al., Focussing protons from a kilojoule laser for intense beam heating using proximal target structures, Scientific reports 10 (1) (2020) 9415.

[11] M. H. Key, Status of and prospects for the fast ignition inertial fusion concepta), Physics of Plasmas 14 (5) (2007) 055502.

[12] T. Bartal, M. E. Foord, C. Bellei, M. H. Key, K. A. Flippo, S. A. Gaillard, D. T. Offermann, P. K. Patel, L. C. Jarrott, D. P. Higginson, M. Roth, A. Otten, D. Kraus, R. B. Stephens, H. S. McLean, E. M. Giraldez, M. S. Wei, D. C. Gautier, F. N. Beg, Focusing of short-pulse high-intensity laser-accelerated proton beams, Nature Physics 8 (2) (2012) 139–142, publisher: Nature Publishing Group.

[13] C. Bellei, M. E. Foord, T. Bartal, M. H. Key, H. S. McLean, P. K. Patel, R. B. Stephens, F. N. Beg, Electron and ion dynamics during the expansion of a laser-heated plasma under vacuum, Physics of Plasmas 19 (3) (2012) 033109.

[14] B. Qiao, M. E. Foord, M. S. Wei, R. B. Stephens, M. H. Key, H. McLean, P. K. Patel, F. N. Beg, Dynamics of high-energy proton beam acceleration and focusing from hemisphere-cone targets by high-intensity lasers, Phys. Rev. E 87 (2013) 013108.

[15] M. King, A. Higginson, C. McGuffey, R. Wilson, G. Schaumann, T. Hodge, J. B. Ohland, S. Gales, M. Hill, S. F. Pitt, et al., Geometry effects on energy selective focusing of laser-driven protons with open and closed hemisphere-cone targets, Plasma Physics and Controlled Fusion 66 (1) (2023) 015001.

[16] K. Bhutwala, C. McGuffey, W. Theobald, O. Deppert, J. Kim, P. M. Nilson, M. S. Wei, Y. Ping, M. E. Foord, H. S. McLean, P. K. Patel, A. Higginson, M. Roth, F. N. Beg, Transport of an intense proton beam from a cone-structured target through plastic foam with unique proton source modeling, Phys. Rev. E 105 (2022) 055206.

[17] J. Derouillat, A. Beck, F. Pérez, T. Vinci, M. Chiaramello, A. Grassi, M. Flé, G. Bouchard, I. Plotnikov, N. Aunai, et al., Smilei: A collaborative, open-source, multi-purpose particle-in-cell code for plasma simulation, Computer Physics Communications 222 (2018) 351–373.
[18] T. D. Arber, K. Bennett, C. S. Brady, A. Lawrence-Douglas, M. Ramsay, N. J. Sircombe, P. Gilles, R. Evans, H. Schmitz, B. A.R, R. C.P, Contemporary particle-in-cell approach to laser-plasma modelling, Plasma Physics and Controlled Fusion 57 (6) (2015) 1–26.
[19] G. R. Werner, T. G. Jenkins, A. M. Chap, J. R. Cary, Speeding up simulations by slowing down particles: Speed-limited particle-in-cell simulation, Physics of Plasmas 25 (12) (2018).
[20] S. Ji, P. F. Hopkins, A reduced speed-of-light formulation of the magnetohydrodynamic-particle-in-cell method, Monthly Notices of the Royal Astronomical Society 516 (4) (2022) 5143–5147.
[21] A. J. Kemp, S. C. Wilks, M. Tabak, Laser-to-proton conversion efficiency studies for proton fast ignition, Physics of Plasmas 31 (2024) 042709.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 122–129

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

Portable test run of ESPResSo on EuroHPC systems via EESSI

Alan O'Cais[a], Kenneth Hoste[b,*], Jean-Noël Grad[c], Caspar van Leeuwen[d], Lara Peeters[b], Satish Kamath[d], Thomas Röblitz[e], Richard Topouchian[e], Bob Dröge[f], Pedro Santos Neves[f], Rudolf Weeber[c]

[a] *University of Barcelona, Martí i Franquès 1, 08028 Barcelona, Spain*
[b] *Department of Information and Communication Technology, Ghent University, Krijgslaan 289 S9, 9000 Ghent, Belgium*
[c] *Institute for Computational Physics, University of Stuttgart, Allmandring 3, 70569 Stuttgart, Germany*
[d] *Compute Services, SURF, Science Park 140, 1098 XG Amsterdam, The Netherlands*
[e] *IT division, University of Bergen, Nygårdsgaten 5, 5015 Bergen, Norway*
[f] *Center for Information Technology, University of Groningen, Smitsborg, Nettelbosje 1, 9747 AJ Groningen, The Netherlands*

Abstract

One of the milestones of the EuroHPC Centre of Excellence MultiXscale is to be able to run the EESSI test suite on at least two different architectures available on EuroHPC Supercomputers. Our initial efforts focused on making the test suite portable across two different supercomputers: Karolina and Vega (the CPU partitions of both are a Zen2 micro-architecture).

More recently we have spent time getting the same test suite working on a more "exotic" architecture, the ARM A64FX architecture of Deucalion (which was in pre-production at the time of the experiment). This has raised some additional complications for EESSI as CernVM-FS (which is used to distribute EESSI) was not yet natively available there.

We show the current scalability of the ESPResSo application using the portable test suite. ESPResSo is already known to have scalability issues for both multi-node and multi-GPU configurations, which are currently being analysed in collaboration with the POP Centre of Excellence. The purpose of this effort was to ensure that we can quickly and automatically record the performance of the application across a range of EuroHPC systems (i.e., ESPResSo acts as a pilot application for the full test suite).

Keywords: HPC; EESSI; software testing; CI/CD

* Corresponding author.
E-mail address: kenneth.hoste@ugent.be

1877-0509
10.1016/j.procs.2025.02.267

1. Introduction

The MultiXscale HPC Centre of Excellence[1] has ongoing development projects on the full spectrum of EuroHPC compute resources which we leverage to help deliver the project goals. One of these project goals is to create a test suite that is portable between the different EuroHPC architectures and can automatically gather performance and scalability information for the target application.

In the context of MultiXscale, target applications are delivered via the European Environment for Scientific Software Installations (EESSI, pronounced as "easy"[2]). For this paper we focus on a single application within the test suite, the Extensible Simulation Package for Research on Soft Matter (ESPResSo[3]), which is a key application within the context of MultiXscale. The test suite itself is built using ReFrame[4] [3] and is available as a Git repository[5].

The purpose of the work is not to show the performance improvement of ESPResSo (this is a work-in-progress [2] and the subject of a further EuroHPC allocation), but to showcase the portability of the test suite, particularly when leveraging EESSI, and its ability to quickly and automatically gather performance and scalability results across multiple resources. In Section 2, we describe the background and setup of the experiments. The results are presented in Section 3, and a brief discussion of these is given in Section 4.

2. Methods

In this section we provide the background to the results presented in Section 3. We briefly outline the context of both how we deliver applications via EESSI, and the application itself, ESPResSo. We also outline the specific adaptation we needed to make to run in parallel on Deucalion, where EESSI was not yet natively available.

2.1. EESSI

EESSI [1] is a collaboration between different European partners in the HPC community to build a common stack of scientific software installations for HPC systems and beyond (including laptops, personal workstations and cloud infrastructure). The development of EESSI is currently co-funded as a component of the EuroHPC Centre of Excellence MultiXscale. For end users, EESSI provides a uniform user experience with respect to available scientific software, regardless of which system they use. This is an important issue when it comes to the portability of a test suite, as the test suite must have access to an optimised version of the application under study.

EESSI aims to provide a software stack that works on wide range of platforms: from personal workstations and laptops to HPC clusters and the cloud. To achieve this, EESSI has to support a wide range of different CPUs, networks, GPUs, and so on. This is an ongoing effort: the most common hardware today (recent Intel, AMD and ARM CPUs, NVIDIA GPUs, Ethernet, Infiniband) is already supported, and the range of supported hardware will be continuously expanded. Additionally, EESSI is designed to work on any Linux distribution, as well as macOS via Lima and Windows via WSL. EESSI not only wants to focus on the performance of the software, but also on automating the workflow for maintaining the software stack, thoroughly testing the installations, and collaborating efficiently.

2.1.1. The EESSI test suite

The EESSI test suite[6] is a collection of tests to check the software installations included in the EESSI software layer are working and performing/scaling as expected, using ReFrame[7]. The collection is openly developed[8] and at the time of writing includes tests for PyTorch, CP2K, ESPResSo, LAMMPS, QuantumESPRESSO, TensorFlow, GROMACS, and the OSU Micro-Benchmarks. The tests are designed to be *portable* across different systems[9].

[1] https://www.multixscale.eu/
[2] https://eessi.io/docs/
[3] https://espressomd.org/
[4] https://reframe-hpc.readthedocs.io/
[5] https://github.com/EESSI/test-suite
[6] https://eessi.io/docs/test-suite/
[7] https://reframe-hpc.readthedocs.io/
[8] https://github.com/EESSI/test-suite
[9] https://eessi.io/docs/test-suite/writing-portable-tests/

2.2. ESPResSo

ESPResSo [4, 5] is a highly versatile molecular dynamics software package optimised for coarse-grained simulations of molecular systems ranging from the nanometer to micrometer scales. It provides a lattice-Boltzmann solver with a particle coupling scheme to model hydrodynamic interactions, long-range solvers for electrostatic and magnetostatic interactions, and Monte Carlo methods to model chemical reactions. These features can be combined to study complex physical processes occurring in soft matter systems that feature reactive species, active matter, or liquid-solid interfaces. ESPResSo has been used to study supercapacitors, polyelectrolytes, charged colloids, ferrofluids, pH-responsive hydrogels, DNA translocation through a nanopore, motile bacteria in porous media, and red blood cells. ESPResSo is free and open-source software published under the GNU General Public License v3. It is parallelized and suitable for desktop computers, university clusters, and supercomputers. Some features can leverage GPU accelerators. Users interact with the software via its Python interface.

Since 14 June 2024, ESPResSo v4.2.2 is available in the EESSI production repository[10], optimised for the 8 target CPU microarchitectures that are fully supported by version 2023.06 of EESSI. This allows running ESPResSo effortlessly on the EuroHPC systems where EESSI is already available, such as Vega[11] and Karolina[12] which are included in this study. On 27 June 2024, an additional installation of ESPResSo v4.2.2 that is optimised for Arm A64FX processors was added, which enables also running ESPResSo efficiently on Deucalion[13], even though EESSI was not available yet system-wide on Deucalion (see Section 2.4).

With the portable test for ESPResSo that is available in the EESSI test suite we can easily evaluate the scalability of ESPResSo across EuroHPC systems, even if those systems have (very) different CPU microarchitectures.

2.2.1. Simulating Lennard-Jones fluids using ESPResSo

Lennard-Jones fluids model interacting soft spheres with a potential that is weakly attractive at medium range and strongly repulsive at short range. Originally designed to model noble gases, this simple setup now underpins most particle-based simulations, such as ionic liquids, polymers, proteins and colloids, where potentials that are strongly repulsive at short ranges are desirable to prevent particles from overlapping with one another. In addition, Lennard-Jones interactions tend to account for a significant portion of the force calculation in atomistic simulations, since they often feature a large excess of solvent molecules compared to solute molecules. Compared to other potentials, the Lennard-Jones interaction is inexpensive to calculate, and its limited range allows us to partition the simulation domain into arbitrarily small regions that can be distributed among many processors.

2.3. Portable test to evaluate the performance of ESPResSo

To evaluate the performance of ESPResSo, we have implemented a portable test for ESPResSo in the EESSI test suite; the results shown in Section 3 were collected using version 0.3.2 of the EESSI test suite.

After installing and configuring the EESSI test suite on Vega, Karolina, and Deucalion, running the Lennard-Jones (LJ) test case with ESPResSo 4.2.2 available in EESSI can be done with:

```
reframe --name "ESPRESSO_LJ.*%module_name=ESPResSo/4.2.2"
```

With this command, the ReFrame runtime will create job scripts like the one reproduced below, submit these from a single task to 8 full nodes (with one task per core, as defined in the ESPResSo test), and extract the relevant performance numbers from the output file. The number of tasks, tasks per node, partition names, and memory arguments are all generated by ReFrame to match the system architecture as described in the ReFrame configuration files for Vega[14], Karolina[15] and Deucalion[16].

[10] https://www.eessi.io/docs/repositories/software.eessi.io/
[11] https://www.izum.si/en/vega-en/
[12] https://www.it4i.cz/en/infrastructure/karolina
[13] https://rnca.fccn.pt/en/deucalion/
[14] https://github.com/EESSI/test-suite/blob/v0.3.2/config/izum_vega.py
[15] https://github.com/EESSI/test-suite/blob/v0.3.2/config/it4i_karolina.py
[16] https://github.com/EESSI/test-suite/blob/v0.3.2/config/macc_deucalion.py

```
#!/bin/bash
#SBATCH --ntasks=<ntasks>
#SBATCH --ntasks-per-node=<ntasks_per_node>
#SBATCH --cpus-per-task=1
#SBATCH --time=5:0:0
#SBATCH --partition <partition_name>
#SBATCH --export=None
#SBATCH --mem=<memory>
source /cvmfs/software.eessi.io/versions/2023.06/init/bash
module load ESPResSo/4.2.2-foss-2023a
mpirun -np 96 python3 lj.py
```

2.4. Running ESPResSo in parallel on Deucalion via EESSI + `cvmfsexec`

While EESSI is already available system-wide on both Vega and Karolina for some time, it was not available yet on Deucalion when these performance experiments were run. Nevertheless, we were able to obtain the optimised installation of ESPResSo for A64FX available in EESSI by leveraging the `cvmfsexec` tool[17], and by creatively implementing two simple shell wrapper scripts, which we will describe in more detail in the following subsections.

2.4.1. `cvmfsexec` wrapper script

The first wrapper script `cvmfsexec_eessi.sh` can be used to run a command in a subshell in which the EESSI CernVM-FS repository (`software.eessi.io`) is mounted via `cvmfsexec`. This script can be used by regular users on Deucalion, it does not require any special privileges beyond the Linux kernel features that `cvmfsexec` leverages, such as user namespaces, and internet access.

We reproduce the contents of `cvmfsexec_eessi.sh` here:

```
#!/bin/bash
if [ -d /cvmfs/software.eessi.io ]; then
    # run command directly, EESSI CernVM-FS repository is already mounted
    "$@"
else
    # run command via in subshell where EESSI CernVM-FS repository is mounted,
    # via cvmfsexec which is set up in a unique temporary directory
    orig_workdir=$(pwd)
    mkdir -p /tmp/$USER
    tmpdir=$(mktemp -p /tmp/$USER -d)
    cd $tmpdir
    git clone https://github.com/cvmfs/cvmfsexec.git > $tmpdir/git_clone.out 2>&1
    cd cvmfsexec
    ./makedist default > $tmpdir/cvmfsexec_makedist.out 2>&1
    cd $orig_workdir
    $tmpdir/cvmfsexec/cvmfsexec software.eessi.io -- "$@"
    # cleanup
    rm -rf $tmpdir
fi
```

The script should be stored in the `bin` subdirectory of your home directory. Make sure the script is executable:

```
chmod u+x ~/bin/cvmfsexec_eessi.sh
```

[17] https://github.com/cvmfs/cvmfsexec

A simple way to test this script is to use it to inspect the contents of the EESSI repository:

```
~/bin/cvmfsexec_eessi.sh ls /cvmfs/software.eessi.io
```

or to start an interactive shell in which the EESSI repository is mounted:

```
~/bin/cvmfsexec_eessi.sh /bin/bash -l
```

The job scripts that were submitted by ReFrame on Deucalion leverage the `cvmfsexec_eessi.sh` script to set up the environment and get access to the ESPResSo v4.2.2 installation that is available in EESSI.

2.4.2. `orted` wrapper script

In order to get multi-node runs of ESPResSo working without having EESSI available system-wide, we also had to create a small wrapper script for the `orted` command that is used by Open MPI to start processes on remote nodes. This is necessary because `mpirun` launches `orted`, which must be run in an environment in which the EESSI repository is mounted. If not, MPI startup will fail with an error like "`error: execve(): orted: No such file or directory`". This wrapper script must be named `orted`, and must be located in a path listed in the `$PATH` environment variable. We placed it in `~/bin/orted`, and added "`export PATH=$HOME/bin:$PATH`" to our `~/.bashrc` login script. It needs to be executable, otherwise it might throw an error that starts with "`An ORTE daemon has unexpectedly failed after launch ...`"

The contents of the `orted` wrapper script are as follows:

```
#!/bin/bash
# first remove path to this orted wrapper from $PATH, to avoid infinite loop
orted_wrapper_dir=$(dirname $0)
export PATH=$(echo $PATH | tr ':' '\n' | grep -v $orted_wrapper_dir | tr '\n' ':')
~/bin/cvmfsexec_eessi.sh orted "$@"
```

Make the wrapper script executable:

```
chmod u+x ~/bin/orted
```

2.4.3. Slurm job script

We can use the `cvmfsexec_eessi.sh` script in a Slurm job script on Deucalion to initialise the EESSI environment in a subshell in which the EESSI CernVM-FS repository is mounted, and subsequently load the module for ESPResSo v4.2.2 and launch the Lennard-Jones fluid simulation via `mpirun`. A job script example using 2 full 48-core nodes on A64FX partition of Deucalion is:

```
#!/bin/bash
#SBATCH --ntasks=96
#SBATCH --ntasks-per-node=48
#SBATCH --cpus-per-task=1
#SBATCH --time=5:0:0
#SBATCH --partition normal-arm
#SBATCH --export=None
#SBATCH --mem=30000M
~/bin/cvmfsexec_eessi.sh << EOF
export EESSI_SOFTWARE_SUBDIR_OVERRIDE=aarch64/a64fx
source /cvmfs/software.eessi.io/versions/2023.06/init/bash
module load ESPResSo/4.2.2-foss-2023a
export SLURM_EXPORT_ENV=HOME,PATH,LD_LIBRARY_PATH,PYTHONPATH
mpirun -np 96 python3 lj.py
EOF
```

The `lj.py` Python script is available as part of the EESSI test suite[18]. Note that when running the EESSI test suite via ReFrame, a job script like this is automatically generated. We are showing it here to highlight how the `cvmfsexec_eessi.sh` wrapper script can be leveraged.

3. Results

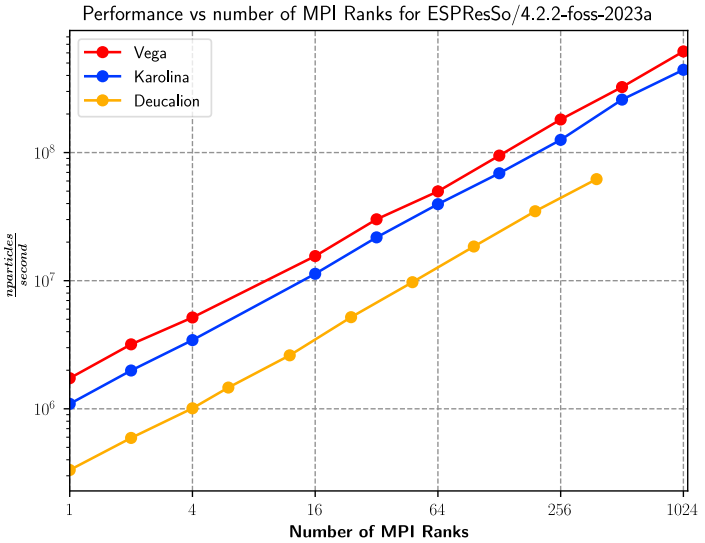


Fig. 1. Strong scaling parallel performance of ESPResSo, expressed in particles integrated per second as it scales with the number of MPI ranks.

The performance results of the tests are collected by ReFrame in a detailed JSON report. This report can easily be machine parsed to generate plots.

In Figure 1, we show that the strong scaling parallel performance of ESPResSo, expressed in particles integrated per second, scales linearly with the number of cores. On Vega using 8 nodes (1024 MPI ranks, one per physical core), ESPResSo 4.2.2 can integrate the equations of motion of roughly 615 million particles every second. On Deucalion using 8 nodes (384 cores), we observe a performance of roughly 62 million particles integrated per second.

In Figure 2, we show that the parallel efficiency of ESPResSo 4.2.2 (weak scaling, 2000 particles per MPI rank) decreases approximately linearly with the logarithm of the number of cores on the three EuroHPC systems we used. These results align well with the benchmark results discussed in the initial scalability report of ESPResSo [2], which validates them to some extent. This is important since they were collected entirely differently in this study, and across a range of different systems.

4. Discussion

Figure 1 shows similar scaling across systems, but with a substantial difference in baseline performance (1 rank, executed on a single core). While the runs on Vega and Karolina where both executed on the same processor model (AMD Epyc 7H12 64-core processor, dual socket nodes), they run at very different clock frequencies. On Vega, boost clocks are enabled, and manual inspection during a run shows clock speeds of around 3.1 GHz. On Karolina, CPUs appear to be underclocked at 2.1 GHz. Performance on the A64FX CPUs (which support ARMv8.2 + SVE instructions) in Deucalion is lower than on the AMD Epyc-based systems. While we have not done an in-depth

[18] https://github.com/EESSI/test-suite/blob/v0.3.2/eessi/testsuite/tests/apps/espresso/src/lj.py

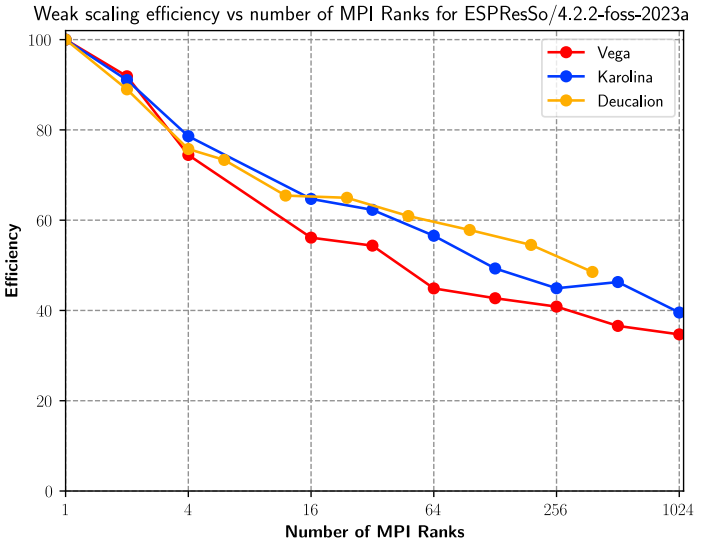


Fig. 2. Weak scaling parallel efficiency of ESPResSo with 2000 particles per MPI rank.

analysis, factors that might have contributed are an even lower clock frequency (2.0 GHz), as well as potentially less efficient optimisation by the compiler. The number of CPU models supporting ARMv8.2a + SVE based instructions is quite limited, and therefore it is likely that less effort has been done on GCC compiler optimisations for this specific architecture.

Figure 2 in particular demonstrates the scalability issue ESPResSo is facing, the improvement of which is the subject of ongoing work within MultiXscale (and in collaboration with the POP Centre of Excellence).

What is worthy noting with respect to the EESSI test suite is that the gathering of the data to prepare the plots of Figure 1 and Figure 2 is entirely automated. The input of the application owners was required during the preparation of the tests and associated data. However, the data itself was gathered by the EESSI team without further input from the application owners. This has the implication that the EESSI test suite has the capability to help simplify (and centralise) the gathering of scaling and performance data for supported applications. This would free the application developers from requiring continuous access to resources for the entire spectrum of EuroHPC platform in order to evaluate the performance of their application during their development processes.

5. Conclusion & future work

We have demonstrated how evaluating the performance of ESPResSo across different EuroHPC systems is simplified by the European Environment for Scientific Software Installations (EESSI). This shared stack of optimised scientific software installations, which includes ESPResSo, enables the implementation of *portable* tests that can be run automatically across different scales – ranging from single-core to multi-node. Together, EESSI and the portable test for ESPResSo that was implemented in the EESSI test suite make assessing the impact of improvements implemented by the developers a lot more efficient, since they significantly reduce the overhead that is typically experienced when running performance scaling experiments across different systems.

In the context of the EuroHPC Centre of Excellence MultiXscale, we are taking additional steps to further reduce the burden of evaluating performance improvements for developers of scientific software like ESPResSo. We are setting up a separate CernVM-FS repository in EESSI where pre-release versions of software can be automatically installed through an efficient developer-friendly procedure. That way, the impact of improvements that were implemented can quickly and efficiently be evaluated across different EuroHPC systems, without having to manually set up, manage, and use separate build environments on each of them.

Acknowledgements

Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and countries participating in the project under grant agreement No 101093169. In particular, we acknowledge the co-funding provided by the Ministerio de Ciencia, Innovación y Universidades, Spain; the Ministerie van Economische Zaken, the Netherlands; and the Departement Economie, Wetenschap & Innovatie, Flemish Governement, Belgium; The contributions of Thomas Röblitz and Richard Topouchian were co-funded by The Research Council of Norway through grant 341493. The contributions of Jean-Noël Grad and Rudolf Weeber were co-funded by the Federal Ministry of Education and Research (Bundesministeriums für Bildung und Forschung, BMBF) under the funding code 16HPC095; the responsibility for the content of this publication lies with the authors.

We also acknowledge the EuroHPC Joint Undertaking for awarding us access to:

- Vega at IZUM, Slovenia
- Karolina at IT4Innovations, Czech Republic
- MeluXina at LuxProvide, Luxembourg
- Discoverer at SofiaTech, Bulgaria
- LUMI at CSC, Finland
- Leonardo at CINECA, Italy
- MareNostrum5 as BSC, Spain
- Deucalion at MACC, Portugal

References

[1] Dröge, B., Rusu, V.H., Hoste, K., van Leeuwen, C., O'Cais, A., Röblitz, T., 2023. EESSI: A cross-platform ready-to-use optimised scientific software stack. Software: Practice and Experience 53, 176–210. doi:10.1002/spe.3075.

[2] Grad, J.N., Weeber, R., 2023. Report on the current scalability of ESPResSo and the planned work to extend it. MultiXscale Deliverable 2.1. EuroHPC Centre of Excellence MultiXscale. doi:10.5281/zenodo.8420222.

[3] Karakasis, V., Manitaras, T., Rusu, V.H., Sarmiento-Pérez, R., Bignamini, C., Kraushaar, M., Jocksch, A., Omlin, S., Peretti-Pezzi, G., Augusto, J.P.S.C., Friesen, B., He, Y., Gerhardt, L., Cook, B., You, Z.Q., Khuvis, S., Tomko, K., 2020. Enabling continuous testing of HPC systems using ReFrame, in: Juckeland, G., Chandrasekaran, S. (Eds.), Tools and Techniques for High Performance Computing, Springer International Publishing, Cham, Switzerland. pp. 49–68. doi:10.1007/978-3-030-44728-1_3.

[4] Weeber, R., Grad, J.N., Beyer, D., Blanco, P.M., Kreissl, P., Reinauer, A., Tischler, I., Košovan, P., Holm, C., 2024. ESPResSo, a versatile open-source software package for simulating soft matter systems, in: Yáñez, M., Boyd, R.J. (Eds.), Comprehensive Computational Chemistry. Elsevier, Oxford, pp. 578–601. doi:10.1016/B978-0-12-821978-2.00103-3.

[5] Weik, F., Weeber, R., Szuttor, K., Breitsprecher, K., de Graaf, J., Kuron, M., Landsgesell, J., Menke, H., Sean, D., Holm, C., 2019. ESPResSo 4.0 – an extensible software package for simulating soft matter systems. European Physical Journal Special Topics 227, 1789–1816. doi:10.1140/epjst/e2019-800186-9.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 130–139

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

Dynamic recognition of the nucleosome core particle by select chromatin factors

Hatice Döşeme[a,b], Tuğçe Uluçay[c], Seyit Kale[a,d,1]

[a]*Izmir Biomedicine and Genome Center, Dokuz Eylül University Health Campus, Balçova, Izmir 35330, Türkiye*

[b]*Izmir International Biomedicine and Genome Institute, Dokuz Eylül University Health Campus, Balçova, Izmir 35330, Türkiye*

[c]*Izmir Institute of Technology, Gülbahçe Campus, Urla, Izmir 35430, Türkiye*

[d]*Faculty of Medicine, Department of Biophysics, Izmir Katip Çelebi University, Çiğli, Izmir 35620, Türkiye*

Abstract

The intricate interactions between the nucleosome core particle and chromatin-binding proteins control essential biological functions templated by DNA. The nucleosome is a symmetrical and disc-shaped nucleoprotein which binds several chromatin factors in a 2:1 stoichiometry. We report computational evidence for a DNA-sequence-driven emergence of asymmetry whereby the nucleosome binding affinities of the chromatin factors are altered on each side even though the protein factors bind chemically equivalent proteinous interfaces of the nucleosome. Furthermore, none of these proteins interact directly with the nucleosomal DNA. Using atomistic molecular dynamics simulations, we surveyed five chromatin factors that are known to bind the nucleosome in a 2:1 stoichiometry. In four factors, we found that the nucleosomal gyre that binds DNA strongly is also more preferred. These factors are Sir3, PRC1, RCC1, and SAGA-DUB. However, a fifth chromatin factor, 53BP1, prefers the gyre with the weaker DNA binding with higher affinity. We argue that this tunability in nucleosome affinity could be related to the function of the chromatin interactors as 53BP1 could prefer loose DNA gyres to execute its DNA repair function.

Keywords: chromatin; nucleosome ; molecular dynamics ; computational biophysics

[1]* Corresponding author. ORCID: https://orcid.org/0000-0001-7903-8543
E-mail address: seyit.kale@ibg.edu.tr

10.1016/j.procs.2025.02.268

1. Introduction

Eukaryotic genome is packed into a highly organized fiber known as chromatin. The nucleosome core particle (NCP), a nucleoprotein complex composed of around 147 base pairs of DNA wrapped around an octameric protein core, constitutes the fundamental repeating architectural unit of chromatin. This core domain is composed of four pairs of highly conserved proteins known as histones H2A, H2B, H3, and H4 [1].

Histone post-translational modifiers and ATP-dependent chromatin remodeling complexes are the two primary groups of protein complexes that are recruited to alter the structural and functional dynamics of chromatin. Many post-translational modifications (PTMs) occur at histone N-terminal tails that protrude from the nucleosomal core. These PTMs include methylation of arginine residues, phosphorylation of serine and threonine residues, and acetylation and methylation of lysine residues. The set of PTMs that each nucleosome possesses determines the range of nucleosomal functions including the recruitment of regulatory proteins, bending and wrapping the nucleosomal and internucleosomal DNA, and sliding or eviction of nucleosome during translation, replication, or repair [1,2]. This context-dependence is crucial for the development of novel medications targeting chromatin [3].

Dysregulation of these DNA-templated mechanisms can lead to serious clinical conditions such as cancer, neurological disorders, metabolic problems, and cardiovascular diseases. Clinical trials are currently being conducted to examine the effectiveness of epigenetic medications, focusing on nucleosome's interactions with chromatin factors. A thorough grasp of the molecular mechanisms behind nucleosome recognition is essential to progress the development of these drugs and gain control over ensuing changes in chromatin state [4,6]. Effector proteins have precise loci to interact with the nucleosomal DNA and histone proteins. Because histone tails can store a range of reversible post-translational modifications, such as acetylation and methylation, they offer dynamic alterations in a context-dependent manner. Proteins that interact with nucleosomes often have several reader domains to recognize these alterations [5,6].

On its two equivalent and opposing histone surfaces, the nucleosome exhibits distinct binding sites, the most prominent of which being the "acidic patch" spanned by several H2A and H2B residues (Fig. 1) [11]. This patch facilitates chromatin compaction by interacting with the histone H4 tail of adjacent nucleosomes. Proteins that bind to the acidic patch typically have a motif known as the "Arginine anchor", a pair of positively charged Arginine residues in proximity of a few Angstroms from each other to engage with the acidic residues of this epitope. The H4-H2B cleft and other areas of the histone core surface are also involved in interactions between chromatin factors and nucleosomes [7]. The linker histone H1 is bound by nucleosomal DNA, which then cooperates with nucleosome-binding domains. Effector proteins have been discovered to interact with many nucleosome epitopes, illustrating the complexity of interactions between nucleosomes and proteins, owing to advancements in the understanding of nucleosome binding. However, the nucleosome poses a major challenge in the field of structural biology because its reconstitution *in vitro* can be demanding in terms of labor and biochemical expertise [8,9].

Structural studies involving chromatin typically make use of a synthetic DNA sequence known as Widom "601" [10]. This sequence binds the NCP more strongly on the left gyre of the core octamer than on the right gyre [12]. In this context, we define "left" and "right" gyres, where the former precedes the latter in the DNA sequence of the sense strand. Recently, we have reported computational evidence that an anti-chromatin antibody imitating chromatin factor binding has a higher nucleosome affinity toward the left gyre than the right one (Fig. 2) [14].

The focus of this work is to understand to what extent the behavior we observed in the antibody-nucleosome interaction can be generalized to physiological chromatin factors. For this, we have selected five protein complexes: Sir3, PRC1, RCC1, SAGA-DUB, and 53BP1. All these factors share three common features: 1) they interact with the nucleosome primarily via histones and in a very limited manner with the nucleosomal DNA, 2) they bind the nucleosome in a 2:1 stoichiometry, i.e., two chromatin factors per nucleosome, and 3) high-resolution atomistic experimental structural models are available for all of them in complex with the nucleosome. We find that four factors bind theleft gyre more strongly, i.e., the gyre that also binds DNA more strongly. Only one, specifically a factor known as 53BP1 prefers the weaker gyre. We argue that this could be implicated in the DNA repair function of this protein for which more readily accessible DNA could be preferred.

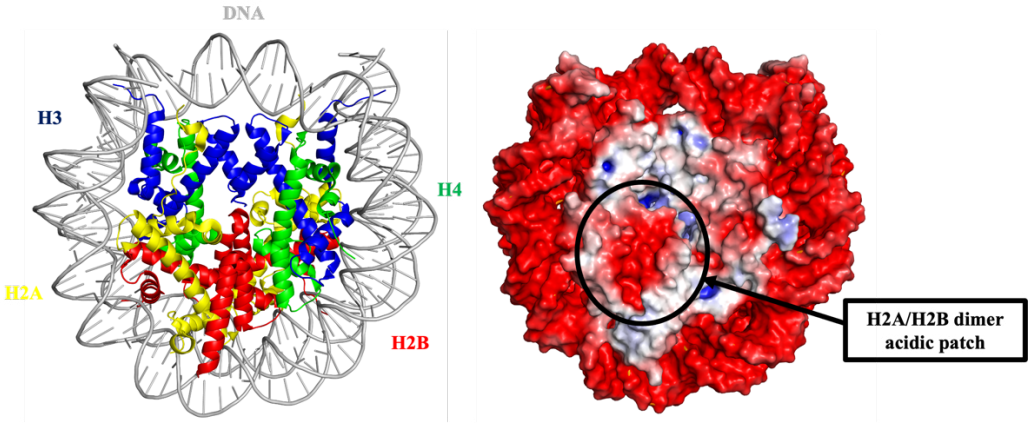


Fig. 1. Left: the structure of the nucleosome (PDB ID: 3LZ0) [19]. Right: charge density of the nucleosome. Negative charges are in red, positives in blue. Acidic patch is indicated.

RCC1

In 2010, a study from the Tan lab revealed the first crystallographic image of a protein domain bound to the nucleosome [13]. Nucleus rearrangement is impacted by RCC1, a protein factor essential for mitosis and directs Ran GTPase recruitment to the nucleosome. The study described the intricate connections between RCC1 and the nucleosome, including the way it binds to the acidic patch through a loop containing the arginine anchor. In contrast to LANA, the first protein whose structure was reported in complex with the nucleosome, the important and conserved connection between arginine residues and the acidic patch was visible thanks to the sufficiently high resolution. This discovery brought to light the complex structure of nucleosomes as interaction platforms of chromatin [13].

PRC1

The polycomb repressive complex 1 (PRC1) is an example of how effector protein orientation in nucleosome complex formation is accurate because it specifies binding modalities and structurally supports certain effector protein functions through synergistic interactions. The PRC1-nucleosome structure was reported from the same lab, i.e., Song Tan's, that also reported the RCC1 complex [13]. PRC1 structure showed how the H2A residue K119 was specifically ubiquitinated and exposed on the nucleosome surface. To achieve specificity, PRC1 employs two distinct binding strategies: it engages with the acidic patch via the arginine anchor of its Ring1B/Bmi1 subunit and nucleosomal DNA via the E2 subunit UbcH5c. These mechanisms determine the specific location of the catalytic core of the ubiquitin transporting E2 to its target H2AK119 [15].

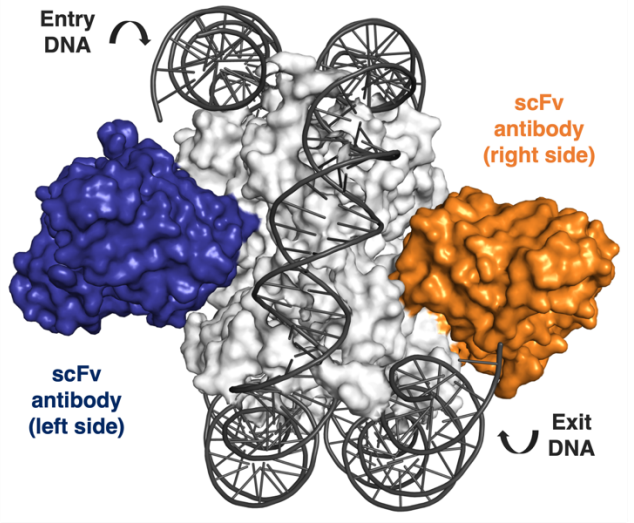


Fig. 2. Binding of two anti-chromatin antibodies (blue and orange) to the nucleosome core particle. Histones are in white, DNA in black. (Adapted from [14]).

SAGA-DUB

The DUB module is a crucial element of the SAGA complex which modulates chromatin to control gene expression. It is responsible for histone H2B deubiquitination, a crucial epigenetic step in gene activation. Studies examined the intricate identification and deubiquitination of H2B by this module using a wide range of structural and biochemical approaches. The structure of the SAGA-DUB in complex with the nucleosome provided detailed insight as to how the DUB module confers selectivity toward the nucleosomal H2B deubiquitination. Moreover, the identification of essential histone H2B residues necessary for deubiquitination was made possible; alterations in these residues significantly reduced the DUB module's activity and chromatin affinity. The structural analysis showed that the C-terminal helix of H2B, which contains ubiquitinated residues, and the active site of the Ubp8 subunit are close 3 to one another, suggesting a direct contact during deubiquitination [16].

Sir3

Sir3 binds nucleosome via its BAH domain, which then dimerize to generate silenced arrays of nucleosomes in yeast cells. The structure of the BAH domain of Sir3 reveals two Arginine residues forming very strong contacts with the acidic patch [17].

53BP1

53BP1 is a complex involved in the repair of double strand breaks. The three-dimensional structure of 53BP1 in complex with the nucleosome suggests critical recognition of not only the nucleosome via the acidic patch, but also the recognition of critical and multiple histone post-translational modifications including the ubiquitylation of H2A at several residues and the methylation of H4. The pronounced multi-valency in the way 53BP1 interacts with DNA and other nucleosome components indicates a highly specialized and well-orchestrated DNA repair response that necessitates the engagement of several nucleosomal components [18].

2. Materials and Methods

We retrieved the atomic coordinates of five nucleosome bound chromatin factors from the Protein Data Bank (PDB). These chromatin binding factors are: 1-53BP1 (PDB ID: 5KGF), 2-SAGA DUB (PDB ID: 4ZUX), 3- PRC1 (PDB ID: 4R8P), 4-RCC1 (PDB ID: 3MVD), and 5-Sir3 (PDB ID: 3TU4). In each of these complexes we extended the nucleosomal DNA ends such that the final sequence contains 149 base pairs. We modelled the DNA using PyMOL,

and wherever necessary we introduced mutations via the Web 3DNA web server such that all complexes contain the same Widom 601 DNA sequence. We used PDB annotations to identify missing loops unresolved due to high flexibility. We repaired these missing loops using SuperLooper2 and PyMOL. We solved each structure in a cubic water box in the presence of 161.5 mM NaCl and 5 mM Mg^{++} ions. The choice of the NaCl concentration reflects the physiological salt concentrations as mimicked in wet-lab experiments using PBS solution. The addition of the low amount of Mg^{++} ion is intended to mimic the mitotic chromatin conditions after breakdown of nuclear wall. We introduced the ions using sub-routines of the GROMACS software, version 2019 [23]. We used the CHARMM36m [20,21] force field together with the OPC water model [22] for the description of the biochemical material and the solvent environment, respectively. The choice of this force field and water combination is justifiable by good experimental agreement in our prior studies using the nucleosome core particle [14,24]. Using again GROMACS version 2019 [23], we gathered production trajectories for every complex. Each system was first energy minimized using a combination of conjugate gradient and steepest descent methods. The systems were then equilibrated in the NVT ensemble for 1 ns 100 K and at 310 K. Production trajectories for each of the five nucleosomal systems were collected in the NPT ensemble using an integration timestep of 2 fs for a total of 500 ns for all. For statistical validation, one chromatin factor system (Sir3) was selected and re-run in 5 replicas, each of 100 ns long, using different initial random number seeds. Another control run was performed, again using the Sir3 system, in the presence of excess Mg^{++} (~35 mM). Constant temperature and pressure are maintained using the velocity-rescaling thermostat [25] and the Parrinello-Rahman barostat [26], respectively. Stabilities of the chromatin systems are analyzed using root mean square distance (RMSD) analyses. Binding free energies are estimated using a molecular dynamics integrated version of the Prodigy model [27]. Relative binding strengths are also estimated by counting the numbers of proximal non-hydrogen atom pairs. Here, a proximal contact is defined by the distance cutoff of 5 Å. All production trajectories are collected at the physiological temperature of 310 K and the atmospheric pressure of 1 atm. All analyses are calculated using a combination of in-house Python scripts, PyMOL, and VMD.

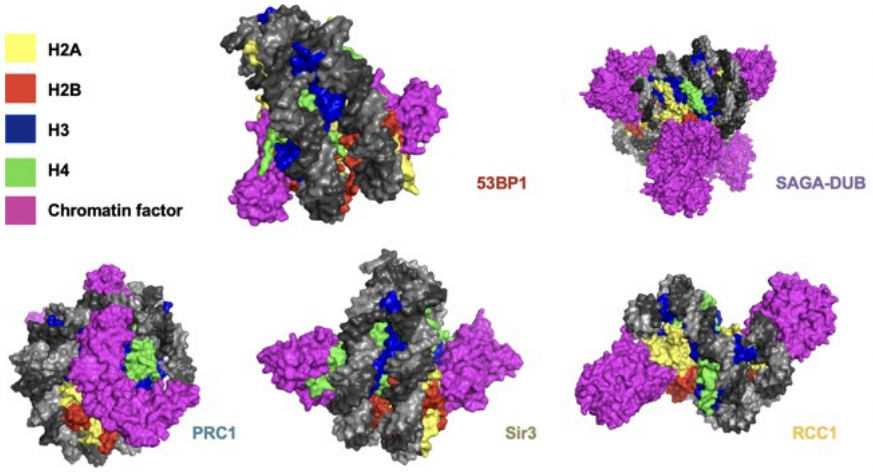


Fig. 3. The structures of five different chromatin factors (magenta) in complex with the NCP.

3. Results

3.1. Four of the five chromatin factors bind the "left" nucleosome gyre more strongly than the "right" gyre

Mean pairwise contact counts and distance analyses between the histones and the chromatin factors suggest that four of the five complexes, i.e., Sir3, PRC1, RCC1, and SAGA-DUB, bind the "left" nucleosomal gyre more strongly than the "right" one (Fig. 4A). Protein-protein binding affinity analysis using Prodigy (Fig. 4B) suggests that this preferential binding exhibits a spectrum of free energy differences, the most prominent difference observed in Sir3. Extension of the molecular dynamics simulations involving Sir3 500 ns pronounces this difference (Fig. 4A and 4B). Also, in Sir3, we found that the preferential binding on the "left" gyre is not affected by the presence or absence of Magnesium ions in solution. RMSD analyses indicate that all nucleosome-chromatin factor complexes are sufficiently stable over the simulation timescales used (Supplementary Figure 1).

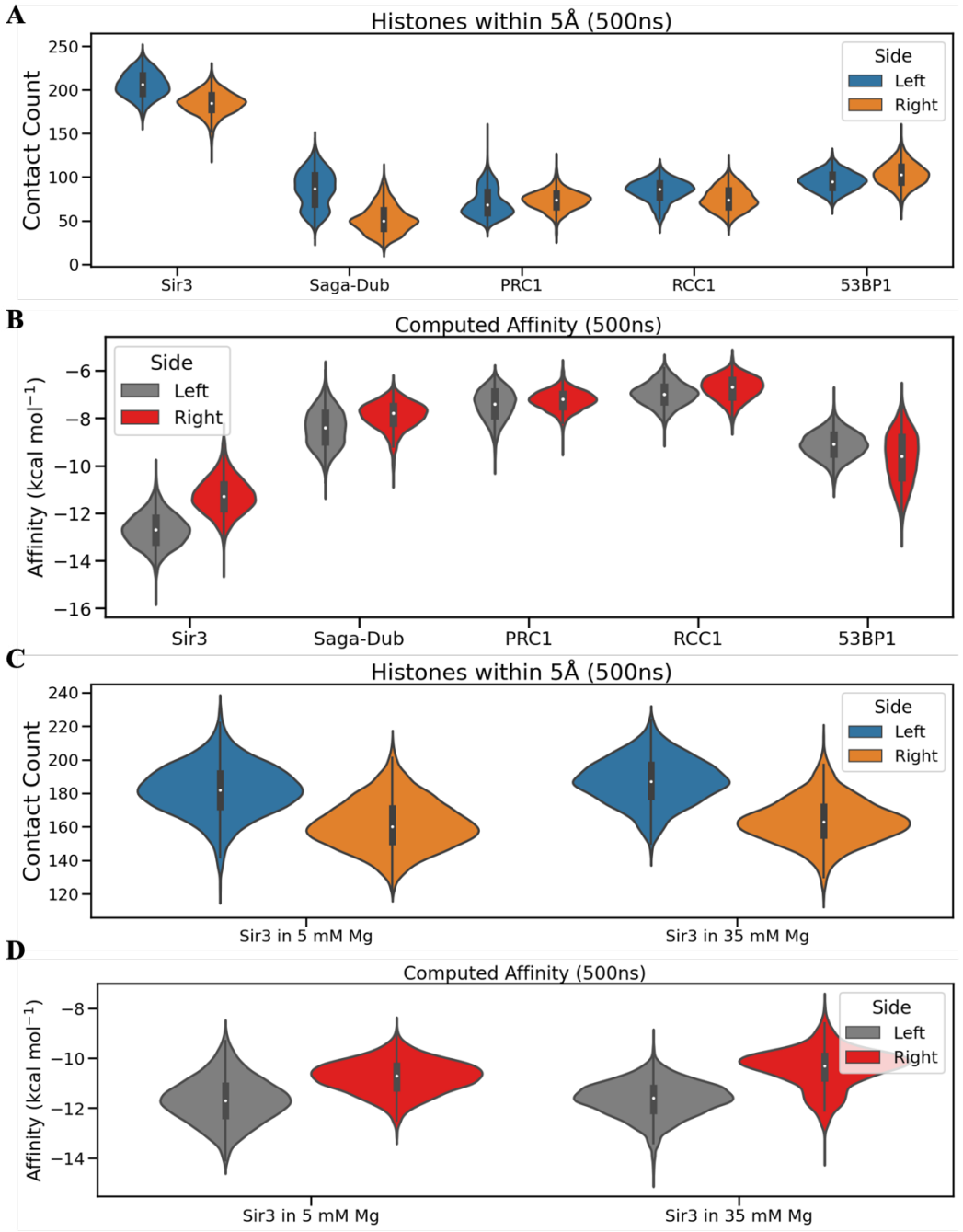


Fig. 4. A and **C**. Average pairwise contacts between the heavy atoms of the chromatin factors and histone octamer. Contacts are defined by a 5 Å distance cutoff between non-hydrogen atom pairs and averaged over snapshots separated by 100 ps intervals. Each simulation is 500 ns long. **B** and **D.** Estimated NCP binding affinities of chromatin factors for the left and the right façades.

3.2. 53BP1 binds the "right" gyre more strongly than the "left" gyre

Mean pairwise contact counts and distance analyses between the histones and the chromatin factors suggest that, unlike the other four chromatin factors, 53BP1 exhibits a stronger binding affinity toward the "right" nucleosome gyre (Fig. 4A, 4C and Table 1). Prodigy analysis indicates that this preference leads to free energy difference of nearly 1.5 kcal/mol (Fig. 4B and 4D, Table 2). Upon binding the nucleosome, the loop containing the Arginine finger motif residue R1627 undergoes local secondary structural changes toward a more stable helical structure on the "right" side but not on the "left" (Fig. 5A-B). This Arginine is also in closer proximity and more stably situated to and with respect to acidic patch residues H2A E61 and E92 on the "right" side but not on the "left" (Fig. 5C).

Table 1. Average pairwise contacts between heavy atoms of the chromatin factors and the NCP histones. Means and standard deviations are indicated. PDB IDs are in parentheses.

Chromatin Factor	Left	Right
53BP1 (PDB ID: 5KGF)	95±10.9	103±13.8
SAGA-DUB (PDB ID: 4ZUX)	86±21.3	52±15.9
Sir3 (PDB ID: 3TU4)	206±14.7	185±13.4
RCC1 (PDB ID: 3MVD)	84±13.3	75±12.7
PRC1 (PDB ID: 4R8P)	74±16.5	71±11.4

Table 2. Average binding affinities (in kcal/mol) between heavy atoms of the chromatin factors and the NCP histones. Means and standard deviations are indicated. PDB IDs are in parentheses.

Chromatin Factor	Left	Right
53BP1 (PDB ID: 5KGF)	−9.12±0.57	−9.68±1.08
SAGA-DUB (PDB ID: 4ZUX)	−8.42±0.8	-7.9±0.6
Sir3 (PDB ID: 3TU4)	−12.71±0.77	−11.26±0.69
RCC1 (PDB ID: 3MVD)	−6.98±0.5	−6.73±0.4
PRC1 (PDB ID: 4R8P)	−7.43±0.6	−7.24±0.42

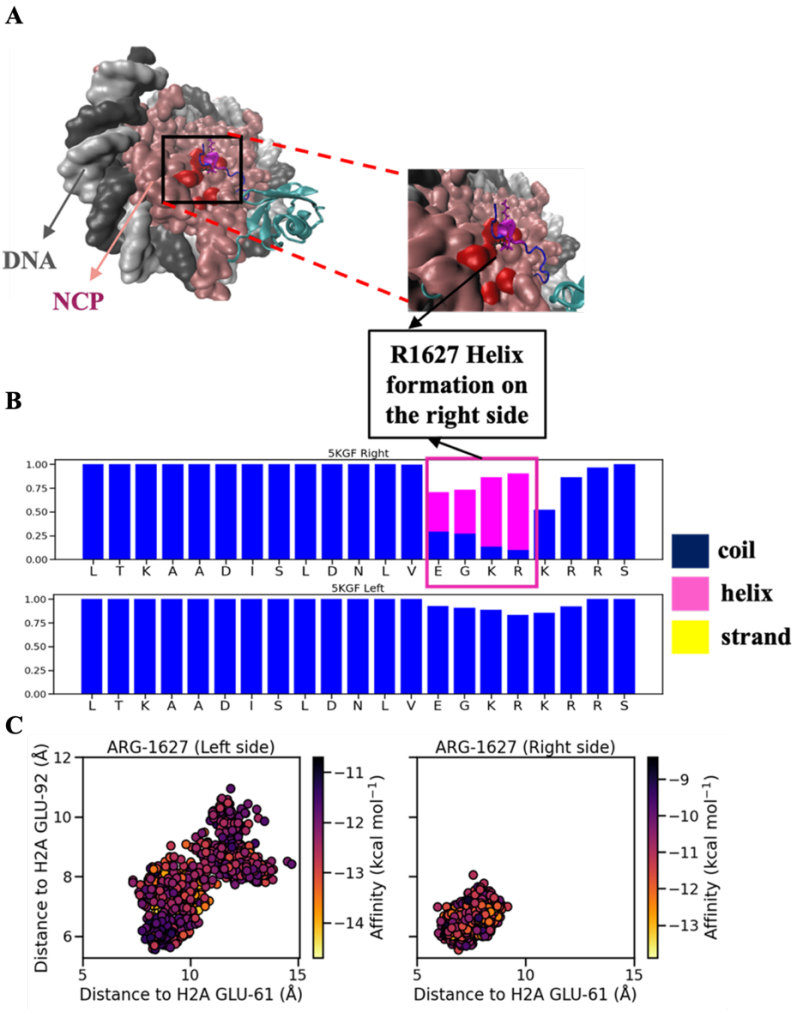


Fig. 5. Preferential binding of 53BP1 to the right nucleosomal façade is associated with local structural alterations. **A.** Zoom-in view of 53BP1 on the right NCP façade. Note the local alpha-helix formation around the main epitope R1627 (annotated in black). **B.** Local secondary structure propensities of the left and right copies of 53BP1. Focus is on the region that contains key Arginine interactors. **C.** Distances between the mass centers of R1627 to two critical acidic patch residues, E61 (abscissae) and E92 (ordinates).

3.3. Replicate runs validate the ergodicity of the simulations

Finally, we replicated one of the chromatin factor simulations, specifically that of Sir3, five times each run being 100 ns long, to sum to the comparable duration of 500 ns as performed for all systems. Averaging over all five short replicas, we confirm that Widom 601 nucleosome bound Sir3 still binds preferentially the 'left' gyre more strongly than the 'right' one. As shown in Figure 6, both contact analyses and the estimated binding free energies support the original preference observed in the 500 ns long simulation. This argument supports the fact that the production trajectories are ergodic and sufficiently well-equilibrated.

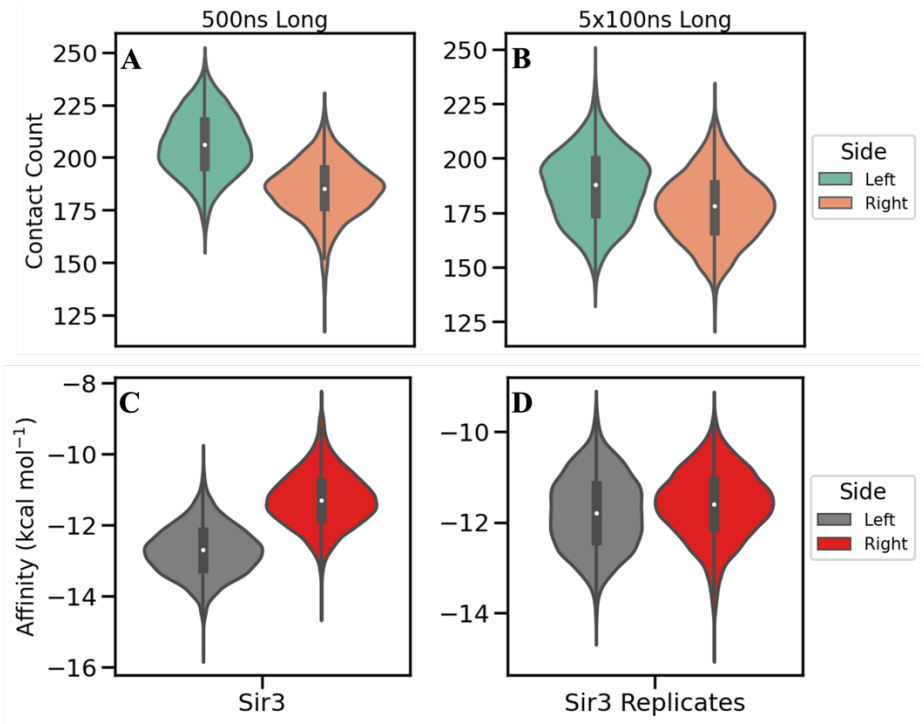


Fig. 6. A and **B.** Average pairwise contacts between the heavy atoms of the Sir3 replicates and histone octamer. Contacts are defined by a 5 Å cutoff and averaged over snapshots separated by 100 ps intervals with total 500 ns. **C** and **D.** Estimated NCP binding affinities of Sir3 replicates for the left and the right façades. Plots in **A** nd **C** are over a 500 ns long simulation. Plots in **B** and **D** are from five 100 ns long simulation.

4. Discussion, Future Remarks and Conclusion

We report here computational evidence that the sequence of the nucleosomal DNA can modulate the affinity of chromatin factors to the nucleosome. To study this, we undertook sub-microsecond atomistic molecular dynamics simulations involving chromatin factor nucleosome complexes where each chromatin factor binds the nucleosome in pairs and on chemically identical façades of the nucleosome. We find that four out of the five chromatin factors prefer the gyre that also binds the DNA more strongly. The fifth chromatin factor, 53BP1, prefers the other gyre, i.e., the gyre that binds the DNA weakly. Detailed molecular investigation suggests that the strongly binding copy of this factor undergoes local structural changes pronouncing and stabilizing this preference. We hypothesize that the weak-side preference of 53BP1 not only emphasizes the diversity of chromatin interactors in terms of structure, but also in terms of function. This factor is implicated in the repair of double strand breaks for which a more accessible nucleosomal DNA could be preferred.

Acknowledgements

This work was supported by The Scientific and Technological Research Council of Türkiye (TÜBİTAK), Turkey and the European Molecular Biology Organization (EMBO) Installation Grant no 5056 (awarded to S.K.). Molecular dynamics trajectories are generated and analyzed using the high-performance computing resources provided by the Izmir Biomedicine and Genome Center, TÜBİTAK ULAKBIM High Performance and Grid Computing Center (TRUBA resources), theDutch national supercomputer Snellius, and the MareNostrum5 of the Barcelone Supercomputing Center. Access to Snellius was provided by the PRACE Distributed European Computing Initiative DECI-17 grant EPICENTROMERE (awarded to S.K.). Access to MareNostrum5 was provided by EuroHPC JU grant EHPC-AI-2024A01-088 (awarded to S.K.). Financial support by TÜBİTAK does not mean that the content of the publication has been approved in a scientific sense by TÜBİTAK.

Data availability

Supplementary Figure 1 and the data utilized in this study, including molecular dynamics input, topology and parameter files can be accessed on Zenodo (under: https://doi.org/10.5281/zenodo.14055957, doi: 10.5281/zenodo.14055957).

Supplementary Figure 1: RMSDs of nucleosomal DNA **(A),** core histones **(B),** left bound chromatin factors **(C),** and right bound chromatin factors **(D)** over the 500 ns long trajectories.

References

[1] Sokolova, V., Sarkar, S., and Tan, D. (2023). Histone variants and chromatin structure, update of advances. Preprint at Elsevier B.V., 10.1016/j.csbj.2022.12.002 10.1016/j.csbj.2022.12.002.

[2] Zhou, K., Gaullier, G., and Luger, K. (2019). Nucleosome structure and dynamics are coming of age. Preprint at Nature Publishing Group, 10.1038/s41594-018-0166-x 10.1038/s41594-018-0166-x.

[3] McGinty, R.K., and Tan, S. (2016). Recognition of the nucleosome by chromatin factors and enzymes. Preprint at Elsevier Ltd, 10.1016/j.sbi.2015.11.014 10.1016/j.sbi.2015.11.014.

[4] Widom, J., & Klug, A. (1985). Structure of the 300A chromatin filament: X-ray diffraction from oriented samples. Cell, 43(1), 207–213. https://doi.org/10.1016/0092-8674(85)90025-x.

[5] Luger, K., Dechassa, M.L., and Tremethick, D.J. (2012). New insights into nucleosome and chromatin structure: An ordered state or a disordered affair? Preprint, 10.1038/nrm3382 10.1038/nrm3382.

[6] Horn, V., and van Ingen, H. (2020). Recognition of Nucleosomes by Chromatin Factors: Lessons from Data-Driven Docking-Based Structures of Nucleosome-Protein Complexes. In Chromatin and Epigenetics (IntechOpen). 10.5772/intechopen.81016.

[7] Shaytan, A.K., Landsman, D., and Panchenko, A.R. (2015). Nucleosome adaptability conferred by sequence and structural variations in histone H2A-H2B dimers. Preprint at Elsevier Ltd, 10.1016/j.sbi.2015.02.004 10.1016/j.sbi.2015.02.004.

[8] Deák, G., Wapenaar, H., Sandoval, G., Chen, R., Taylor, M. R. D., Burdett, H., Watson, J. A., Tuijtel, M. W., Webb, S., & Wilson, M. D. (2023). Histone divergence in trypanosomes results in unique alterations to nucleosome structure. Nucleic acids research, 51(15), 7882–7899. https://doi.org/10.1093/nar/gkad577.

[9] Thoma, F., Koller, T., & Klug, A. (1979). Involvement of histone H1 in the organization of the nucleosome and of the salt dependent super structures of chromatin. The Journal of cell biology, 83(2 Pt 1), 403–427. https://doi.org/10.1083/jcb.83.2.403.

[10] Lowary, P. T., & Widom, J. (1998). New DNA sequence rules for high affinity binding to histone octamer and sequence directed nucleosome positioning. Journal of molecular biology, 276(1), 19–42. https://doi.org/10.1006/jmbi.1997.1494.

[11] Kalashnikova, A.A., Porter-Goff, M.E., Muthurajan, U.M., Luger, K., and Hansen, J.C. (2013). The role of the nucleosome acidic patch in modulating higher order chromatin structure. Preprint at Royal Society, 10.1098/rsif.2012.1022 10.1098/rsif.2012.1022.

[12] Ngo, T.T.M., Zhang, Q., Zhou, R., Yodh, J.G., and Ha, T. (2015). Asymmetric unwrapping of nucleosomes under tension directed by DNA local flexibility. Cell 160, 1135–1144. 10.1016/j.cell.2015.02.001.

[13] Makde, R. D., England, J. R., Yennawar, H. P., & Tan, S. (2010). Structure of RCC1 chromatin factor bound to the nucleosome core particle. Nature, 467(7315), 562–566. https://doi.org/10.1038/nature09321.

[14] Doğan, D., Arslan, M., Uluçay, T., Kalyoncu, S., Dimitrov, S., and Kale, S. (2021). CENP-A Nucleosome is a Sensitive Allosteric Scaffold for DNA and Chromatin Factors. J Mol Biol 433. 10.1016/j.jmb.2020.166789.

[15] McGinty, R.K., Henrici, R.C., and Tan, S. (2014). Crystal structure of the PRC1 ubiquitylation module bound to the nucleosome. Nature 514, 591–596. 10.1038/nature13890.

[16] Morgan, M. T., Haj-Yahya, M., Ringel, A. E., Bandi, P., Brik, A., & Wolberger, C. (2016). Structural basis for histone H2B deubiquitinationby the SAGA DUB module. Science (New York,N.Y.), 351(6274),725–728. https://doi.org/10.1126/science.aac568.

[17] Lancaster, K.M., Roemelt, M., Ettenhuber, P., Hu, Y., Ribbe, M.W., Neese, F., Bergmann, U., and DeBeer, S. (2011). X-ray emission spectroscopy evidences a central carbon in the nitrogenase iron-molybdenum cofactor. Science (1979) 334,974–977. 10.1126/science.1206445.

[18] Wilson, M.D., Benlekbir, S., Fradet-Turcotte, A., Sherker, A., Julien, J.P., McEwan, A., Noordermeer, S.M., Sicheri, F., Rubinstein, J.L., and Durocher, D. (2016). The structural basis of modified nucleosome recognition by 53BP1. Nature 536, 100–103. 10.1038/nature18951.

[19] Vasudevan, D., Chua, E. Y. D., & Davey, C. A. (2010). Crystal structures of nucleosome core particles containing the '601' strong positioning sequence. Journal of molecular biology, 403(1), 1–10. https://doi.org/10.1016/j.jmb.2010.08.039.

[20] Best, R.B., Zhu, X., Shim, J., Lopes, P.E.M., Mittal, J., Feig, M., et al., (2012). Optimization of the additive CHARMM allatom protein forcefield

targeting improved sampling of the backbone phi, psi and side-chain chi(1) and chi(2) dihedral angles. J. Chem Theory Comput., 8, 3257 - 3273.
[21] Huang, J., Rauscher, S., Nawrocki, G., Ran, T., Feig, M., de Groot, B.L., et al., (2017). CHARMM36m: an improved force field for folded and intrinsically disordered proteins. Nature Methods, 14, 71–73.
[22] Izadi, S., Anandakrishnan, R., Onufriev, A.V., (2014). Building water models: a different approach. J. Phys. Chem. Letters, 5, 3863–3871.
[23] Van der Spoel, D., Lindahl, E., Hess, B., Groenhof, G., Mark, A.E., Berendsen, H.J.C., (2005). GROMACS: fast, flexible, and free. J. Comput. Chem., 26, 1701–1718.
[24] Boopathi R., Danev R., Khoshouei M., Kale S., Nahata S., Ramos L., Angelov D., Dimitrov S., Hamiche A., Petosa C., Bednar J., (2020). Phase-plate cryo-EM structure of the Widom 601 CENP-A nucleosome core particle reveals differential flexibility of the DNA ends, *Nucleic Acids Research*, Volume 48, Issue 10, 04 June 2020, Pages 5735–5748, https://doi.org/10.1093/nar/gkaa246.
[25] Bussi G, Donadio D, Parrinello M. Canonical sampling through velocity rescaling. *J Chem Phys*. 2007;126(1):014101. doi:10.1063/1.2408420
[26] Parrinello, M., Rahman, A., (1981). Polymorphic transitions in single-crystals - a new molecular-dynamics method. J.Appl. Phys., 52, 7182–7190.
[27] Xue LC, Rodrigues JP, Kastritis PL, Bonvin AM, Vangone A. PRODIGY: a web server for predicting the binding affinity of protein-protein complexes. *Bioinformatics*. 2016;32(23):3676-3678. doi:10.1093/bioinformatics/btw5.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 140–149

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC User Day

Towards a European HPC/AI ecosystem: a community-driven report

Petr Taborsky[a,*], Iacopo Colonnelli[b], Krzysztof Kurowski[c], Rakesh Sarma[d], Niels Henrik Pontoppidan[e], Branislav Jansík[f], Nicki Skafte Detlefsen[a], Jens Egholm Pedersen[g], Rasmus Larsen[h], Lars Kai Hansen[a]

[a]*Technical University of Denmark, Anker Engelunds Vej 1, Bygning 101A, 2800 Kongens Lyngby, Denmark*
[b]*University of Torino, Computer Science Dept., Corso Svizzera 185, 10149, Torino, Italy*
[c]*Poznań Supercomputing and Networking Center, Jana Pawła II 10, 61-139 Poznań, Poland*
[d]*Forschungszentrum Jülich GmbH, Jülich Supercomputing Centre, Wilhelm-Johnen-Straße, 52425 Jülich, Germany*
[e]*Eriksholm Research Centre, Rørtangvej 20, 3070 Snekkersten, Denmark*
[f]*IT4Innovations, VSB – Technical University of Ostrava, Ostrava, Czech Republic*
[g]*KTH Royal Institute of Technology, Brinellvägen 8, 114 28 Stockholm, Sweden*
[h]*Alexandra Instituttet, Rued Langgaards Vej 7, 2300 Copenaghen, Denmark*

Abstract

The rapid advancements in AI and Machine Learning necessitate a robust computational infrastructure to support cutting-edge research and industrial applications. From the academic and industrial AI community perspective, voiced in the recent ELISE project, the European AI platform is recommended to center around the EuroHPC growing ecosystem. It should be user-driven, easily accessible, powerful, and compliant with European regulations. AI-optimized and dedicated supercomputers for the European AI community are also coming, in addition to upgrading partitions of existing EuroHPC systems to 'AI enabled' stage. Related calls have been initiated in September 2024. Further, conventional EuroHPC systems are suggested to be extended with quantum computing, edge AI, and neuromorphic computing to cater to AI models deployed on network edge devices and sustainability in the long run. The challenges are presented in three case studies, ranging from training Transformers on HPC to LLMs trained federally across three different Euro HPC systems to recent results on hybrid classical-quantum application. This paper concludes with case studies results-informed next steps believed to benefit AI practitioners and the broader AI community.

Keywords: Artificial Intelligence; High-Performance Computing; HPC; ELISE; ELLIS; EuroHPC Joint Undertaking; Quantum Computing; Federated Learning

* Corresponding author.
E-mail address: ptab@dtu.dk

1877-0509
10.1016/j.procs.2025.02.269

1. Introduction

The European Union is well positioned to accelerate its Artificial Intelligence (AI) and Machine Learning (ML) research by leveraging advancements in High-Performance Computing (HPC) and Quantum Computing (QC), with prospects of a more sustainable Neuromorphic platform in the long run. Among other activities, the European R&D community, represented by ELLIS Community and the European Commission, initiated the ELISE project to investigate and recommend to AI practitioners and researchers the options for a joint pan-European AI R&D platform for the near future. This paper presents excerpts from the recent ELISE project report, comprising inputs from 100+ R&D teams, hardware providers (e.g., NVIDIA), and industry partners (e.g., OTICON) that are the most relevant to the EuroHPC community. In particular, it focuses on EuroHPC systems.

HPC systems have been pivotal in scientific research, providing the computational power for complex simulations and data analysis. With the rise of AI and ML, a broader R&D community (e.g., ELLIS) is interested in utilizing HPC systems for these fields. This report examines the feasibility and implications of using HPC systems like MeluXina, LUMI, or the upcoming JUPITER exascale supercomputer for AI/ML research in Europe.

The article analyzes two large-scale case studies taken from the AI community, i.e., a large neural network training on a single EuroHPC facility (Sec. 2), a federated training of a Large Language Model (LLM) across three EuroHPC systems (Sec. 3), and one hybrid classical-quantum case study (Sec. 4), demonstrating some challenges that industrial and scientific AI practitioners have to face and proposing a way forward.

2. Training Large Neural Networks on HPC system

Traditional HPC applications are large-scale scientific simulations from diverse domains, like life sciences, weather forecasting, quantum chemistry, and physics [32]. Typically, these applications rely on floating-point operations (double precision, via CPUs and, increasingly, GPUs) to run predetermined models that generate data, often organized in a few large data files. On the other hand, AI applications rely on given data to fit a model, i.e., data produce a model using some ML probabilistic algorithm. The forward-backward pass in the backpropagation algorithm [38] typically leverages hardware accelerators (e.g., GPUs or TPUs) to perform a set of fast and possibly noisy matrix multiplications. In the meantime, a small set of CPUs performs preprocessing and transfer tasks on lots of little data chunks called mini-batches. Given that, AI and simulations are quite opposite approaches that naturally prefer different settings of the underlying systems. From the AI perspective, the following is beneficial:

- *Heterogeneous accelerators*. While traditional HPC systems rely on CPUs and possibly GPUs, primarily integrating accelerators (e.g., GPUs or TPUs) is essential for optimizing AI/ML workloads [18, 7].
- *Mixed precision*. While simulation and numerical differential equation solvers used in HPC workflows benefit high precision, training neural networks may prefer more noisy iterations to fewer precise ones [14, 33, 42].
- *Parallel processing*. AI/ML tasks benefit model and data parallelisms, which typically require hardware-specific and topology-specific code optimizations to be effectively implemented on HPC systems [3, 7].
- *Large accelerator memory*. Training AI/ML models involves handling large datasets, necessitating substantial memory resources. Moreover, it is beneficial when memory is 'close' to a processing unit [26].
- *Distributed file system and bandwidth*. The distributed file system and interconnect in existing HPC facilities may not be optimized for AI/ML tasks. For example, transferring thousands of small data chunks across processing units may not suit the file system used, and I/O often becomes the training time bottleneck [7].

2.1. Why to Use HPC Systems for AI/ML?

Despite HPC systems not being necessarily AI-centric, it is still pragmatic and often the best choice of industrial or academic researchers to run AI tasks on them, all things considered:

- *Enhanced computational power*. HPC systems offer unparalleled computational capabilities essential for training large AI/ML models. For instance, LUMI, the fastest supercomputer in Europe, can perform at 379.7 petaflops. So, even suboptimized code runs faster than on smaller GPU clusters, see Fig. 1.

- *Scalability*. HPC systems are made to handle large-scale computations, making them suitable for AI/ML tasks that require significant parallel processing.
- *Existing infrastructure*. Leveraging existing HPC infrastructure can reduce the need for additional investments in new hardware, facilitating a more cost-effective approach to AI/ML research.
- *Energy Efficiency*. Modern HPC systems like JUPITER are designed with energy efficiency in mind, which is crucial for the sustainability of large-scale AI/ML operations.
- *Data privacy compliance*. EuroHPC systems often provide GDPR compliance 'by default', and the upcoming AI Factories will advise on even higher standards, including the AI Act.

2.2. Training Large Neural Network on MeluXina EuroHPC

The previous paragraph gave a general reason for using HPCs for AI. Next, we dig deeper into the AI challenges on EuroHPC systems, where things are further complicated by the diversity of these systems and the diverse nature of AI architectures used for AI tasks (e.g., transformers vs. xLSTMs). In 2021, a dedicated and detailed study [18] executed a set of Scientific Machine Learning (SciML) tasks across several HPC architectures in the US, analyzing the challenges of applying AI tasks on HPC. According to the study: "[. . .] Input and activation sizes limit batching and will ultimately mandate the exploitation of model parallelism; AI-optimized GPUs running SciML demand more PCIe, NVMe, and Lustre bandwidth than currently provided; Local NVMe used to feed SciML training workloads does not provide clear performance benefits at scale and should be evaluated against centralized fabric-attached storage or strong scaling with static partitioning of training data; Data scientists should structure models to exploit unused resources to reduce time per epoch. [. . .]".

2.3. Experiments, Results & Recommendations

To corroborate the findings above on EuroHPC systems and to obtain hands-on user experience, the ELISE project team executed the following minimalistic experiments in 2022. In addition, they also provided insight into how large an efficiency gap of a naive straightforward approach is: take a 'laptop' code and run it on better (HPC) hardware. The obtained results are summarized in Fig. 1.

Experimental setup. **HW**: European 'Tier 0' MeluXina EuroHPC[1], 'Tier 1' Danish life sciences supercomputer Computerome HPC, and 'Tier 2' University GPU cluster node. **Model**: Wave2Vec2.0 [1]. It represents a large enough transformer-based model ($\simeq 10^8$ parameters) to impose a challenge for training on a 'standard' Tier-2 GPU cluster. **Task**: To fine-tune a pre-trained (self-supervised on LibriSpeech dataset) speech-to-text model on the PolyAI/minds14 dataset. **Data**: PolyAI/minds14, the transcribed noisy audio files with intents extracted from a commercial system in the e-banking domain, associated with spoken examples in 14 diverse language varieties, see [13], also available at: https://paperswithcode.com/dataset/minds14. **Code**: Wave2Vec2 on HF, a Hugging Face implementation of the Wave2Vec2.0 model offering reproducibility.

Results & Recommendations. The two major challenges identified in the experiment can be summarized as follows. They demonstrate that while an infamous 'number of GPUs' may be a bottleneck for the largest models with optimized code, tackling challenges of many R&D AI tasks is more tangible and often within the reach of AI practitioners.

GPU memory (HBM) bottlenecks
HPC systems offer extensive resources, yet they show that it is not straightforward to use them efficiently. For instance, larger GPU DRAM allows practitioners to load and process larger mini-batches per GPU. Our experiments and the

[1] MeluXina EuroHPC provides NVIDIA Ampere A100 GPUs with 40GB HBM and AMD EPYC CPU. Tier 1 scratch Storage uses a Lustre file system (400 G/s). Its Accelerator Module offers 200 GPU nodes, each featuring 2 AMD Rome CPUs (32 cores @ 2.35 GHz - 128HT cores total) and 4 NVIDIA A100-40 GPUs. These nodes have 512 GB of RAM, a local SSD of 1.92 TB, and 2 HDRcards connecting them to the InfiniBand network. Additional supporting experiments on Computerome HPC (DK) using NVIDIA V100 (Volta) GPUs and 'Tier 2' University GPU cluster node with NVIDIA Titan X GPU and 12GB DRAM were conducted.

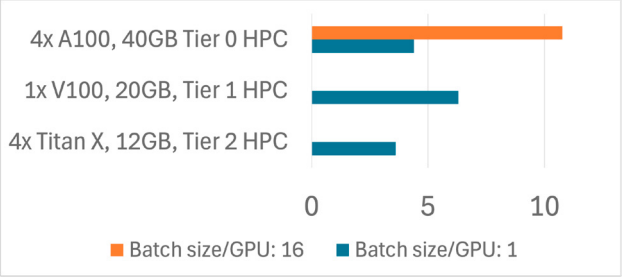


Fig. 1. **Migrating AI code across HPCs without optimizing it for underlying HW is inefficient and sub-optimal**. Throughput (number of processed training samples per second per GPU) of vanilla 'off-the-shelf' Hugging Face code during fine-tuning of the pre-trained Wave2vec2.0 transformer model are shown. Fine-tuning was done on three different HPC systems, ranging from local GPU cluster (Tier-2) to EU level HPC (Tier-0). The code was not optimized for any of the three HPC systems used. Results show that migrating code from the local university GPU cluster (Tier 2 HPC) to (Tier 0 HPC) without a change, i.e., keeping the same number of GPUs, only provides negligible gains. Significant gains are indicated (orange vs. blue) when batch size per GPU is increased from 1 to 16. However, despite using approximately half of GPU DRAM in the case of batch 16, the GPUs were heavily underutilized during training, operating at 6% of their peak performance on average. While the total gain from increasing batch size 16x is less than 3x in this case, it also leaves room for improvement in a range of 1 order of magnitude. **Throughput metric (x-axis)** = the number of training samples processed per second by a single GPU (an average over GPUs, 1 hour of training, and 5 runs).

study mentioned above [18] demonstrated that significant gains can be obtained by increasing batch size per GPU (Fig. 1, orange vs. blue). However, in the case of Wave2Vec2.0 fine-tuning, with all data samples ($\simeq$ 100 MBs) loaded on every GPU, we experienced GPUs running idle most of the time, waiting for gradients synchronization. This overhead was due to the Wave2Vec2.0 transformer with more than 10^8 parameters being large enough to make GPU-GPU gradient reduction take more than ten times longer than doing forward-backward passes[2]. This situation is common for fine-tuning pre-trained models for specific applications and training models from scratch. The good news is that code optimization may lead to a more than an order of magnitude reduction in training time.

Due to many models utilizing a 'funnel' architecture, the largest bottleneck will be the few widest layers. Thus, layer parallelization may not reduce the bottleneck, and intra-layer parallelization or other techniques may be required [18, 45]. Instead, when performance is bound by file system or interconnection bandwidth, it is possible to increase model complexity (e.g., making it deeper), balance hyperparameters such as learning rate and batch size, and choose a more advanced optimizer (e.g., see PyTorch [35]) to reduce the total number of iterations. Generally, a creative approach is advised. EuroHPC JU not only offers training and courses, e.g., on LUMI, but also provides so-called 'Development Access' and, recently, 'AI and Data Intensive Access' to EuroHPC facilities for these purposes.

I/O (Storage & Interconnect)

Handling and processing large datasets efficiently requires robust data management systems. Current HPC systems may not be fully optimized for the specific needs of AI/ML, such as the rapid access and processing of large volumes of unstructured data that often come in many small files scattered over the file system [21]. Besides high GPU memory and GPU-to-GPU bandwidth within one compute node, a node-to-node interconnect plays an immense role depending on AI application. For instance, training ResNet of 50 layers and 25×10^6 parameters using 32-bit precision on mini-batch of 32 on *A*100 GPU or higher would fully utilize throughput of $\approx$ 900 GB/s or higher, while the inter-node fabric currently available at HPC centers are, e.g., $\approx$ 50GB/s per each AMD MI250x GPU module (LUMI-G) at the time of writing[3]. While data parallelization may be hindered by node-to-node communication bottlenecks, model parallelization may also have limited effects due to many models utilizing a 'funnel' architecture, as discussed above.

While EuroHPC infrastructure updates are in progress, models will likely keep increasing their size. Thus, AI practitioners may consider other techniques, such as a gradient accumulation on GPUs over several batches (often used to overcome batch size limits imposed by available GPU memory) combined with an adjusted learning rate [40] before the all-reduce operation that concludes the round. This idea can be further extended to asynchronous training.

[2] To eliminate other communication costs (e.g., inter-node data transfers), we only used 4 GPUs on the same compute node for fine-tuning.
[3] Latest generation of NVLink(Switch) provides up to 1.8TB/s theoretical throughput

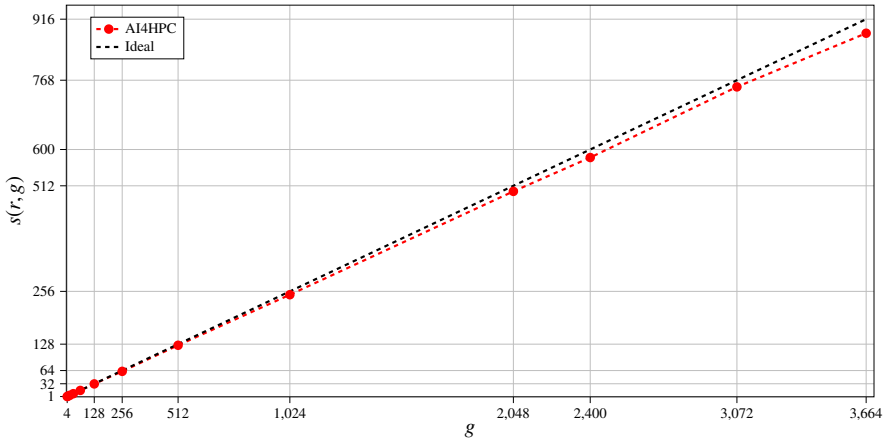


Fig. 2. AI4HPC scaling on the JUWELS system was demonstrated on 3,664 NVIDIA A100 GPUs with 96% efficiency. Here, a U-Net model consisting of 52 million parameters is trained with a synthetic dataset using the *Horovod* backend. The black line shows the ideal speed-up. g denotes the number of GPUs, while $s(r, g)$ is the speed-up.

Such approaches have common traits with federated learning, which also allows to tackle communication bottlenecks by reducing the size and frequency of inter-node communications (see Sec. 3).

2.4. Towards an HPC-optimized library for AI

The increasing focus on including accelerators in the EuroHPC hosting sites requires the next generation of AI codes to be *HPC-ready*. The challenge for the upcoming exascale systems is potentially even higher, as they should demonstrate good scalability with increasing communication overhead under more node-to-node interconnects.

In order to address these challenges and to better utilize HPC systems, the AI4HPC library was developed in the EU-funded CoE RAISE project. AI4HPC is targeted to allow users to seamlessly and efficiently train their AI models in a distributed setting on the EU HPC infrastructure while including various code optimizations to improve training performance. The library has already been built and tested across various European HPC centers. At the time of writing this manuscript, the tested sites are JUWELS at Jülich Supercomputing Center (JSC), JURECA at JSC, DEEP-EST at JSC, LUMI at IT Center for Science (CSC), CTE-AMD at Barcelona Supercomputing Center (BSC), Leonardo at CINECA, and JEDI at JSC, the first module of the upcoming exascale supercomputer JUPITER.

AI4HPC includes data manipulation routines, the collection of various ML architectures, optimization routines for efficient training, the Hyperparameter Optimization (HPO) module, and monitoring and performance benchmarking tools. The library includes multiple distributed training backends, which users can exploit for their training workloads. The integrated backends are PyTorch DDP, Horovod, HeAT and DeepSpeed. In terms of large-scale performance measurement, the scalability of the library has been demonstrated (shown in Fig. 2) on the JUWELS system with up to 96% efficiency on 3,664 NVIDIA A100 GPUs.

As mentioned in the challenges above, apart from performance requirements, submitting jobs on HPC systems requires building the correct environment to enable execution. This task involves understanding the modules and their continuously updated versions. For the AI4HPC library, a proper execution environment is automatically created for each configured HPC center. Integration of new centers in the library is straightforward. The CoE RAISE project also developed another tool, LAMEC, which specifically manages the environment and generates job submission scripts for multiple HPC centers in a simplified GUI. LUMI, Vega, Karolina, and Leonardo are already integrated into LAMEC.

3. Cross-facility Deep Learning

If squeezing every last drop of computing performance from larger and larger HPC centers is pivotal for sustaining the scales of modern AI, exploring cross-facility federated science [10] is fundamental for many practical reasons:

- *Reaching even larger scales*. If a complex application can be decomposed into (almost) embarrassingly parallel modules, these modules can be efficiently offloaded to different HPC facilities, increasing concurrency and reducing time-to-solution. Typical AutoML tasks, e.g., hyperparameter optimization and neural architecture search, fall into this category [17].
- *Enhancing resource utilization*. During peak load or maintenance periods, jobs can linger in a pending state for a long time. The introduction of a federated meta-scheduler [23] or a decentralized scheduling plane [28] that distributes jobs across multiple HPC facilities would benefit both users, reducing time-to-solution, and systems, reducing idle times in underloaded machines.
- *Exploiting data locality*. Since the advent of physics-informed neural networks [37], Deep Learning has often been coupled with large-scale scientific simulations to improve accuracy and reduce time-to-solution [27, 20]. In situ data processing [4], i.e., analysing data where they are generated, is a promising approach to avoid network communications and reduce inference latency [9]. A federated HPC ecosystem would allow scientists to set up largely distributed training processes, allocating model replicas near data sources.
- *Ensuring data privacy*. In some cases, data cannot be moved from their original, trusted location for privacy and security reasons. Still, cross-silo federated learning approaches [30] that train models on multiple datasets without disclosing them can benefit all involved parties. With the advent of Deep Learning in sensitive sectors like healthcare [11] and finance [34], supporting this kind of scenario is becoming crucial.
- *Guaranteeing fairness*. Foundation models [8] with trillions of parameters necessitate an entire exascale data center for training from scratch [29]. However, prolonged exclusive resource allocations to a single European initiative can undermine the principle of fair resource usage, disadvantaging smaller, national projects. Distributing computation among different centers can ensure a more equitable European HPC ecosystem.

The challenges of cross-facility science have been studied for various application domains, from astrophysics to genomics [43], to molecular dynamics [36], to Deep Learning [5], and orchestrating cross-site experiments on pairs of European HPC facilities has already proven feasible. In [36], the authors run a large-scale plasma simulation analysis using a sparse grid combination technique to mitigate the curse of dimensionality. The experiment ran on top of two facilities, i.e., HAWK at HLRS (Stuttgart, DE) and SuperMUC at LRZ (Garching, DE). In [5], a LLaMAv2-7B model is trained from scratch using a federated learning approach to reduce the inter-site communication overhead. Again, the experiment involved two facilities: Leonardo at CINECA (Bologna, IT) and Karolina at IT4I (Ostrava, CZ). Both cases involve small federations of homogeneous (x86-64 CPUs, NVIDIA GPUs) and relatively close HPC facilities.

3.1. Experiments, Results & Recommendations

In order to assess the EuroHPC readiness for large-scale cross-facility experiments, we ran an extended version of the federated learning experiment described in [5]. This time, we tried a federated training of a LLaMAv3 8B model on top of three different HPC sites: Leonardo in Bologna, Italy (Intel CPUs, NVIDIA GPUs), LUMI in Kajaani, Finland (AMD CPUs, AMD GPUs), and MeluXina in Luxembourg (AMD CPUs, NVIDIA GPUs). Compared to the previous attempts, this is a far more challenging setting, with three geographically far machines mounting heterogeneous accelerators. Fig. 3 compares the land surface covered by the three European cross-facility experiments, showing how the one discussed here is by far the largest European cross-HPC experiment described in the literature. This work focuses on the challenges encountered while setting up the federation and orchestrating the application's life cycle. A detailed description of the technical aspects behind this experiment and the obtained results will be provided elsewhere.

Results & Recommendations. Standard distributed training processes can be modelled as Bulk Synchronous Programming (BSP) workloads [44], where each superstep locally computes the forward pass and backpropagation on each node, communicates the gradients to all other nodes, and globally synchronizes the process to compute the result-

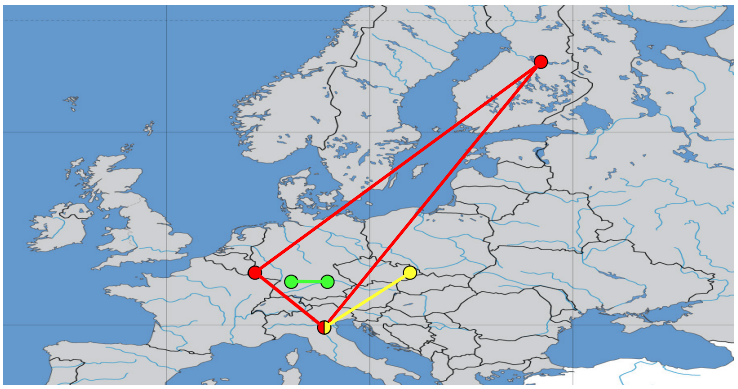


Fig. 3. Map of the European cross-facility experiments: in green the plasma simulation analysis on SuperMUC and HAWK [36], in yellow the federated learning experiment on Leonardo and Karolina [5], and in red the federation among Leonardo, MeluXina, and LUMI HPC facilities. The last configuration requires data to be exchanged across a total of about 5.200 km. The total land surface covered is about 680.000 km^2, more or less the 16% of the EU surface area.

ing gradient [2]. Federated learning workloads minimize cross-cluster data transfers by only requiring synchronization and communication of model weights at the end of each FL round [5]. However, since BSP workloads contain global barriers that are very sensitive to stragglers, performance fluctuations in different HPC facilities significantly undermine the overall performance of the training workload. **Challenge**: Reduce cross-site performance fluctuations, which depend on several factors:

- *Heterogeneous hardware and software stacks*. Deep learning workloads heavily rely on hardware accelerators, especially GPUs. Even if different accelerators can scale very well for a wide range of dataset sizes, the absolute duration of a training step varies significantly between different hardware, with AMD GPUs on LUMI being up to 6 times slower than NVIDIA GPUs on Leonardo and MeluXina. This difference is due to the different hardware and heterogeneous software stacks available in different computing facilities. **Solution**: Provide unified AI software collections on different facilities optimised for the underlying hardware accelerators. Better support for HPC package managers (e.g., Spack [12]) and software containers (e.g., Singularity [25]) could also help in achieving (performance) portability.
- *Unbalanced queuing times*. Different HPC facilities reach peak load during different periods, making queuing times highly variable among sites. Plus, large HPC facilities are complex systems that still need frequent maintenance periods, making it difficult to find all involved sites up and running simultaneously. **Solution**: Implement a cross-facility orchestration plane. Several meta-scheduling [23] or distributed scheduling [28] algorithms can be derived from grid computing frameworks and combined with a vendor-agnostic compatibility layer [15] to avoid lock-in. Plus, cross-facility workflow management systems like StreamFlow [6], PyCOMPSs [41], or JAWS [22] can help orchestrate cross-HPC workloads and offer non-functional requirements out of the box, e.g., portability, reproducibility, fault tolerance, provenance tracking, and secure efficient data transfers.
- *Slow and unstable inter-site network*. Relying on the public Internet for heavy data transfers leads to suboptimal and highly variable transfer times that slow down the communication phase of BSP supersteps. With huge, trillion-parameter models, this overhead can become significant. **Solution**: Procure a dedicated high-speed interconnection plane among EuroHPC facilities.

Another challenge we faced during the federation setup was the extreme heterogeneity in HPC resource access. The PRACE program already allows cross-facility resource requests. However, the way different sites manage resource grants varies significantly in several aspects: monthly vs bulk allocation of compute hours, core hours vs node hours grants, and significantly different amounts of computing hours granted by different facilities. **Challenge**: Revise the resource allocation program with HPC federations as first-class citizens, allowing users to submit cross-facility experiment proposals.

4. Quantum/Classical AI in EuroQCS-Poland

Until recently, there has been 'classical' digital computing and quantum computing world. Despite its impressive theoretical yet elusive benefits, quantum computing (QC) has been approached by academics and industry primarily using quantum simulators. Only recently, hybrid-quantum computing, which combines classical and quantum computation within one task, has been shown by experiments and benchmarks to be practically helpful. In practice, 'hybrid' computation is a back-and-forth collaboration approach where different aspects of a problem are passed between the quantum and classical tools best suited for each stage, thus accelerating the overall process and delivering a performance boost.

EuroHPC is also working to integrate quantum computing into its broader supercomputing ecosystem, funding research projects and building infrastructure to support this hybrid approach. The idea is to develop quantum-ready HPC systems where different quantum computers can be efficiently integrated with classical supercomputers. This initiative offers a novel interpretation of quantum computers as accelerator platforms in genuine HPC environments in Europe. The foreseen integration will require essential R&D developments towards a hybrid software stack managing both HPC and QC workloads. This effort positions Europe at the forefront of developing quantum-accelerated HPC systems to address next-generation computational challenges.

Owned by the EuroHPC JU, the first quantum system EuroQCS-Poland will be hosted in 2025 at the Poznan Supercomputing and Networking Center (PSNC) and integrated into the local HPC infrastructure, allowing for remote access via the co-located supercomputer connected to the PIONIER NREN and Pan-European GEANT networks. The quantum system will be a digital, gate-based quantum computer based on trapped ions offering 20-plus physical qubits delivered by AQT [16].

To select the best quantum system for EuroQCS-Poland, a set of application benchmarks, including AI/ML algorithms, have been developed to evaluate the overall performance of available European quantum computing technologies [24]. The well-known MNIST dataset, containing images of handwritten digits collected for image classification, has been used as input. The Quantum Support Vector Machine (QSVM) was proposed as AI/ML benchmark as it is an algorithm that applies a quantum kernel to a Support Vector Machine (SVM) for classification and regression [39]. In a nutshell, SVM finds an optimal hyperplane between classes, but when data is not linearly separable, a kernel function maps it to a higher-dimensional space. The proposed QSVM use case was successfully tested on a trapped-ion quantum computer to create a feature map, potentially identifying complex patterns that classical methods could not identify.

5. Conclusion and future work

The rapid advancements in AI and Machine Learning necessitate a robust computational infrastructure to support cutting-edge research and industrial applications on a European scale. The EuroHPC systems are well suited for AI advancements, yet they still require substantial effort from AI practitioners to exploit their resources efficiently. The ELISE Horizon Europe project's recent report recommends that the EuroHPC systems move further toward a community-driven R&D ecosystem that is easily accessible and powerful and has links to quantum, edge AI, and neuromorphic computing. While edge computing is an increasingly important deployment platform for many AI applications, such as hearing devices [19], neuromorphic computing constitutes an emerging and energy-efficient alternative to von Neumann architectures [31].

Extending the existing EuroHPC systems with the upcoming *hybrid classical-quantum systems* and an AI-optimized supercomputer, announced recently with AI Factories, will address several challenges with porting AI tasks to HPC. In addition, by connecting to over 100 international research teams and platform providers over the last three years, the ELISE project formulated a set of recommendations for the R&D community to advance AI-driven innovation. These include supporting the *active participation* of the AI R&D community in designing the platform, e.g., within the newly set up EuroHPC User Forum, where the AI-focused R&D community is currently underrepresented.

Moreover, the AI community and the EuroHPC JU should join forces to support a *platform-agnostic* format of AI models. This effort will enable a smooth transition and *federation* of training, research and models across EuroHPC systems, whose current diversity may become a crucial bottleneck for R&D in Europe in the future and should be addressed, for instance, adopting some of the solutions suggested in this work.

Acknowledgements

This project was funded by ELISE EU Grant agreement ID: 951847. We would like to acknowledge the participation of over a hundred anonymous research teams, from academia, government, and private sectors, incl. banking, automotive, telecommunications, etc., who took part in our workshops, surveys, and work groups including following: MLOPs Summer school, DTU; On-line Crash Course for LLMs on EuroHPCs; MLOPs Course for Industry, DTU; European AI Platform workshop, European Networks of AI Excellence and the Center of Excellence in Exascale Computing, CoE RAISE; AI on LUMI HPC; VISION, RIAG, INFRAG meeting; First HPC User Day; AI Platform of the Future, Workshop No.1 & 2; BEST summer school. Your willingness to share your experiences and perspectives provided essential data and nuanced understanding that have been crucial. The same goes to Nvidia, Dr. rer. nat. Maria Athelogou and Frédéric Parienté. Special thanks to the EuroHPC JU sites for providing access to HPC systems, which made this project possible. And to European Commission and EuroHPC JU representatives, Mladen Skelin, Jan Hückmann, Dr. Daniel Opalka, Dr. Lilit Axner, Miguel Rubio and many others for inviting the ELISE team to coordination and discussion meetings, making the project influential and actionable. We are also grateful to the administrative and technical staff for their continuous assistance and to our peer reviewers for their constructive feedback.

References

[1] Baevski, A., Zhou, H., Mohamed, A., Auli, M., 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. arXiv:2006.11477.

[2] Ben-Nun, T., Hoefler, T., 2019. Demystifying parallel and distributed deep learning: An in-depth concurrency analysis. ACM Comput. Surv. 52, 65:1–65:43. doi:10.1145/3320060.

[3] Brewer, W., Behm, G., Scheinine, A., Parsons, B., Emeneker, W., Trevino, R.P., 2020. Inference benchmarking on hpc systems, in: 2020 IEEE High Performance Extreme Computing Conference (HPEC), IEEE. pp. 1–9.

[4] Choi, J.Y., Chang, C., Dominski, J., Klasky, S., Merlo, G., Suchyta, E., Ainsworth, M., Allen, B., Cappello, F., Churchill, M., Davis, P.E., Di, S., Eisenhauer, G., Ethier, S., Foster, I.T., Geveci, B., Guo, H., Huck, K.A., Jenko, F., Kim, M., Kress, J., Ku, S., Liu, Q., Logan, J., Malony, A.D., Mehta, K., Moreland, K., Munson, T.S., Parashar, M., Peterka, T., Podhorszki, N., Pugmire, D., Tugluk, O., Wang, R., Whitney, B., Wolf, M., Wood, C., 2018. Coupling exascale multiphysics applications: Methods and lessons learned, in: 14th IEEE International Conference on e-Science, e-Science 2018, Amsterdam, The Netherlands, IEEE Computer Society. pp. 442–452. doi:10.1109/ESCIENCE.2018.00133.

[5] Colonnelli, I., Birke, R., Malenza, G., Mittone, G., Mulone, A., Galjaard, J., Chen, L.Y., Bassini, S., Scipione, G., Martinovič, J., Vondrák, V., Aldinucci, M., 2024. Cross-facility federated learning. Procedia Computer Science 240, 3—-12. doi:10.1016/j.procs.2024.07.003.

[6] Colonnelli, I., Cantalupo, B., Merelli, I., Aldinucci, M., 2021. Streamflow: Cross-breeding cloud with HPC. IEEE Trans. Emerg. Top. Comput. 9, 1723–1737. doi:10.1109/TETC.2020.3019202.

[7] De Sensi, D., Pichetti, L., Vella, F., De Matteis, T., Ren, Z., Fusco, L., Turisini, M., Cesarini, D., Lust, K., Trivedi, A., et al., 2024. Exploring gpu-to-gpu communication: Insights into supercomputer interconnects. arXiv preprint arXiv:2408.14090 .

[8] Devlin, J., Chang, M., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT, Association for Computational Linguistics. pp. 4171–4186. doi:10.18653/V1/N19-1423.

[9] Do, T.M.A., Pottier, L., Caíno-Lores, S., da Silva, R.F., Cuendet, M.A., Weinstein, H., Estrada, T., Taufer, M., Deelman, E., 2021. A lightweight method for evaluating *in situ* workflow efficiency. J. Comput. Sci. 48, 101259. doi:10.1016/J.JOCS.2020.101259.

[10] Enders, B., Bard, D., Snavely, C., Gerhardt, L., Lee, J., Totzke, B., Antypas, K., Byna, S., Cheema, R., Cholia, S., Day, M.R., Gaur, A., Greiner, A., Groves, T.L., Kiran, M., Koziol, Q., Rowland, K., Samuel, C., Selvarajan, A., Sim, A., Skinner, D., Thomas, R.C., Torok, G., 2020. Cross-facility science with the superfacility project at LBNL, in: 2nd IEEE/ACM Annual Workshop on Extreme-scale Experiment-in-the-Loop Computing, XLOOP@SC 2020, Atlanta, GA, USA, IEEE. pp. 1–7. doi:10.1109/XLOOP51963.2020.00006.

[11] Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., Dean, J., 2019. A guide to deep learning in healthcare. Nature Medicine 25, 24–29. doi:10.1038/s41591-018-0316-z.

[12] Gamblin, T., LeGendre, M.P., Collette, M.R., Lee, G.L., Moody, A., de Supinski, B.R., Futral, S., 2015. The spack package manager: bringing order to HPC software chaos, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2015, Austin, TX, USA, ACM. pp. 40:1–40:12. doi:10.1145/2807591.2807623.

[13] Gerz, D., Su, P.H., Kusztos, R., Mondal, A., Lis, M., Singhal, E., Mrkšić, N., Wen, T.H., Vulić, I., 2021. Multilingual and cross-lingual intent detection from spoken data. arXiv:2104.08524.

[14] Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016. Deep learning. MIT press Cambridge.

[15] Hategan-Marandiuc, M., Merzky, A., Collier, N.T., Maheshwari, K., Ozik, J., Turilli, M., Wilke, A., Wozniak, J.M., Chard, K., Foster, I.T., da Silva, R.F., Jha, S., Laney, D.E., 2023. PSI/J: A portable interface for submitting, monitoring, and managing jobs, in: 19th IEEE International Conference on e-Science, e-Science 2023, Limassol, Cyprus, IEEE. pp. 1–10. doi:10.1109/E-SCIENCE58273.2023.10254912.

[16] Humble, T.S., McCaskey, A., Lyakh, D.I., Gowrishankar, M., Frisch, A., Monz, T., 2021. Quantum computers for high-performance computing. IEEE Micro 41, 15–23. doi:10.1109/MM.2021.3099140.

[17] Hutter, F., Kotthoff, L., Vanschoren, J. (Eds.), 2019. Automated Machine Learning - Methods, Systems, Challenges. The Springer Series on Challenges in Machine Learning, Springer. doi:10.1007/978-3-030-05318-5.

[18] Ibrahim, K.Z., Nguyen, T., Nam, H.A., Bhimji, W., Farrell, S., Oliker, L., Rowan, M., Wright, N.J., Williams, S., 2021. Architectural requirements for deep learning workloads in hpc environments, in: 2021 International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS), IEEE. pp. 7–17.
[19] Iftikhar, S., Gill, S.S., Song, C., Xu, M., Aslanpour, M.S., Toosi, A.N., Du, J., Wu, H., Ghosh, S., Chowdhury, D., et al., 2023. Ai-based fog and edge computing: A systematic review, taxonomy and future directions. Internet of Things 21, 100674.
[20] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S.A.A., Ballard, A.J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A.W., Kavukcuoglu, K., Kohli, P., Hassabis, D., 2021. Highly accurate protein structure prediction with alphafold. Nature 596, 583–589. doi:10.1038/s41586-021-03819-2.
[21] Khan, A., Paul, A.K., Zimmer, C., Oral, S., Dash, S., Atchley, S., Wang, F., 2022. Hvac: Removing i/o bottleneck for large-scale deep learning applications, in: 2022 IEEE International Conference on Cluster Computing, pp. 324–335. doi:10.1109/CLUSTER51413.2022.00044.
[22] Kirton, E., Foster, B., Froula, J.L., Sul, S.J., Trong, S., Kollmer, A., Melara, M., Rowland, K., Rath, G., USDOE, 2020. Joint genome institute analysis workflow service (jaws) v2.0. doi:10.11578/dc.20210617.3.
[23] Kurowski, K., Nabrzyski, J., Oleksiak, A., Weglarz, J., 2008. A multicriteria approach to two-level hierarchy scheduling in grids. J. Sched. 11, 371–379. doi:10.1007/S10951-008-0058-8.
[24] Kurowski, K., Rydlichowski, P., Wojciechowski, K., Pecyna, T., Slysz, M., 2023. Application performance benchmarks for quantum computers. CoRR abs/2310.13637. doi:10.48550/ARXIV.2310.13637, arXiv:2310.13637.
[25] Kurtzer, G.M., Sochat, V., Bauer, M.W., 2017. Singularity: Scientific containers for mobility of compute. PLOS ONE 12, 1–20. doi:10.1371/journal.pone.0177459.
[26] Kwon, Y., Rhu, M., 2018. A case for memory-centric hpc system architecture for training deep neural networks. IEEE Computer Architecture Letters 17, 134–138. doi:10.1109/LCA.2018.2823302.
[27] Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., Ravuri, S., Ewalds, T., Eaton-Rosen, Z., Hu, W., Merose, A., Hoyer, S., Holland, G., Vinyals, O., Stott, J., Pritzel, A., Mohamed, S., Battaglia, P., 2023. Learning skillful medium-range global weather forecasting. Science 382, 1416–1421. doi:10.1126/science.adi2336.
[28] Lu, K., Subrata, R., Zomaya, A.Y., 2007. On the performance-driven load distribution for heterogeneous computational grids. J. Comput. Syst. Sci. 73, 1191–1206. doi:10.1016/J.JCSS.2007.02.007.
[29] Ma, Z., He, J., Qiu, J., Cao, H., Wang, Y., et al., 2022. BaGuaLu: targeting brain scale pretrained models with over 37 million cores, in: ACM PPoPP, pp. 192–204. doi:10.1145/3503221.3508417.
[30] McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A., 2017. Communication-efficient learning of deep networks from decentralized data, in: Singh, A., Zhu, X.J. (Eds.), Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, Fort Lauderdale, FL, USA, PMLR. pp. 1273–1282.
[31] Mead, C., 2023. Neuromorphic engineering: In memory of misha mahowald. Neural Computation 35, 343–383.
[32] NICULESCU, V., 2019. High performance computing in big data analytics. Applied Medical Informatics .
[33] Noh, H., You, T., Mun, J., Han, B., 2017. Regularizing deep neural networks by noise: Its interpretation and optimization. Advances in Neural Information Processing Systems 30.
[34] Özbayoglu, A.M., Gudelek, M.U., Sezer, O.B., 2020. Deep learning for financial applications : A survey. Appl. Soft Comput. 93, 106384. doi:10.1016/J.ASOC.2020.106384.
[35] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in pytorch .
[36] Pollinger, T., Craen, A.V., Niethammer, C., Breyer, M., Pflüger, D., 2023. Leveraging the compute power of two HPC systems for higher-dimensional grid-based simulations with the widely-distributed sparse grid combination technique, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2023, Denver, CO, USA, ACM. pp. 84:1–84:14. doi:10.1145/3581784.3607036.
[37] Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707. doi:10.1016/J.JCP.2018.10.045.
[38] Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323, 533–536. doi:10.1038/323533a0.
[39] Slysz, M., Kurowski, K., Waligóra, G., Węglarz, J., 2023. Exploring the capabilities of quantum support vector machines for image classification on the mnist benchmark, in: Computational Science – ICCS 2023, Springer Nature Switzerland, Cham. pp. 193–200.
[40] Smith, S.L., Kindermans, P., Ying, C., Le, Q.V., 2018. Don't decay the learning rate, increase the batch size, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net.
[41] Tejedor, E., Becerra, Y., Alomar, G., Queralt, A., Badia, R.M., Torres, J., Cortes, T., Labarta, J., 2017. PyCOMPSs: Parallel computational workflows in python. Int. J. High Perform. Comput. Appl. 31, 66–82. doi:10.1177/1094342015594678.
[42] Thomas, V., Pedregosa, F., Merriënboer, B., Manzagol, P.A., Bengio, Y., Le Roux, N., 2020. On the interplay between noise and curvature and its effect on optimization and generalization, in: International Conference on Artificial Intelligence and Statistics, PMLR. pp. 3503–3513.
[43] Tyler, N., Jr., R.A.K., Bard, D., Nugent, P., 2022. Cross-facility workflows: Case studies with active experiments, in: IEEE/ACM Workshop on Workflows in Support of Large-Scale Science, WORKS 2022, Dallas, TX, USA, IEEE. pp. 68–75. doi:10.1109/WORKS56498.2022.00014.
[44] Valiant, L.G., 1990. A bridging model for parallel computation. Commun. ACM 33, 103–111. doi:10.1145/79173.79181.
[45] Zeng, Z., Liu, C., Tang, Z., Chang, W., Li, K., 2021. Training acceleration for deep neural networks: A hybrid parallelization strategy, in: 2021 58th ACM/IEEE Design Automation Conference (DAC), pp. 1165–1170. doi:10.1109/DAC18074.2021.9586300.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 150–158

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

Relativistic MHD simulations of merging and collapsing stars

Agnieszka Janiuk[a,*], Ireneusz Janiuk[b], Dominika Ł. Król[c], Piotr Plonka[d], Gerardo Urrutia[a], Joseph Saji[a]

[a] *Center for Theoretical Physics, Polish Academy of Sciences, Al. Lotników 32/46, 02-668, Warsaw, Poland*
[b] *Self-employed*
[c] *Astronomical Observatory of the Jagiellonian University, Orla 171, 30-244 Kraków, Poland*
[d] *Astronomical Observatory of the Warsaw University, Al. Ujazdowskie 4, 00-478 Warsaw, Poland*

Abstract

We explore the compact object mergers and massive star collapse leading to bright transients in high energy range. Electromagnetic gamma ray bursts and radioactivity of kilonovae may be accompanied by gravitational waves, To simulate collapsing stars and compact merger remnants., we use numerical scheme HARM (High Accuracy Relativistic MHD). Current code branches developed by our team are HARM-EOS, with tabulated equation of state, and HARM-SELFG, with evolving Kerr metric and self-gravity. We focus on configurations of magnetic field important for powering electromagnetic jets. We also explore the nuclear heating in the post-merger ejecta and calculate synthetic lightcurves to be compared with observed kilonovae. Code performance and grid settlements have been tested and optimized on LUMI during development grant, and now it is ready to be launched in high-resolution 3D setups. These endeavors require substantial resources, available at the LUMI supercomputer.

Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day

Keywords: relativistic astrophysics; compact stars; black holes; magnetohydrodynamics

1. Scientific rationale

We investigate the fate of a collapsing stellar core, which is the final state of evolution of a massive, rotating star. Such stars explode as type I b/c supernovae, observed in association with long gamma ray bursts (GRBs) [1]. The core of the star is potentially forming a black hole, which is embedded in a dense, rotating, and possibly highly magnetized envelope. We study the process of collapse using General Relativistic MHD simulations, and we account for the growth of the black hole mass and its spin, as well as related evolution of the spacetime metric. We found before [2, 3] that some particular configurations of the initial black hole spin, the content of angular momentum in the stellar core, and the magnetic field strength, are favored for producing a bright electromagnetic transient (GRB). On the other hand, the typical configurations do not lead to a transient electromagnetic explosion and end up in a direct

* Corresponding author.
E-mail address: agnes@cft.edu.pl

1877-0509
Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day
10.1016/j.procs.2025.02.270

collapse to a black hole. The event may be accompanied by some residual variability induced by changing accretion rate.

We recently confirmed the important role of self-gravity in the stellar core, that modifies distribution of density and pressure in the collapsar [3]. We quantified the relative strength of the interfacial instabilities, such as Self-Gravity Interfacial (SGI) instability and Rayleigh-Taylor (RT), which may account for the production of an in-homogeneous structure. The physical models of these instabilities may change substantially, if the 3D effects, and non-axisymmetric modes are considered. The new simulations will investigate these 3D effects in detail in the future project.

The second part of our research is related to evolution of thermodynamic quantities and modeling of outflows from post-merger accretion disk in short GRBs. The physical evolution is sensitive to the implemented equation of state (EOS). In the code we have implemented a module with tabulated equation of state, known as Helmholtz [4]. Inversion of quantities conserved by the GR MHD scheme is done over the tabulated energy, pressure and electron fraction (i.e. chemical composition). Outflows driven by magnetic fields can be very neutron rich, hence the initial chemical composition affects creation of heavy unstable isotopes of elements beyond Iron and Nickel. These outflows will then radioactively decay and power the emissions called a kilonova.

The third aim of the project is to embed the complex chemical structure of dense hot matter in the collapsar model. The tabluated EOS, used instead of the previously assumed polytrope equation, is expected to change the physics of collapsar, In the case of entropy generation via shocks and at instability interfaces. Hence the work on efficient implementation of nuclear EOS was needed, and we will elaborate on our previous implementations [5, 6].

2. Numerical Code

We develop our own implementation of HARM, in various branches. The acronym stands for High Accuracy Relativistic Magneto-hydrodynamics [7] and solves the set of hyperbolic equations of GR MHD by the finite volume method, with the constrained transport and classical HLL Riemann solver. The flow evolution is described by equations of the continuity, energy-momentum conservation and magnetic induction, in the GRMHD scheme

$$\nabla_\mu(\rho u^\mu) = 0, \qquad \nabla_\mu(T^{\mu\nu}) = 0, \qquad \nabla_\mu(u^\nu b^\mu - u^\mu b^\nu) = 0 \tag{1}$$

$$T^{\mu\nu} = T^{\mu\nu}_{gas} + T^{\mu\nu}_{EM} \tag{2}$$

$$T^{\mu\nu}_{gas} = \rho h u^\mu u^\nu + p g^{\mu\nu} = (\rho + u + p) u^\mu u^\nu + p g^{\mu\nu}, \qquad T^{\mu\nu}_{EM} = b^2 u^\mu u^\nu + \frac{1}{2} b^2 g^{\mu\nu} - b^\mu b^\nu, b^\mu = u^*_\nu F^{\mu\nu} \tag{3}$$

Here, the stress-energy tensor, $T^{\mu\nu}$, is comprised of the gas and electromagnetic terms, u^μ is the four-velocity of the gas, u is the internal energy, ρ is the density, p is the pressure, h is fluid's enthalpy, and b^μ is the magnetic four vector. $F^{\mu\nu}$ is the Faraday tensor and ${}^*F$ is its dual. Note that in a force-free approximation, we have electric field satisfying condition $E_\nu = u^\nu F^{\mu\nu} = 0$.

In HARM, the unit convention is adopted such that $G = c = M = 1$. Thus the black hole mass will scale the simulations (e.g. gravitational radius $r_g = GM_{BH}/c^2$ or time $t_g = GM_{BH}/c^3$, are our units of length and time in the code).

The scheme solves numerically the above system of equations in the general form

$$\partial_t U(P) = -\partial_i F^i(P) + S(P) \tag{4}$$

where $U(P)$ is vector of conserved variables, F^i are fluxes through cell boundaries, and $S(P)$ are source terms.

HARM-EOS branch is designed to incorporate detailed microphysics in the astrophysical plasmas, essential for compact stars environment. Equation of State (EOS) of an ideal gas with analytic form was used in the original code, while our group at CTP PAS developed the code version which uses a relativistic, partially-degenerate Fermi gas EOS, computed numerically and tabulated. We started from the two-parameter EOS, with $p(\rho, T)$ and $u(\rho, T)$ implemented in [5]. Here the EOS depended only on density ρ and temperature T. Notice that we need to rescale the quantitities to physical units, such as gcm^{-3}, from dimensionless HARM code quantities.

The most recent advancement of the code is self-consistently using a general three-parameter EOS, and the neutrino leakage scheme. We use the EOS adapted from Helmholtz tables, with $p(\rho, T, Y_e)$ and $u(\rho, T, Y_e)$, where Y_e is the

electron fraction. The latter is defined as the ratio between electron (or, equivalently, proton) and baryon number densities, and gives a measure of the matter neutronisation (at high gas densities, some protons transform to neutrons, via weak interactions). This newly implemented EOS is usable for a wide range of densities and temperatures. The evolving electron fraction is giving an additional source term to the energy equation.

Noticeably, the conserved variables are not the same as those used for the EOS calculation. Instead, in every time-step, the code has to make (typically twice) an inversion between the conserved and primitive variables. There exist a number of inversion schemes, based on specific transformations between these five independent variables (their explicit form can be found e.g. in [8]). The neutrino leakage scheme computes a gray optical depth estimate along radial rays for electron neutrinos, antineutrinos, and heavy-lepton neutrinos (nux), and then computes local energy and lepton number loss terms. The scheme is based on equations provided by [9]. The source code of the leakage scheme has been downloaded from *https://stellarcollapse.org* and we implemented it in our version of the GR MHD code, HARM-EOS.

Another branch of the code is called HARM-SELFG. The module incorporates self-gravity force, which acts in addition to the gravitational potential of a compact core (black hole) onto which the material is accreting. The self-gravity force is non-negligible in the system when the surrounding material is more massive than the central object. This condition is satisfied in case of collapsing massive star. In such star, once the newly born black hole is created, both the mass and angular momentum accreted onto the event horizon. They contribute to the growth of black hole and hence the update the relevant Kerr metric coefficients is needed [10]. In addition, the metric is further modified by the perturbation acting in the region above the horizon due to the self-gravity force, felt by the gas at a given distance from the horizon. These perturbative terms are calculated from the stress–energy tensor. Therefore, in addition to the two equations governing the growth of black hole mass and spin via the mass and angular momentum transfer through the horizon, as given below, [2], we now add perturbative terms to mass and angular momentum, computed at every radius above the event horizon.

$$\dot{M}_{BH} = \int d\theta d\phi \ \sqrt{-g}\, T^r{}_t, \qquad \dot{J} = \int d\theta d\phi \ \sqrt{-g}\, T^r{}_\phi, \tag{5}$$

$$\delta M_{BH}(t,r) = 2\pi \int_{r_{hor}}^{r} T^r_t \sqrt{-g} d\theta, \qquad \delta J(t,r) = 2\pi \int_{r_{hor}}^{r} T^r_\phi \sqrt{-g} d\theta, \tag{6}$$

$$\delta a = \frac{J + \delta J(r)}{M_{BH} + \delta M_{BH}(r)} - a^i, \qquad a^i = a^{i-1} + \Delta a. \tag{7}$$

(in the above, we use spherical, Boyer-Lindquist, coordinates, and $\sqrt{-g}$ is the square root of metric determinant).

The terms computed in addition to mass and angular momentum changes as these self-gravity perturbations, are integrated at each grid point in the radial direction and at each time. They affect the change of Kerr metric coefficients, which are sensitive to the mass and spin updates. The dimensionless black hole spin, related to its specific angular momentum $a = J/M_{BH}$, is given by $s = a/M_{BH} = J/M_{BH}^2$, and must be in the range between 0 and 1. It evolves as a result of black hole mass, M_{BH}, and its angular momentum, J changes in time, which evolve due to accretion of mass under the horizon. It is additionally changed by the self-gravity of the collapsing core. The numerical method used in HARM-SELFG has been described in detail in [3].

To sum up, we develop two branches of the code HARM:

- HARM-EOS, used in case of small mass accretion onto a fixed black hole
- HARM-SELFG, used in case of large mass accretion, onto a growing black hole

2.1. Data formats

Simulation input is provided in the initial conditions routine and configuration file. Output Data are stored either in ASCII or in HDF5 or in VTK format. For full 3D runs, each dump file containing essential variables is about 1.2 GB size. Typical run produces about 1000 dump files, plus a number of ascii format data with extra diagnostics. A single HDF5 file with grid coordinates is generated, at the initialization. Frequency of dumping given in the init depends on the time-step (i.e. subject to resources). In addition, the code generates restart files stored every 4-6 hrs.

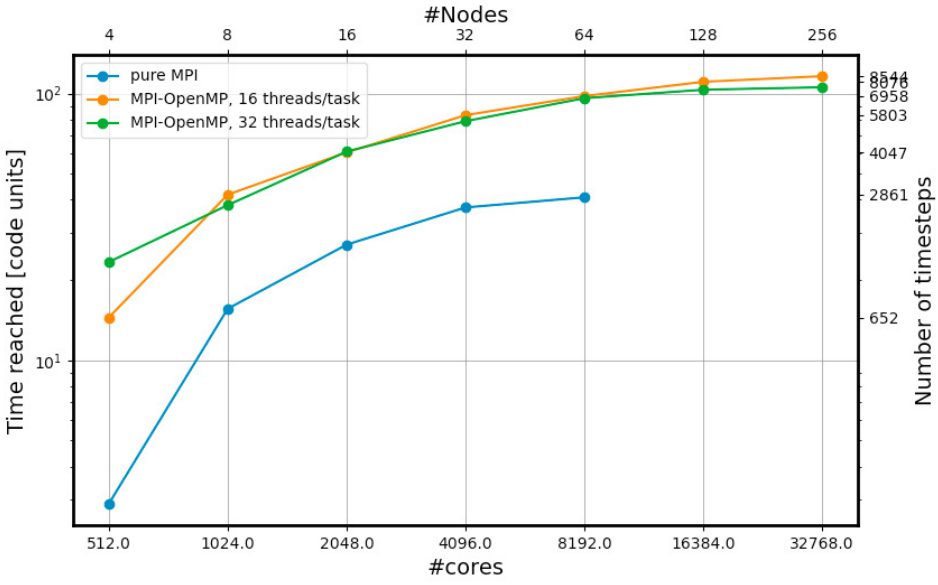


Fig. 1. Scalability test on Lumi supercomputer. We plot nummber of steps reached within 1 hour of real-time simulation (right axis) and time of simulation reached in code units (left axis), as the function of number of nodes/cores (marked in top/bottom axes).

Previous restart file is overwritten, once the new file is created. Each file contains all code variables and is about 4.2 MB. New code development (thanks to our DEV grant) features now a single restart file generated from all threads, to avoid performance problems on large CPU number with LUMI. The post-processing is performed on our local machines, with dedicated Python scripts or interactive tools (such as VisIt). During a run, periodic checks and quick post-processing is done to ensure correct performance. Full post-processing is done after the simulation ends. Data are downloaded via ‘scp‘ and stored on local disks. After post-processing, only selected data are retained on the server and compressed to a few files, resulting in a manageable file number and cleaning storage space for subsequent simulations.

2.2. Code scalability

Our HARM code is designed to be flexible and portable. With over two decades of development, it has consistently demonstrated to perform well on different supercomputing hardware, including Cyfronet Ares and Prometheus systems. The performance test results presented below have been obtained through the tests of the newest version of the HARM-EOS code at two supercomputing systems: at the Okeanos supercomputer at Warsaw Interdisciplinary Center for Mathematical Modeling, where most of runs of HARM-EOS branch have been performed recently, and in the LUMI supercomputer, where the same version has been tested on a much larger number of CPUs. This was possible due to our DEV-grant utilized this summer. The tests have not been completed so far, due to the maintenance break of the facility.

Code HARM-EOS has been tested on LUMI supercomputer with version based on pure MPI parallelization. Scalability tests indicated decreasing efficiency for $N \geq 64$ nodes. Recently, a hybrid version with OpenMP has been prepared and preliminary tests were made on Okeanos supercomputer at Warsaw ICM. In addition, test on Lumi supercomputer with more nodes were also performed. We noted that in case of 6 and 8 threads per task, scaling efficiency can be increased, and it is better than for the pure MPI case. The various parameter sweeps do not change results significantly. Results of tests are shown in Figure 1.

Conclusions from the recent scalability tests are as follows.

- Hybrid model OpenMP + MPI seems more efficient for any number of nodes, as tested on LUMI.
- It is possible that efficiency of OpenMP depends on the thread division, as well as resolution. Tests with various resolutions in radial and azimuthal directions are planned.
- The simulations have adaptive time step, that is not constant. Initial phase of the simulations is slower, due to the need for initial condition relaxation. Weak scaling tests are planned to see how the time to reach relaxed solution varies for a fixed problem size with the number of cores.
- Detailed comparisons of code efficiency between LUMI and Okeanos is needed. Nodes in LUMI are more efficient, but on Okeanos individual cores might be stronger (to be tested).
- Dumping frequency (i.e. our chosen frequency of saving checkpoints) may slightly affect performance. In case of large number of threads, writing the dumps in VTK format (integrated over all threads) may take up to several minutes.

3. Exemplary results

3.1. Initial conditions

Our code is flexible in using various initial conditions that provide density, internal energy, velocity and magnetic fields distribution in the computational domain. In the context of gamma ray burst central engines, we usually invoke slowly rotating, sub-Keplerian fluids, surrounding the central Kerr black hole.

Post-merger accretion disk systems are initialized with a barotropic, constant angular momentum torus located at the equatorial plane and embedded in poloidal magnetic field (with vector potential A_ϕ being its only non-zero component). By construction, the equilibrium torus model describes a steady-state hydrodynamical fluid around a Kerr hole but in consequence of the magnetic field action and dynamo mechanism, time evolved configurations are not in equilibrium. The magneto-rotational instability transports the angular momentum outwards and mediates accretion (see details in [11] and references therein). The model is constrained by the geometric size of such torus, whose initial parameters are the inner radius (i.e. distance between the cusp and the black hole horizon) and radius of the pressure maximum.

Collapsar scenario is initialized with a spherical distribution of density and radial velocity in the star, and we use here the transonic accretion solution. Free parameter of this configuration is the gas temperature, or sound speed at infinity. It is equivalent to the chosen location of the sonic point, r_s, from the black hole horizon, r_{hor}. At r_s the infalling gas velocity is equal to the local speed of sound, while at r_{hor} it reaches speed of light. There is no need to perturb the spherically distributed gas to allow accretion – it starts accreting in the consequence of the gravitational pull of the black hole. So in contrast to the above equlibrium torus initial condition, here we rather need to slow down the material, and keep it rotating in near the equatorial plane, in order to explain longer activity of the engine than that implied by a pure free fall. In practice, we endow the torus with a small angular momentum, which is normalized by the value of specific angular moementum and specific energy at the ISCO (innermost stable circular orbit). Details and formalism can be found e.g. in [12] and references therein. In the currently studied setups, we additionally perturb this quasi-spherical structure with magnetic fields, of various chosen initial configurations (dipole, uniform, etc.). The latter is needed to properly model the jets launched from collapsing stars.

3.2. EOS module

To study the evolution of a post-merger disk addressed to the engine of a short GRB, we use the HARM-EOS module. The microphysics of the torus formed just after the disruption of a neutron star is determined by the nuclear temperatures and densities, hence we are using Fermi gas EOS and account for partial degeneracy of nucleons and electrons or positrons. What is more important, in this way we can follow the process of neutronisation of plasma, as quantified by the so called electron fraction.

$$Y_{\rm e} = \frac{n_{\rm e^-} - n_{\rm e^+}}{n_{\rm b}}, \tag{8}$$

where in the nominator, the number density of electrons and positrons, is balanced by protons, due to the charge neutrality.

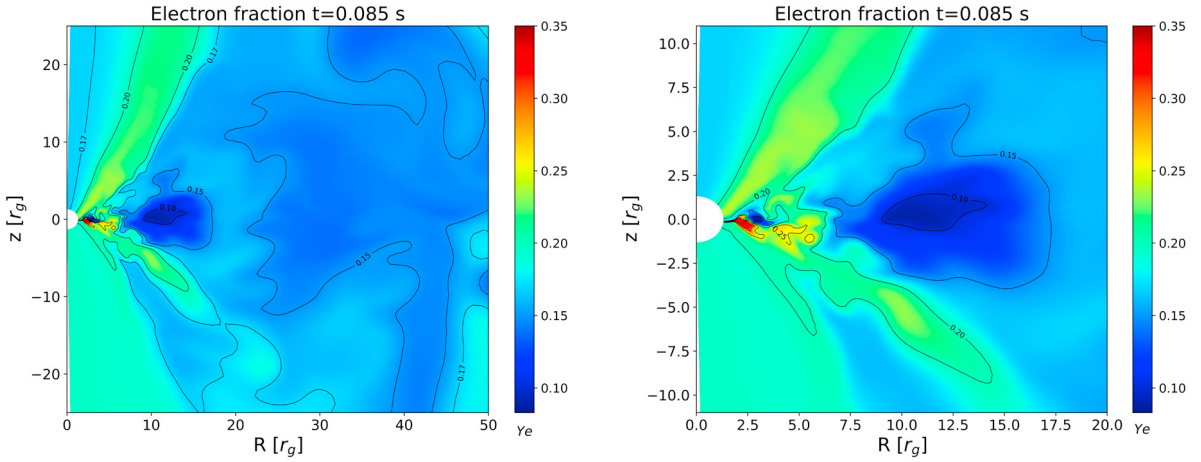


Fig. 2. Snapshots from the simulation of the GRB central engine. Plots show its chemical composition parameter - the electron fraction, in a zoomed-out (left) and zoomed-in (right) views.

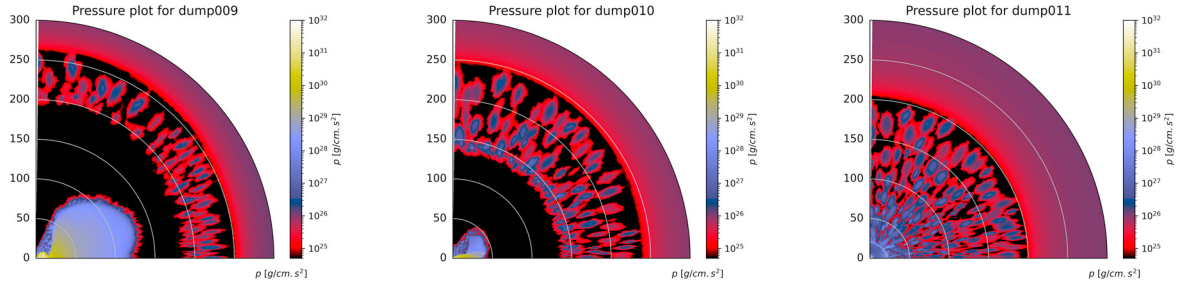


Fig. 3. Pressure inhomogeneities arising in the collapsing stellar core due to the self-gravity effect. Simulation ran in axisymmetric, 2D setup, with code HARM-SELFG. Three consecutive snapshots shown from left to right, are separated by 1000 code time units.

In Figure 2, we show the result of a post-merger system simulation. Parameters of the model are BH mass of $3M_\odot$, its dimensionless spin $a = 0.9375$, and initial gas-to-magnetic pressure ratio $\beta = 50$ at the pressure maximum radius, located at $r_{max} = 9r_g$. The disk mass is about $0.1M_\odot$. The dense and hot disk launches fast wind outflows ($v/c \sim 0.11 - 0.23$) with a broad range of electron fraction $Y_e \sim 0.1 - 0.4$. The details of simulation are sensitive to engine parameters: BH spin and magnetisation of the disk [6].

3.3. SELFG module

As an effect of self-gravity we observe density inhomogeneities and formation of the accretion shocks in our simulations. Models are parameterized by the value initial black hole spin, and rotation parameter of the collapsar. In most runs, there appears an equatorial outflow of matter, which reaches radii of up to about 80 r_g and is then stalled in the transonic shock. In addition, inhomogeneities in the pressure and density at the chosen time intervals, are detected. We illustrate them in the plots shown in Figure 3.

3.4. Jet launching

In both sets of simulations, the ultimate goal is to produce ultra-relativistic outflows from the central engine, that will be ultimately responsible for the observed high energy emissions and address the astrophysical significance of our calculations. The engines embedded in a low density environment, are found to produce such outflows. We map

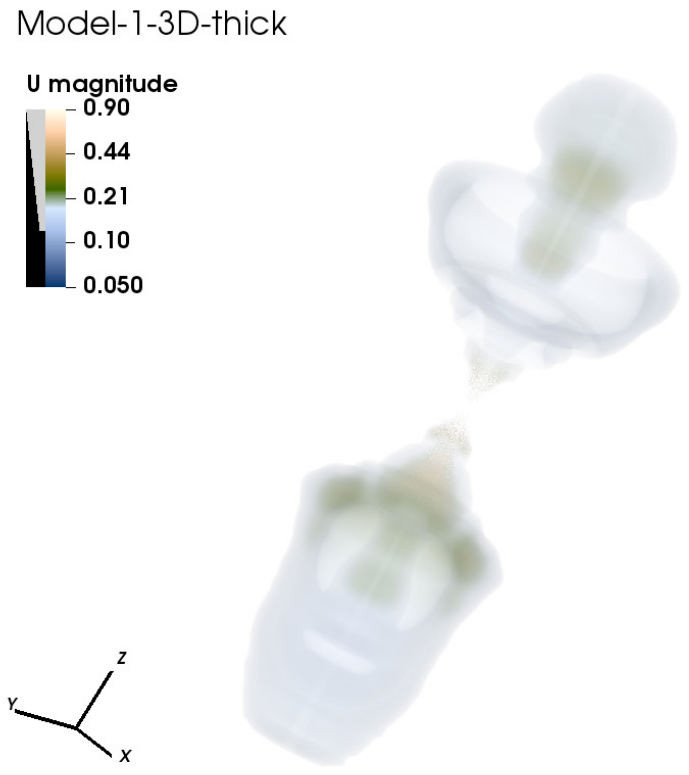


Fig. 4. Volume rendering mas of the velocity magnitude in the outflows from central engine. The 3D model was computed with HARM-EOS module.

the velocity distribution at the base of the outflow (jet) and find it is reaching a large fraction of the speed of light. The velocity distribution in a jet is shown in Fig. 4.

The engines embedded in a more dense and massive environment, such as the collapsing stellar envelope, have more difficulty in breaking out from the massive shells. Much stronger magnetic fields are needed to address this problem. It is planned for our future research project to produce successful jets form form collapsing stars, able to break out from their massive and self-gravitating envelopes.

4. Related work

General Relativistic Magneto-Hydrodynamic simulation techniques (GR MHD) are important tool to study various astrophysical systems, such as neutron star mergers, core collapse supernovae, and accretion onto black holes. The conservative MHD scheme evolves numerically a set of differential equations for the fluid variables, that are distributed over a numerical grid. In General Relativity, the evolution of the system may be studied in either a stationary backgroud specatime, like e.g. in the HARM scheme [7], see also [13], [14], or in a fully dynamical spacetime [15] [16] [17] [18]. Conservative GR MHD schemes solve a set of conservation equations of the form defined in Section 2 by Equation 4.

The fluxes and source terms are functions of primitive variables, therefore at every time step in the simulation, the given vector of conserved variables has to be inverted to obtain primitive variables, in order to evolve the conserved. The recovery procedure is an important part of every conservative GR MHD scheme, taking into account that $U(P)$ is a set of nonlinear equations itself. In contrast to Newtonian hydrodynamics, in General Relativity there is no analytic inversion between these vectors, and this set must be solved numerically, at each time step.

The thermodynamic state of the plasma is described by the so-called equation of state (EOS). Most of the old GR MHD codes utilise an ideal gas EOS, that is a polytrope, and propose specific recovery schemes that are adequate to it [19]. Further complications are found if the EOS is not polytropic one, but aims to describe a complex microphysics, such as e.g. in the core-collapse supernovae, or compact star mergers. Here the modern approach is to employ tabulated, composition-dependent, finite-temperature EOS's. They are either two or three-parameter dependent, where the third parameter describing the gas composition is called an 'electron fraction' and relates number density of electrons to protons and neutrons, while the weak interactions and neutrino emissions are taken into account.

In recent years, only a handful of GR MHD codes which use a composition-dependent EOS exist [20] [21]. Our public release of the code, HARM-COOL [5] was one of such codes as well.

Within this project, we aim to utilise a newer (private) branch of the code under development, named here the HARM-EOS. It has been used already in recent publication [22], where 2-dimensional simulations in GR MHD have been made to address the problem of black hole hyperaccretion. Recently, the code and has undergone preliminary numerical tests on Okeanos supercomputer at ICM Warsaw to serve for 3D models of kilonovae, as well as on the LUMI via development access grant, as described above.

Another aspect of our project is to incorporate the effects of self-gravity in the HARM scheme. As mentioned above, the code works in a stationary background spacetime. In principle, the self-force effect would require solving full set of Einstein Equations, which is currently done only with most advanced schemes, such as Carpet [18]. Some most recent core-collapse supernova simulations account for the growth of the black hole and change of its spin in a dynamical metric [23]. Those simulations are feasible only in 2D setup though, due to computational demands. In our case, to incorporate the effects of self gravity of the disk, we chose the Teukolsky equation [24]. This equation describes gravitational, electro-magnetic, scalar and neutrino field perturbations of a rotating Kerr black hole (see initial idea by [25]). The code HARM-SELFG has been developed as a separate branch of our HARM implementation. It is utilizing a numerical implementation of [26], who showed that the perturbation due to a particle on a bound orbit around black hole described by Teukolsky equation affects the Kerr parameters describing the mass and angular momentum of the black hole for the metric 'outside' the particle's orbit and vanishes 'inside' the orbit. Hence, we compute volume integrals of the corresponding stress-energy tensor components and add them as perturbation to the Kerr metric coefficients at any given radius above the horizon (see schematic figure below). By definition, the problem does not assume spherical symmetry. The potential wells may therefore appear off-axis, in the whole region 'outside' the orbit of a given fluid element. Some results were presented in our suite of 2D GR MHD simulations of collapsars [3]. We have proven that the effects of self-force are very strong in the initial phase of the collapse, and they significantly influence the time evolution of the black hole mass and its dimensionless spin. We also found interesting features in the self-gravitating collapsing star that present interfacial instabilities whose growth rate is stronger than the classical Rayleigh-Taylor instability. To the best of our knowledge, such instabilities have not been studied in the context of collapsars, while they have been known from the studies of proto-planetary disks [27].

5. Conclusions and future work

It has to be noted that the GR MHD simulations with three-parameter EOS in 3D need enormous resources in terms of CPU time, hence an efficient use of such codes is subject to resources available. With the current and future access to the European HPC systems we plan to develop, test, and make publicly available an integrated code HARM-EOS-SELFG to provide a universal tool for studying collapsing and merging compact stars. We plan to prepare a user friendly environment for running specific types of simulations, addressed to the Gamma Ray Bursts progenitors. We also plan to specify outputs and work packages sustainable for further post-processing. A particular focus will be given to the observable phenomena, such as the lightcurves of the GRBs and their counterparts at lower energies, as well as the late-time ejecta profiles.

Acknowledgements

We acknowledge support from the Polish National Science Center under grant 2023/50/A/ST9/00527. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2024/017013, and Warsaw University

Interdisciplinary Center for Mathematical Modeling for access to their facilities through grant 994-1723. We finally acknowledge HPC-Europe computational facilities and allocation on LUMI supercomputer through grant EHPC-DEV-2024D03-076.

References

[1] T. Piran (2014) "The physics of gamma-ray bursts" *Reviews of Modern Physics* **76** 1143
[2] D. Krol, A. Janiuk (2021) "Accretion-induced Black Hole Spin-up Revised by Numerical General Relativistic MHD",*Astrophysical Journal* **912** 132
[3] A. Janiuk, N. Shahamat, D. Krol (2023) "Self-gravitating collapsing star and black hole spin-up in long gamma ray bursts " *Astronomy & Astrophysics* **677** 19
[4] F.X. Timmes, D. Swesty (2000) "The Accuracy, Consistency, and Speed of an Electron-Positron Equation of State Based on Table Interpolation of the Helmholtz Free Energy" *Astrophysical Journal Supplement Series* **126** 501
[5] A. Janiuk (2019) "The r-process Nucleosynthesis in the Outflows from Short GRB Accretion Disks" *Astrophysical Journal* **882** 163
[6] F.H. Nouri, A. Janiuk, M. Przerwa (2023) "Studying Postmerger Outflows from Magnetized-neutrino-cooled Accretion Disks" *Astrophysical Journal* **944** 220
[7] C. Gammie, J. McKinnet, G. Toth (2003) "HARM: A Numerical Scheme for General Relativistic Magnetohydrodynamics" *Astrophysical Journal* **589** 444
[8] D. Siegel, et al. (2018) "Recovery Schemes for Primitive Variables in General-relativistic Magnetohydrodynamics" *Astrophysical Journal* **859** 71
[9] A. Janiuk, N. Shahamat, D. Krol (2023) "Self-gravitating collapsing star and black hole spin-up in long gamma ray bursts " *Astronomy & Astrophysics* **677** 19
[10] A. Janiuk, P. Sukova, I. Palit (2018) "Accretion in a Dynamical Spacetime and the Spinning Up of the Black Hole in the Gamma-Ray Burst Central Engine" *Astrophysical Journal* **868** 68
[11] K. Sapountzis, A. Janiuk (2019) "The MRI Imprint on the Short-GRB Jets" *Astrophysical Journal* **873** 12
[12] A. Murguia-Berthier, et al. (2020) "On the Maximum Stellar Rotation to form a Black Hole without an Accompanying Luminous Transient" *Astrophysical Journal Letters* **90** 24
[13] S. Komissarov (2005) "Observations of the Blandford-Znajek process and the magnetohydrodynamic Penrose process in computer simulations of black hole magnetospheres" *Mon.Not.Roy.Astron.Soc.* **359** 801
[14] C.J. White, J.M. Stone, C.F. Gammie (2016) "An Extension of the Athena++ Code Framework for GRMHD Based on Advanced Riemann Solvers and Staggered-mesh Constrained Transport" *Astrophysical Journal Supplement Series* **225** 22
[15] M.D. Duez, et al. (2005) "Relativistic magnetohydrodynamics in dynamical spacetimes: Numerical methods and tests" *Physical Review D* **72** 024028
[16] B. Giacomazzo, L. Rezzolla (2007) "WhiskyMHD: a new numerical code for general relativistic magnetohydrodynamics" *Classical and Quantum Gravity* **24** 235
[17] K. Kiuchi, Kyutoku K., Shibata M. (2012) "Three-dimensional evolution of differentially rotating magnetized neutron stars" *Physical Review D*, **86** 064008
[18] Z.B. Etienne, et al. (2015) "IllinoisGRMHD: an open-source, user-friendly GRMHD code for dynamical spacetimes" *Classical and Quantum Gravity* **32** 175009
[19] S.C. Noble, et al. (2006) "Primitive Variable Solvers for Conservative General Relativistic Magnetohydrodynamics" *Astrophysical Journal* **641** 626-637
[20] C. Palenzuela, et al. (2015) "Effects of the microphysical equation of state in the mergers of magnetized neutron stars with neutrino cooling" *Physical Review D* **92** 044045
[21] D.M. Siegel, B.D. Metzger (2017) "Three-Dimensional General-Relativistic Magnetohydrodynamic Simulations of Remnant Accretion Disks from Neutron Star Mergers: Outflows and r -Process Nucleosynthesis" *Physical Review Letters* **119** 231102
[22] A.A. Zdziarski, et al. (2024) "What Is the Black Hole Spin in Cyg X-1?" *Astrophysical Journal Letters* **967** 9
[23] T. Kuroda, M. Shibata (2024) " Numerical relativity simulations of black hole and relativistic jet formation" *Monthly Notices of the Royal Astronomical Society: Letters* **533** 107
[24] S.A. Teukolsky (1972) " Rotating Black Holes: Separable Wave Equations for Gravitational and Electromagnetic Perturbations" *Physical Review Letters* **29** 1114
[25] I. Palit, A. Janiuk, P. Sukova "Effects of self-gravity of the accretion disk around rapidly rotating black hole in long GRBs" *Acta Physica Polonica B Proceedings Supplement* **13** 261
[26] M. van de Meent (2017) "Self-Force Corrections to the Periapsis Advance around a Spinning Black Hole"
[27] P.J. Armitage (2011) "Dynamics of Protoplanetary Disks" *Annual Review of Astronomy and Astrophysics* **49** 195

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 159–163

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC user day

EuroHPC JU Infrastructures and Their Use in Science and Technology

1. EuroHPC JU: Current state of the Infrastructures

In our first edition of proceedings we have published the first overview of EuroHPC JU systems' usage by European projects for the year 2023..
In this edition we overview the state of the art for the year 2024. During the 2024 two additional EuroHPC JU systems became available for the European users: The MareNostrum5 pre-exascale system in Spain and the Deucalion petascale system in Portugal. Thus the following EuroHPC supercomputers were operational: LUMI in Finland, Leonardo in Italy, MareNostrum5 in Spain, Vega in Slovenia, MeluXina in Luxembourg, Karolina in Czech Republic, Deucalion in Portugal and Discoverer in Bulgaria.

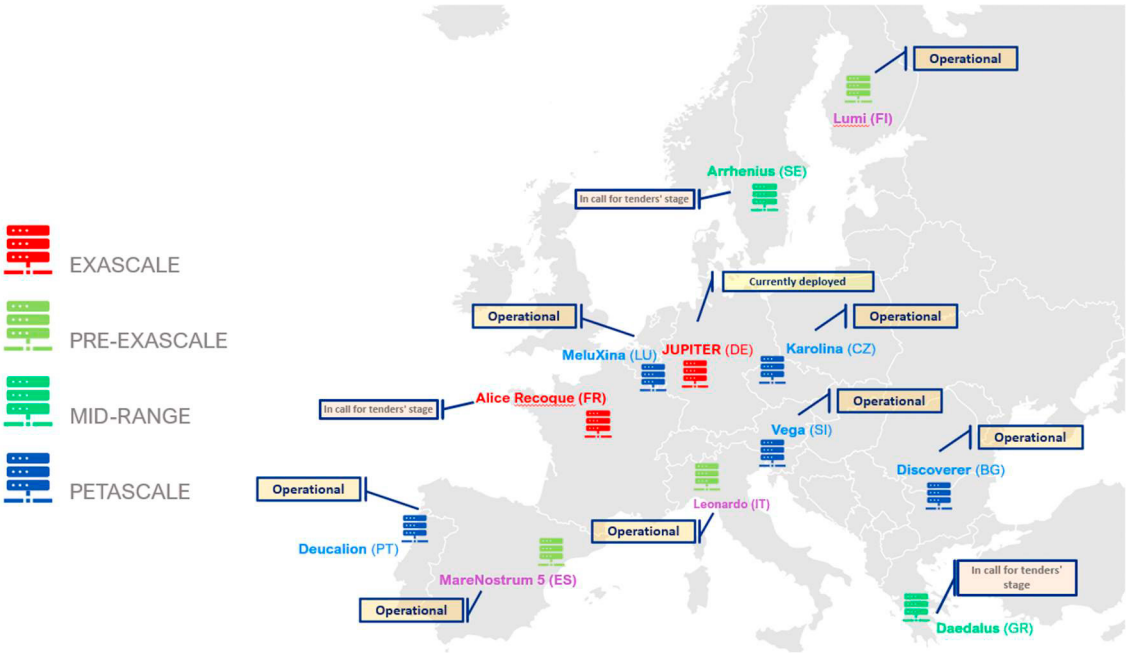


Figure 1: EuroHPC JU Supercomputers as of December 2024

Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day
10.1016/j.procs.2025.02.271

As of the beginning of 2025, JUPITER, the first European Exascale system, is in the construction phase, and three more supercomputers are in preparation as depicted in Figure 1: the second European exascale system, Alice Recoque to be hosted by the Jules-Verne consortium in France, Arrhenius, the mid-range system in Sweden, and Daedalus, another mid-range system in Greece.

There are five Access Modes, that is types of calls, for the EuroHPC supercomputers and below we summarise each other these for 2024

1.1. Benchmark and Development Access Modes

In 2024, around 700 projects were granted access to EuroHPC JU systems via Benchmark and Development calls. In comparison with 2023 these increased with 300. These calls allowed projects to test their software on different architectures of the system, improve them, benchmark and optimize them.

1.2. Regular and Extreme Access Modes

In 2024, over 100 projects accessed the EuroHPC JU systems via the Regular and Extreme Scale Access calls. This number is comparable to the 96 reported in 2023. These modes require the involvement of a large number of external experts which support the Access Resource Committee to conclude the final ranking of proposals based on the quality of the proposals. The result of this rigorous process is that a number of applications may be rejected due to achieving a lower ranking. This is because the JU must allocate resources in order of ranking, with the highest ranked proposals receiving resources first.

The proposals belong to various domains, including Biochemistry, Bioinformatics, Life Sciences, Physiology and Medicine, Chemical Sciences and Materials, Solid State Physics, Computational Physics: Universe Sciences, Fundamental Constituents of Matter and Engineering, Mathematics and Computer Sciences.

1.3. AI and Data Intensive Access Modes

In 2024, EuroHPC JU launched a new access mode dedicated specifically to AI projects. More about this access mode will be detailed in our future books of proceedings.

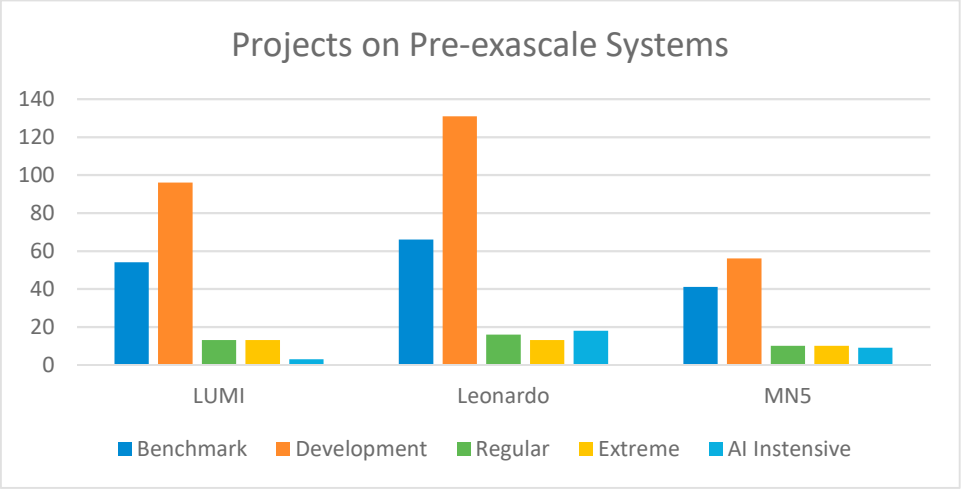


Figure 2: Division of projects active during 2024 divided per type of access per petascale system.

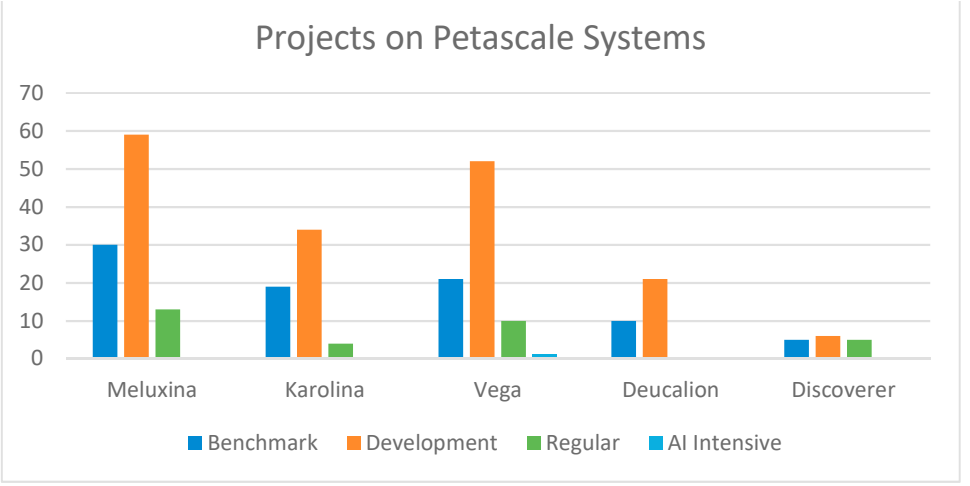


Figure 3: Division of projects active during 2024 divided per type of access per pre-exascale system.

It is important to note that both MareNostrum5 and Deucalion became available to users in the middle of 2024 therefore the projects to these systems were allocated only during part of the year.

1.4. Special Access Modes

EuroHPC JU can grant special access free of charge to strategic European Union initiatives considered to be essential for the public good, or in emergency and crisis management situations.

In 2022, the Destination Earth project, also known as DestinE, became the first such initiative to be granted Special Access. DestinE has used resources from LUMI, Leonardo, MareNostrum 5 and MeluXina EuroHPC supercomputers. The project develops a highly accurate digital model of the complex Earth system – a digital twin (DT) – to monitor, simulate and predict environmental change and human impact to support more sustainable developments and support corresponding European policies supporting the European Green Deal.

2. Usage of systems by projects through EuroHPC JU Access calls

As shown in Figure 4, during the year 2024, 872 projects were given access to EuroHPC JU supercomputers coming from four different EuroHPC JU access call modes: AI and Data Intensive, Benchmark, Development, Regular and Extreme. This is an increase with almost 300 projects in comparison to 596 during the year 2023. Most of these projects similar to 2023 were from Italy, Germany, Austria, Spain, France, and Sweden. However, in 2024 we also noticed new countries accessing EuroHPC JU systems such as North Macedonia, Montenegro and Latvia.

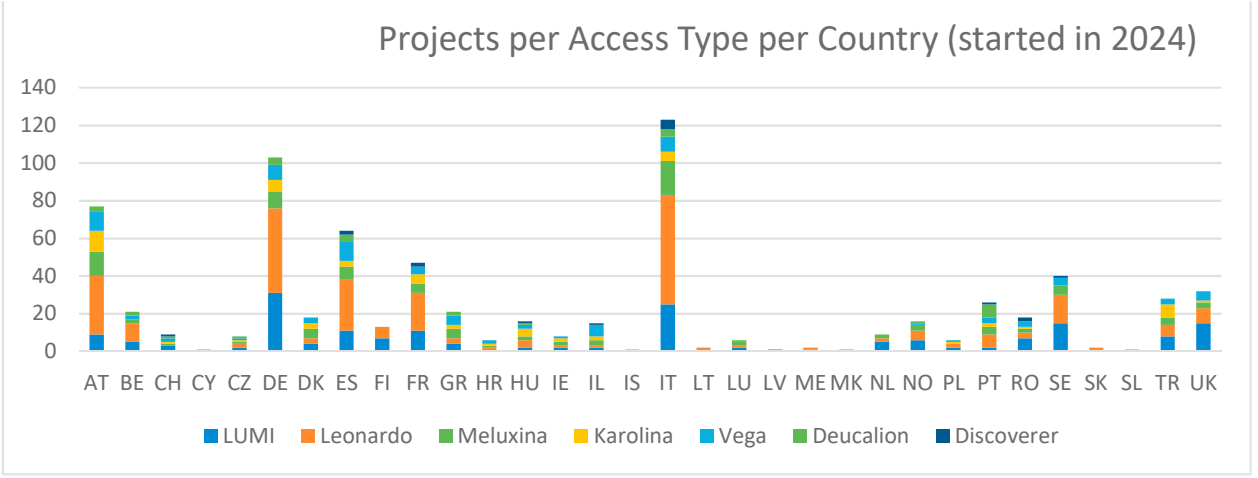


Figure 4: The active EuroHPC JU projects during the year 2024 on EuroHPC JU supercomputers (Note the data for MN5 has not been provided)

We have also analysed the type of applicants accessing the EuroHPC systems, categories between Academic, industrial or form the public sector. Figure 2 shows this division per system.

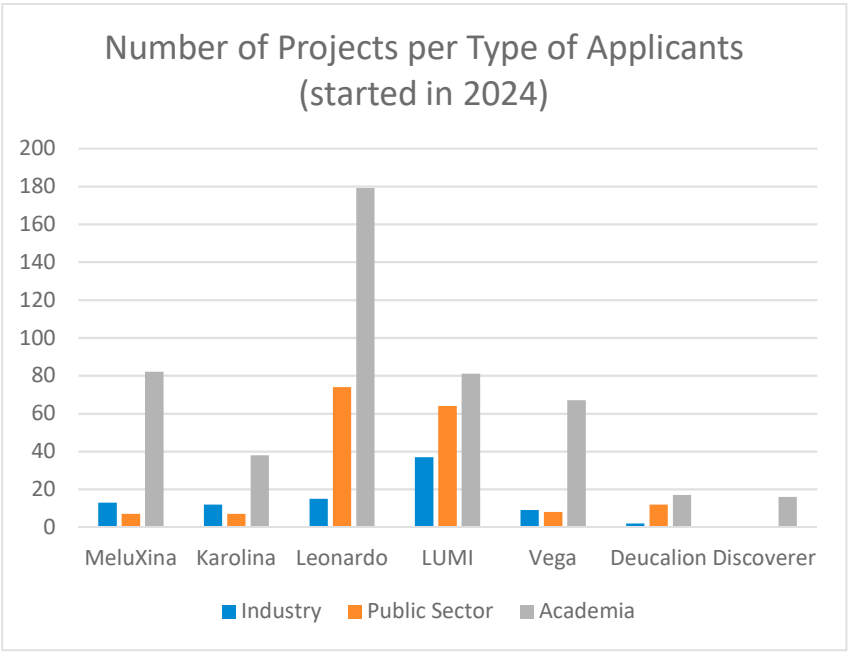


Figure 5: Industrial, academic and the public sector projects per system. (Note the data for MN5 has not been provided)

Considering the current major investments of both European Commission and all EU countries in AI, as well as the recent implementations of AI factories it is of outermost importance for EuroHPC JU to analyse the AI usage of EuroHPC JU systems. Below we detail the number of AI projects that accessed EuroHPC JU systems and highlight the once coming from the private sector.

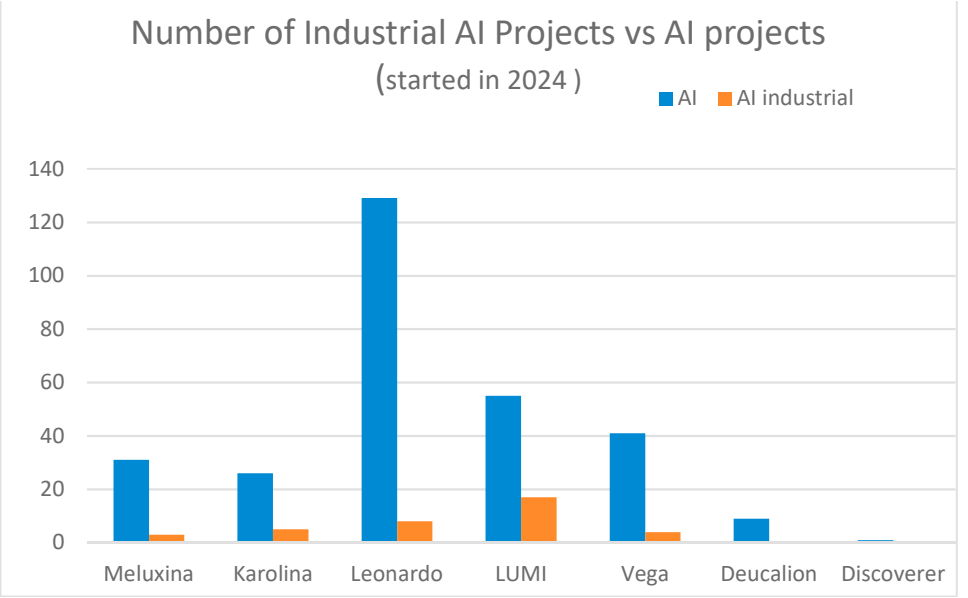


Figure 6: AI and AI industrial projects per system.

In 2024 we have had 329 AI projects accessing EuroHPC JU systems. Thus the number in 2024 doubled in comparison to 2023.

3. Conclusion

To conclude, in 2024 the number of applicants and successful projects increased substantially in comparison to 2023. As for AI projects, these have doubled in quantity. We also noted a substantial increase of interest from the private sector as well as from countries that were not accessing EuroHPC JU systems before. When looking into details of scientific disciplines and industrial sectors using EuroHPC JU systems, we note new type of users emerging with HPC. Examples are applicants from disciplines such as the Social World and its interactions, Human Mobility, Environment and Space, Diagnostic Tools, Therapies and Public Health and Neurosciences and Disorders of the Nervous System.