



OPEN Toward cardiac electrophysiology digital twins with an efficient open source scalable solver on GPU clusters

Lucas Arantes Berg^{1,6}✉, Rafael Sachetto Oliveira^{2,6}✉, Julia Camps³, Lucas Marins Ramalho de Lima⁴, Joventino de Oliveira Campos⁴, Zhinuo Jenny Wang¹, Ruben Doste¹, Alfonso Bueno-Orovio¹, Rodrigo Weber dos Santos^{4,5} & Blanca Rodriguez^{1,5}

Modelling and simulation are essential in biomedicine, and specifically in computational cardiology. Reliable, efficient and accurate solvers are critical. This study presents an open-source, GPU-based cardiac electrophysiology solver for scalable multiscale simulations (MONOALG3D), incorporating conduction system calibration and performance optimization. The solver employs the monodomain equation coupled with the Purkinje network, solved via the finite volume method, featuring a GPU-based linear solver and concurrent simulation dispatch with MPI. We demonstrate a $10.94\times$ speedup over a CPU-based solution and scalability by running 512 simulations on 128 compute nodes. Coarse and fine biventricular mesh simulations with 855, 670 and 6, 845, 360 control volumes are completed in less than 24 min and 303 min, respectively, considering a single beat and a human-based ventricular cellular model with 43 state variables. The proposed open-source solver enhances computational efficiency and physiological fidelity through Purkinje-muscle-junction calibration, enabling large-scale, high-speed cardiac simulations including the conduction system. This work marks a significant step toward fast and scalable cardiac simulations on GPU architectures by providing execution of concurrent simulations with the novel MPI batch feature and calibration of Purkinje coupling parameters, paving the way for integration into a Digital Twin personalisation pipeline, including the conduction system.

Keywords Cardiac digital twin, Purkinje network, Monodomain model, GPU solver, Finite volume method, Open-source software

Computational modelling and simulation techniques in biomedicine have advanced over the last decades, from enabling investigations of disease mechanisms to making in silico trials for therapy evaluation possible^{1–3}. Computational cardiology is a field that exemplifies a substantial amount of progress^{4–7}. Credible computer models of the heart are now available from subcellular to whole-organ dynamics, and efficient and accurate solvers have also been made available enabling large simulation studies. These two advances have driven the field forward towards the realisation of the ‘Digital Twin’ vision in healthcare^{7–10} and in silico trials for therapy evaluation¹¹. For this, cardiac models need to incorporate clinically-relevant features of cardiac function and structure, such as the Purkinje conduction system, fibre orientation, anisotropy, cell coupling, and ECG computation, among others. Moreover, in silico clinical trials and therapy evaluation require consideration of large cohorts of virtual patients. Given ecological and economical limitations in computational resources, simulation software needs to provide an accurate and efficient approximation to the mathematical models describing such phenomena. Finally for reproducibility purposes, the software and models need to be made open-source.

The challenge of the high computational costs associated with the resolution of cardiac models has led to the development of more efficient numerical schemes and the adoption of parallel computing techniques to reduce simulation times. In addition, the inclusion of the Purkinje system is a fundamental step toward physiological

¹Department of Computer Science, University of Oxford, Oxford, UK. ²Department of Computer Science, Universidade Federal de São João del-Rei, São João del-Rei, Brazil. ³Department of Engineering, Universitat Pompeu Fabra, Barcelona, Spain. ⁴Graduate Program in Computational Modelling, Universidade Federal de Juiz de Fora, Juiz de Fora, Brazil. ⁵These authors jointly supervised this work: Rodrigo Weber dos Santos and Blanca Rodriguez. ⁶Lucas Arantes Berg and Rafael Sachetto Oliveira contributed equally to this work. ✉email: arantesberglucas@gmail.com; sachetto@ufsj.edu.br

accuracy and correct modelling of ventricular activation in cardiac digital twin models. In experimental and computational studies, it has been shown that this structure can initiate and maintain certain types of arrhythmias due to altered conduction properties under pathological conditions leading to ectopic beats and reentrant circuits¹².

Equally important, open-source software presents several advantages¹³: transparency (as the source code is freely available to the public, allowing end-users to validate and verify its functionalities), as well as flexibility and modularity (users can adapt and customise the software to their particular needs by adding novel features or implementing new modules that can be shared in a collaborative development environment).

A range of cardiac solvers has been developed to simulate the heart's electrical activity. For example, the open-source software OPENCARP¹⁴ provides a flexible and parallelized finite element framework but currently lacks support for the Purkinje conduction system. Meanwhile, recent GPU-based solvers such as CARDIOMAT¹⁵ offer significant acceleration and include Purkinje modelling, but rely on conditionally stable schemes with strict timestep limits and lack physiological delays at the Purkinje-muscle junctions. While each tool presents trade-offs, GPU cardiac solvers^{16–25} offer improved accessibility and cost-efficiency.

MONOALG3D uses the finite volume method (FVM) to simulate the monodomain model on GPU and/or CPU hardware, and OpenMP and NVIDIA CUDA to respectively parallelise CPU/GPU computations. In this study, we aim to extend and enhance the open-source solver MONOALG3D with novel features. Our goals include: (1) a fully integrated Purkinje network model with a detailed mathematical formulation and Purkinje-muscle-junction parameter calibration; (2) a GPU-based solver for the diffusion linear system, significantly reducing computational time; and (3) an MPI-based batch feature that enables parallel dispatching of simulations across exascale HPC infrastructures. It is important to highlight that with goal (1) we expand on the initial MONOALG3D versions presented in Berg et al.²⁶ and Riebel et al.²⁷, which mostly relies on the Purkinje network generation method and on retrograde propagation through the Purkinje with its potential role in promoting and sustaining complex arrhythmias, respectively. Relative to the original implementation from Oliveira et al.²⁸ the novel features this work presents are: a detailed Purkinje coupling modelling, a GPU-based linear system solver, and an MPI-driven batch simulation system. In addition, we improved the input/output of the simulator by using an optimized output format (Ensign) to reduce disk usage. Together, these improvements facilitate large-scale, high-throughput simulation studies critical for digital twin and personalized medicine applications.

Performance improvements are evaluated on different hybrid CPU/GPU combinations for the device and on two human-based cell models of increasing complexity, as well as compared to space adaptivity features presented in previous work²⁸. Finally, to demonstrate the full capabilities of the proposed solver and verify its scalability on GPU clusters under more realistic scenarios, our last experiment presents a patient-specific application considering a biventricular simulation with the Purkinje system and ECG recordings.

Methods

Monodomain model

The monodomain model is commonly used to describe electrical propagation due to its lower computational cost compared to the bidomain model^{22,29}. In the next equations we present the mathematical models for the myocardium and the Purkinje system, along with their coupling. We use subscripts P for the Purkinje domain, M for the myocardium domain, and $d \in \{P, M\}$ for the full domain.

$$\beta \left(C_m \frac{\partial V_d}{\partial t} + I_{ion_d}(V_d, \vec{\eta}_d) \right) = \nabla \cdot (\sigma_d \nabla V_d) + \beta I_{stim_d} \quad \text{in } \Omega_d \times (0, T), \quad (1)$$

$$\frac{\partial \vec{\eta}_d}{\partial t} = f_d(V_d, \vec{\eta}_d) \quad \text{in } \Omega_d \times (0, T), \quad (2)$$

$$(\sigma_M \nabla V_M) \cdot \vec{n}_M = (\sigma_P \nabla V_P) \cdot \vec{n}_P \quad \text{on } \partial \Gamma_{PMJ} \times (0, T), \quad (3)$$

$$(\sigma_d \nabla V_d) \cdot \vec{n}_M = 0 \quad \text{on } \partial \Gamma_d \times (0, T), \quad (4)$$

$$V_d(X_d, 0) = V_{d,0}(X_d), \quad \eta_d(X_d, 0) = \eta_{d,0}(X_d) \quad X_d \text{ in } \Omega_d, \quad (5)$$

where V_d is the transmembrane potential of either domain, I_{ion_d} the total ionic current associated to the cellular model that depends on state variables $\vec{\eta}_d$, f_d the non-linear system of equations encapsulating the dynamics of the state variables, β the surface-to-volume ratio, C_m the membrane capacitance, σ_d the domain conductivity tensor, and I_{stim_d} an external stimulus. The model is further closed with appropriate Neumann boundary conditions to ensure flux continuity between the myocardium and Purkinje domains as given by Eq. (3), where \vec{n}_d is the normal vector of the myocardium or Purkinje on the surface interface between myocardium and Purkinje, Γ_{PMJ} . For the other boundaries, Γ_d , we simply use standard no-flux boundary conditions. Initial conditions are provided by equation (5). For the Purkinje domain, we consider the one-dimensional form of Eq. (1), while for the myocardium domain, its three-dimensional formulation.

Cardiac tissue is known to be comprised of strongly coupled fibres with anisotropic conduction properties. Such fibres are defined for each myocardial element by three orthonormal vectors $(\vec{f}, \vec{s}, \vec{n})$, where \vec{f} lies on the local fibre or longitudinal direction, \vec{s} on the sheet or transversal direction, and \vec{n} on the normal direction to the fibre. Moreover, associated with each of these vectors, there exist conductivity values σ_f , σ_t , and σ_n , jointly defining the myocardial conductivity tensor as:

$$\sigma_M = (\vec{f} \otimes \vec{f})\sigma_f + (\vec{s} \otimes \vec{s})\sigma_t + (\vec{n} \otimes \vec{n})\sigma_n. \quad (6)$$

Finite volume method applied to the monodomain model

A common technique to efficiently solve the monodomain model is to divide its reaction and diffusion parts using the Godunov operator splitting³⁰. Applied to Eqs. (1)–(2), this leads to the solution of two separate problems: a non-linear system of ordinary differential equations (ODEs)

$$\frac{\partial V_d}{\partial t} = \frac{1}{C_m} [-I_{ion_d}(V_d, \vec{\eta}_d) + I_{stim_d}], \quad (7)$$

$$\frac{\partial \vec{\eta}_d}{\partial t} = f_d(V_d, \vec{\eta}_d), \quad (8)$$

and a parabolic linear partial differential equation (PDE)

$$\beta C_m \frac{\partial V_d}{\partial t} = \nabla \cdot (\sigma_d \nabla V_d). \quad (9)$$

Within the different numerical techniques available, the FVM offers a robust approach for solving the monodomain model due to its foundation on conservative principles and applicability to diverse geometries³¹. This technique discretises the computational domain into control volumes. Each control volume is associated with a variable of interest, and the governing equations are applied to ensure the conservation of this variable across the control volume faces.

Cell model

For the solution of the cellular electrophysiology model described by Eqs. (7,8) MONOALG3D offers different techniques for integration. It supports both the explicit Euler method as well as Rush-Larsen or other methods based on the generalization of matrix exponential, such as the Uniformization approach³².

Myocardium modelling

To spatially discretise the diffusion term in Eq. (9), we consider the relation:

$$J_d = -\sigma_d \nabla V_d, \quad (10)$$

where J_d ($\mu A/cm^2$) represents the density of intracellular current flow.

Integrating over each control volume V_i , applying the divergence theorem and using equation (9), it yields:

$$\beta C_m \int_{V_i} \frac{\partial V_d}{\partial t} dV = - \int_{S_i} J_d \cdot \vec{n}_d dA, \quad (11)$$

where \vec{n}_d represents the normal vector to the domain surface. This equation is the basic term for deriving the linear system of equations associated with the linear PDE.

We now particularise the FVM equations for the myocardium. For simplicity, let us consider a tridimensional uniform mesh, consisting of hexahedra with a fixed space discretisation h_M . Located at the centre of each myocardial volume (i, j, k) is a node with the transmembrane potential V_M as the associated variable of interest. Assuming that the volumetric membrane current represents an averaged value in each hexahedron, and using (11), we then have:

$$\left(\beta C_m \frac{\partial V_M}{\partial t} \right) \Bigg|_{(i,j,k)} = \frac{- \int_{S_i} J_M \cdot \vec{n}_M dA}{h_M^3}. \quad (12)$$

To support spatially varying fibre orientation and anisotropy of the myocardial conductivity tensor, the surface integral calculations in Eq. (12) consider the total sum of flows on the 6 faces of the control volume (each with face area h_M^2) over a 27-neighbours stencil. This gives:

$$h_M^3 \beta C_m \frac{\partial V_M}{\partial t} = h_M^2 \sum_{l=1}^6 J_l. \quad (13)$$

Each J_l in Eq. (13) is implicitly calculated by evaluating the spatial derivatives of V_M via second-order finite differences at timestep $n + 1$, and computing the average conductivity tensor given by (6) at the surfaces of the discretised volume. Altogether, the previous steps lead to a linear system to solve the diffusion equation using the backward Euler method. Additional details are provided in Supplementary Material section A.1 (see Supplementary Figs. S1 and S2).

Purkinje modelling

To model the Purkinje system, we consider the one-dimensional form of the linear PDE given by Eq. (9), with time and space discretisations following an equivalent approach to the myocardial case presented above.

However, in the case of a Purkinje control volume, we have to consider three different possible configurations in our Purkinje networks (normal, branching, or terminal) as shown in Fig. 1. The total flux given by equation (11) is:

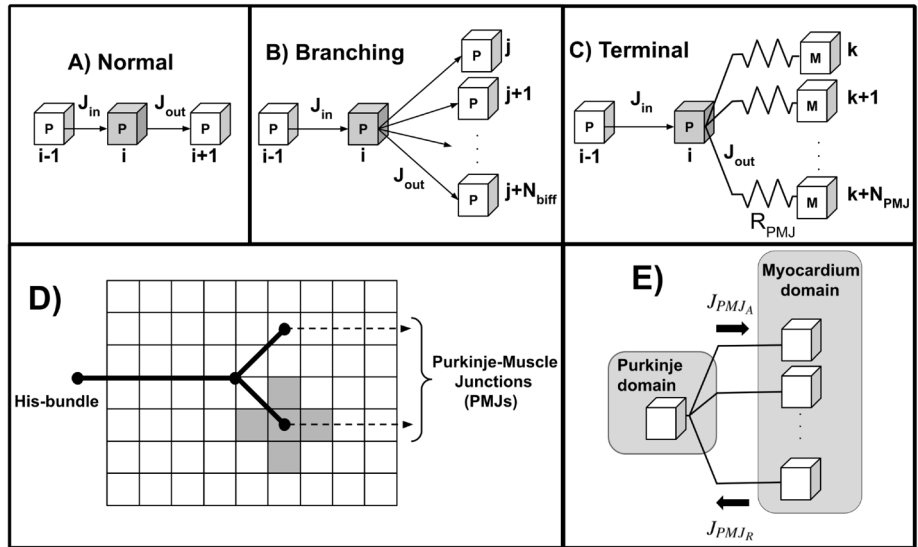


Fig. 1. Illustration of the three possible configurations for a Purkinje control volume and the Purkinje coupling model. In panels (A–C), the control volume for which we are calculating the fluxes is depicted in grey. (A) Normal case, where a Purkinje control volume is associated with no branch. (B) Branching case, where a Purkinje control volume is linked to N_{biff} other Purkinje control volumes. (C) Terminal case, where a Purkinje control volume is coupled to N_{PMJ} myocardium control volumes from the myocardium domain by a fixed resistance R_{PMJ} . Labels P and M indicate the domain where each control volume is located. (D) Simple Purkinje network with single bifurcation, coupling a Purkinje terminal to its five closest myocardium control volumes (coloured in grey). (E) Direction of the flux J_{PMJ} for anterograde (J_{PMJ_A}) and retrograde (J_{PMJ_R}) propagation, respectively.

$$\text{Normal : } J_{tot} = J_{x_{i+1/2}} - J_{x_{i-1/2}}, \tag{14}$$

$$\text{Branching : } J_{tot} = \sum_{j=1}^{N_{biff}} J_{x_j} - J_{x_{i-1/2}}, \tag{15}$$

$$\text{Terminal : } J_{tot} = J_{PMJ} - J_{x_{i-1/2}}, \tag{16}$$

where N_{biff} is the number of Purkinje control volumes linked to the bifurcation, and J_{PMJ} is the flux associated at the Purkinje-muscle-junctions. Following a similar approach to the three-dimensional case, the fluxes $J_{x_{i+1/2}}$, $J_{x_{i-1/2}}$, and J_{x_j} are calculated via finite differences at timestep $n + 1$ alongside the Purkinje conductivity σ_P associated to the surface of the discretised Purkinje control volume using harmonic means.

Purkinje–myocardium coupling

To model the coupling between Purkinje–myocardium domains, we consider an additional flux J_{PMJ} . The electrical stimulus coming from the Purkinje system reaches the myocardium at specialised sites called Purkinje-muscle junctions (PMJs), spreading in the endocardium by a distance of approximately 1 mm between each other³³. Importantly, PMJs are known to exhibit a characteristic asymmetric conduction delay due to electrotonic interactions of around 4–14 ms on the anterograde direction (Purkinje-to-myocardium)³⁴, and of about 2–4 ms when propagation occurs in the retrograde direction (myocardium-to-Purkinje)³⁴. This behaviour is characterised as a source-sink mismatch phenomenon since a single Purkinje terminal may need to activate a bulk of myocardium tissue^{35,36}.

Typically, the PMJ coupling is modelled by a fixed resistance, linking a Purkinje element to several myocardium elements³⁷. We follow this approach by modelling the flux J_{PMJ} ($\mu A/cm^2$) using a fixed resistance R_{PMJ} and by coupling a single Purkinje control volume to its N_{PMJ} closest myocardium control volumes, as shown in Fig. 1D. Moreover, the PMJ flux is given as a non-homogeneous Neumann boundary condition by:

$$J_{PMJ} = \frac{1}{h_P^2} \sum_{k=1}^{N_{PMJ}} \frac{(V_P - V_{M_k})}{R_{PMJ}}, \tag{17}$$

where the sign of the flux determines if J_{PMJ} exerts its action in the anterograde or retrograde direction (see Fig. 1E).

Numerical scheme

For the iterative solution of the coupled model, we start by solving the reaction terms describing the Purkinje and myocardium cellular models, given by the non-linear systems of ODEs in Eqs. (7)–(8). We consider here the forward Euler method for simplicity, albeit MONOALG3D is equipped with more advanced ODE schemes (such as Rush-Larsen and adaptive forward Euler). This gives:

$$C_m \frac{V_d^{n+1/2} - V_d^n}{\Delta t} = [-I_{ion_d}(V_d^n, \eta_d^n) + I_{stim_d}], \quad (18)$$

$$\frac{\eta_d^{n+1/2} - \eta_d^n}{\Delta t} = f_d(V_d^n, \eta_d^n). \quad (19)$$

PMJ fluxes are computed next based on Eq. (17), as:

$$J_{PMJ}^{n+1/2} = \frac{1}{h_P^2} \sum_{k=1}^{N_{PMJ}} \frac{(V_P^{n+1/2} - V_{M_k}^{n+1/2})}{R_{PMJ}}. \quad (20)$$

The diffusion terms of the Purkinje and myocardium domains involves the solution of the linear system:

$$\beta C_m \frac{V_d^{n+1} - V_d^{n+1/2}}{\Delta t} = \nabla \cdot (\sigma_d \nabla V_d^{n+1}) + J_{PMJ}^{n+1/2} h_P^2. \quad (21)$$

Observe that by computing J_{PMJ} at time $n + 1/2$, we decouple the Purkinje and Myocardium domains. This enhances the solver's modularity, allowing different classes to be used for each domain, but at the cost of numerical stability. While a fully implicit solution of the PDE is unconditionally stable, decoupling the two domains results in a conditionally stable scheme, where R_{PMJ} constrains the maximum time step.

ECG calculations

An approximation for the ECG can be computed by assuming that the tissue is immersed in an unbounded volume conductor³⁸. The surface potential can be then calculated using the equation:

$$\phi_e = \frac{1}{4\pi\sigma_b} \int_{\Omega} \frac{\beta I_m}{\|\vec{r}\|} d\Omega, \quad (22)$$

where σ_b is the bath conductivity, and \vec{r} is the distance vector between source and field points, the latter essentially the electrode positions of the virtual ECG leads. The source term βI_m is given by the solution of the diffusive term $\nabla \cdot (\sigma \nabla V_m)$, which is available in every timestep.

To efficiently implement this new functionality in MONOALG3D, we implemented the calculations of the Eq. (22) using OpenMP in CPUs or CUDA on GPUs environments.

Performance efficiency strategies

Solving diffusion on GPUs

In previous work²⁸, the linear system linked to the diffusion term in Eq. (9) was exclusively solved in the CPU using an OpenMP version of the conjugate gradient (CG) method. To enable the solution of large linear systems on GPUs, we first converted its sparse matrix representation from the ALG format to a Compressed Sparse Row (CSR) data structure compatible with the `cusparse` library. This allows to directly solve the CG on the GPU by using this data structure together with the methods implemented in the `cuBLAS` library. It is worth noting that the biconjugate gradient (BCG) method is also available in this new version.

New output format

To minimise disk space usage and improve output performance, we provide novel support for EnSight files as new output format. This format is also compatible with multiple visualisation tools, such as Paraview, allowing most post-processing workflows to be kept unchanged.

MPI batch

Finally, for sensitivity analysis and uncertainty quantification studies, MONOALG3D provides a novel feature for the concurrent dispatch of multiple simulations using the message passing interface (MPI) standard. Given a baseline simulation and a range of parameters, the solver generates automated configuration files for all possible combinations of input parameters. Each configuration file is then dispatched in parallel using MPI. This allows to upscale more efficiently the number of jobs running in HPC environments, enabling for instance to perform hundreds of simultaneous simulations for a given patient using a wide range of parameters. Such a feature is an important step towards in silico trials, drug therapy, and risk assessment studies.

Computational simulations

Two sets of experiments were used to evaluate the improvements implemented in MONOALG3D: a benchmark cuboid mesh to quantify performance improvements; a patient-specific mesh coupled to a Purkinje network (see Fig. 2), as an exemplar of application towards cardiac digital twinning.

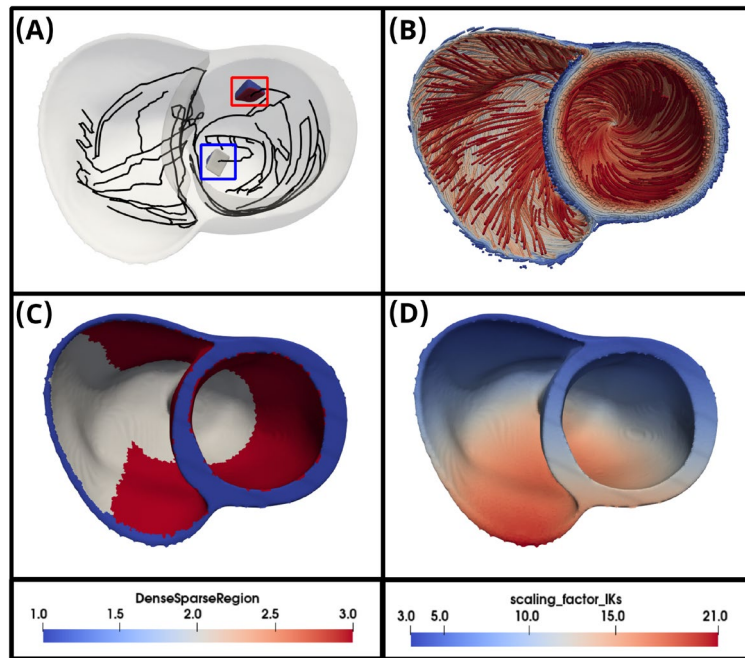


Fig. 2. Human-based biventricular mesh used for patient-specific simulations. (A) Biventricular mesh coupled to the Purkinje network generated with the extra branching procedure from²⁶. The Purkinje network is coloured in black and ventricular wedges used for the Purkinje coupling calibration are highlighted by red and blue squares. (B) Fibre orientation field of vector f . (C) Heterogeneity in conductivity tensor σ_M , where blue denotes fully orthotropic conductivity values ($\sigma_f, \sigma_t, \sigma_n$), while white and red represent dense and sparse endocardial regions with isotropic conductivity values of σ_{dense} and σ_{sparse} , respectively. (D) Scaling factor map of I_{Ks} current in the *ToR-ORd* model used for T-wave personalisation⁴⁶. This figure was generated using the last version of *MONOALG3D* (https://github.com/rsachetto/MonoAlg3D_C) and visualised with the Paraview tool version 5.13.2.

All our numerical experiments were performed on the Polaris supercomputer provided by the Argonne Leadership Computing Facility, a 560 node HPE Apollo 6500 Gen 10+ system. Each computing node is equipped with a 2.8 GHz AMD EPYC Milan 7543P 32 core CPU with 512 GB of DDR4 RAM and four NVIDIA A100 GPUs with 40,960 megabytes of memory each.

Benchmark myocardium cuboid

A numerical test, adapted from Niederer et al.³⁹, was conducted with minor domain size modifications to: (1) evaluate GPU speedups in solving non-linear ODEs and the parabolic PDE; (2) assess space adaptivity effects on efficiency; and (3) compare disk space usage between EnSight and VTK formats.

The test used a $1 \times 1 \times 1 \text{ cm}^3$ myocardium cuboid with transverse anisotropic conduction $\sigma_{\parallel} = 1.334 \text{ mS/cm}$, $\sigma_{\perp} = 0.176 \text{ mS/cm}$, monodomain parameters $\beta = 1400 \text{ cm}^{-1}$, $C_m = 1 \text{ }\mu\text{F/cm}^2$, and human-based ventricular models: *ten Tusscher* (12 state variables)⁴⁰ and *ToR-ORd* (43 state variables)⁴¹. Stimulation was applied in a $0.15 \times 0.15 \times 0.15 \text{ cm}$ region for 2 ms at 53 pA/pF.

The nonlinear ODEs were solved using the Rush-Larsen scheme⁴² with $\Delta t = 0.01 \text{ ms}$, while the parabolic PDE used $\Delta t = 0.02 \text{ ms}$ for a total of 1500 ms. Without space adaptivity, uniform discretisation at $h_M = 250 \text{ }\mu\text{m}$ led to 64,000 control volumes. Adaptive resolutions ranged from $h_{M_{min}} = 250 \text{ }\mu\text{m}$ to $h_{M_{max}} = 500 \text{ }\mu\text{m}$, with refinement/de-refinement bounds at 10.01/10.00 and adaptation every 10 timesteps.

The benchmark was tested across eight CPU/GPU configurations:

- A+OC+PC: Adaptive, ODEs on CPU, PDE on CPU;
- A+OG+PC: Adaptive, ODEs on GPU, PDE on CPU;
- A+OC+PG: Adaptive, ODEs on CPU, PDE on GPU;
- A+OG+PG: Adaptive, ODEs on GPU, PDE on GPU;
- OC+PC: Non-adaptive, ODEs on CPU, PDE on CPU;
- OC+PG: Non-adaptive, ODEs on CPU, PDE on GPU;
- OG+PC: Non-adaptive, ODEs on GPU, PDE on CPU;
- OG+PG: Non-adaptive, ODEs on GPU, PDE on GPU.

Mesh geometry and transmembrane potential were saved every 100 timesteps. More details on the benchmark setup are provided in Supplementary Fig. S3 section A.2.

Patient-specific model with Purkinje network

To demonstrate the full capabilities of the proposed GPU cardiac solver, we conducted a simulation study within a patient-specific biventricular model incorporating a Purkinje network. The study had three objectives: (1) evaluate solver performance in realistic scenarios; (2) calibrate Purkinje coupling parameters R_{PMJ} and N_{PMJ} to physiological anterograde PMJ delays; and (3) verify solver scalability for concurrent GPU simulations.

We used a human biventricular mesh (76-year-old female, 87 kg, 107 cm^3 volume) reconstructed from MRI⁴³, previously applied in clinical ECG personalization^{44–46}. Figure 2, presents further anatomical details, including its Purkinje network coupling (Fig. 2A), fiber orientation field (Fig. 2B), subendocardial Purkinje coupling layers (Fig. 2C), and I_{Ks} scaling factor map for T-wave personalization⁴⁵ (Fig. 2D).

For cellular electrophysiology, we used the *ToR-Ord* human-based ventricular model⁴¹ with modifications for T-wave personalization⁴⁶: 50% I_{Kr} scaling, $5 \times I_{Ks}$ scaling⁴⁷, and reducing τ_{jca} from 75 to 60 ms. The Purkinje domain was modeled with the human-based Purkinje *Trovato* model⁴⁸. Both ODE systems were solved via the Rush-Larsen scheme with $\Delta t = 0.01\text{ ms}$. PDEs used the same discretisation step for a total simulation time of 600 ms. The stimulus protocol consisted of a single pulse applied at the His bundle ($N_{cells} = 25$) with 40 pA/pF amplitude and 2 ms duration. For more details about the mesh configuration refer to Supplementary Material sections A.3 and A.4.

Purkinje-muscle-junctions calibration

To validate the Purkinje module and enable a physiological range for its coupling parameters, an initial PMJ calibration experiment was conducted using two ventricular wedges from the considered biventricular mesh, as highlighted in Fig. 2A. Both wedges are activated by a single Purkinje terminal and are located at different regions of the left ventricle (see Fig. 2A), in order to analyse the effects of the isotropic endocardial conductivities (σ_{dense} and σ_{sparse}) on the anterograde PMJ delay. As so, the first wedge is located within the sparse endocardial region, while the second is in the dense endocardial region. Different space discretisations were also tested for both the Purkinje (100 and 250 μm) and the myocardial (250 and 500 μm) domains. Conductivities were calibrated using monodomain cable simulations to match conduction velocities (CVs) with an automatic parametrization strategy⁴⁹, as summarised in Supplementary Table S3 (Supplementary Material section A.4). For the Purkinje coupling parameters, we considered a range of 25 equispaced values in [100, 2500] $k\Omega$ for the PMJ resistance, R_{PMJ} , and 10 equispaced values in [10, 100] for the number of myocardium control volumes linked to a terminal Purkinje control volume, N_{PMJ} .

Altogether, we performed 2000 simulations considering all possible combinations of wedges, space discretisations, and Purkinje coupling parameters. Simulations were executed for a total time of 50 ms, as we were only interested in measuring anterograde PMJ delays. These took $\approx 30\text{ s}$ to run enabled by the OG+PG setup. The PMJ delay was calculated as the time difference between the terminal Purkinje control volume and the closest myocardium control volume linked to it reaching a transmembrane potential threshold of -40 mV .

Large-scale biventricular simulations

To evaluate performance, we tested two myocardium space discretisations: a *coarse* mesh ($h_M = 500\text{ }\mu\text{m}$) with 855, 670 control volumes and a *fine* mesh ($h_M = 250\text{ }\mu\text{m}$) with 6, 845, 360 volumes. The Purkinje domain used a fixed $h_P = 250\text{ }\mu\text{m}$ with 7, 948 volumes. The control volumes for the *fine* and *coarse* meshes are regular hexahedrons with fixed space discretisation h_M . Similarly, the control volumes of the Purkinje network are also represented as regular hexahedrons with a fixed space discretisation h_P . Solver scalability was assessed across three simulation setups:

- 1N1S: 1 node, 1 simulation;
- 1N4S: 1 node, 4 concurrent simulations;
- 128N512S: 128 nodes, 512 concurrent simulations.

A large-scale simulation study calibrated R_{PMJ} and N_{PMJ} for the full biventricular case within a physiological range using MONOALG3D's MPI batch processing (see Supplementary Fig. S4 in Supplementary Material section A.5). We ran 512 concurrent simulations, varying R_{PMJ} and N_{PMJ} based on the initial Purkinje-muscle-junctions calibration simulations over the ventricular wedges. It is important to highlight that for each concurrent biventricular simulation 8 CPU cores and 1 GPU are utilized, as this setup demonstrated the best performance. For the *coarse* mesh, R_{PMJ} spanned [100, 1300] $k\Omega$ (32 values) and N_{PMJ} [15, 50] (16 values). For the *fine* mesh, R_{PMJ} ranged from [500, 2300] $k\Omega$. ECG comparison with clinical data was performed using Pearson's correlation coefficient across all 8 leads (I, II, V1–V6).

Results and discussion

Myocardium cuboid benchmark

An initial test was conducted using the OC+PC configuration, which uses entirely the CPU without space adaptivity to solve both the ODE and PDE systems, to evaluate the optimum number of OpenMP threads for the selected HPC facility. Five simulations were executed per number of threads, considering the human ventricular cellular *ToR-Ord* model (Fig. 3A). The best total execution times were found for an optimal number of 8 OpenMP threads, leading to a $\approx 6.66 \times$ efficiency speedup, and enabling benchmark execution times around 30 min.

Similarly, input/output efficiency was optimised by considering 5 broadly adopted scientific formats: VTK-text (ASCII), VTK-binary, VTK-binary-compressed, EnSight-text (ASCII), and EnSight-binary (Fig. 3B). The results from this analysis highlight substantial savings in output file size when saving each model state variable (transmembrane potential in our case) in EnSight-binary format. This resulted in file storage sizes of merely 0.19 gigabytes, while VTK-text required around 6 gigabytes to store the same outputs. Therefore, the EnSight-

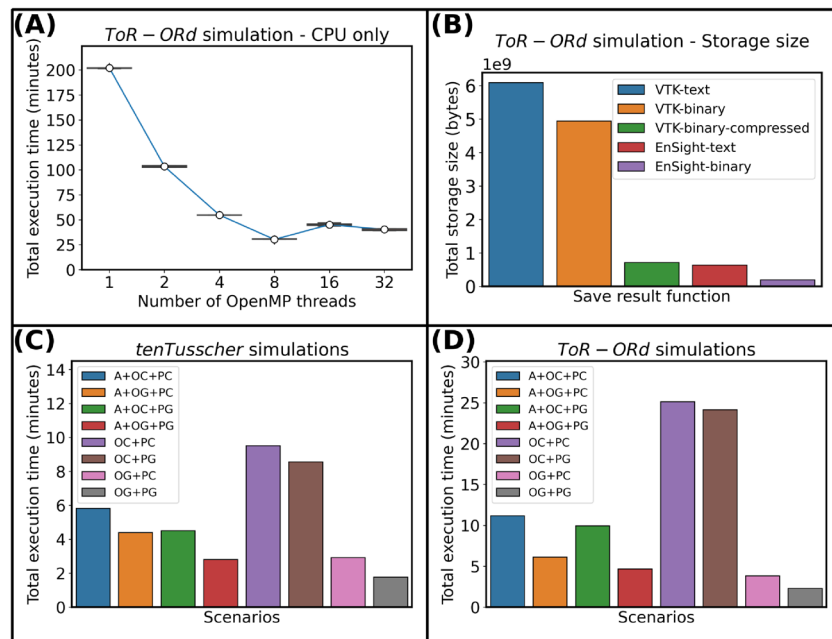


Fig. 3. Results for the myocardium cuboid benchmark. **(A)** Execution time of the *ToR-ORD* simulation considering an OC+PC solver. **(B)** Disk usage for storing the mesh geometry and with the myocardial transmembrane potential for the *ToR-ORD* simulation using different file formats. **(C)** Total execution time for each of the 8 scenarios when using the *ten Tusscher* model. **(D)** Total execution time for each of the 8 scenarios with the *ToR-ORD* model.

binary can save approximately $31\times$, $25\times$, $3.5\times$ and $3.23\times$ more disk space when compared to VTK-text, VTK-binary, VTK-binary-compressed, and EnSight-text formats, respectively.

We then evaluated the solver's performance for each of the 8 considered combinations of CPU/GPU architectures (see section *Computational simulations/Benchmark myocardium cuboid*), using either the *ten Tusscher* (Fig. 3C) or the *ToR-ORD* (Fig. 3D) cellular models. Based on the analysis above, 8 OpenMP threads were used in all the cases. The efficiency results presented in Fig. 3C for the *ten Tusscher* model yielded a maximum simulation time of ≈ 9 min for the OC+PC scenario (i.e., solving the entire problem in the CPU, without space adaptivity). Space adaptivity allowed the problem to be solved under 6 min in the CPU (A+OC+PC scenario), and in less than 3 min if both problems are solved in the GPU (A+OG+PG scenario). The largest efficiency improvement was however found when the problem was solved entirely in the GPU without space adaptivity (OG+PG scenario), decreasing the simulation time under 2 min. Equivalent results are presented in Fig. 3D for the *ToR-ORD* model for all the CPU/GPU configurations. Solving the simulation entirely on the CPU without space adaptivity (OC+PC scenario) yielded the most demanding execution time of ≈ 25 min, while a $10.94\times$ efficiency gain and a total simulation time below 3 min were attained by exploiting the full GPU implementation (OG+PG scenario).

The results above indicate that space adaptivity (scenarios A+OC+PC, A+OG+PC, A+OC+PG and A+OG+PG) did not lead to any improvements in performance when compared to solving the fully refined mesh entirely on the GPU (scenario OG+PG). This behaviour can be attributed to the computational overhead associated with spatial adaptivity, specifically reassembling the matrix of the PDE and updating the grid data structures. In contrast, preloading the fully refined mesh and transferring all the data structures for both the ODEs and PDE to the GPU at the onset of a simulation offers significant advantages in terms of memory usage and computational performance when a fixed spatial discretisation is used. Therefore, the *ten Tusscher* model consumed around 436 megabytes of GPU memory when solved with the OG+PG scenario, while the *ToR-ORD* approximately 452 megabytes. This approach not only reduces data transfer between the host and the GPU, but also minimises memory allocation operations. Consequently, the computation at each time step is more regular than when space adaptivity is used, leading to an improved overall performance.

In addition, the joint analysis of the two considered cellular models revealed that solving the ODE system on the GPU was responsible for most of the performance gains. This becomes apparent by comparing scenarios OC+PC and OC+PG. For the *ten Tusscher* model, scenario OC+PG is just $1.11\times$ faster than scenario OC+PC, while for *ToR-ORD* a speedup of $1.04\times$ is obtained. However, when the ODE system was solved on the GPU (scenario OG+PC), a $3.26\times$ gain was attained for the *ten Tusscher* model when compared to scenario OC+PC, while this improvement was even more pronounced for *ToR-ORD*, and around $6.60\times$ gain. This result is primarily underlain by differences in algebraic complexity between both cellular models: the *ten Tusscher* model consists of 12 state variables, compared to 43 in *ToR-ORD*. Additionally, the *ToR-ORD* model involves a greater number of algebraic expressions, making it well-suited for GPU-based computations. Detailed execution times for both models can be found in the Supplementary Tables S1 and S2 section A.2. Finally, based on the

convergence analysis for the cuboid benchmark executed using the latest versions of MONOALG3D and OPENCARP and presented in the Supplementary Material section A.7, we verified that space discretisations of $500\ \mu\text{m}$ and $250\ \mu\text{m}$ provide relative errors below 220% and 45%, respectively, on both solvers when compared to a more refined mesh resolution of $50\ \mu\text{m}$ (see Supplementary Fig. S6). Therefore, spatial discretisation above $250\ \mu\text{m}$ should be avoided, especially under pathological scenarios, and the usage of a mesh resolution of $500\ \mu\text{m}$ for the biventricular mesh was purely motivated to demonstrate scalability of MONOALG3D in the present work.

Purkinje-muscle-junctions calibration reveals that anterograde delay is affected by different factors

Figure 4 summarises how different factors affect the anterograde PMJ delay. First, refining the myocardial mesh reduces the delay, as the PMJ volume shrinks, we decrease the source-sink mismatch³⁶. This effect is observable in pairwise comparisons such as Figures 4A–C, B–D, E–G, and F–H.

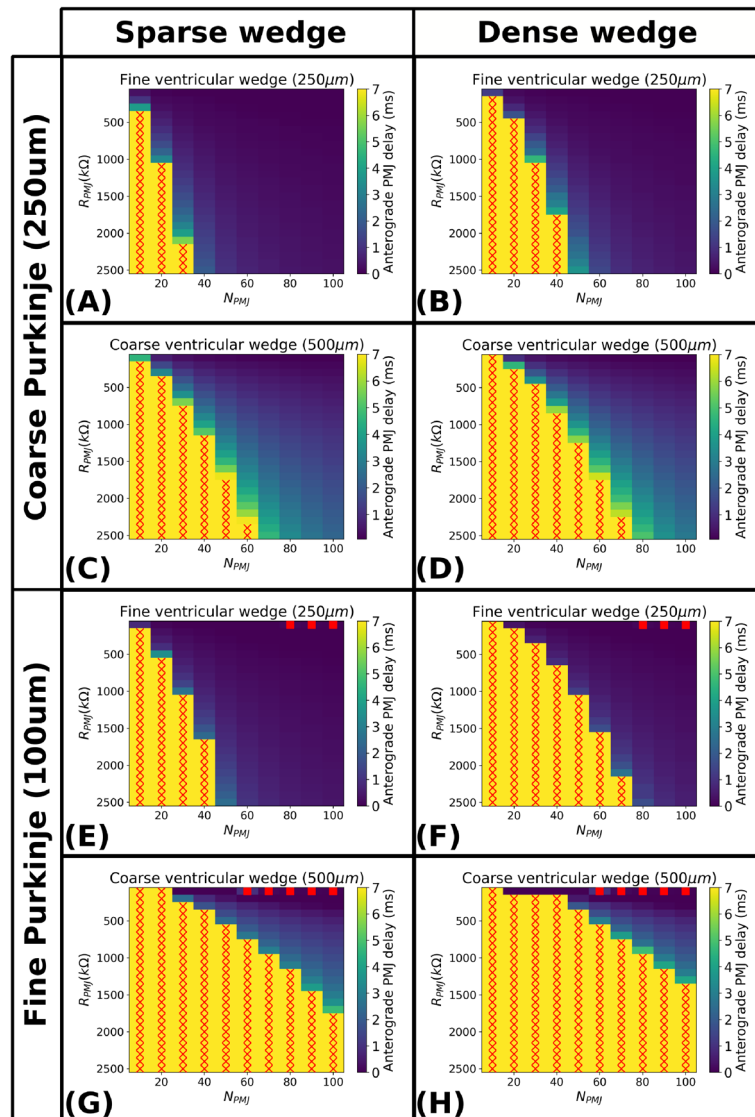


Fig. 4. Heatmaps of anterograde PMJ delays for varying Purkinje coupling parameters R_{PMJ} and N_{PMJ} at different space discretisations. Red crosses denote propagation block at the PMJ site, while red squares denote instabilities associated to solving the Purkinje coupling explicitly. (A) Sparse endocardial region at fine discretisation, coupled to a coarse Purkinje fibre. (B) Dense endocardial region at fine discretisation, coupled to a coarse Purkinje fibre. (C) Sparse endocardial region at coarse discretisation, coupled to a coarse Purkinje fibre. (D) Dense endocardial region at coarse discretisation, coupled to a coarse Purkinje fibre. (E) Sparse endocardial region at fine discretisation, coupled to a fine Purkinje fibre. (F) Dense endocardial region at fine discretisation, coupled to a fine Purkinje fibre. (G) Sparse endocardial region at coarse discretisation, coupled to a fine Purkinje fibre. (H) Dense endocardial region at coarse discretisation, coupled to a fine Purkinje fibre. Fine spatial discretisation: $h_M = 250\ \mu\text{m}$, $h_P = 500\ \mu\text{m}$. Coarse spatial discretisation: $h_M = 500\ \mu\text{m}$, $h_P = 250\ \mu\text{m}$. Conductivity values as per Supplementary Table S3 (Supplementary Material section A.4).

Mesh	OpenMP threads					
	1	2	4	8	16	32
<i>coarse</i> ($h_M = 500 \mu m$)	44.93	29.49	19.00	14.95	13.61	13.43
<i>fine</i> ($h_M = 250 \mu m$)	371.96	269.51	225.55	223.93	235.22	235.88

Table 1. Execution times (minutes) for the patient-specific simulations using the OG+PG scenario with a varying number of OpenMP threads. OG+PG: Non-adaptive, ODEs on GPU, PDE on GPU.

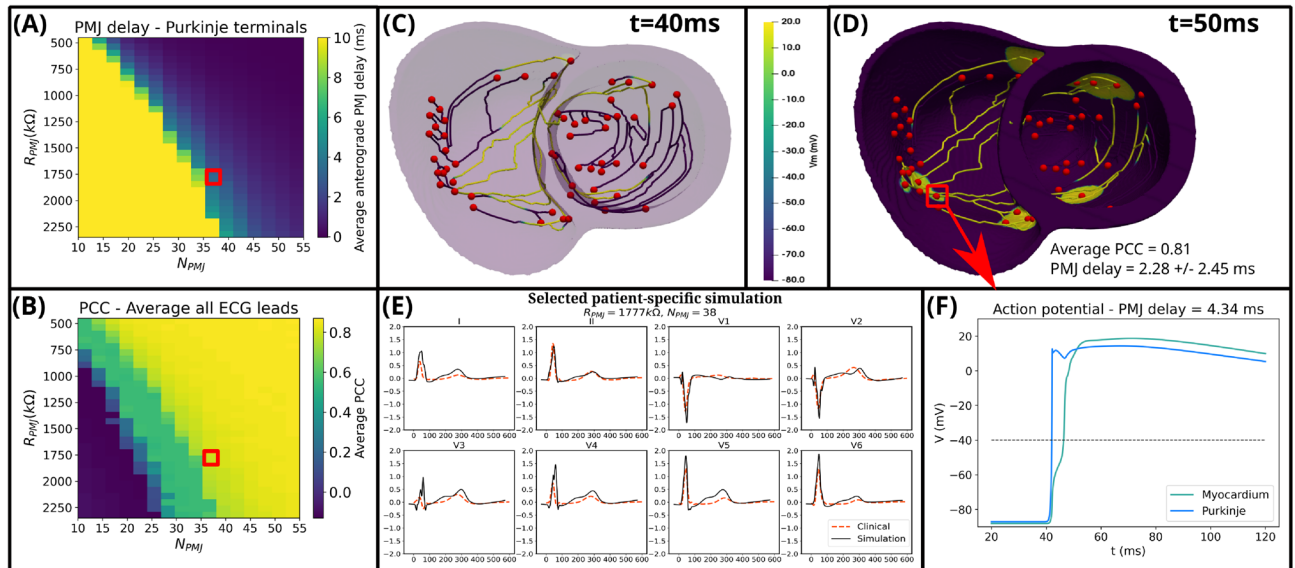


Fig. 5. Results for the 512 patient-specific simulations with the *fine* mesh. (A) Average anterograde PMJ delay across all Purkinje terminals. (B) Average PCC across all leads between the clinical and simulated ECG. Selected Purkinje coupling parameters ($R_{PMJ} = 1777 k\Omega$, $N_{PMJ} = 38$) are highlighted by red squares. (C,D) Selected patient-specific simulation with an average PCC of 0.81 in ECG reconstruction and average anterograde PMJ delay of 2.28 ± 2.45 ms, at times $t = 40$ and $t = 50$ ms, respectively. (E) Comparison between clinical and simulated ECGs. (F) Action potential upstrokes for the PMJ site highlighted in panel (D) with anterograde PMJ delay of 4.34 ms, at the terminal Purkinje volume (blue) and its closest coupled myocardium volume (lime). The 3D models in panels (C,D) were generated using the last version of MONOALG3D (https://github.com/rsachetto/MonoAlg3D_C) and visualised with the Paraview tool version 5.13.2.

Second, Purkinje mesh resolution also influences the delay. Finer discretisation increases the delay due to reduced current per control volume (less source). Conversely, coarser meshes with larger Purkinje control volumes, deliver more current through the coupling interface, accelerating myocardial depolarisation. This trend appears in Fig. 4A–E, B–F, C–G, and D–H.

Third, endocardial conductivity impacts the delay, with denser regions, i.e., high conductivity regions, producing longer delays (Fig. 4A,B). Higher conductivity leads to greater current dispersion before threshold is reached, delaying activation. Similar effects appear in comparisons Fig. 4C,D,E,F,G,H. In addition, we report some instabilities (red squares in Fig. 4) due to the PMJ coupling being solved explicitly in Eq. (20).

Lastly, anterograde PMJ delay depends on coupling parameters. Increasing R_{PMJ} raises the delay and may even block propagation (red crosses in Fig. 4) due to reduced current flow. In contrast, increasing N_{PMJ} , the number of coupled myocardial volumes, lowers the delay by stimulating larger regions of the myocardium.

Large-scale biventricular simulations with anterograde PMJ delay calibration are a step towards cardiac digital twinning

An initial simulation that varied the number of OpenMP threads using the OG+PG scenario was executed to verify the optimal number of threads to be used for the *coarse* and *fine* meshes. Based on the results in Table 1, all patient-specific simulations were executed entirely on the GPU using 8 OpenMP threads, exploiting one GPU device for both the Purkinje and myocardium domains as this configuration demonstrated the best performance.

Figure 5 illustrates selected simulation results at *fine* mesh resolution for the identified optimal set of Purkinje coupling parameters from a total of 512 executions. This set was chosen based on similarity between the simulated and clinical ECG signals across all leads, as well as replicating a range of physiological anterograde PMJ delays across all Purkinje terminals when activating the myocardium. Nevertheless, the existence of other

combinations of Purkinje coupling parameters yielding a similar activation pattern (Fig. 5A,B) indicates that a range of cardiac digital twins can be approximated by the considered parameters, R_{PMJ} and N_{PMJ} . Ventricular activation started ≈ 48 ms after His-bundle pacing (Fig. 5C,D), with the whole biventricular domain being activated in around 110 ms (end of simulated QRS complex, Fig. 5E), also within the physiological range of 80–120 ms for healthy subjects³³. Moreover, the inclusion of heterogeneity in I_{Ks} from⁴⁶ generated a reasonable approximation for the T-wave (Fig. 5E), with an average PCC of 0.81 across all the 8 independent ECG leads (I, II, V1–V6). The results for the *coarse* mesh simulations are presented in the Supplementary Material section A.6 (see Supplementary Fig. S5).

The effects of anterograde PMJ delays were correctly recovered as shown in Fig. 5D, as well as in Fig. 5F for a PMJ site on the right ventricle. As it can be seen in the latter, the closest myocardial control volume to the Purkinje terminal could not be activated instantaneously, due to the source-sink mismatch between the Purkinje terminals and myocardial cells^{35,36,50}. The myocardial control volume only became entirely depolarised approximately 4.34 ms after the stimulus reached the PMJ site, exhibiting an initial brief spike followed by a distinct blunted depolarisation (Fig. 5F). This behaviour, reported in different experimental studies^{51,52} and attributed to the electrotonic effects at the junctions.

In terms of scalability, Table 2 summarises execution times for our considered submission scenarios. For the 1N1S scenario (1 computing node, 1 simulation), total execution times for the *coarse* and *fine* discretisations were around 15 and 221 min, respectively. Similar execution times without significant performance loss were observed when 4 concurrent simulations were executed in the same compute node (scenario 1N4S), with average total times around 22 and 210 min for the *coarse* and *fine* meshes, respectively. A similar behaviour was observed using the MPI batch feature for the 128N512S scenario, with execution times of approximately 22 and 212 min for *coarse* and *fine* resolutions, respectively. Considering the times for the MPI process to start and end, these values were around 23 and 302 min, respectively. From the results in Table 2, it also transpires that the most demanding component is the solution of the ODE system of the myocardium, contributing around 11 and 94 min for the *coarse* and *fine* mesh resolutions, respectively. This is explained due to the larger number of control volumes of this domain compared to the Purkinje one, making this section of the problem more computationally demanding and the overall bottleneck.

In addition, to highlight the performance gains achieved with the novel GPU-based linear system, we ran the *coarse* mesh simulation using the OG+PC and OG+PG configurations with varying numbers of OpenMP threads. For the OG+PC scenario, the total execution time was approximately 191.45 min with 8 OpenMP threads and 96.14 min with 32 threads. In contrast, the OG+PG configuration significantly reduced the total runtime to around 14.97 min and 13.47 min for 8 and 32 threads, respectively. Focusing on the time spent solving the myocardium PDE system, OG+PC required 178.07 min and 83.39 min, whereas OG+PG completed the same task in just 4.25 min and 3.56 min, when using 8 and 32 threads, respectively. In terms of GPU memory consumption for the *coarse* and *fine* meshes, approximately 1146 megabytes and 5544 megabytes are allocated in a single GPU when solved with the OG+PG configuration.

Additional results on the performed patient-specific simulations are presented in Fig. 6, where 9 representative simulations using the *fine* mesh show how the Purkinje coupling parameters can impact ventricular activation. Furthermore, in Fig. 6A there is further evidence supporting the existence of multiple possible simulations with a similar ECG. These results also indicate that, while sustaining analogous ECGs, different combinations of Purkinje coupling parameters can generate distinct distributions of PMJ delays across the Purkinje terminals. Based on that, certain Purkinje terminals exhibit more variability in their associated PMJ delays, which might indicate a more important role of such PMJs to the whole ventricular activation.

Another interesting result is presented in Fig. 6B, which illustrates the results from 22 patient-specific simulations using the *fine* mesh, all with an average PCC of 0.81 for the ECG. The analysis of local activation times (LATs) from the closest coupled myocardium control volume to each Purkinje terminal also reveals that, in almost all terminals, the average LAT does not significantly change (see left panel Fig. 6B). This is however with the exception of a small number of Purkinje terminals, which present larger LAT variability of up to 9.11 ms. Moreover, the analysis of PMJ delays revealed an even greater variability across Purkinje terminals (see right panel Fig. 6B), while still producing a similar ECG morphology. These findings further support the concept of

Mesh	Scenario	Total	Write	ECG	M_{ODE}	M_{PDE}	P_{ODE}	P_{PDE}	MPI_{end}
	1N1S	15.15	0.08	0.07	7.78	4.27	0.20	1.01	–
<i>coarse</i>	1N4S	22.24 ± 0.21	0.07 ± 0.00	0.11 ± 0.00	10.94 ± 0.11	7.23 ± 0.09	0.23 ± 0.00	1.05 ± 0.01	23.15
	128N512S	22.41 ± 0.32	0.07 ± 0.00	0.11 ± 0.02	11.06 ± 0.17	7.24 ± 0.15	0.23 ± 0.01	1.06 ± 0.01	23.55
	1N1S	221.49	0.52	0.42	97.62	87.70	0.26	1.13	–
<i>fine</i>	1N4S	210.66 ± 0.32	0.50 ± 0.00	0.42 ± 0.00	93.80 ± 0.18	84.81 ± 0.05	0.39 ± 0.01	1.13 ± 0.00	213.13
	128N512S	211.94 ± 7.33	0.49 ± 0.01	0.42 ± 0.00	93.87 ± 0.50	85.89 ± 5.56	0.37 ± 0.03	1.13 ± 0.01	302.29

Table 2. Execution times (minutes) for the patient-specific simulations for the 3 submission scenarios considering a total simulation time of $t_{max} = 600$ ms. For the *coarse* mesh the total time is around 15.15, 22.24 and 22.41 minutes for the 1N1S, 1N4S and 128N512S scenarios, respectively, while for the *fine* mesh the total time is approximately 221.49, 210.66 and 211.94 minutes for the same scenarios. *Total*: total time; *Write*: writing time; *ECG*: ECG computation time; M_{ODE}/M_{PDE} : time to solve the myocardium ODE/PDE system; P_{ODE}/P_{PDE} : time to solve the Purkinje ODE/PDE system; MPI_{end} : time for the MPI process to finish.

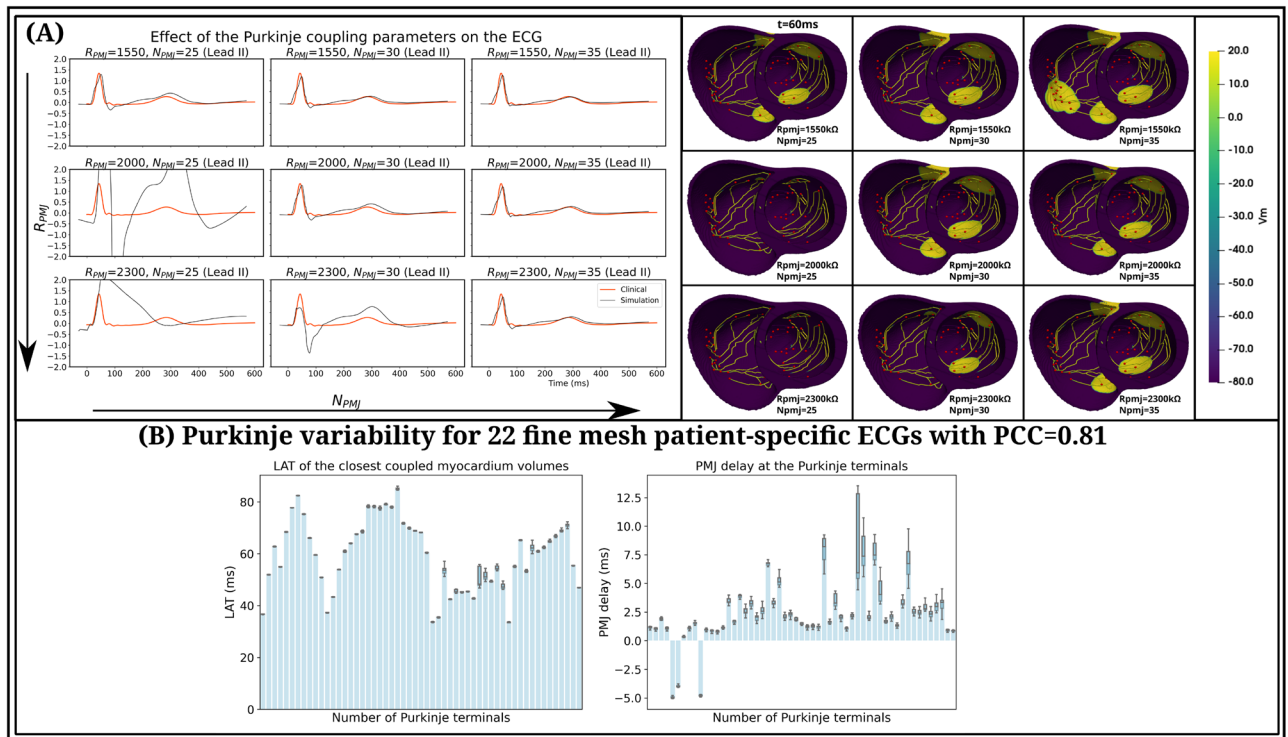


Fig. 6. (A) Effects of Purkinje coupling parameters R_{PMJ} and N_{PMJ} on the simulated ECG (Lead II) and on biventricular activation, considering the *fine* mesh. The left panel illustrates 9 simulated ECGs (black traces) for different combinations of R_{PMJ} and N_{PMJ} , compared against the clinical one (in red). The right panel illustrates the activation pattern at time $t = 60$ ms for each of the 9 corresponding simulations. (B) Comparison between 22 patient-specific simulations using the *fine* mesh and average PCC of 0.81 for different combinations of the Purkinje coupling parameters. Left panel, boxplots of local activation times (ms) from the closest coupled myocardial control volumes to each of the 55 terminals located in the generated Purkinje network, while in the right panel boxplots for the anterograde PMJ delay (ms) for each Purkinje terminal. The 3D models in panel (A) were generated using the last version of `MONOALG3D` (https://github.com/rsachetto/MonoAlg3D_C) and visualised with the Paraview tool version 5.13.2.

non-uniqueness in ventricular activation patterns, an observation recently highlighted by Gradits et al.⁵³. This suggests that, for a given patient, multiple plausible cardiac digital twin configurations may exist.

Finally, another relevant finding from these simulations is the existence of an almost linear relation between the Purkinje coupling parameters and the appearance of propagation block, as analysed by Figs. 5A and S5A (Supplementary Material section A.6).

Based on these results, the novel MPI batch feature illustrates the scalability and efficiency of the proposed cardiac solver to conduct patient-specific studies under cluster GPU environments and is a step towards cardiac digital twinning, enabling the correct adjustment of sensitive Purkinje coupling parameters to physiological ranges for anterograde PMJ delays.

Limitations

This study employed a regular voxel-based (cubic) discretisation of the cardiac geometry, which can introduce staircase approximations along curved surfaces. We acknowledge this geometric simplification; however, because our simulations are based on the monodomain model, a reaction–diffusion formulation for transmembrane potential and focus exclusively on activation propagation without extracellular stimulation, prior studies indicate that such surface approximations have only a modest influence on bulk conduction metrics^{54,55}. To ensure robustness, a mesh-refinement analysis was performed, showing activation time variations below 5.56% for a mesh resolution of $100 \mu\text{m}$, that can be verified in Supplementary Material section A.7. Therefore, we consider the staircase effect to have negligible impact on our main findings. Nevertheless, for applications involving extracellular potentials, defibrillation, or torso coupling, a boundary-fitted or smoothed mesh representation would be more appropriate.

Conclusion

In this work, we presented an open-source, high-performance GPU-based solver for cardiac electrophysiology simulations. Our main objective was to enhance the efficiency and scalability of large-scale simulations while incorporating physiologically relevant features such as the Purkinje conduction system. Through systematic evaluation across diverse CPU/GPU configurations, we demonstrated that executing both ODE and PDE

solvers on the GPU significantly accelerates computations, especially for complex human-based cellular models. Compared to our previous work, we found that transferring all required data structures to the GPU at initialization yields better performance than space adaptivity strategies. Additionally, patient-specific simulations with calibrated anterograde PMJ delays highlight the solver's suitability for realistic digital twin applications. The solver also demonstrated excellent scalability, with the ability to run 512 simulations concurrently across 128 compute nodes, achieving per simulation execution times close to the one using a single node and with linear speedups in terms of throughput. This scalability, combined with the solver's support for detailed conduction modelling and efficient output formats, positions it as a practical and accessible tool for large-scale in silico studies. To our knowledge, this is the first open-source solver to offer high-throughput simulations with detailed Purkinje-muscle coupling. These capabilities lay the foundation for digital twin frameworks and virtual patient cohorts, marking a step toward in silico clinical trials and personalized therapy evaluation.

Data availability

The source code for the open-source cardiac solver is publicly available at https://github.com/rsachetto/MonoAIg3D_C. All the necessary configuration files, custom functions, post-processing scripts used during the current study, as well as, the biventricular mesh and Purkinje networks used for the patient-specific simulations are publicly available at the Zenodo repository <https://doi.org/10.5281/zenodo.15041476> to allow reproducibility.

Received: 8 April 2025; Accepted: 22 December 2025

Published online: 18 February 2026

References

- Jean-Quartier, C., Jeanquartier, F., Jurisica, I. & Holzinger, A. In silico cancer research towards 3R. *BMC Cancer* **18**, 1–12 (2018).
- Delp, S. L. et al. OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE Trans. Biomed. Eng.* **54**, 1940–1950 (2007).
- Musuamba, F. T. et al. Scientific and regulatory evaluation of mechanistic in silico drug and disease models in drug development: Building model credibility. *CPT Pharmacometr. Syst. Pharmacol.* **10**, 804–825 (2021).
- Fenton, F. H., Cherry, E. M., Hastings, H. M. & Evans, S. J. Multiple mechanisms of spiral wave breakup in a model of cardiac electrical activity. *Chaos* **12**, 852–892 (2002).
- Lopez-Perez, A., Sebastian, R. & Ferrero, J. M. Three-dimensional cardiac computational modelling: methods, features and applications. *Biomed. Eng. Online* **14**, 1–31 (2015).
- Dasi, A. et al. In-silico drug trials for precision medicine in atrial fibrillation: From ionic mechanisms to electrocardiogram-based predictions in structurally-healthy human atria. *Front. Physiol.* **13**, 966046 (2022).
- Oliveira, R. S. et al. Ectopic beats arise from micro-reentries near infarct regions in simulations of a patient-specific heart model. *Sci. Rep.* **8**, 16392 (2018).
- Li, L. et al. Towards enabling cardiac digital twins of myocardial infarction using deep computational models for inverse inference. *IEEE Trans. Med. Imaging* **1**, 1 (2024).
- Corral-Acero, J. et al. The Digital Twin to enable the vision of precision cardiology. *Eur. Heart J.* **41**, 4556–4564 (2020).
- Niederer, S. A., Sacks, M. S., Girolami, M. & Willcox, K. Scaling digital twins from the artisanal to the industrial. *Nat. Comput. Sci.* **1**, 313–320 (2021).
- Viceconti, M. et al. In silico trials: Verification, validation and uncertainty quantification of predictive models used in the regulatory evaluation of biomedical products. *Methods* **185**, 120–127 (2021).
- Haissaguerre, M., Vigmond, E., Stuyvers, B., Hocini, M. & Bernus, O. Ventricular arrhythmias and the His-Purkinje system. *Nat. Rev. Cardiol.* **13**, 155 (2016).
- Kavanagh, P. *Open Source Software: Implementation and Management* (Elsevier, 2004).
- Plank, G. et al. The openCARP simulation environment for cardiac electrophysiology. *Comput. Methods Progr. Biomed.* **208**, 106223 (2021).
- Biasi, N., Seghetti, P., Parollo, M., Zucchelli, G. & Tognetti, A. A Matlab toolbox for cardiac electrophysiology simulations on patient-specific geometries. *Comput. Biol. Med.* **185**, 109529 (2025).
- Viola, F., Del Corso, G., De Paulis, R. & Verzicco, R. GPU accelerated digital twins of the human heart open new routes for cardiovascular research. *Sci. Rep.* **13**, 8230 (2023).
- Del Corso, G., Verzicco, R. & Viola, F. A fast computational model for the electrophysiology of the whole human heart. *J. Comput. Phys.* **457**, 111084 (2022).
- Kaboudian, A., Cherry, E. M. & Fenton, F. H. Real-time interactive simulations of complex ionic cardiac cell models in 2D and 3D heart structures with GPUs on personal computers. In *2021 Computing in Cardiology (CinC)*, vol. 48, 1–4 (2021).
- Vasconcellos, E. C., Clua, E. W., Fenton, F. H. & Zamith, M. Accelerating simulations of cardiac electrical dynamics through a multi-GPU platform and an optimized data structure. *Concurr. Comput. Pract. Exp.* **32**, e5528 (2020).
- Mena, A., Ferrero, J. M. & Matas, J. F. R. GPU accelerated solver for nonlinear reaction-diffusion systems, application to the electrophysiology problem. *Comput. Phys. Commun.* **196**, 280–289 (2015).
- Xia, Y., Wang, K. & Zhang, H. Parallel optimization of 3d cardiac electrophysiological model using GPU. *Comput. Math. Methods Med.* **2015**, 862735 (2015).
- Amorim, R. M. & dos Santos, R. W. Solving the cardiac bidomain equations using graphics processing units. *J. Comput. Sci.* **4**, 370–376 (2013).
- de Barros, B. et al. Simulations of complex and microscopic models of cardiac electrophysiology powered by multi-GPU platforms. *Comput. Math. Methods Med.* **2012**, 824569 (2012).
- Rocha, B. M. et al. Accelerating cardiac excitation spread simulations using graphics processing units. *Concurr. Comput. Pract. Exp.* **23**, 708–720 (2011).
- Sachetto Oliveira, R. et al. Comparing CUDA, OpenCL and OpenGL implementations of the cardiac monodomain equations. In *International Conference on Parallel Processing and Applied Mathematics* 111–120 (Springer, 2011).
- Berg, L. A. et al. Enhanced optimization-based method for the generation of patient-specific models of Purkinje networks. *Sci. Rep.* **13**, 11788 (2023).
- Riebel, L. L. et al. In silico evaluation of cell therapy in acute versus chronic infarction: role of automaticity, heterogeneity and purkinje in human. *Sci. Rep.* **14**, 21584 (2024).
- Sachetto Oliveira, R. et al. Performance evaluation of GPU parallelization, space-time adaptive algorithms, and their combination for simulating cardiac electrophysiology. *Int. J. Numer. Methods Biomed. Eng.* **34**, e2913 (2018).
- Vergara, C. et al. A coupled 3D–1D numerical monodomain solver for cardiac electrical activation in the myocardium with detailed Purkinje network. *J. Comput. Phys.* **308**, 218–238 (2016).

30. Sundnes, J. et al. On the computational complexity of the bidomain and the monodomain models of electrophysiology. *Ann. Biomed. Eng.* **34**, 1088–1097 (2006).
31. Mazumder, S. *Numerical Methods for Partial Differential Equations: Finite Difference and Finite Volume Methods* (Academic Press, 2015).
32. Gomes, J. M. et al. Uniformization method for solving cardiac electrophysiology models based on the markov-chain formulation. *IEEE Trans. Biomed. Eng.* **62**, 600–608 (2015).
33. Zipes, D. P., Jalife, J. & Stevenson, W. G. *Cardiac Electrophysiology: From Cell to Bedside E-book* (Elsevier, 2017).
34. Behradfar, E., Nygren, A. & Vigmond, E. J. The role of Purkinje-myocardial coupling during ventricular arrhythmia: a modeling study. *PLoS ONE* **9**, e88000 (2014).
35. Wiedmann, R. T., Tan, R. C. & Joyner, R. W. Discontinuous conduction at Purkinje-ventricular muscle junction. *Am. J. Physiol. Heart Circ. Physiol.* **271**, H1507–H1516 (1996).
36. dos Santos, R. W. et al. ATX-II effects on the apparent location of M cells in a computational model of a human left ventricular wedge. *J. Cardiovasc. Electrophysiol.* **17**, S86–S95 (2006).
37. Vigmond, E. J. & Clements, C. Construction of a computer model to investigate sawtooth effects in the Purkinje system. *IEEE Trans. Biomed. Eng.* **54**, 389–399 (2007).
38. Bishop, M. J. & Plank, G. Bidomain ECG simulations using an augmented monodomain model for the cardiac source. *IEEE Trans. Biomed. Eng.* **58**, 2297–2307 (2011).
39. Niederer, S. A. et al. Verification of cardiac tissue electrophysiology simulators using an N-version benchmark. *Phil. Trans. R. Soc. A* **369**, 4331–4351 (2011).
40. ten Tusscher, K. H., Noble, D., Noble, P.-J. & Panfilov, A. V. A model for human ventricular tissue. *Am. J. Physiol. Heart Circ. Physiol.* **286**, H1573–H1589 (2004).
41. Tomek, J. et al. Development, calibration, and validation of a novel human ventricular myocyte model in health, disease, and drug block. *Elife* **8**, e48890 (2019).
42. Rush, S. & Larsen, H. A practical algorithm for solving dynamic membrane equations. *IEEE Trans. Biomed. Eng.* **1**, 389–392 (1978).
43. Banerjee, A. et al. A completely automated pipeline for 3D reconstruction of human heart from 2D cine magnetic resonance slices. *Phil. Trans. R. Soc. A* **379**, 20200257 (2021).
44. Camps, J. et al. Inference of ventricular activation properties from non-invasive electrocardiography. *Med. Image Anal.* **73**, 102143 (2021).
45. Camps, J. et al. Digital twinning of the human ventricular activation sequence to Clinical 12-lead ECGs and magnetic resonance imaging using realistic purkinje networks for in silico clinical trials. *Med. Image Anal.* **94**, 103108 (2024).
46. Camps, J. et al. Harnessing 12-lead ECG and MRI data to personalise repolarisation profiles in cardiac digital twin models for enhanced virtual drug testing. *Med. Image Anal.* **100**, 103361 (2025).
47. Doste, R., Coppini, R. & Bueno-Orovio, A. Remodelling of potassium currents underlies arrhythmic action potential prolongation under beta-adrenergic stimulation in hypertrophic cardiomyopathy. *J. Mol. Cell. Cardiol.* **172**, 120–131 (2022).
48. Trovato, C. et al. Human Purkinje in silico model enables mechanistic investigations into automaticity and pro-arrhythmic abnormalities. *J. Mol. Cell. Cardiol.* **142**, 24–38 (2020).
49. Costa, C. M., Hoetzel, E., Rocha, B. M., Prassl, A. J. & Plank, G. Automatic parameterization strategy for cardiac electrophysiology simulations. In *Computing in Cardiology 2013* 373–376 (IEEE, 2013).
50. Li, J., Zhang, H. & Boyett, M. Numerical analysis of conduction of the action potential across the Purkinje fibre-ventricular muscle junction. *Comput. Cardiol.* **1**, 265–268 (2016).
51. Mendez, C., Mueller, W. J. & Uguaiaga, X. Propagation of impulses across the Purkinje fiber-muscle junctions in the dog heart. *Circ. Res.* **26**, 135–150 (1970).
52. Matsuda, K. Configuration of the transmembrane potential of the Purkinje-ventricular fiber junction and its analysis. In *Electrophysiology and Ultrastructure of the Heart* 177–187 (1967).
53. Grandits, T., Gillette, K., Plank, G. & Pezzuto, S. Accurate and efficient cardiac digital twin from surface ECGs: Insights into identifiability of ventricular conduction system. *Med. Image Anal.* **1**, 103641 (2025).
54. Potse, M., Dubé, B., Vinet, A. & Cardinal, R. A comparison of monodomain and bidomain propagation models for the human heart. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society* 3895–3898 (IEEE, 2006).
55. Mountris, K. A. & Pueyo, E. Cardiac electrophysiology meshfree modeling through the mixed collocation method. *Appl. Sci.* **13**, 11460 (2023).

Acknowledgements

This work was funded by a Wellcome Trust fellowship in Basic Biomedical Sciences to B.R. (214290/Z/18/Z), the EPSRC project CompBioMedX (EP/X019446/1), the CompBioMed2 Centre of Excellence in Computational Biomedicine grant agreements No. 675451 and No. 823712. L.A.B acknowledges support from CNPq grant No. 446127/2024-8 and the Federal University of Juiz de Fora. R.S.O. acknowledges support from Fapemig grant No. APQ-00748-18 and UFSJ. R.W.S. acknowledges support from Fapemig grant APQ-02445-24, “Coordenação de Aperfeiçoamento de Pessoal de Nível Superior” (CAPES), “Financiadora de Estudos e Projetos” (Finep), National Council for Scientific and Technological Development (CNPq), the National Laboratory for Scientific Computing (LNCC/MCTI, Brazil) for providing HPC resources of the SDumont supercomputer SINAPAD, and the Federal University of Juiz de Fora (UFJF). A.B.O. acknowledges support from UK Research and Innovation grant No. 10110728. The U.S. Department of Energy’s (DOE) Innovative and Novel Computational Impact on Theory and Experiment (INCITE) Program awarded access to Polaris, under contract No. DE-AC02-06CH11357. The project that gave rise to these results received the support of a fellowship from “la Caixa” Foundation (ID 100010434) to J.C. (code LCF/BQ/PI25/12100029). For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) public copyright licence to any Author Accepted Manuscript version arising from this submission.

Author contributions

L.A.B.: Conceptualization, Methodology, Software, Investigation, Formal analysis, Writing—original draft, Writing—review & editing. R.S.O.: Conceptualization, Methodology, Software, Investigation, Formal analysis, Writing—original draft, Writing—review & editing. J.C.: Software, Formal analysis, Writing—review & editing. L.M.R.L.: Software, Formal analysis, Writing—review & editing. J.O.C.: Software, Formal analysis, Writing—review & editing. Z.J.W.: Software, Formal analysis, Writing—review & editing. R.D.: Data curation, Formal analysis, Writing—review & editing. A.B.O.: Formal analysis, Writing—review & editing. R.W.S.: Conceptualization, Methodology, Investigation, Formal analysis, Writing—original draft, Writing—review & editing. B.R.:

Conceptualization, Formal analysis, Funding acquisition, Resources, Writing—review & editing.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-025-33709-w>.

Correspondence and requests for materials should be addressed to L.A.B. or R.S.O.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2025