

Centre of Excellence
MICROCARD-2



Deliverable 3.1 (confidential)

AMG and BDDC scalability report

PROJECT INFORMATION	
Grant agreement number	101172576
Project acronym	MICROCARD-2
Project title	Numerical modeling of cardiac electrophysiology at the cellular scale
Start date of the project	1 November 2024
Duration of the project	30 months
Call	HORIZON-EUROHPC-JU-2023-COE-03 – Centres Of Excellence For Exascale HPC Applications
Topic	HORIZON-EUROHPC-JU-2023-COE-03-01
Scientific coordinator	Mark Potse (UBx)

DELIVERABLE INFORMATION	
Deliverable number	3.1
Dissemination level	sensitive
Work package	3
Tasks	3.1, 3.3
WP leader	Luca F. Pavarino
Lead beneficiary	UNIPV
Authors	Luca F. Pavarino, Simone Scacchi
Reviewers	
Due date of Deliverable	31 October 2025 (M12)
Submission date	31 October 2025



EuroHPC
Joint Undertaking



**Co-funded by
the European Union**

Contents

PROJECT SUMMARY	4
DELIVERABLE SUMMARY	4
1 Introduction	6
2 The cardiac EMI model	7
2.1 Discretization in time and space	8
2.1.1 Time Discretization	8
2.1.2 Space discretization	8
3 EMI-AMG preconditioners	9
3.1 Simplified regular geometries	10
3.1.1 EMI-AMG weak scaling	11
3.1.2 EMI-AMG strong scaling	11
3.2 Realistic myocyte geometries	15
4 3D EMI-BDDC preconditioners	17
4.1 Simplified regular geometries	17
4.1.1 EMI-BDDC weak scaling	18
4.1.2 BDDC Robustness w.r.t. conductivity coefficients	19
4.2 Realistic myocyte geometries	19
References	20

Authors Luca F. Pavarino, Simone Scacchi

History of changes

version	date	change history	authors	organisation
V1	30/09/2025	Initial version	Luca F. Pavarino, Simone Scacchi	UNIPV
V2	27/10/2025	ready for submission	Luca F. Pavarino, Simone Scacchi	UNIPV

List of beneficiaries

nr	organisation name	short name	country	PI
1	Université de Bordeaux	UBx	France	Mark Potse
2	Université de Strasbourg	UNISTRA	France	Vincent Loechner
3	Simula Research Laboratory AS	SIMULA	Norway	Aslak Tveito
4	Università degli studi di Pavia	UNIPV	Italy	Luca Pavarino
5	Technische Universität München	TUM	Germany	Hartwig Anzt
6	Karlsruhe Institute of Technology	KIT	Germany	Axel Loewe
7	Zuse Institute Berlin	ZIB	Germany	Martin Weiser
8	MEGWARE Computer Vertrieb und Service GMBH	MEGWARE	Germany	Axel Auweter
9	Inria	Inria	France	Olivier Aumage
10	Università di Trento	UTrento	Italy	Simone Pezzuto

PROJECT SUMMARY

Cardiac function is coordinated by an electric system whose disorders are among the most frequent causes of death and disease. Numerical models of this complex system are mature and widely used, but to match observations in aging and diseased hearts they need to move from a continuum approach to a representation of individual cells and their interconnections. This makes the problem more complex, harder to solve, and four orders of magnitude larger, necessitating exascale computers.

The EuroHPC-2019 MICROCARD project developed a simulation platform that can meet this challenge, by a joint effort of HPC experts, numerical scientists, biomedical engineers, and biomedical scientists, from academia and industry. The Centre of Excellence MICROCARD-2 will consolidate and scale up the MICROCARD results enabling digital twins of cardiac tissue.

We will further develop MICROCARD's numerical schemes, moving to second-order spatial discretization. We will develop mixed-precision preconditioners and data compression to reduce communication bandwidth. The highly successful efforts made in MICROCARD towards automated compilation of high-level model descriptions into optimized, energy-efficient system code for different CPUs and GPUs will be extended to upcoming architectures. We will continue efforts to robustify parallel remeshing software and add necessary functionality for parallel mesh partitioning and production of realistic synthetic tissue meshes needed for simulations.

The platform will be benchmarked with realistic test cases and be made accessible for a wide range of users with tailored workflows. It will be adaptable to similar biological systems such as nerves, and several of our products such as improved solvers, preconditioners, remeshers, and partitioners will be reusable in a wide range of applications.

DELIVERABLE SUMMARY

This Deliverable D3.1 reports the results obtained in the WP3 Task 3.1 – Dual-primal preconditioners for hybrid CPU/GPU architectures (UNIPV, ZIB, TUM, Simula, M1–30). We report on both Algebraic MultiGrid (AMG) and Balancing Domain Decomposition by Constraints (BDDC) preconditioners for the cardiac Extracellular - Membrane - Intracellular (EMI) system.

Using a composite Discontinuous Galerkin (DG) discretization, we introduce an EMI-AMG solver for the three dimensional EMI model. Our investigation includes the EMI-AMG scalability performance, both weak and strong, and evaluates its numerical robustness under ischemic conditions, addressing the challenges of heterogeneous media. Numerical tests exploit state-of-the-art pre-exascale supercomputers with hybrid CPU-GPU architectures. The results indicate better scalability performance of the EMI-AMG solver on CPUs compared to GPUs. However, the best solution times achieved using GPUs are up to 40x faster than those obtained on CPUs.

We then construct and analyze a BDDC preconditioner for the solution of three dimensional composite Discontinuous Galerkin discretizations of reaction-diffusion systems of ordinary and partial differential equations arising in the cardiac EMI Model. The modeling of each individual

cardiac cell results in discontinuous global solutions across cell boundaries, requiring the careful construction of dual and primal spaces for the BDDC preconditioner. We provide a scalable condition number bound for the preconditioning operator and validate the theoretical results with extensive numerical experiments.

1 Introduction

Our work in MICROCARD-2 WP3 built on our preliminary work in 2D and idealized cell geometries, which led to two 2023 papers [1, 2] and the Technical Report [3]. This preliminary work showed that scalable and efficient dual-primal preconditioners, such as Balancing Domain Decomposition by Constraints (BDDC), can be extended to the more recent EMI models targeted in the MICROCARD EuroHPC project and the MICROCARD-2 CoE [4, 5]. These preliminary results were based in turn on our previous work on parallel scalable preconditioners for classical cardiac models such as the Monodomain and Bidomain models, see [6].

Here we report on our MICROCARD-2 work in Task 3.1, focusing on both Algebraic Multigrid (AMG) and BDDC preconditioners for cell-by-cell EMI models, leveraging our previous experience with the PETSc and Ginkgo parallel libraries. We refer to [7] for more details on our EMI-AMG results and to [8] for our 3D EMI-BDDC results. The BDDC implementation in this Task 3.1 employs two levels, while multilevel BDDC approaches will be considered in Task 3.5.

The results obtained show that EMI-BDDC is scalable in the number of subdomains (cells), quasi-optimal in the ratio H/h (local problem size), and robust with respect to the main problem parameters such as time step size and EMI conductivities. Parallel tests on realistic 3D geometries indicate that these results hold also in more complex settings, but additional tests are required to assess the exascale performance of EMI-BDDC preconditioners on hybrid CPU/GPU architectures.

Our EMI-AMG preconditioners were implemented and tested with the PETSc parallel library [9], one of the parallel HPC backends of OpenCARP. The results show a good performance on the parallel machines we used (e.g. the EuroHPC/CINECA LEONARDO supercomputer), achieving fast execution times on GPU nodes. The results also show that the EMI-AMG preconditioners are not as scalable as the EMI-BDDC preconditioners, but additional test are required to assess their exascale performance.

In addition to the PETSc parallel library [9], we also used the Kaskade finite element library [10] from ZIB to generate the regular structured meshes and associated discrete EMI systems in the preliminary work [2] and the 3D extension [8]. The use of another OpenCARP backend, GINKGO [11], focusing on GPU portability, has been studied in WP2. Since our preconditioners need to be tailored to both the specific choice of space-time discretization of the cell-by-cell model developed in WP1 and the iterative solvers developed in WP2, we often worked in collaboration with WP1 and WP2.

2 The cardiac EMI model

We recall here the formulation of the cardiac cell-by-cell/EMI model [5, 12–14] adopted in this report. Let $\Omega \in \mathbb{R}^3$ be the cardiac tissue domain, composed by $N + 1$ subdomains, representing N cells Ω_i $i = 1, \dots, N$ immersed in the extracellular space, denoted by Ω_0 . The cells are interconnected through the gap junctions, which are protein channels allowing the exchange of ionic species between the cells, while the ionic channels in the cellular membrane allow the ionic current flow between the cells and the extracellular space.

The EMI model consists of the following system of PDEs coupled with a stiff ODE system:

$$\begin{cases} -\operatorname{div}(\sigma_0 \nabla u_0) = I_{\text{app}} & \text{in } \Omega_0, \\ -\operatorname{div}(\sigma_i \nabla u_i) = 0 & \text{in } \Omega_i, \quad i = 1, \dots, N \\ -n_i^T \sigma_i \nabla u_i = C_m \frac{\partial v_{ij}}{\partial t} + F(v_{ij}, w) & \text{on } E_{ij} = \overline{\Omega}_i \cap \overline{\Omega}_j \subset \partial \Omega_i \\ \frac{\partial w}{\partial t} - R(v_{ij}, w) = 0 & \text{on } E_{i0}, \quad i = 1, \dots, N \end{cases} \quad (1)$$

where $v_{ij} = u_i - u_j$ represents the jump in the value of the electric potentials u_i between cells i and j across the common boundary E_{ij} between Ω_i and Ω_j , σ_i is the conductivity coefficient for the cell Ω_i , n_i is the outward normal to Ω_i , C_m is the membrane capacitance per unit area of the membrane surface, and

$$F(v_{ij}, w) = \begin{cases} I_{\text{ion}}(v_{ij}, w) & \text{on the cellular membrane,} \\ G(v_{ij}) & \text{on the gap junctions,} \end{cases} \quad (2)$$

where I_{ion} is the ionic current flowing through the cellular membrane and G is the gap junction current, assumed to be a linear function of the potential jump v . In order to guarantee the existence of solution, the applied current I_{app} must satisfy the compatibility condition $\int_{\Omega_0} I_{\text{app}} dx = 0$. Since the potential is unique up to a constant, we fix the constant by imposing zero average over Ω_0 for the extracellular component of the solution u_0 .

The last two equations in (1) represent the membrane model, an ODE system which describes the ion flow dynamics. Here, we denote with w the gating variables, related to the behavior of the ionic channels through the cellular membrane. R in this setting is a (often non-linear) function of v_{ij} and w , describing the variation in time of the above quantities. In this work, we adopt the Aliev-Panfilov membrane model [15].

Integrating by parts over Ω_i the first two equations in (1) and by applying the third equation on the membrane, we derive the weak formulation of the i -th problem: find $u_i(\cdot) : (0, T) \rightarrow H^1(\Omega_i)$ such that

$$\int_{\Omega_i} \sigma_i \nabla u_i \nabla \phi_i dx + \sum_{\substack{j \in \mathcal{E}_i^0 \\ j \neq i}} \int_{E_{ij}} \left(C_m \frac{\partial v_{ij}}{\partial t} + F(v_{ij}, w) \right) \phi_i ds = 0, \quad \forall \phi_i \in H^1(\Omega_i).$$

Summing the contributions from all $N + 1$ subdomains, the global problem is given by:

$$\sum_{i=1}^N \int_{\Omega_i} \sigma_i \nabla u_i \nabla \phi_i dx + \frac{1}{2} \sum_{i=1}^N \sum_{\substack{j \in \mathcal{E}_i^0 \\ j \neq i}} \int_{E_{ij}} \left(C_m \frac{\partial [\![u]\!]_{ij}}{\partial t} + F([\![u]\!]_{ij}, w) \right) [\![\phi]\!]_{ij} ds = 0,$$

where we denote $\llbracket u \rrbracket_{ij} = u_i - u_j$ as the jump due to the discontinuity between the values of the electric potential u_i in subdomain Ω_i and u_j in the adjacent subdomain Ω_j along the common boundary $E_{ij} \subset \partial\Omega_i$.

2.1 Discretization in time and space

We report here the time and space discretizations; more details and further motivations can be found in our previously published work [1].

2.1.1 Time Discretization

As time discretization, we use an implicit-explicit (IMEX) scheme, where the diffusion term is treated implicitly and the reaction term explicitly. We subdivide the time interval $[0, T]$ into K intervals and, by defining the time step $\tau = t^{(k+1)} - t^{(k)}$, for $k = 0, \dots, K$, we derive the following scheme:

$$w^{(k+1)} = w^{(k)} + \tau R(\llbracket u^{(k)} \rrbracket_{i0}, w^{(k)}) \quad i = 1, \dots, N,$$

$$\begin{aligned} & \sum_{i=1}^N \sum_{\substack{j \in \mathcal{E}_i^0 \\ j \neq i}} \int_{E_{ij}} C_m \llbracket u^{(k+1)} \rrbracket_{ij} \llbracket \phi \rrbracket_{ij} ds + 2\tau \sum_{i=1}^N \int_{\Omega_i} \sigma_i \nabla u_i^{(k+1)} \nabla \phi_i dx \\ &= \sum_{i=1}^N \sum_{\substack{j \in \mathcal{E}_i^0 \\ j \neq i}} \int_{E_{ij}} C_m \llbracket u^k \rrbracket_{ij} \llbracket \phi \rrbracket_{ij} ds - \tau \sum_{i=1}^N \sum_{\substack{j \in \mathcal{E}_i^0 \\ j \neq i}} \int_{E_{ij}} F(\llbracket u^{(k)} \rrbracket_{ij}, w^{(k+1)}) \llbracket \phi \rrbracket_{ij} ds. \end{aligned}$$

We define:

$$\begin{aligned} a_i(u_i, \phi_i) &:= \int_{\Omega_i} \sigma_i \nabla u_i \nabla \phi_i dx, \\ p_i(u_i, \phi_i) &:= \frac{1}{2} \sum_{\substack{j \in \mathcal{E}_i^0 \\ j \neq i}} \int_{E_{ij}} C_m \llbracket u \rrbracket_{ij} \llbracket \phi \rrbracket_{ij} ds, \\ f_i(\phi_i) &:= \frac{1}{2} \sum_{\substack{j \in \mathcal{E}_i^0 \\ j \neq i}} \int_{E_{ij}} \left(C_m \llbracket u \rrbracket_{ij} \llbracket \phi \rrbracket_{ij} - \tau F(\llbracket u \rrbracket_{ij}, w^{(k+1)}) \llbracket \phi \rrbracket_{ij} \right) ds, \\ d_i(u_i, \phi_i) &:= \tau a_i(u_i, \phi_i) + p_i(u_i, \phi_i). \end{aligned}$$

2.1.2 Space discretization

For the space discretization, we used a finite-element setting, employing three-dimensional linear elements (trilinear Q1 hexahedral elements for regular structured meshes or bilinear P1 tetrahedral elements for realistic unstructured meshes). Let $V_i(\bar{\Omega}_i)$ be the regular finite element

space of piecewise continuous linear functions in $\overline{\Omega}_i$ and define $V(\Omega) = V_0(\overline{\Omega}_0) \times \cdots \times V_N(\overline{\Omega}_N)$ as the global finite element space. In this scenario the general problem reads as follows: find $u = \{u_i\}_{i=0}^N \in V(\Omega)$ such that

$$d_h(u, \phi) = f(\phi), \quad \forall \phi = \{\phi_i\}_{i=0}^N \in V(\Omega), \quad (3)$$

where $d_h(u, \phi) := \sum_{i=0}^N d_i(u_i, \phi_i) = \sum_{i=0}^N (\tau a_i(u_i, \phi_i) + p_i(u_i, \phi_i))$, $f(\phi) := \sum_{i=0}^N f_i(\phi_i)$. Thus, the problem (3) can be written in matrix form as

$$\mathcal{K}\mathbf{u} = \mathbf{f}, \quad \text{with } \mathcal{K} = \sum_{i=0}^N \mathcal{K}_i, \quad \mathcal{K}_i = \tau A_i + M_i \quad (4)$$

where A_i and M_i are the local stiffness and mass matrices, respectively. We notice that the mass matrix is built only on the common interfaces E_{ij} between two subdomains Ω_i and Ω_j . The matrices \mathcal{K}_i , A_i , and M_i are all square matrices, with dimensions corresponding to the number of degrees of freedom (DOFs) [1].

3 EMI-AMG preconditioners

Algebraic Multigrid (AMG) algorithms [16] solve a given linear system by cycling through levels of “coarse,” smaller linear systems and using interpolated updates to improve the solution on the original space. This grid correction addresses low-frequency components of the error, while the higher-frequency components are efficiently mitigated through relaxation, namely the application of an iterative solver, e.g., Gauss-Seidel or Jacobi.

In a two-level AMG scheme, the residual equation $\mathcal{K}e = r$, where e and r are the approximation error and the residual, respectively, is solved on a coarser grid, the solution is interpolated back to the finer grid, and the approximation is updated as $u^{(k)} \leftarrow u^{(k)} + e^{(k)}$. This idea extends recursively to multilevel schemes by constructing a hierarchy of grids $\{\Gamma^{(k)}\}_{k \in \mathbb{N}}$, derived from the graph associated with the matrix \mathcal{K} . The multilevel scheme requires:

1. A hierarchy of M grids $\Gamma^1 \supset \Gamma^2 \supset \cdots \supset \Gamma^M$, from finest to coarsest.
2. Corresponding grid operators $\mathcal{K}^1, \mathcal{K}^2, \dots, \mathcal{K}^M$.
3. Interpolation operators P^1, P^2, \dots, P^{M-1} .
4. Restriction operators R^1, R^2, \dots, R^{M-1} .
5. Smoothers S^1, S^2, \dots, S^{M-1} .

In the first phase of the algorithm, called *setup phase*, all these objects are constructed. Then, the solution proceeds with a recursively defined cycle, often implemented following a V-cycle scheme, as detailed in [17].

The relaxation usually employs a Richardson iteration of the form $u_{j+1}^{(k)} = u_j^{(k)} + \omega S^{-1}(f - \mathcal{K}u_j^{(k)})$, where S can be $\text{diag}(\mathcal{K})$ (Jacobi), the lower part of \mathcal{K} (Gauss-Seidel) or the ILU approximation of \mathcal{K} , and ω is the relaxation factor.

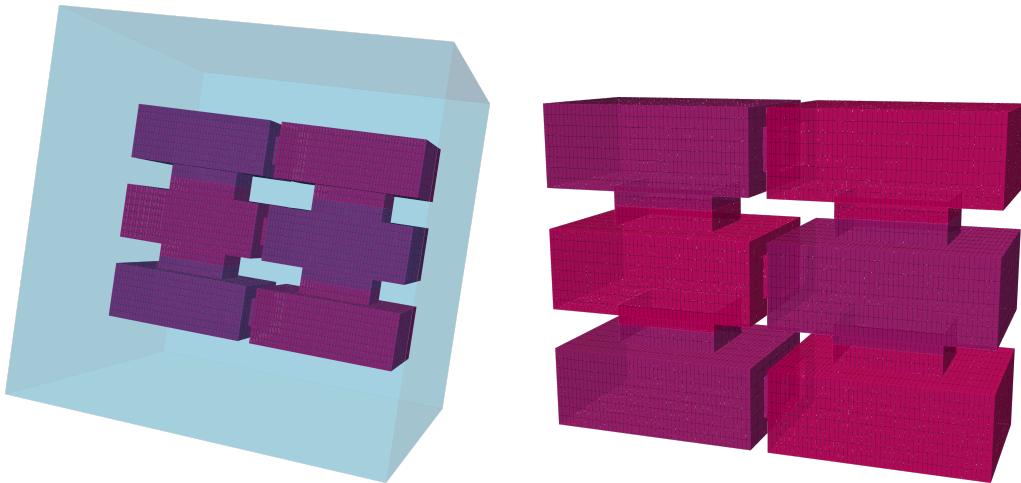


Figure 1: The considered three-dimensional geometry: $3 \times 2 \times 1$ cells surrounded by the extracellular liquid. Notice the interconnection between cells due to gap junctions, here represented as smaller blocks. On the right, a closeup of the cells discretization.

In this report, we solve the symmetric positive definite EMI system using the Preconditioned Conjugate Gradient (PCG) method, with two widely used AMG implementations as preconditioners. Both are integrated into PETSc, the Portable, Extensible Toolkit for Scientific Computation [9], which provides a comprehensive suite of scalable parallel solvers for linear and nonlinear equations, ODE integrators, and optimization algorithms. The first preconditioner is GAMG, PETSc's native AMG solver, while the second is BoomerAMG, accessible via a wrapper for the Hypre library [18].

These AMG implementations differ significantly in their multigrid strategies. BoomerAMG adopts a classical approach based on a strong threshold for coarsening, while GAMG relies on a smoothed aggregation strategy. Several algorithms exist for selecting the coarse space in both implementations, but we employ the default options. GAMG uses a modified Maximal Independent Set (MIS) algorithm, whereas Hypre employs a Hybrid Maximal Independent Set (HMIS) algorithm [17]. These algorithms are involved in the definition of a *threshold parameter*, which rules the coarsening of the original iteration matrix, affecting the overall performance of the solver in terms of iterations and solution time per timestep. For our problem, we fixed the threshold (GAMG) and the strong threshold (Hypre) values to 0.02 and 0.25 respectively.

3.1 Simplified regular geometries

In order to test the numerical performance of the AMG implementations for the EMI model (EMI-AMG), simplified regular geometries were considered first. These consist in a sequence of repetitive parallelepipeds (idealizing the myocytes, thus they hold a 6:1 proportion ratio between their lengths and heights/depths) connected through smaller blocks that represent ionic gap junctions; the obtained geometry is immersed in a cube, which in turns represent the extracellular space. A visual representation of such geometry is depicted in figure 1; such geometries were generated with an *in-house* Fortran-90 code.

3.1.1 EMI-AMG weak scaling

Weak scaling tests were performed for the three configurations with different ionic models, in order to test the robustness of the EMI-AMG preconditioner with respect to different membrane properties. The three configurations are denoted by EMI-AP (EMI with Aliev-Panfilov ionic model), EMI-LRR (EMI with Luo-Rudy-Ranjan ionic model) and EMI-Is-LRR (EMI-LRR with augmented potassium concentration in a central square of cells in order to simulate the effect of ischemia). Further details on the numerical setup can be found in [7]. The local size of the problem, namely the number of cells per processor, is fixed, and PCG iterations and mean solution time per simulation timestep are studied varying the number of processors. The EMI-AP tests were performed also on GPU, both on Marconi100 and LEONARDO, while the other tests were performed on Galileo100. On GPU, we considered from 1 up to 64 devices, corresponding to 1 up to 8 computing nodes. Each GPU handles $2 \times 1 \times 1$ cells (plus the extracellular space, which is fixed in size), for a total of 105k DOFs. On 64 GPUs the problem size is around 3 millions of DOFs. Regarding the CPU case, we considered from 4 up to 512 CPUs and each unit of 4 CPUs handles $4 \times 2 \times 1$ cells, corresponding roughly to 281k DOFs.

In [table 1](#) we report the results of the tests on Marconi100, while in [table 2](#) we report results of the tests on LEONARDO for the EMI-AP case. We observe a general increase in the mean solution time per timestep when the number of processors is increased, instead of the almost constant mean solution time expected by an optimal scaling. However, the increase in the solution time does not exceed one order of magnitude, with generally better scaling in the CPU case, but significantly improved solution times in the GPU case. We can also observe a comparison of the performance between the Hypre GPU case on Marconi100 and LEONARDO, where the updated hardware on the latter machine yields improved scaling, solution time and overall performance compared to the former, limiting also the effects of load imbalance which can be regarded as one of the possible issues affecting the scalability.

In [figure 2](#), we show the results of the weak scaling tests for the non-ischemic and EMI-Is-LRR case, respectively, both tested only on CPU. Again, we do not observe optimal scalability, although the increments in terms of mean solution time per timestep are below one order of magnitude. Furthermore, while Hypre seems to dominate the performance both in terms of iterations and time up to 64 processors, GAMG shows generally better results for a larger number of processors, for both the ischemic and non-ischemic case. As a general trend, Hypre tends to show a more significant increment in the number of iterations, compared to GAMG.

3.1.2 EMI-AMG strong scaling

The strong scaling tests feature tests on both GPU and CPU on Marconi 100 and LEONARDO, for the EMI-AP case, and tests on CPU on Galileo100 for the two LRR cases. In the following, we have fixed the global size of the problem to an $8 \times 8 \times 1$ cells domain, with a total number of about 1.5 million DOFs, and we have measured the number of solver iterations and the mean solution time per timestep. The GPU tests were performed from 4 (maximum number of devices on a single node) up to 64 (16 nodes) GPUs, while for the CPU case we considered from 4 up to 512 processors, having a maximum number of 32 CPUs per node.

GPU/CPU	Cells	DOFs	Hypre GPU		Hypre CPU		GAMG CPU	
			it	t_{sol} (s)	it	t_{sol} (s)	it	t_{sol} (s)
1	$2 \times 1 \times 1$	105148	20.9	0.06				
2	$2 \times 2 \times 1$	150953	23.8	0.12				
4	$4 \times 2 \times 1$	281803	27.5	0.14	18.1	6.0	28.4	8.0
8	$4 \times 4 \times 1$	453301	31.1	0.15	21.8	6.0	31.5	7.4
16	$8 \times 4 \times 1$	874777	40.1	0.22	28.0	7.6	37.5	8.6
32	$8 \times 8 \times 1$	1537325	47.5	0.36	32.7	9.3	39.3	8.7
64	$16 \times 8 \times 1$	3019381	65.7	0.72	45.7	12.8	48.4	10.6
128	$16 \times 16 \times 1$	5622685			59.8	17.2	52.4	10.9
256	$32 \times 16 \times 1$	11143213			89.8	26.4	70.4	14.8
512	$32 \times 32 \times 1$	21462653			106.9	32.8	80.1	19.1

Table 1: Weak scaling on Marconi100, EMI-AP model.

GPU/CPU	Cells	DOFs	Hypre GPU		Hypre CPU		GAMG CPU	
			it	t_{sol} (s)	it	t_{sol} (s)	it	t_{sol} (s)
1	$2 \times 1 \times 1$	105148	21.0	0.04				
2	$2 \times 2 \times 1$	150953	23.9	0.07				
4	$4 \times 2 \times 1$	281803	27.9	0.09	18.1	4.8	29.8	7.4
8	$4 \times 4 \times 1$	453301	31.5	0.10	21.7	5.8	31.3	7.1
16	$8 \times 4 \times 1$	874777	40.2	0.13	28.1	10.7	38.3	10.3
32	$8 \times 8 \times 1$	1537325	47.6	0.15	32.7	11.3	40.4	13.0
64	$16 \times 8 \times 1$	3019381	66.1	0.22	45.9	16.6	50.3	17.4
128	$16 \times 16 \times 1$	5622685			59.9	38.3	54.6	51.6
256	$32 \times 16 \times 1$	11143213			88.1	99.6	71.8	87.1
512	$32 \times 32 \times 1$	21462653			100.5	116.2	81.9	157.9

Table 2: Weak scaling on LEONARDO, EMI-AP model.

GPU/CPU	Hypre GPU		Hypre CPU		GAMG CPU	
	it	t_{sol} (s)	it	t_{sol} (s)	it	t_{sol} (s)
4	47.3546	0.5723	32.0979	54.0009	40.2087	60.7175
8	47.0599	0.5925	33.1868	28.8100	38.7712	29.9661
16	47.5944	0.2755	33.6513	15.2780	38.7742	15.3349
32	47.5045	0.2283	32.6374	9.3404	39.1838	8.7190
64	47.6444	1.1812	33.2338	4.8057	39.4905	4.4844
128			36.9311	3.8548	39.8442	2.5530
256			37.8222	1.9635	40.0839	1.3454
512			39.0539	3.9485	40.0689	1.7245

Table 3: Strong scaling on Marconi100, EMI-AP model. The global problem is solved for $8 \times 8 \times 1$ cells (≈ 1.5 million DOFs).

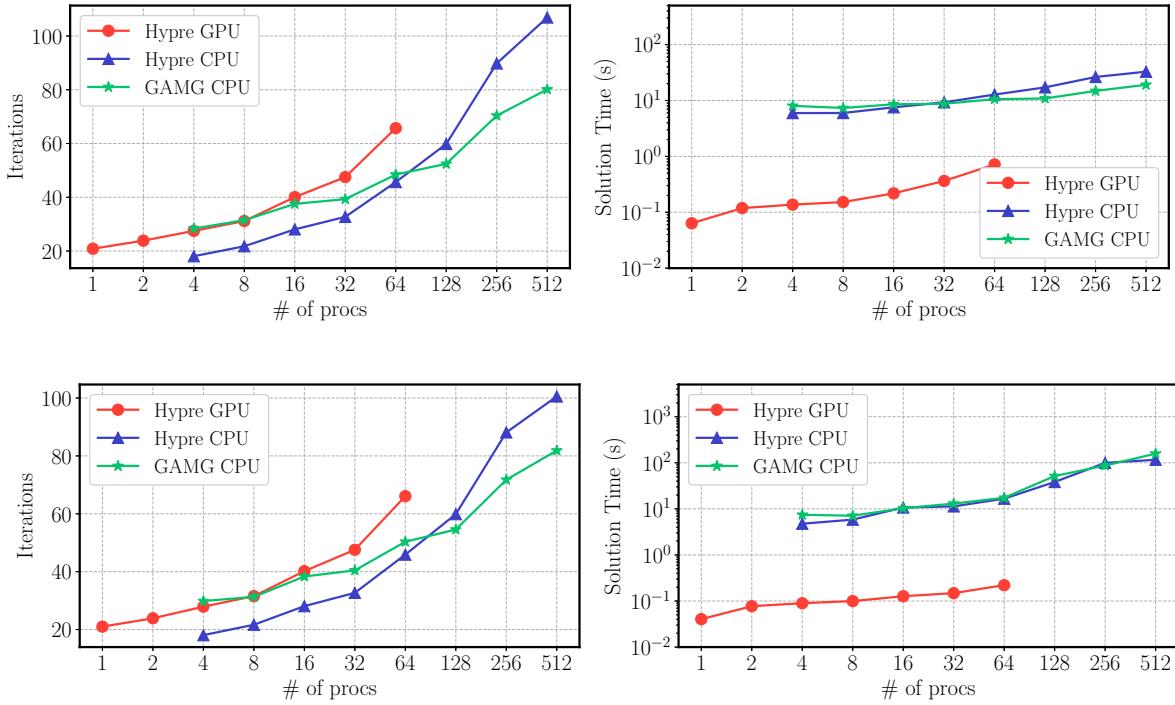


Figure 2: Weak scaling on Marconi100 (top) and LEONARDO (bottom). For each case, we report the average PCG iterations (left) and solution time (right) per timestep as a function of the number of processors.

GPU/CPU	Hypre GPU		Hypre CPU		GAMG CPU	
	it	t_{sol} (s)	it	t_{sol} (s)	it	t_{sol} (s)
4	47.6064	0.3512	32.0979	44.4694	40.4396	59.0130
8	47.4426	0.2434	33.0410	28.6452	40.3776	32.1137
16	47.7552	0.1802	33.6334	21.9026	40.4765	20.2133
32	47.5664	0.1907	32.6823	11.2943	40.2777	13.1915
64	48.1918	0.1695	33.2987	6.5457	40.1988	7.3561
128			36.5125	10.1462	40.7562	10.6399
256			37.7193	15.4397	40.9650	7.9058
512		39.1159	66.1487	41.7672	24.8621	

Table 4: Strong scaling on Leonardo, EMI-AP model. The global problem is solved for $8 \times 8 \times 1$ cells (≈ 1.5 million DOFs).

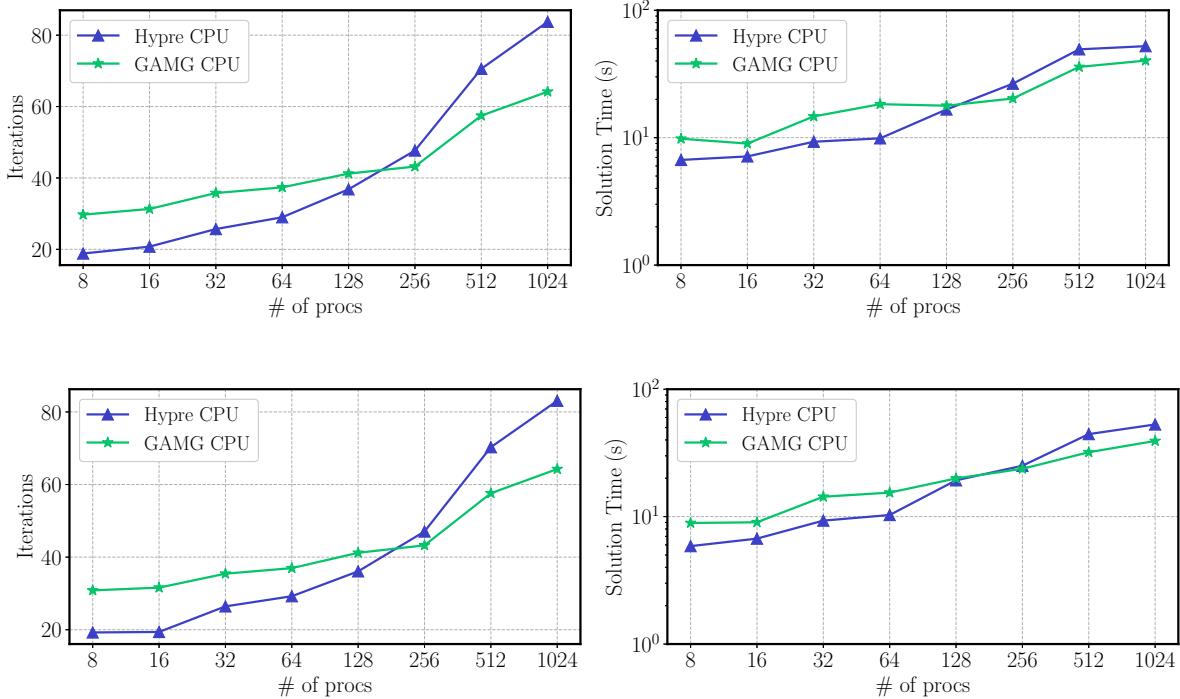


Figure 3: Weak scaling on G100 for EMI-LRR (top) and EMI-Is-LRR (bottom). For each case, we report the average PCG iterations (left) and solution time (right) per timestep as a function of the number of processors.

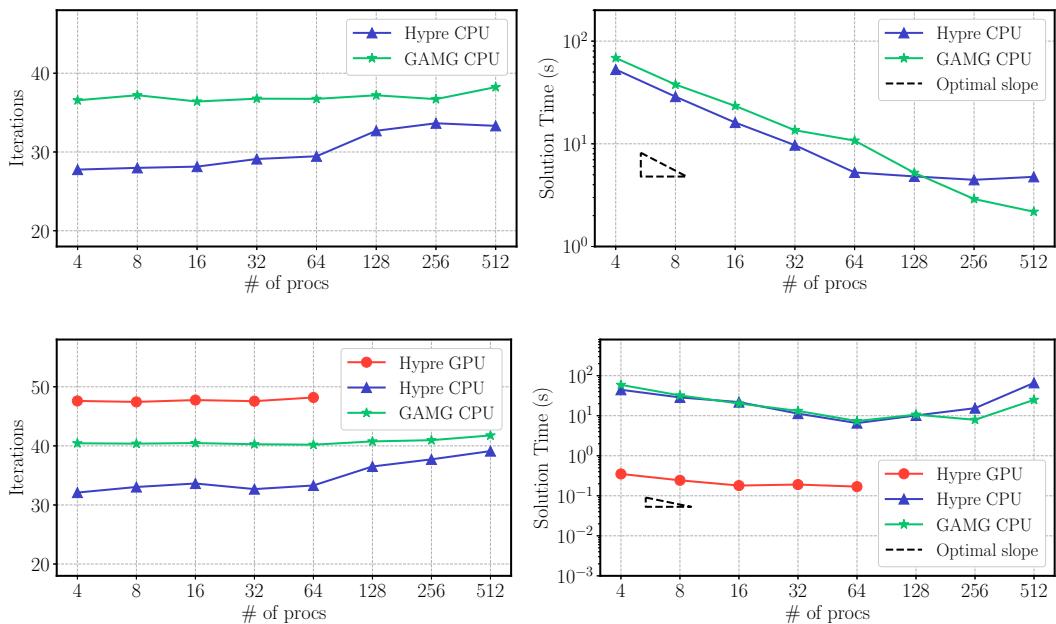


Figure 4: Strong scaling for EMI-Is-LRR on Galileo100 (top) and for EMI-AP for LEONARDO (bottom). For each case, we report the average PCG iterations (left) and solution time (right) per timestep as a function of the number of processors.

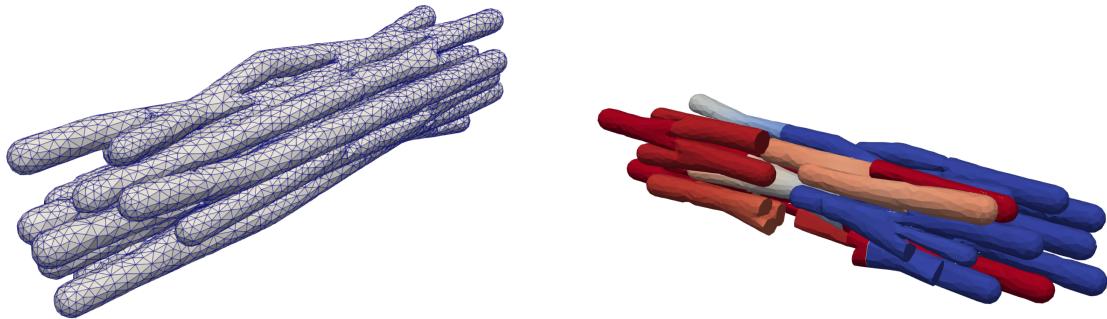


Figure 5: Realistic three-dimensional cardiomyocytes geometry (left) and EMI potentials u_k (right). The mesh is composed of 35 myocytes (about 37k DOFs).

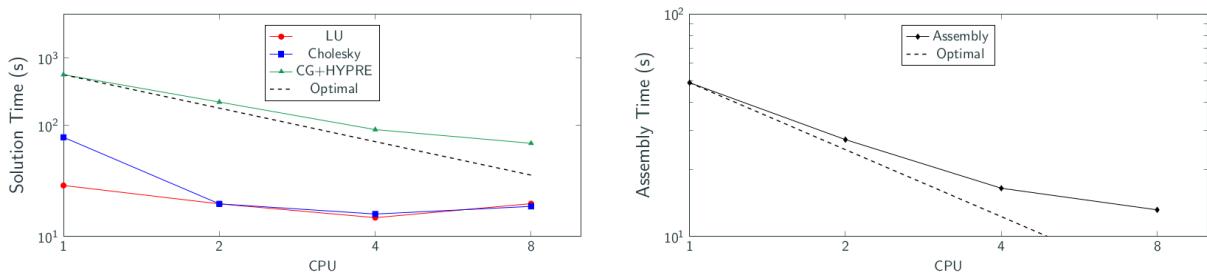


Figure 6: Strong scaling results for realistic three-dimensional cardiomyocytes geometry; solution time (left), assembly time (right). The problem has been solved on a mesh composed of 35 myocytes (about 37k DOFs).

In table 3 and in figure 4 (top), we report the test results for the EMI-Is-PRR case on Galileo100. In table 4 and in figure 4 (bottom) the same tests are reported for LEONARDO and a comparison between the two GPUs architectures employed is highlighted. We notice that the number of iterations is almost constant for all the implementations tested, as expected, but we do not achieve a halving of the mean solution time per timestep as we would expect for an optimal strong scaling. Nevertheless, we obtain sub-optimal scaling for most of the tests, especially on CPU, even if some load imbalance issues tend to worsen the performance of the solver in terms of solution time when considering an higher number of processors. A sub-optimal, but still remarkable, scaling is obtained for the GPU case on LEONARDO, and in general GPUs, both on Marconi100 and LEONARDO, show highly improved performances in terms of solution time compared to the CPU case, roughly an order of magnitude less.

3.2 Realistic myocyte geometries

In figure 6, we report the results of preliminary tests on a laptop with an 8-core M1 processor for a more realistic three-dimensional tissue geometry shown in figure 5.

We also carried out preliminary tests with hypre AMG (PETSc options `-ksp_type cg -pc_type hypre -pc_hypre_type boomeramg -ksp_monitor_singular_value`) running on an 18-core workstation for two versions of four realistic three-dimensional geometries and report the tests in [table 6](#). The tested meshes come in two versions and were built similarly to the one in [figure 5](#): Version 1 presents more rounded corners of the domain boundaries, while in Version 2 they are all sharp. Version 2 has fewer elements; however the mesh quality is lower. The quality is measured as

$$Q = \alpha \frac{V}{\left(\sum_{i=1}^6 L_i^2\right)^{3/2}} \quad (5)$$

where the L_i are the lengths of the edges of the tetrahedron, V its volume, and the factor $\alpha = 72\sqrt{3}$ is to arrange that an optimal, equilateral tetrahedron has $Q = 1$ [19]. A flat tetrahedron has $Q = 0$. Quality statistics of the test meshes are reported in [table 5](#).

mesh	Version 1		Version 2	
	Q_{\min}	$Q < 0.2$	Q_{\min}	$Q < 0.2$
(a)	0.0528	0.007%	0.0090	0.032%
(b)	0.0003	0.011%	0.0089	0.054%
(c)	0.0135	0.011%	0.0008	0.053%
(d)	—	—	0.00607	0.156

Table 5: Properties of the realistic three-dimensional cardiomyocytes geometries (generated with two different algorithms, Version 1 and 2) tested with hypre EMI-AMG (results in [table 6](#)). Q denotes the ratio between volume and edge lengths. We report the minimum Q value (Q_{\min}) and the fraction of elements with quality lower than 0.2. We do not have those information about the (d) mesh.

Results show that no differences occur from a preconditioned solver perspective. Current work is focusing on newer and larger meshes with improved element quality.

mesh	myocytes	extr. regions	Version 1			Version 2		
			DOFs	it	cond	DOFs	it	cond
(a)	76	76	354,298	9	1.74	334,178	9	1.60
(b)	141	141	643,463	9	1.74	599,920	9	1.88
(c)	331	331	1,442,792	10	1.86	1,339,531	10	1.89
(d)	409	409	—	—	—	2,956,099	11	2.69

Table 6: Hypre EMI-AMG results for Version 1 and 2 of the three-dimensional cardiomyocyte meshes. Version 1 and 2 refers to two different mesh generation algorithms. We report the number of myocytes, the number of splitting of the extracellular region and, for each Version, the number of degree of freedom (dof), the average number of iteration (it) and the average condition number (cond).

4 3D EMI-BDDC preconditioners

We report here a summary of our main results for a BDDC preconditioner for the solution of three dimensional composite Discontinuous Galerkin discretizations of reaction-diffusion systems of ordinary and partial differential equations arising in the cardiac EMI Model. More details on the structure of BDDC preconditioners and results can be found in [1, 3, 8]. The modeling of each individual cardiac cell results in discontinuous global solutions across cell boundaries, requiring the careful construction of dual and primal spaces for the BDDC preconditioner. Key statistics, such as the number of linear iterations and condition number, resulted to be quite satisfactory in terms of boundedness and trend through all the numerical tests, showing effectiveness and robustness of BDDC preconditioner in accelerating the solution of the EMI problem.

4.1 Simplified regular geometries

As in section 3.1, we consider an artificial, repetitive geometry that splits a cube into two subdomains, an intracellular subdomain in the center with connections to the outside via all faces of the cubes and an extracellular domain around it (see figure 7). This way, each intracellular subdomain has both an interface to an extracellular subdomain (via a cell membrane) and to other intracellular subdomains (via gap junctions). These geometries were generated using the finite element library Kaskade [10] and allow to consider coupling of myocytes in all three directions, while the software considered in section 3.1 allowed only for a planar distribution of those simplified myocytes.

For this study, we considered a linear gap junction $\frac{v_{ij}}{R_g}$ where $R_g = 4.5 \times 10^{-4}$ for the interfaces between adjacent intracellular subdomains and the Aliev-Panfilov ionic model for the gating variables between intra- and extracellular subdomains. If not stated otherwise, the conductivity coefficients σ_i were fixed to $3 \frac{\text{mS}}{\text{cm}}$ for intra- and $20 \frac{\text{mS}}{\text{cm}}$ for extra-cellular subdomains. To improve load balancing, we decomposed the extracellular domain into subdomains using regular continuous finite elements. On the interfaces between those subdomains, we did not need to consider any discontinuities. We used the BDDC implementation in the software library Ginkgo [11] as a preconditioner for a Conjugate Gradient (CG) method, where we always chose a zero initial guess. The stopping criterion evaluated the L^2 -norm of the relative residual against a preset threshold of 10^{-6} , except for table 7, where we considered the absolute residual norm of 10^{-8} . Iteration matrices and related right-hand sides were computed using the finite element library Kaskade 7 [10]. The times reported referred to the solving time of the preconditioned solver. Numerical results were given for the solution of the linear system at time step 0.01 ms. All tests were performed on the CPU partition of the EuroHPC machine Karolina¹ on compute nodes with two AMD Zen 2 EPYC™ 7H12 CPUs, totalling 128 CPU cores and 256 GB of main memory per node.

¹<https://www.it4i.cz/en/infrastructure/karolina>

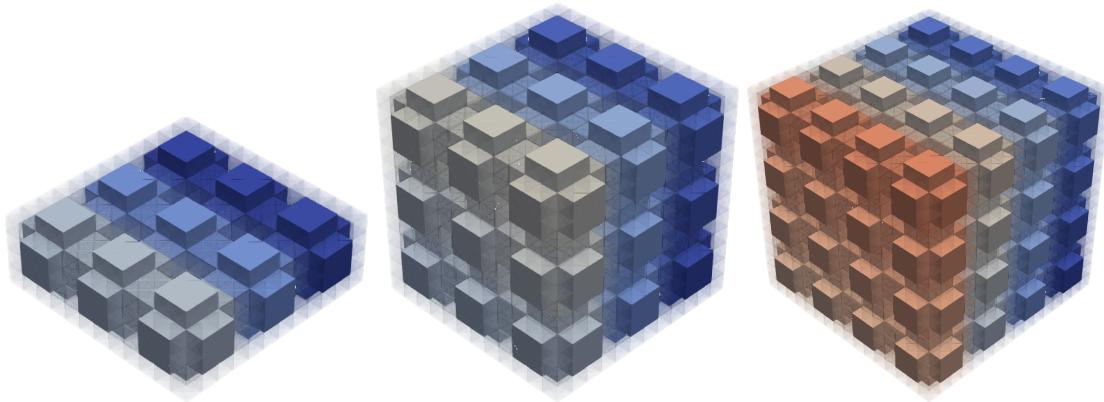


Figure 7: BDDC test geometry with $3 \times 3 \times 1$ (left), $3 \times 3 \times 3$ (middle) and $4 \times 4 \times 4$ (right) cells. Each cell is an intracellular subdomain inside a cube that can be stacked in all dimensions, resulting in a mesh where all intracellular subdomains have interfaces with extracellular space via a cell membrane model and with other intracellular domains via a linear gap junction.

Cells	Subds	DOFs	VEF				VE			
			it	cond	CD	t_{sol}(s)	it	cond	CD	t_{sol}(s)
3x3x1	18	4493	7	2.3158	73	2.9	14	17.4721	40	2.5
3x3x2	36	8718	7	2.2712	207	3.3	16	12.2331	123	2.8
3x3x3	54	12943	7	2.2587	341	3.8	15	11.9005	206	3.3
4x4x4	128	30209	7	2.2605	919	5.4	15	11.7062	567	6
5x5x5	250	58461	7	2.2602	1929	5.8	15	11.3865	1204	6.2
6x6x6	432	100405	7	2.2602	3491	9.1	15	11.6549	2195	6.6
7x7x7	686	158747	7	2.2602	5725	15.6	15	11.6975	3618	15.5

Table 7: BDDC weak scalability for an increasing number of cells from $3 \times 3 \times 1$ to $7 \times 7 \times 7$. Each cube is discretized with 1024 tetrahedra, i.e. 512 for each intra- and extracellular subdomain. We report the number of subdomains (Subds), the global dimension (DOFs) of the linear problem, the number of preconditioned CG iterations (it), a condition number estimate (cond) computed with the Lanczos estimate, the dimension of the coarse problem (CD), and the time in seconds needed for a preconditioner application. The stopping criterion tolerance for this test is a residual norm of 10^{-8} . Two different coarse space are compared: vertices + edge and face averages (VEF) and vertices + edge averages (VE).

4.1.1 EMI-BDDC weak scaling

For a weak scaling study, we considered the linear system in an EMI model simulation at the same time step (0.01 ms) for a growing number of subdomains, starting with $3 \times 3 \times 1$ cells and reaching up to $7 \times 7 \times 7$ cells. The observed convergence behavior aligns with the theory: as we increased the number of subdomains but leave $\frac{H}{h}$ constant, the number of iterations needed to converge as well as the condition number estimate remained roughly constant, see table 7.

To evaluate the stability of the method, we considered a setup where we ran the solver with random right-hand-side vectors, in order to confirm that (as expected from the theory) the

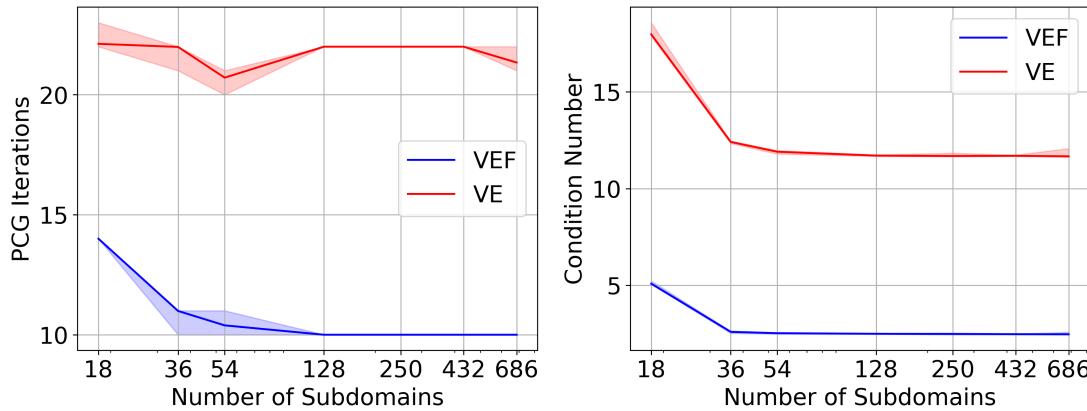


Figure 8: CG+BDDC (PCG) iterations needed to converge to a relative residual tolerance of 10^{-6} (left) and condition number estimates (right) for random right-hand side vectors. The results colored in blue consider a full primal space containing vertex values as well as edge and face averages. The results colored in red consider only vertex values and edge averages in the primal space. The solid lines show the mean over 100 different random right-hand sides, the colored areas represent the range of iterations or condition numbers for each test case, respectively.

Krylov method convergence was independent of the right-hand side and initial guess. For this study, we generated 100 different right-hand-side vectors filled with random values in the range $(-1, 1)$ for each of the test cases and recorded the iteration count needed to converge to a relative residual norm tolerance of 10^{-6} . Figure 8 confirmed the expectation that the choice of right-hand side did not impact the convergence of our method significantly. In terms of compute time, we saw two major jumps: the first when inter-node communication over the network was needed starting at 128 subdomains, and the second when the solution of the coarse problem started dominating the runtime. The latter effect could possibly be alleviated by employing a third level of BDDC on the coarse problem to improve scalability.

4.1.2 BDDC Robustness w.r.t. conductivity coefficients

We also studied the BDDC robustness of the condition number and iteration counts with respect to the conductivity coefficients σ_i . Figure 9 shows the convergence behavior for random conductivity coefficients in the extracellular and intracellular subdomains. As the extracellular subdomains together represent one continuous space, we assigned the same coefficient to all of them, while each of the intracellular subdomains was assigned a random conductivity coefficient, resulting in constant conductivity coefficients within each subdomain. One could observe the same behavior as for fixed σ_i with a slightly wider range of needed iterations.

4.2 Realistic myocyte geometries

For completeness, we recall here our preliminary results already reported in [2, 3] for the EMI BDDC preconditioner in 3D tests using four more realistic cardiac meshes of different size,

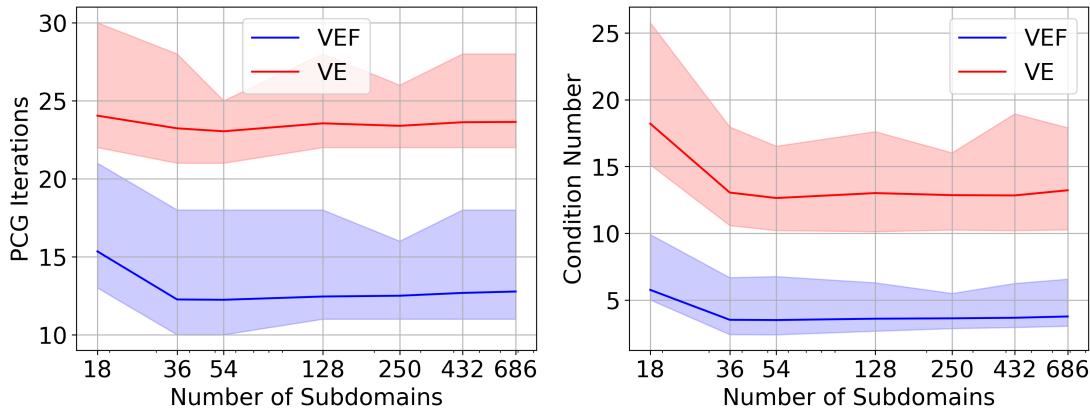


Figure 9: CG+BDDC (PCG) iterations needed to converge to a relative residual tolerance of 10^{-6} (left) and condition number estimates (right) for random conductivity coefficients $\sigma_i \in (1, 20) \frac{\text{mS}}{\text{cm}}$. For each test case, we generate the preconditioner 100 times with different, random conductivity coefficients, the solid lines show the mean of the iterations needed to converge and the condition number estimates.

whose details are listed in [table 5](#). These tests we performed using the BDDC implementation in Kaskade [10].

We tested both unpreconditioned CG and BDDC-preconditioned CG for solving the full systems. As expected, BDDC reduced the iteration count dramatically, and lead to resolution-independent convergence due to the bounded condition number of the preconditioned system, see [table 8](#). Despite the increased computational effort per iteration introduced by the preconditioner, the computational time was also reduced by a large factor. More details can be found in [2].

DOFs	CG-noPrec			BDDC		
	it	$t_{\text{sol}}(\text{s})$	cond	it	$t_{\text{sol}}(\text{s})$	cond
33,698	1111	6.595	24,390	11	0.105	52
75,122	2388	58.711	83,610	12	0.307	134
132,930	2500	69.238	32,560	12	0.630	157
207,122	5209	608.205	138,080	10	0.771	229
297,698	4606	400.290	305,810	10	1.245	196

Table 8: Iteration count, CPU time and condition number (cond) estimate for unpreconditioned and BDDC-preconditioned CG on problems of different size.

References

- [1] Ngoc Mai Monica Huynh, Fatemeh Chegini, Luca Franco Pavarino, Martin Weiser, Simone Scacchi. Convergence analysis of BDDC preconditioners for composite DG discretizations of the cardiac cell-by-cell model. SIAM Journal on Scientific Computing, 2023;45(6):A2836–A2857.

- [2] Fatemeh Chegini, Algiane Froehly, Ngoc Mai Monica Huynh, Luca F Pavarino, Mark Potse, Simone Scacchi, Martin Weiser. Efficient numerical methods for simulating cardiac electrophysiology with cellular resolution. In COUPLED 2023-10th International Conference on Computational Methods for Coupled Problems in Science and Engineering, 2023.
- [3] MICROCARD WP5 Deliverable D5.2. Report on scalable multilevel DD and MG preconditioners. Technical report, MICROCARD EuroHPC Project, 2024.
- [4] Mark Potse. The MICROCARD project. In EuroHPC Summit 2024-The European High Performance Computing, 2024.
- [5] Mark Potse. Modeling the heart cell by cell: the MICROCARD project. In Cardiac Physiome meeting 2024, 2024.
- [6] Piero Colli Franzone, Luca Franco Pavarino, Simone Scacchi. Mathematical Cardiac Electrophysiology. Springer, 2014.
- [7] Edoardo Centofanti, Ngoc Mai Monica Huynh, Luca F Pavarino, Simone Scacchi. Parallel algebraic multigrid solvers for composite discontinuous Galerkin discretization of the cardiac EMI model in heterogeneous media. Computer Methods in Applied Mechanics and Engineering, 2025;442:118001.
- [8] Fritz Goebel, Ngoc Mai Monica Huynh, Fatemeh Chegini, Luca Pavarino, Martin Weiser, Simone Scacchi, Hartwig Anzt. A BDDC preconditioner for the cardiac EMI model in three dimensions. arXiv preprint arXiv:2502.07722, 2025;.
- [9] Satish Balay, William D. Gropp, Lois Curfman McInnes, Barry F. Smith, et al. PETSc users manual. Technical Report ANL-95/11 - Revision 3.20, Argonne National Laboratory, 2024.
- [10] Sebastian Götschel, Anton Schiela, Martin Weiser. Kaskade 7—a flexible finite element toolbox. Computers & Mathematics with Applications, 2021;81:444–458.
- [11] Hartwig Anzt, et al. GINKGO: A modern linear operator algebra framework for high performance computing. ACM Transactions on Mathematical Software, 2022;48(1):1–33.
- [12] Pierre-Elliott Bécue, Mark Potse, Yves Coudière. A three-dimensional computational model of action potential propagation through a network of individual cells. In 2017 Computing in Cardiology (CinC), pages 1–4. IEEE, 2017.
- [13] Aslak Tveito, Karoline H Jæger, Miroslav Kuchta, Kent-Andre Mardal, Marie E Rognes. A cell-based framework for numerical modeling of electrical conduction in cardiac tissue. Frontiers in Physics, 2017;5:48.
- [14] Aslak Tveito, Kent-Andre Mardal, Marie E Rognes. Modeling excitable tissue: The EMI framework. Springer Nature, 2021.
- [15] Rubin R Aliev, Alexander V Panfilov. A simple two-variable model of cardiac excitation. Chaos, Solitons & Fractals, 1996;7(3):293–301.
- [16] Ulrich Trottenberg, Cornelius W Oosterlee, Anton Schuller. Multigrid methods. Academic press, 2001.
- [17] Hans De Sterck, Ulrike Meier Yang, Jeffrey J Heys. Reducing complexity in parallel algebraic multigrid preconditioners. SIAM Journal on Matrix Analysis and Applications,

- 2006;27(4):1019–1039.
- [18] Robert D Falgout, Jim E Jones, Ulrike Meier Yang. The design and implementation of hypre, a library of parallel high performance preconditioners. In Numerical solution of partial differential equations on parallel computers, pages 267–294. Springer, 2006.
 - [19] G. Balarac, F. Basile, P. Bénard, F. Bordeu, J.-B. Chapelier, L. Cirrottola, G. Caumon, C. Dapogny, P. Frey, A. Froehly, G. Ghigliotti, R. Larauftie, G. Lartigue, C. Legentil, R. Mercier, V. Moureau, C. Nardoni, S. Pertant, M. Zakari. Tetrahedral remeshing in the context of large-scale numerical simulation and high performance computing. MathematicS In Action, 2022;11:129–164.