A Distributed Memory Framework for the EMI Model

Fatemeh Chegini, Aurel Neic, Martin Weiser, Fritz Göbel, Axel Loewe,
Marie Houillon, Hartwig Anzt, Mark Potse, and
other contributing members of the MICROCARD consortium will be added

Abstract

The Extracellular-Membrane-Intracellular (EMI) model enables cardiac electrophysiology simulations with cellular resolution, treating extracellular and intracellular domains separately, with the electrical potentials driven by membrane dynamics. Its implementation in distributed-memory environments requires some care to scale to very large problems, in particular due to the non-standard structure of a differential-algebraic equation with dynamics restricted to the non-manifold membrane surfaces. We present a scalable computational framework for EMI simulations in the openCARP software, from mesh handling, tiered indexing, and surface mesh extraction to semi-implicit time integration and distributed-memory solvers. The implementation uses a consistent chain of transfer operators to couple volumetric potentials with transmembrane voltage, supporting heterogeneous ionic membrane models, canonical output, and distributed-memory solvers. We also describe the hash-based distributed protocol for matching membrane faces across MPI ranks and the parallel construction of the corresponding data structures.

Numerical examples demonstrate the correctness of the implementation under several excitation scenarios and localized heterogeneities and its applicability to physiologically relevant problems. Weak and strong scaling experiments, together with timing analyses of mesh processing, matrix construction, and EMI-model initialization relative to simulation time, show efficiency and scalability on modern HPC architectures.

FC: @Martin: I am reading the paper from the beginning, and wherever we need to modify any section based on (restructuring EMI mesh), I will make the text in blue.

FC: @Martin: I am done for now. I modified the relevant paragraphs based on the restructuring of the EMI mesh.

FC: @To the others: Martin and I are still working on the paper. For your information, Martin will be on vacation for the next two weeks, and I will be on vacation for the next five weeks, until the end of August. We may still update the paper occasionally during this period.

1 Introduction

The Extracellular-Membrane-Intracellular (EMI) model has become an important formulation of cardiac electrophysiology when high-fidelity resolution of microstructural detail is required [33, 18, 35]. In contrast to homogenized continuum models such as the monodomain and bidomain formulations [9, 8], the EMI model represents intracellular, extracellular, and membrane potentials on disjoint geometric regions, leading to a differential-algebraic reaction-diffusion system. By explicitly representing individual myocytes and the surrounding extracellular space, it captures microscale heterogeneities and allows to study effects related to cell shape, membrane properties, and cell-to-cell coupling [33].

The EMI model is usually discretized by finite elements and implicit-explicit time stepping with operator splitting. However, moving from homogenized models to geometrically resolved cellular scales leads to extremely large problems when considering tissue volumes of relevant size, calling for massively parallel distributed simulation. It also introduces substantial challenges for parallel finite element implementation and mesh generation at the microscale [23]. In particular, interface entities may be distributed across different MPI partitions and must be consistently represented in distributed memory, independently of the number of processors and the partitioning method used in the simulation. In distributed memory computations, interface vertices and faces must therefore be decoupled, indexed consistently, and paired across MPI ranks before the coupled EMI problem can be assembled and solved.

An additional challenge associated with distributed memory implementations of the EMI model is the development of scalable solvers and preconditioners for the large linear systems arising from the semi-implicit time discretization of the problem. Efficient numerical solution strategies for EMI-type systems have already been investigated in simpler settings [17]. Several recent works have addressed the theoretical analysis and numerical validation of advanced preconditioners for EMI discretizations, including AMG [30, 6], BDDC [16, 12], and related Schwarz-type preconditioners [5]. Alternative formulations and discretization strategies, including boundary integral and CutFEM approaches, have also been proposed for EMI simulations [11, 3]. These studies demonstrated robust convergence and scalability properties on idealized cardiac geometries and relatively small-scale benchmark problems. However, extending such solver technologies to cellularly resolved cardiac structures with realistic geometries discretized by highly non-uniform unstructured meshes remains a challenge for large-scale HPC simulations.

We present a distributed-memory implementation of the EMI model in openCARP [25, 27], extending the previous shared-memory EMI implementation in `Kaskade 7` [7, 36] to distributed-memory architectures. The implementation constructs a derived topologically disconnected EMI mesh on which standard P_1 finite elements represent the subdomain-wise continuous finite element space for electrical potentials. It also retains the links between matching membrane faces and builds surface representations for face-based membrane quantities. Transfer matrices connect these volume and surface representations. The implementation is integrated with the LIMPET library [27] of ionic membrane models. Solvers and preconditioners are provided by the PETSc [22] and Ginkgo [2] linear algebra backends.

We demonstrate the application to relevant physiological problems and present performance and scalability studies, including solver comparisons.

The remainder of the paper is structured as follows. Sec. 2 presents the EMI equations and their finite-element discretization in space and time. Sec. 3 describes the distributed implementation of the discretized EMI model, including construction of the disconnected EMI mesh, interface reconstruction, retained membrane face links, distributed indexing, matrix assembly, and coupling between volume and membrane unknowns. Sec. 4 is dedicated to EMI simulation settings in openCARP, including volume conductivity regions, solver backends, membrane-model assignment, and stimulation protocols. Numerical experiments and performance results are presented in Sec. 5. We conclude in Sec. 6 and outline directions for future research.

2 The EMI model and its finite element discretization

We begin with the formal model definition and the mathematical description of the spatial and temporal discretization. Implementation-specific data structures and parallel assembly details are provided in Sec. 3.

2.1 The EMI model

The EMI model consist of a spatial component that describes the geometry and spatial relationships of potentials, currents, and membrane states, and a dynamics part formulating the temporal evolution of membrane states and transmembrane voltage.

2.1.1 The spatial model

Let the computational domain $\Omega \subset \mathbb{R}^3$ be subdivided into $n+1$ pairwise disjoint subdomains Ω_i, such that $\bigcup_{i=0,\dots,n} \overline{\Omega_i} = \overline{\Omega}$. The subdomain Ω_0 represents the only extracellular space, and will generally be denoted by Ω_e. The remaining n subdomains represent myocytes, which will generally be denoted as Ω_i with $\mathrm{i} \geq 1$. Let $\Gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$ be the joint interface of two adjacent subdomains and let $\Gamma = \bigcup_{i,j=0,\dots,n} \Gamma_{ij}$ denote the membrane interface separating myocytes from each other or from the extracellular space. In this paper, the term membrane includes both myocyte–extracellular interfaces and intercalated disc interfaces between adjacent myocytes. The EMI formulation seeks potentials $u_i(t) \in V_i = H^1(\Omega_i)$ and membrane states $w_{ij}(t) \in W_{ij} = L^2(\Gamma_{ij})$. The potential difference across Γ_{ij} is the transmembrane voltage $v_{ij} = u_i|_\Gamma - u_j|_\Gamma$. Note that $v_{ij} = -v_{ji}$; the sign is fixed by the convention $i > j$.

The EMI equations consist of electrostatics in each subdomain,

$$-\operatorname{div}(\sigma_i \nabla u_i) = I_i^\mathrm{stim} \qquad \text{in } \Omega_i \quad \text{for } i = 0, \dots, n, \tag{1}$$

with scalar diffusion σ_i, an optionally applied stimulus current density I_i^stim, and Neumann boundary conditions

$$-n_i \cdot \sigma_i \nabla u_i = I_{ij} \qquad \text{on } \Gamma_{ij}, \quad i, j = 0, \dots, n,\ i \neq j, \tag{2}$$

where n_i is the outward unit normal of Ω_i. The transmembrane current density

$$I_{ij} = C_\mathrm{m} \frac{\partial v_{ij}}{\partial t} + I_{ij}^\mathrm{ion}(v_{ij}, w_{ij}) - I_{ij}^\mathrm{stim} \tag{3}$$

consists of a capacitance current, an ion current depending on the transmembrane voltage v_{ij} and the state w_{ij} of the membrane, and, if applied, a stimulus current I_{ij}^stim. We require $I_{ij}^\mathrm{ion}(v_{ij}, w_{ij}) = -I_{ji}^\mathrm{ion}(v_{ji}, w_{ji})$, $I_{ij}^\mathrm{stim} = -I_{ji}^\mathrm{stim}$, and $w_{ij} = w_{ji}$, such that the outflux of Ω_i across Γ_{ij} coincides with the influx to Ω_j and charges are conserved. The membrane state w_{ij} comprises gating variables of the ion channels and ion concentrations, is independent of the membrane orientation, and evolves pointwise according to

$$\frac{\partial w_{ij}}{\partial t} = R_{ij}(v_{ij}, w_{ij}) \qquad \text{on } \Gamma_{ij}. \tag{4}$$

In addition, we prescribe a vanishing flux on the domain boundary, i.e.,

$$n_i^T \sigma_i \nabla u_i = 0 \qquad \text{on } \partial\Omega \cap \partial\Omega_i, \tag{5}$$

and normalized potentials by

$$\sum_{i=0}^{n} \int_{\Omega_i} u_i \,\mathrm{d}x = 0. \tag{6}$$

Note that the volumetric stimulus currents I_i^stim in (1) have to satisfy the compatibility condition

$$\sum_{i=0}^{n} \int_{\Omega_i} I_i^\mathrm{stim} \,\mathrm{d}x = 0,$$

as is usual for pure Neumann problems. In contrast, the transmembrane stimulus currents I_{ij}^stim satisfy the compatibility by construction.

Table 1: Overview of the EMI model parameters

Symbol	description	Value	Unit
σ_i	intracellular ($i > 0$) conductivity	400	$\mathrm{S\,m^{-1}}$
σ_0	extracellular conductivity	2×10^3	$\mathrm{S\,m^{-1}}$
C_m	membrane capacitance	0.01	$\mathrm{F\,m^{-2}}$
R_g	resistance of intercalated discs	300×10^{-6}	$\Omega\,\mathrm{m}$

2.1.2 Ionic models

Ionic models define the pointwise ion current densities $I_{ij}^{\mathrm{ion}} : \mathbb{R} \times \mathbb{R}^{m_{ij}} \to \mathbb{R}$ and the membrane evolution $R_{ij} : \mathbb{R} \times \mathbb{R}^{m_{ij}} \to \mathbb{R}^{m_{ij}}$. Many different models exist, with varying complexity, state dimension m_{ij}, and physiological meaning, most of which are available in the LIMPET library. In principle, every membrane part Γ_{ij} can have its own interface model. The principal distinction is not the indexing of the interface, but the assigned model: intercalated disc interfaces use a gap-junction law, whereas myocyte–extracellular interfaces usually use one of the full ionic membrane models.

If $ij > 0$, the interface Γ_{ij} represents the intercalated disc separating two myocytes. These intercalated discs contain gap junctions, pore proteins that create direct connections between myocytes through which ions can diffuse passively. Following Plonsey [28], we represent the gap junction current density

$$I_{ij}^{\mathrm{g}} = R^{\mathrm{g}} v_{ij}, \tag{7}$$

in proportion to the transmembrane voltage, where R^{g} is the junctional conductivity. This simple ionic model is stateless, i.e., $m_{ij} = 0$, and trivially satisfies the sign consistency requirement.

For $i > j = 0$, the interface Γ_{i0} represents a membrane separating a myocyte from the extracellular space. For such interfaces, I_{i0}^{ion} and R_{i0} are defined by one of the many ionic models from the LIMPET library. The sign consistency requirement has to be imposed explicitly by ordering myocytes and extracellular space. Note that the ionic model depends on the interface part Γ_{i0}, such that several different ionic models can be involved in a single simulation.

2.2 Finite element discretization

2.2.1 Weak Form and Semi-Implicit Time Discretization

Let $\phi = (\phi_0, \ldots, \phi_n) \in V = \bigoplus_{i=0}^{n} V_i$ denote an arbitrary test function, where each component $\phi_i \in H^1(\Omega_i)$. After integration by parts, the interface flux balances (2) contribute a surface integral term on Γ:

$$\sum_{i=0}^{n} \int_{\Omega_\mathrm{i}} (\sigma_\mathrm{i} \nabla u_\mathrm{i} \cdot \nabla \phi_\mathrm{i} - \phi_i I_i^{\mathrm{stim}}) \,\mathrm{d}\Omega + \sum_{i>j} \int_{\Gamma_{ij}} (\phi_\mathrm{i} - \phi_\mathrm{j}) I_{ij} \,\mathrm{d}\Gamma = 0. \tag{8}$$

Inserting the transmembrane current (3) with a first order discretization of the transmembrane voltage's time derivative yields

$$\begin{aligned} &\sum_{i=0}^{n} \int_{\Omega_\mathrm{i}} (\sigma_\mathrm{i} \nabla u_\mathrm{i}^{n+1} \cdot \nabla \phi_\mathrm{i} - \phi_i (I_i^{\mathrm{stim}})^n) \,\mathrm{d}\Omega \\ &\quad + \sum_{i>j} \int_{\Gamma_{ij}} (\phi_\mathrm{i} - \phi_\mathrm{j}) \left(C_\mathrm{m} \frac{u_i^{n+1} - u_j^{n+1} - u_i^n + u_j^n}{\Delta t} + I_\mathrm{ion}(u_i^n - u_j^n, w^n) - (I_{ij}^{\mathrm{stim}})^n \right) \mathrm{d}\Gamma = 0 \end{aligned} \tag{9}$$

for all test functions $\phi \in V$ and time steps $n \in \mathbb{N}$, where we have treated the stationary electrostatics implicitly and the ion and stimulus currents explicitly.

2.2.2 Finite Element Spaces

Let Ω be covered by a simplicial triangulation $\mathcal{T}_h^\Omega$ compatible with the subdomain structure, such that each simplex $T \in \mathcal{T}_h^\Omega$ is contained in the closure of exactly one subdomain. For the potentials, we consider piecewise linear finite elements on $\mathcal{T}_h^\Omega$:

$$V_h = \{\varphi_h \in V \mid \forall T \in \mathcal{T}_h^\Omega : \varphi_h|_T \in \mathcal{P}_1\}. \tag{10}$$

In triple points, where two myocytes and the extracellular space meet, three different potential values are present, such that the transmembrane voltage is not single-valued at the vertex. Thus, a P_1 vertex-based representation of the transmembrane voltage and, consequently, of the ionic states w, is ambiguous. Instead, we use a piecewise constant representation on the induced triangulation $\mathcal{T}_h^\Gamma$ of the membrane interface, where the transmembrane voltage is face-wise well-defined:

$$W_h = \{\psi_h \in L^2(\Gamma) \mid \forall F \in \mathcal{T}_h^\Gamma : \psi_h|_F \in \mathcal{P}_0\}. \tag{11}$$

For both spaces, we use canonical bases consisting of partial hat functions $\phi_{ki} \in V_h$ associated with subdomain–vertex incidences (x_k, Ω_i) such that $x_k \in \mathcal{N} \cap \overline{\Omega}_i$, and piecewise constant functions $\psi_k \in W_h$ associated with membrane faces $F_k \in \mathcal{T}_h^\Gamma$, respectively. The implementation-specific construction of the corresponding mesh and index sets is described in Sec. 3.

2.2.3 Fully discrete systems

Restricting the weak formulation (9) to the finite element space V_h in a standard Galerkin approach turns it into the large-scale linear equation system

$$\begin{aligned} \mathbf{A}\,\delta\mathbf{u}^n &= \mathbf{r}^n \\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \delta\mathbf{u}^n \end{aligned}$$

for the increment $\delta\mathbf{u}^n$ of the potentials' coefficient vector $\mathbf{u}$ for time $n+1$. The system matrix $\mathbf{A} = \mathbf{K} + \frac{1}{\Delta t}\mathbf{M}_\Gamma$ contains the block-diagonal stiffness matrix $\mathbf{K}$ for the electrostatics in each subdomain and the membrane-based mass matrix $\mathbf{M}_\Gamma$ coupling the subdomains.

Assuming the volumetric stimulus currents $I_i^{\text{stim}} \in V_h$ to be given as a coefficient vector $\mathbf{I}_\Omega^{\text{stim}}$ and the transmembrane stimulus currents $I_{ij}^{\text{stim}} \in W_h$ to be given as coefficient vector $\mathbf{I}_\Gamma^{\text{stim}}$, the right-hand side vector $\mathbf{r}^n$ is assembled as

$$v^n = \mathbf{B}\mathbf{u}^n \tag{12}$$

$$\mathbf{r}^n = -\left(\mathbf{S}(\mathbf{I}_{\text{ion}}^n(v^n, w^n) + (\mathbf{I}_\Gamma^{\text{stim}})^n) + \mathbf{K}\mathbf{u}^n\right) + \mathbf{M}(\mathbf{I}_\Omega^{\text{stim}})^n, \tag{13}$$

where $\mathbf{M}$ is the block-diagonal mass matrix, $\mathbf{B}$ the gather matrix, and $\mathbf{S}$ the scatter matrix. The latter two translate between the volume space V_h and the membrane space W_h. Each row of $\mathbf{B}$ contains of the six entries $\begin{bmatrix}1/3 & 1/3 & 1/3 & -1/3 & -1/3 & -1/3\end{bmatrix}$, scattered according to the global indices of the two incident potentials' degrees of freedom located on the three vertices of the membrane face $F \in \mathcal{T}_h^\Gamma$ such that the mean transmembrane voltage

$$v_F = \frac{1}{|F|}\int_F (u_i - u_j)\,\mathrm{d}s$$

on $F \subset \Gamma_{ij}$ is obtained. Similarly, each column of $\mathbf{S}$ contains the six entries

$$\begin{bmatrix}|F|/3 & |F|/3 & |F|/3 & -|F|/3 & -|F|/3 & -|F|/3\end{bmatrix}^T,$$

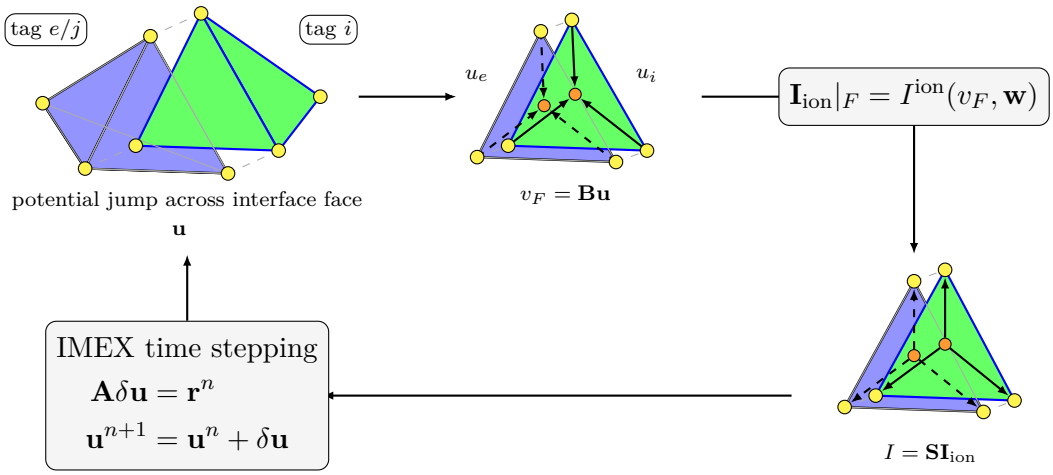


Figure 1: Complete EMI computational cycle at each time step. From potentials $\mathbf{u}$, the mean transmembrane voltage v_F is obtained on each membrane face, the ionic current $\mathbf{I}_{\text{ion}}$ is evaluated, and then scattered back to the volume finite-element space V_h to form the right-hand-side vector $\mathbf{r}^n$ for the next time step.

such that each face current density $I_{\text{ion}}|_F$ contributes its share

$$\pm \int_F \varphi_h I_{\text{ion}} \, \mathrm{d}s$$

to the transmembrane flux tested by the ansatz function φ_h. Here, the sign depends on whether the support of the ansatz function φ_h is contained in Ω_i or in Ω_j, and the entries are scattered into the column again according to the global indexing of the potentials' ansatz functions. The complete computational cycle, from volume-based potentials to interface-based transmembrane voltage, ionic current evaluation, and subsequent scattering to the right hand side, is illustrated in Fig. 1.

2.3 Unit Consistency

Geometries and physical model parameters are provided as input to the simulation code in different units. Here, we provide a dimension and unit analysis for the assembled matrices and right hand side vectors.

The stiffness and mass matrices inherit their units from the volume and interface integrals. For first-order basis functions, the volumetric stiffness entries satisfy

$$(\mathbf{K})_{ij} = \int_\Omega \nabla\phi_i \cdot \sigma \nabla\phi_j \, d\Omega, \tag{14}$$

so that

$$[\nabla\phi_i] = [\nabla\phi_j] = \mu\mathrm{m}^{-1}, \qquad [d\Omega] = \mu\mathrm{m}^3, \qquad [\sigma] = \mathrm{mS}\,\mu\mathrm{m}^{-1},$$

and therefore

$$[\mathbf{K}] = \mathrm{mS}.$$

Likewise, the interface capacitance matrix

$$(\mathbf{M}_\Gamma)_{ij} = \int_\Gamma C_m(\phi_i - \phi_j)(\phi_j - \phi_j) \, \mathrm{d}\Gamma \tag{15}$$

is assembled on interface faces only, which gives

$$[\mathbf{M}_\Gamma] = \frac{\mu\mathrm{F}}{\mathrm{cm}^2}\mu\mathrm{m}^2.$$

In the semi-implicit operator

$$\mathbf{A} = \mathbf{K} + \frac{1}{\Delta t}\mathbf{M}_\Gamma,$$

the units are

$$\left[\frac{1}{\Delta t}\mathbf{M}_\Gamma\right] = \frac{1}{\mathrm{ms}}\frac{\mu\mathrm{F}}{\mathrm{cm}^2}\mu\mathrm{m}^2.$$

Since $\mu\mathrm{F} = \mathrm{mS}\cdot\mathrm{ms}$ and $\mathrm{cm}^2 = 10^8\mu\mathrm{m}^2$, this term has the same conductance unit as $\mathbf{K}$ up to the expected geometric conversion factor from cm^2 to $\mu\mathrm{m}^2$. Consequently,

$$[\mathbf{A}\,\delta\mathbf{u}] = \mathrm{mS}\,\mathrm{mV} = \mu\mathrm{A},$$

which matches the right-hand side in equation (12). This can be checked directly from

$$\mathbf{r}^n = -\left(\mathbf{S}\mathbf{I}_{\mathrm{ion}}^n + \mathbf{K}\mathbf{u}^n\right) + \mathbf{M}I_{\mathrm{vol}}^n.$$

$\mathbf{I}_{\mathrm{ion}}^n$ is a membrane current density with units

$$[\mathbf{I}_{\mathrm{ion}}^n] = \mu\mathrm{A}/\mathrm{cm}^2.$$

Since $\mathbf{S}$ contributes the interface face area, we obtain

$$[\mathbf{S}\mathbf{I}_{\mathrm{ion}}^n] = \mu\mathrm{m}^2\frac{\mu\mathrm{A}}{\mathrm{cm}^2} = 10^{-8}\mu\mathrm{A}.$$

For the stiffness contribution,

$$[\mathbf{K}\mathbf{u}^n] = \mathrm{mSmV} = \mu\mathrm{A}.$$

Hence the right-hand side is a current vector,

$$[\mathbf{r}^n] = \mu\mathrm{A},$$

with the factor 10^{-8} again reflecting the conversion from cm^2 in the face-current density to $\mu\mathrm{m}^2$ in the geometric interface measure.

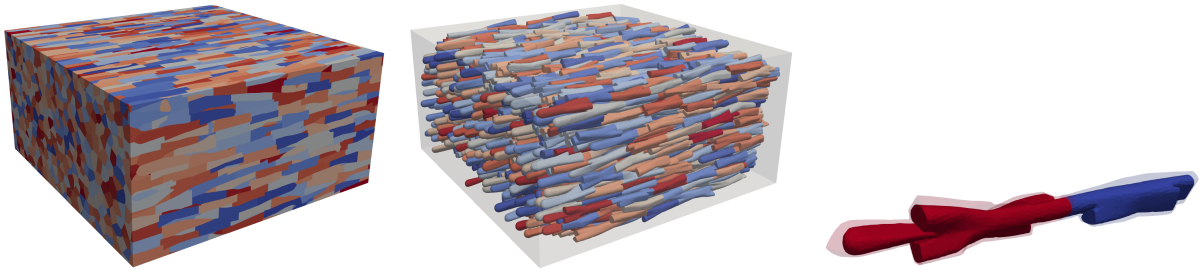
3 Distributed implementation of the EMI model

This section describes how the algebraic problems resulting from the discretization described in Sec. 2 are realized in openCARP. The presentation follows the implementation data path from the tagged input mesh to the derived disconnected EMI volume mesh, retained membrane face links, surface representations, and distributed algebraic system.

3.1 Mesh partitioning and distribution

The input to the EMI preprocessing pipeline is a volumetric mesh with tagged subdomains Ω_i. While openCARP supports several mesh element types, we restrict the presentation to simplicial meshes for brevity. The mesh is read from the standard openCARP point and element files: the point file stores vertex coordinates and provides the global vertex numbering, while the element file stores for each tetrahedron $T \in \mathcal{T}_h^\Omega$ the four incident vertex numbers and a region tag ρ_T. Tetrahedra with identical region tag form a region $r_\rho = \bigcup_{T\in\mathcal{T}_h^\Omega:\rho_T=\rho} T$. Each myocyte forms one region, while the larger extracellular space is typically subdivided in many regions to allow a balanced distribution, see Fig. 2. Files with suffixes `*.extra` and `*.intra` list extracellular region tags $\mathcal{E}$ and intracellular region tags $\mathcal{I}$.

The parallel efficiency of a distributed EMI simulation is strongly influenced by how volume and membrane degrees of freedom are assigned to MPI ranks. This distribution determines



(a) Tagged EMI mesh

(b) Myocytes with surrounding continuous extracellular space

(c) Myocytes surrounded by extracellular subregions

Figure 2: Extracellular regions used for tag-aware partitioning. Intracellular myocytes remain explicit, while the surrounding extracellular domain is split into smaller tagged regions to facilitate load-balanced distribution across MPI ranks. The volume of the mesh is $0.5\,\mathrm{mm}^3$ with 2710 myocytes and 4568 extracellular regions.

which data remain local to a rank during mesh preprocessing, matrix assembly, time stepping, matrix-vector products, and preconditioner application, and which data have to be exchanged through MPI calls. A purely graph-based partition of the mesh may cut through myocytes or through the set of faces belonging to one membrane interface. In that case, operations that are naturally local to one physical region, such as applying local solver blocks in BDDC preconditioners [12, 16], become cross-rank operations. The EMI implementation therefore treats the region tags as the primary partitioning units and uses MPI communication only for the interface information that crosses region boundaries. Consequently, there is a, generally not injective, mapping $\mathfrak{r}$ from region tags to MPI ranks.

A simple internal partitioning assigns regions equally to ranks without respecting region adjacency, which can lead to load imbalances and high communication costs. More sophisticated partitionings can be read from external `*.part` files providing the region-to-rank assignment. One way to create efficient partitionings is to apply standard graph partitioners to the adjacency graph of the regions, with vertex weights given by the number of mesh vertices in the corresponding region and edge weights given by the number of faces shared by two adjacent regions [19, 26].

3.2 EMI mesh construction and indexing

The finite element assembly and output routines in openCARP are mesh-centric: the data needed for assembly, distributed linear algebra, and output are attached to mesh objects; the relevant numberings are listed in Tab. 3. For the EMI model, we therefore construct a derived topologically disconnected volume mesh, referred to as the EMI mesh, and a corresponding vertex numbering. On this mesh, standard P_1 finite elements represent V_h, continuous within each subdomain but discontinuous across membrane interfaces. A vertex of the EMI mesh corresponds to a vertex–subdomain incidence in the input mesh. The input mesh is used during preprocessing to provide reference vertex indices, region tags, and coordinates; the solve and solution output are then performed on the derived EMI mesh.

For each input vertex $x \in \mathcal{N}$, we first determine the set of tags of incident tetrahedra,

$$\rho_x = \{\, \rho_T \mid x \in T,\ T \in \mathcal{T}_h^\Omega \,\}.$$

This set cannot in general be computed from rank-local information, because the tetrahedra incident to x may be distributed across several MPI ranks. The implementation therefore exchanges partial tag sets keyed by the reference vertex id and merges them. The number of potential DOFs represented at x is

$$m_x = |\mathcal{I} \cap \rho_x| + \min(1, |\mathcal{E} \cap \rho_x|),$$

Code c_x	Incident regions	Meaning for EMI preprocessing	♯DOFs m_x
0	one region only, or only extracellular regions	Interior/continuous vertex. The vertex index is used for the DOF index.	1
1	one intracellular region and at least one extracellular tag	Membrane vertex.	2
2	two intracellular tags and no extracellular tag	Intercalated disk vertex.	2
3	two intracellular tags and at least one extracellular tag	Complex junction vertex where two myocytes and extracellular space meet.	3

Table 2: Vertex location codes used for EMI DOF index definition. The codes are computed from the globally merged set of incident region tags at each vertex and define how many local potential values must be represented there. Vertices with more than two incident myocytes are considered non-physiological and are currently not supported.

where $\mathcal{I}$ and $\mathcal{E}$ are the intracellular and extracellular tag sets. The resulting vertex classes are summarized in Tab. 2.

For each input vertex $x \in \mathcal{N}$ with $m_x > 1$, the EMI mesh contains m_x colocated vertices at the same spatial coordinates. The input vertex index is retained for one of these vertices, and $m_x - 1$ additional vertex indices are selected from the extended index range. For membrane and complex junction vertices, $c_x \in \{1, 3\}$, the additional indices are assigned to the intracellular vertices, while the input vertex index is assigned to the extracellular vertex. For intercalated-disk vertices, $c_x = 2$, where no extracellular subdomain is incident, the smaller-tag intracellular side keeps the original index and the larger-tag side receives the additional index. The tetrahedral connectivity is then rewritten so that each tetrahedron uses the vertex index associated with its own subdomain. Thus, the mesh geometry is not changed, but the mesh connectivity is topologically disconnected across membrane interfaces.

The EMI mesh vertex numbering J^c is chosen to remain as close as possible to the input mesh vertex numbering, because ionic-model assignments, stimulation data, and reconstructed face keys refer to input-mesh. Single-valued DOFs keep their reference vertex id,

$$J^c_{ki} = k \qquad \text{for non-duplicated vertices and for the extracellular side.}$$

For each intracellular subdomain Ω_i with $i > 0$, let

$$\eta_i = \{\, (x_k, \Omega_i) \mid x_k \in \overline{\Omega_i} \text{ and side } i \text{ needs an additional interface DOF} \,\}, \qquad n_i = |\eta_i|.$$

Let $s_{ki} \in [0, n_i[\cap \mathbb{N}$ be the local ordering in η_i, obtained by sorting by the reference vertex id k. The appended canonical indices are then

$$J^c_{ki} = |\mathcal{N}| + \sum_{j=1}^{i-1} n_j + s_{ki}.$$

The inverse map

$$d \mapsto k$$

from each EMI vertex index d to its input vertex index k is retained for coordinate insertion, face matching, and output.

After the derived EMI mesh has been constructed, openCARP registers `NBR_SUBMESH` on this mesh; in this context `NBR_SUBMESH` is the canonical compact nodal/DOF numbering J^c of the EMI volume mesh.

Numbering	Meaning in the EMI implementation
`NBR_REF`	reference vertex ids inherited from the input mesh. These ids are stable geometric keys used during preprocessing, matching of duplicated faces, coordinate insertion, and mapping of input data.
`NBR_ELEM_REF`	element numbering inherited from the input mesh.
`NBR_SUBMESH`	canonical compact nodal/DOF numbering J^c of the EMI mesh. The same openCARP numbering name can also be registered on derived surface meshes, where it denotes the canonical nodal numbering of that surface mesh.
`NBR_PETSC`	algebraic DOF numbering J^a generated from the parallel layout. This is the numbering used by distributed matrices and vectors.

Table 3: Numberings used by the EMI implementation in openCARP.

3.3 Interface links and surface meshes

The second preprocessing task is to retain the link between the duplicated membrane sides. These links are needed to assemble the membrane mass matrix and the transfer/coupling matrices B and S, and to index the face-based space W_h. The input mesh does not contain a separate EMI surface mesh. Interface faces are therefore reconstructed from tetrahedral connectivity: candidate faces are enumerated from local volume elements, classified by the incident region tags, and matched by partition-independent keys.

For a face F, let ρ_F denote the set of region tags of the incident tetrahedra. Then

$$F \in \mathcal{T}_h^\Gamma \quad \Longleftrightarrow \quad (|\rho_F \cap \mathcal{I}| = 1 \wedge |\rho_F \cap \mathcal{E}| = 1) \vee |\rho_F \cap \mathcal{I}| = 2.$$

In the first case F separates a myocyte from the extracellular space; in the second case F separates two myocytes. Only interior faces with two adjacent tetrahedra are considered for EMI interface extraction. Boundary faces have only one incident volume tag and are therefore not classified as membrane faces.

The face key is built from the sorted reference vertex ids and the adjacent region tags, for instance

$$\kappa(F) = (\mathrm{sort}(k_1, k_2, k_3), \{\rho^-, \rho^+\})$$

for a triangular face with reference vertices k_1, k_2, k_3. Reference vertex ids are used here because they identify the physical face independently of MPI ownership and independently of the duplicated EMI DOF ids. Neighboring tetrahedra of one interface face may be owned by different ranks, so the implementation exchanges compact face records and merges equal keys. This creates a partition-independent link between the two oriented computational copies of a physical membrane face.

Let Γ denote the two-sided computational interface and γ the unique physical interface. The unique-face surface mesh is obtained by identifying the two oriented copies of each physical face,

$$F^+ \sim F^- \quad \Longleftrightarrow \quad F^+ \text{ and } F^- \text{ are the two oriented copies of the same physical face,}$$

and therefore

$$\gamma = \Gamma/\!\sim \; = \{\, [F]_\sim \mid F \in \Gamma \,\}, \qquad [F]_\sim = \{F^+, F^-\}.$$

The basis of W_h is indexed by these unique physical interface faces. Ionic states are stored once per equivalence class $[F]_\sim$ on γ, whereas coupling to the volume DOFs is assembled on the oriented faces $F^\pm$ in Γ.

The resulting surface meshes are derived mesh objects used for membrane and gap-junction evaluation, transfer-operator construction, and output. Each surface element carries the two neighboring volume-region tags. These tag pairs are also used to select heterogeneous ionic models. Intracellular– extracellular pairs and intracellular–intracellular pairs are both referred

to as membrane faces in this paper; they differ only in the assigned interface model, not in the indexing mechanism.

3.4 Algebraic numbering and output

For optimal locality in distributed linear algebra, the derived EMI mesh is given an algebraic PETSc numbering J^a. This numbering assigns all DOFs owned by one MPI rank to a contiguous global row interval. First we assign each DOF (x, Ω_i) to a volume region. This is trivial for myocytes, which each form a region of their own, but requires a rule for vertices incident to more than one extracellular region. For extracellular DOFs, we choose the incident extracellular region with the smallest tag:

$$r_{x,\Omega_i} = \begin{cases} r, & i > 0 \wedge \Omega_i \subset r, \\ r_{\min(\rho_x \cap \mathcal{E})}, & i = 0. \end{cases}$$

The number of DOFs in region r is

$$n_r = |\{\, (x_k, \Omega_i) \mid r_{x_k,\Omega_i} = r \,\}|,$$

and the number of DOFs on rank $\mathfrak{r}$ is

$$n_{\mathfrak{r}} = \sum_{\mathfrak{r}_r = \mathfrak{r}} n_r.$$

The counts $n_{\mathfrak{r}}$ are exchanged with `MPI_Allgather`. Rank $\mathfrak{r}$ then assigns algebraic indices from the interval

$$\Big[\sum_{q<\mathfrak{r}} n_q, \sum_{q\le\mathfrak{r}} n_q\Big[.$$

Within this interval, DOFs are sorted first by region tag and then by reference vertex id. This algebraic numbering is the `NBR_PETSC` numbering used for distributed matrices and vectors.

Ghost DOFs are local copies of globally numbered DOFs owned by another MPI rank. They are needed when local element assembly, surface evaluation, or transfer-operator construction touches data across a partition boundary. They are not additional physical unknowns and do not receive independent canonical or PETSc ownership indices.

Coefficient vectors are stored in the rank-dependent algebraic ordering J^a, whereas output fields are written in the partition-independent canonical ordering J^c. Before writing volume or surface data, the implementation applies an algebraic-to-canonical permutation,

$$\pi_{a\to c} : J^a \mapsto J^c, \qquad \mathbf{u}^c = P_{a\to c}\mathbf{u}^a.$$

Thus the volume potential is output on the decoupled EMI volume mesh, while membrane quantities are output on the unique-face surface mesh. Changing the number of MPI ranks changes ownership and communication during the run, but it does not change the meaning of the exported EMI volume mesh or unique-face surface mesh.

3.5 Matrix assembly

In this section, we describe the construction of the stiffness and membrane mass contributions, together with the discrete operators used in the EMI framework. The volume stiffness terms are assembled on the derived topologically decoupled EMI volume mesh introduced in Section 3.2. The membrane terms use the retained links between the duplicated membrane faces and the corresponding face-based surface meshes. These links define the algebraic path between the volumetric DOFs and the state variables and currents stored at interface barycenters.

3.5.1 Coupling and transfer operators

The framework relies on several primary coupling operators, which are assembled during initialization using the connectivity information between the decoupled EMI volume mesh, the two-sided computational interface Γ, and the unique-face physical interface γ.

The coupling operators define the algebraic path between the decoupled volume unknowns and the face-based membrane quantities. The signed operator $\mathbf{B}$ gathers oriented voltage differences from volume DOFs to the both-face interface layout. The unsigned operator $\mathbf{B}_+$ uses the same geometric connectivity but omits the sign change; it is used when selecting or distributing current and stimulus quantities rather than computing an oriented voltage jump. The scatter operator $\mathbf{S}$ maps both-face current densities back to the volume right-hand side, with target DOFs and signs determined from the face and counter-face tags. Thus $\mathbf{B}$ and $\mathbf{B}_+$ are not separate subdomain copies. They are two algebraic maps on the same both-face connectivity, while $\mathbf{S}$ uses the same side information in the reverse direction.

For an oriented interface face $F \in \Gamma$ with N_v vertices and adjacent volume DOFs

$$d_\ell^+(F),\ d_\ell^-(F), \qquad \ell = 1, \dots, N_v,$$

the nonzero entries of the signed gather and scatter operators are

$$B_{F,d_\ell^+} = \frac{1}{N_v}, \qquad B_{F,d_\ell^-} = -\frac{1}{N_v},$$

and

$$S_{d_\ell^+,F} = \frac{|F|}{N_v}, \qquad S_{d_\ell^-,F} = -\frac{|F|}{N_v}.$$

The unsigned gather operator uses the same DOFs but removes the orientation sign,

$$B^+_{F,d_\ell^+} = B^+_{F,d_\ell^-} = \frac{1}{N_v}.$$

- **Volume stiffness matrices $(\mathbf{K}_i, \mathbf{K}_e)$:** The stiffness matrices for the intracellular and extracellular compartments are assembled on the decoupled EMI volume mesh by restricting the integration to elements carrying the corresponding region tags. Here, m and n denote the indices of volumetric basis functions φ_m and φ_n associated with the global EMI volume DOFs. The matrix entries are given by

$$(\mathbf{K}_i)_{mn} = \int_{\Omega_\mathrm{i}} \sigma_\mathrm{i} \nabla\varphi_m \cdot \nabla\varphi_n \,\mathrm{d}x, \qquad (\mathbf{K}_e)_{mn} = \int_{\Omega_\mathrm{e}} \sigma_\mathrm{e} \nabla\varphi_m \cdot \nabla\varphi_n \,\mathrm{d}x. \tag{16}$$

  This selective assembly uses one derived EMI volume mesh, but the duplicated interface vertices ensure that basis functions are continuous only within their own subdomain. No separate intracellular and extracellular volume meshes are required.

- **Gather matrix (B):** This operator maps volumetric potentials $\mathbf{u}$ to the transmembrane voltage at interface-face barycenters. On membrane faces, $\mathbf{B}$ is partitioned into intracellular and extracellular components, $\mathbf{B}_\mathrm{i}$ and $\mathbf{B}_\mathrm{e}$, such that the barycentric voltage vector is given by $v_\mathrm{b} = \mathbf{B}_\mathrm{i}\mathbf{u}_\mathrm{i} - \mathbf{B}_\mathrm{e}\mathbf{u}_\mathrm{e}$. For a triangular interface face F with vertices (n_1, n_2, n_3), the interpolation is defined as

$$v_b(F) = \frac{1}{N_v}\sum_{j=1}^{N_v} u_\mathrm{i}(n_j) - \frac{1}{N_v}\sum_{j=1}^{N_v} u_\mathrm{e}(n_j), \tag{17}$$

  where $N_v = 3$. In its sparse matrix representation, each row of $\mathbf{B}$ contains weights of $\pm 1/3$ for the six interface DOFs. The operator $\mathbf{B}$ is a rectangular matrix of dimension $F \times N$, where F denotes the number of face barycenters on the computational interface Γ.

- **Unsigned Interpolation Operator ($\mathbf{B}_+$):** This operator is constructed analogously to $\mathbf{B}$, but without the sign difference between intracellular and extracellular contributions. It is primarily used for current injection and averaging. To account for the redundant two-sided representation of the computational interface Γ, a scaling factor of 0.5 is applied when mapping face currents back to physical values, thereby compensating for the duplication inherent in the two-sided representation. The dimension of $\mathbf{B}_+$ is the same as that of $\mathbf{B}$.

- **Scatter matrix (S):** Following the ionic update, the computed face currents I_b are distributed back to the volumetric right-hand side. For a vertex n shared by a set of triangular interface faces $\mathcal{F}_n$, the scattered current contribution is

$$I_{\text{node}}(n) = \sum_{F \in \mathcal{F}_n} \frac{1}{N_v} A_F I_b(F). \tag{18}$$

  In matrix form, $\mathbf{S}$ is the scattering operator from interface-face quantities to volumetric DOFs and therefore has dimension $N \times F$. It has the adjoint dimension of $\mathbf{B}$, but it is assembled as a separate operator because the area weights, side information, and signs are determined from the retained face and counter-face links.

- **Membrane mass contribution ($\mathbf{M}_\Gamma$):** To implement the semi-implicit time integration, the membrane capacitance term is assembled into the left-hand side. It is defined on interface faces and inserted into the volume system through the retained side-to-volume DOF links:

$$(\mathbf{M}_\Gamma)_{ij} = \frac{1}{2} \int_\Gamma C_m \psi_i \psi_j \, \mathrm{d}s. \tag{19}$$

  The factor of $1/2$ is required in the two-sided interface representation to avoid double counting when integrating over both sides of the interface. In the monolithic system this face contribution is inserted into the rows and columns of the adjacent volume DOFs.

While the unique physical interface γ is optimal for ionic ODE evaluation and output, the implementation retains the two-sided computational interface Γ as the runtime coupling layout. This is not only a temporary assembly device. At every time step, volume potentials are gathered to interface quantities, ionic and gap-junction currents are lifted back to the two-sided layout, and the resulting currents are scattered to the correct intracellular or extracellular volume DOFs. Keeping one entry per adjacent side makes the side-to-volume association local and explicit. A formulation based only on the unique-face interface would have to attach both neighboring volume connectivities to a single face and synchronize that information whenever the two sides are owned by different MPI ranks.

The combination of the two layouts separates two concerns. The unique-face layout avoids redundant ionic-state storage and provides reproducible face-wise output. The both-face layout preserves the orientation and side information needed by $\mathbf{B}$, $\mathbf{B}_+$, $\mathbf{S}$, and the surface mass contribution during the repeated time-stepping operations. Consequently, the potential jump is evaluated and scattered through sparse matrix-vector operations, while the global system remains compatible with standard distributed-memory solver backends such as PETSc and Ginkgo.

3.5.2 System matrix assembly

After the membrane update in the operator-splitting scheme, openCARP assembles and solves the parabolic EMI potential problem as a single coupled algebraic system on the decoupled volume mesh. The unknowns of this system are the intracellular and extracellular potentials. The ionic state variables are advanced separately on the face-based membrane representation

and enter the potential solve only through the right-hand side. Thus, the coupling represented in the system matrix is the capacitive membrane contribution between potential DOFs on adjacent subdomains, rather than an implicit solve for the ionic ODE variables.

The system matrix $\mathbf{A}$ is assembled as

$$\mathbf{A} = \mathbf{K} + \frac{1}{\Delta t}\mathbf{M}_\Gamma, \tag{20}$$

where the block-diagonal stiffness matrix $\mathbf{K}$ represents diffusion within the subdomains Ω_i of the decoupled EMI mesh. The individual blocks of $\mathbf{K}$ are symmetric positive semidefinite because they represent pure Neumann subproblems; before membrane coupling, their nullspace dimension is determined by the number of connected subdomain blocks.

The coupling between the blocks is realized through the membrane mass contribution $\mathbf{M}_\Gamma$. For each membrane facet F, the local contribution to $\mathbf{M}_\Gamma$ assumes the form

$$(\mathbf{M}_\Gamma)_F = \begin{pmatrix} \mathbf{M}_F & -\mathbf{M}_F \\ -\mathbf{M}_F & \mathbf{M}_F \end{pmatrix}, \tag{21}$$

where $\mathbf{M}_F$ is the standard surface mass matrix for the triangular facet and the block structure corresponds to the two incident subdomains of F. This structure ensures that the potential difference $v = u_i - u_j$ is correctly coupled to the capacitive current, thereby coupling the two domains across the membrane. After the membrane mass contributions are added, the remaining nullspace is the constant global potential mode associated with the pure-Neumann problem. How the solvers remove the remaining constant-potential nullspace, and how this affects admissible stimulation protocols, is described in Section 4.4.1.

The assembly of the system matrix $\mathbf{A} = \mathbf{K} + \frac{1}{\Delta t}\mathbf{M}_\Gamma$ is the central computational step, since volumetric and surface operators defined on different derived mesh objects must be inserted into a single linear system. This is handled through the retained interface links: each oriented surface vertex is associated with an input reference vertex, an adjacent region tag, the corresponding decoupled EMI volume DOF, and finally an algebraic PETSc row.

The initialization phase proceeds as follows:

1. **Volume operator insertion:** The system matrix is first populated with the stiffness operator $\mathbf{K} = \mathbf{K}_\mathrm{i} \oplus \mathbf{K}_\mathrm{e}$. Because the EMI volume mesh is topologically decoupled, interface vertices on the two sides of a membrane face carry distinct global DOFs in the PETSc parallel space.

2. **Surface-to-volume mapping:** For every oriented interface facet, the local surface vertices are mapped to their corresponding global PETSc rows using the retained reference-vertex and adjacent-tag information. This mapping provides the correspondence between surface entities and volume DOFs required for the membrane coupling terms.

3. **Monolithic insertion:** The coupling blocks of the membrane mass contribution $(\mathbf{M}_\Gamma)_F$ are inserted directly into the global left-hand side, thereby coupling the intracellular and extracellular potentials through the capacitive interface term and yielding a single fully coupled linear system.

3.6 Temporal Evolution and Solution Update

The evolution of the EMI system is governed by a semi-implicit discretization that transforms the continuous interface dynamics into a sequence of linear system solves. At each time step, the main EMI parabolic equation is solved only after the membrane quantities have been evaluated and the resulting source terms have been incorporated into the global right-hand side.

The sequence of operations performed at each time step, illustrated in Figure 1, is as follows:

1. **State projection to membrane faces:** The current volumetric potentials $\mathbf{u}^n$ are interpolated to interface-face barycenters using the assembled gather path associated with $\mathbf{B}$. This yields the transmembrane voltages $v_\flat^n$ on the membrane representation used for the ionic update. The internal conversion from the two-sided coupling layout to the unique-face membrane layout is part of this assembled gather path and is not written as a separate operator here.

2. **Ionic Evaluation (LIMPET):** On γ, the LIMPET library updates the gating variables $\mathbf{w}$ and computes the ionic and gap-junction currents $\mathbf{I}_{\text{ion}}^{n+1}$.

3. **Current scattering:** The face-based ionic and gap-junction currents are inserted into the volumetric right-hand side vector $\mathbf{r}^n$ through the assembled scatter operator $\mathbf{S}$, according to equation (12). This scatter operation includes the side information, orientation signs, and area weights needed to distribute membrane currents to the correct decoupled volume DOFs.

4. **Parabolic Solve:** With the right-hand side fully assembled, the framework solves the EMI parabolic equation
$$\mathbf{A}\delta\mathbf{u} = \mathbf{r}^n$$
and updates the solution by
$$\mathbf{u}^{n+1} = \mathbf{u}^n + \delta\mathbf{u}.$$

4 EMI simulation settings in openCARP

The EMI model is integrated into openCARP [27, 25, 15] as a physics module that uses the existing HPC infrastructure of the framework while addressing the specific topological requirements of microscale electrophysiology.

The implementation is centered around a parabolic solver class that manages preprocessing, operator assembly, linear algebra, and time integration. During preprocessing, the tagged input mesh is converted into the derived decoupled EMI volume mesh and the associated surface meshes described in Section 3.2. The EMI module then interoperates with the standard openCARP stimulus interface, the LIMPET ionic model library, and the existing asynchronous output infrastructure. This design allows a consistent transition from homogenized continuum models to cellular-resolved EMI simulations, provided that a mesh with cellular resolution and structure as well as the corresponding region tags are supplied.

In this section, we describe the solver backends available for the EMI system, the integration of membrane models through LIMPET, and the stimulation mechanisms supported by the implementation.

4.1 Volume Conductivity Regions from Element Tags

Volume conductivities in the EMI implementation are assigned from the region tags of the input volume mesh. The EMI code reuses the standard openCARP `gregion` mechanism, but interprets it on the single decoupled EMI volume mesh rather than on separate intracellular and extracellular grids. Two `gregion` entries are reserved as defaults: `gregion[0]` collects all extracellular tags read from the `*.extra` file, and `gregion[1]` collects all intracellular myocyte tags read from the `*.intra` file. User-defined heterogeneous conductivity regions then start at `gregion[2]`. These entries may list individual element tags or tag ranges using the standard openCARP ID-set syntax; the parser expands this specification to the integer tag list stored in `gregion[i].ID`.

Let $\mathcal{E}$ and $\mathcal{I}$ be the extracellular and intracellular tag sets, and let

$$G_r := \texttt{gregion[r].ID}, \qquad r \geq 2,$$

denote the user-defined heterogeneous conductivity tag sets. The default sets used after applying these overrides are

$$G_0 = \mathcal{E} \setminus \bigcup_{r\geq 2} G_r, \qquad G_1 = \mathcal{I} \setminus \bigcup_{r\geq 2} G_r.$$

For an element T with tag ρ_T, the material region is

$$r(T) = \max\{\, r \mid \rho_T \in G_r \,\},$$

and the stiffness integrator uses the conductivity $\sigma_{r(T)}$.

During EMI initialization, any tag explicitly assigned to a user-defined `gregion[i]`, $i > 1$, is removed from the corresponding extracellular or intracellular default set. The remaining extracellular tags keep the material parameters of `gregion[0]`, and the remaining intracellular tags keep the material parameters of `gregion[1]`. This gives a compact way to represent scars, border zones, or other conductivity heterogeneities: the mesh generator marks those volumes with distinct element tags, and the input file assigns these tags or tag ranges to `gregion[2]`, `gregion[3]`, and so on with their own conductivities.

For the current EMI volume discretization, the conductivity used in each `gregion` is isotropic. The scalar value is taken from `gregion[i].g_bath`, multiplied by `gregion[i].g_mult`, converted from $\mathrm{S\,m^{-1}}$ to $\mathrm{mS\,\mu m^{-1}}$, and copied to the intracellular, extracellular, and bath eigenvalue slots used by the generic electric integrator. After these material records have been constructed, each element of the decoupled EMI mesh is assigned an element-wise material-region id. The stiffness integrator then selects the conductivity of each volume element through this region id. If a tag is accidentally assigned to multiple non-default `gregion` entries, the highest configured region id is used, following the standard openCARP material-region precedence rule.

4.2 Membrane Models and LIMPET Integration

The implementation supports interface-specific membrane models through tag-pair strings of the form $t_i : t_j$, which identify physical interface faces by their adjacent volume-region tags. Users can assign different membrane or gap-junction models through the corresponding input file entries. The face-based ionic initializer parses this configuration and assigns the selected LIMPET models to the barycenters of the unique-face surface mesh. This makes it possible to represent heterogeneous membrane behavior in different tagged regions. Since ionic states are stored on interface faces rather than on volume vertices, multiple membrane models can be used without modifying the decoupled EMI volume mesh or its volumetric operators.

The tag pairs used in this configuration are produced by the distributed surface-reconstruction step. For each unique interface face, preprocessing records the two adjacent volume-region tags in an interface-tag lookup. The face ionics setup then converts this tag-pair information into the element-wise region mask passed to LIMPET. Region 0 is used as the default membrane region, region 1 as the default gap-junction region, and additional `imp_region_emi` entries can override these defaults for selected tag pairs such as `"12:5"`. A tag pair whose two entries are both intracellular is classified as a gap junction; otherwise an intracellular–extracellular pair is classified as a membrane face. This makes the multi-ionic model assignment depend on the physical adjacency encoded by the input tags, not on processor-local surface indices.

For a unique interface face $F \in \gamma$, let

$$\tau(F) = \{\rho_{T^-}, \rho_{T^+}\}$$

be the unordered pair of adjacent volume tags. Without user overrides, the LIMPET region mask is

$$r_\gamma(F) = \begin{cases} 1, & \tau(F) \subset \mathcal{I}, \\ 0, & |\tau(F) \cap \mathcal{I}| = 1,\ |\tau(F) \cap \mathcal{E}| = 1. \end{cases}$$

If an entry `imp_region_emi[r]`, $r \geq 2$, lists the tag pair $\tau(F)$, then the override is applied and $r_\gamma(F) = r$.

4.3 Solver Integration with PETSc and Ginkgo

The EMI parabolic solve is dispatched through the abstract linear algebra layer, which decouples the EMI assembly and time-stepping logic from any specific solver backend. PETSc and Ginkgo provide concrete realizations of the same distributed matrix, vector, and solver interfaces, so that backend selection and solver configuration can be changed at runtime without modifying the EMI code path.

PETSc solver: The resulting linear system is solved through the PETSc library [22]. In the numerical experiments presented in this work, we employ the parallel implementation of the preconditioned conjugate gradient (CG) method provided by PETSc. The solver configuration is exposed through the standard openCARP input files, allowing runtime selection of Krylov methods and preconditioners without modifications to the EMI assembly. As algebraic multigrid (AMG) preconditioners, we consider both the native PETSc GAMG implementation, based on smoothed aggregation [34], and the BoomerAMG preconditioner available through the Hypre interface [14]. GAMG is based on a smoothed-aggregation multigrid strategy, where aggregates are constructed from the matrix graph using a modified maximal independent set (MIS) coarsening procedure. In contrast, BoomerAMG follows a more classical AMG approach based on strength-of-connection measures and Hybrid Maximal Independent Set (HMIS) coarsening, where strongly connected nodes are identified through a user-defined threshold parameter. Although the two approaches rely on different coarsening and interpolation strategies, both provide scalable distributed-memory preconditioning for the symmetric EMI parabolic system and can be configured directly through PETSc runtime options. In the numerical experiments of Section 5, we compare the performance of PETSc-GAMG and PETSc-Hypre (BoomerAMG) in terms of iteration count and overall time-to-solution.

Ginkgo solver: The EMI system can also be solved with the Ginkgo backend [2], a modern C++ linear operator framework that targets distributed memory and accelerator architectures with the same algorithmic code path. The backend adapter represents the assembled EMI matrix, whose rows and columns follow the algebraic numbering of the decoupled EMI volume mesh, as a distributed compressed-sparse-row operator and exposes distributed vectors through the same abstract linear-algebra interface used by the rest of the EMI time-stepping code. At runtime, the adapter instantiates the selected Krylov method, preconditioner, and stopping criterion from a JSON options file passed through `-parab_options_file`, so that no recompilation is needed when the solver method, the preconditioner, or its parameters are modified. The active Ginkgo executor is selected through a separate command-line flag (`-ginkgo_exec`), which determines whether matrices and vectors live on the host (`omp`/`reference`) or on a GPU (`cuda`, `hip`, or `dpcpp`); the same assembled distributed matrix and the same EMI time-stepping loop are reused on either side. The available preconditioners include classical block Jacobi, algebraic multigrid, and one-level Schwarz preconditioners [5] with configurable local solvers, so that the EMI parabolic solve can be tuned independently of the assembly when targeting different hardware architectures.

4.4 Stimulation Protocols and Regional Targeting

The EMI implementation reuses the standard openCARP stimulus interface, in which each stimulus is defined by a temporal protocol, a waveform, a physical type, and an electrode specification. In the EMI setting, the supported stimulation classes fall into three groups: intracellular stimuli acting on the intracellular potential u_i, extracellular stimuli acting on the extracellular potential u_e, and transmembrane stimuli contributing directly to the membrane forcing term I_{stim}.

Standard volumetric stimulation protocols are applied directly to the DOFs of the decoupled EMI volume mesh. During initialization, each stimulus is restricted to the appropriate subset of DOFs: extracellular stimuli act on extracellular DOFs, intracellular stimuli on intracellular DOFs, and transmembrane stimuli are converted from the nodal openCARP stimulus representation to the face-wise membrane representation. This allows region-aware stimulation on the decoupled mesh and supports direct comparison of different stimulation strategies within the same EMI framework.

A key feature of the implementation is the treatment of interface vertices. At a membrane vertex, the derived EMI mesh stores separate intracellular and extracellular DOFs with the same physical coordinates. When a stimulus is applied to a selected myocyte region, it is routed to the intended intracellular DOFs at the interface without directly exciting the corresponding extracellular DOFs. In membrane-driven cases, this also allows the potential jump v to be driven without numerical leakage into neighboring compartments.

For region-aware stimulus application, the implementation uses the unsigned interpolation operator $\mathbf{B}_+$ on the computational interface Γ. In contrast to $\mathbf{B}$, this operator does not introduce a sign change between the two sides of the interface, which makes it suitable for marking intracellular or extracellular interface DOFs separately. In complex multi-compartment junctions, where one physical point is represented by several decoupled EMI DOFs, this avoids ambiguities caused by the original shared vertex. Volumetric intracellular and extracellular current stimuli are accumulated directly on the corresponding nodal DOFs of the decoupled EMI volume mesh, whereas potential-based stimuli are enforced through the Dirichlet boundary-condition manager attached to the monolithic EMI system.

4.4.1 Reference potential and current compatibility

The stimulation setup determines whether the EMI parabolic solve has a fixed reference potential. If no Dirichlet or ground constraint is prescribed, the EMI system has the constant-potential nullspace inherited from the global Neumann problem. This nullspace reflects the fact that the absolute potential level is arbitrary: adding the same constant to all intracellular and extracellular potentials on the decoupled EMI volume mesh does not change any transmembrane voltage, flux balance, or membrane current. The implementation detects this case during matrix construction. When no Dirichlet boundary condition is enforced on the left-hand side, the solver setup marks the matrix as singular with a constant nullspace.

The two supported linear-algebra backends handle this information differently. For PETSc, the EMI solver attaches an explicit constant `MatNullSpace` object to the system matrix before the Krylov solver is configured. PETSc can therefore treat the matrix as singular with a known null space during the solve. After each solve, the computed potential increment is additionally projected to the zero-mean gauge by subtracting its global algebraic mean,

$$\delta\mathbf{u} \leftarrow \delta\mathbf{u} - \frac{1}{N}\sum_{k=1}^{N} \delta u_k \, \mathbf{1}.$$

This keeps the absolute potential level bounded and makes the gauge choice reproducible without

modifying the physically relevant voltage differences.

For the Ginkgo backend, the current wrapper does not attach an explicit matrix null-space object. Instead, it applies the same zero-mean projection after the Krylov solve whenever the EMI matrix is flagged as singular. Thus PETSc uses both solver-level null-space information and post-solve gauge normalization, whereas Ginkgo currently uses post-solve gauge normalization only. In both cases, the projected solution remains in the same physical equivalence class because only the constant mode is removed.

This treatment constrains admissible volumetric current stimuli. In the no-ground case, intracellular and extracellular current stimuli must satisfy the Neumann compatibility condition: the total injected current must vanish. The implementation therefore rejects unbalanced `I_in` or `I_ex` stimuli unless they are paired through a balanced source-sink specification or a reference/ground condition is supplied. Transmembrane stimuli satisfy this compatibility requirement by construction, because they appear as equal and opposite interface contributions between adjacent subdomains.

For transmembrane stimuli, the mathematical EMI model requires a membrane current density I_{ij}^{stim} defined on interface facets. The standard openCARP stimulus infrastructure, however, first constructs a nodal stimulus vector on the mesh. In the EMI implementation this nodal representation is therefore converted to the face-wise membrane representation before it is combined with the ionic current. We denote this assembled nodal-to-face projection by $\mathbf{P}_{\gamma}^{\mathrm{stim}}$. It maps the openCARP nodal stimulus layout directly to the unique-face membrane layout γ, including the unsigned face averaging and the factor required to compensate for the internal two-sided interface representation. If $\mathbf{I}_{\mathrm{nodal\_stim}}^{n}$ denotes the openCARP nodal stimulus vector, the face-wise stimulus used by the EMI membrane update is

$$\mathbf{I}_{\gamma,\mathrm{stim}}^{n} = \mathbf{P}_{\gamma}^{\mathrm{stim}}\, \mathbf{I}_{\mathrm{nodal\_stim}}^{n}.$$

The operator $\mathbf{P}_{\gamma}^{\mathrm{stim}}$ uses the same face-to-volume connectivity as the voltage gather operator $\mathbf{B}$, but without the intracellular/extracellular sign change. Since ionic currents are stored as one piecewise-constant value per surface facet, this operation is a face-average projection from the nodal openCARP stimulus layout to the membrane-current layout. The membrane current entering the right-hand side is then

$$\mathbf{I}_{\gamma}^{n} = \mathbf{I}_{\gamma,\mathrm{ion}}^{n} - \mathbf{I}_{\gamma,\mathrm{stim}}^{n},$$

which is lifted back to the two-sided interface and scattered to the decoupled volume DOFs as described in Section 3.5. Volumetric intracellular and extracellular stimuli are different: they are accumulated directly on the appropriate decoupled volume DOFs. Thus the conversion above is not a change of the EMI model, but the implementation bridge needed because openCARP describes stimuli nodally whereas EMI evaluates membrane forcing on surface facets.

5 Numerical Experiments

In this section, we evaluate the versatility, performance, and robustness of the EMI implementation in openCARP through a series of numerical experiments. The study has three main components: (i) a comparison of different stimulation strategies on the decoupled EMI volume mesh and associated membrane surface mesh, (ii) electrophysiological verification of cellular activation and heterogeneous face-based membrane configurations, and (iii) a detailed characterization of parallel scalability on HPC architectures. For reproducibility, the final version should make the simulation inputs, solver settings, mesh identifiers, and post-processing scripts available together with the reported tables and figures. Before submission, the remaining numerical-section tasks are to fill the pending mesh statistics in Table 6, verify all solver and hardware settings

against the final log files, archive the input parameter files used for each experiment, and confirm that every plotted quantity can be regenerated from the archived simulation output.

5.1 Stimulation and Conduction in a Myocyte Chain

We used a chain of ten myocytes (Fig. 3) to verify the extracellular stimulation implementation and to examine the role of the gap junction resistance R_g on cell-to-cell conduction. Individual myocytes have a body length of 100 µm and a width of 15 µm in this model (center-to-center pitch: 110 µm). Along the largest extensions, we added extrusions of 5 µm with a contact surface (intercalated disc) of $10\times10\,\mu m^2$ to connect neighboring myocytes.

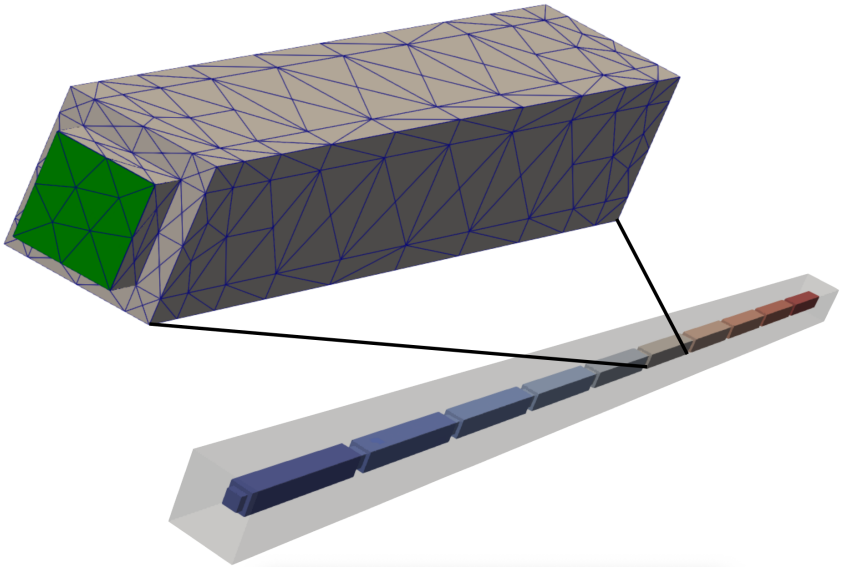


Figure 3: Simple 10-myocyte setup. Myocytes are indicated in different colors and are surrounded by an extracellular bath (gray transparent box). On the top, a single myocyte is shown with its spatial discretization and the interface region between neighbouring myocytes (intercalated disc, Γ_g) highlighted in green.

5.1.1 Extracellular Stimulation

All myocytes used the Courtemanche et al. ionic model [10] with intercalated discs described as a passive linear conductance ($I = (V - V_{rest})/R_g$, $V_{rest} = 0$; openCARP's `Plonsey` membrane law [28]) with gap junction resistance $R_g = 0.0045\,k\Omega\,cm^2$. The strand was surrounded by a rectangular extracellular bath (15 µm padding on all sides; bath conductivity $\sigma_e = 2.0\,S/m$; intracellular conductivity $\sigma_i = 0.4\,S/m$). Three plate electrodes were derived from the mesh bounding box: electrode **A** (anode) spanning the left bath cross-section, electrode **B** (cathode or ground) spanning the right, and electrode **C**, a 10 µm strip centred on the top bath surface at the strand midpoint (Fig. 4). The inner A–B gap is $d_{AB} = 1115\,\mu m$. The target stimulus was a uniform electric field $E = 2\,V/cm$ directed from A to B, delivered as a monophasic rectangular pulse of duration $t_{off} = 2\,ms$ from $t = 0$. Six circuit topologies (Fig. 4, panels (a)–(f)) implemented this target field using different source types and grounding configurations. For the three voltage modes, the applied potential difference was derived from the uniform-field assumption:

$$\Delta\phi_{AB} = E \cdot d_{AB} \approx 223\,mV. \tag{22}$$

The voltage-clamp mode (panel (a)) clamped electrode A to +223 mV with B grounded and reset A to 0 mV at t_{off}; the open-loop voltage mode (panel (b)) applied the same potential

difference but disconnected the source at t_{off}, leaving the electrode potential to evolve freely under tissue dynamics; the balanced voltage mode (panel (c)) drove A and B symmetrically to $\pm 111.5\,\text{mV}$ with electrode C grounded at the strand midpoint as a potential reference. For the three current modes, the injected current was estimated from a parallel-conductor model of the bath cross-section [28]:

$$I = E \cdot (\sigma_{\text{e}}\, A_{\text{e}} + \sigma_{\text{i}}\, A_{\text{i}}), \tag{23}$$

where $A_{\text{i}} = 225\,\mu\text{m}^2$ is the intracellular cross-section and $A_{\text{e}} = 1800\,\mu\text{m}^2$ is the surrounding bath area, yielding $I \approx 0.74\,\mu\text{A}$. The current-injection mode (panel (d)) injected $+I$ at A with B grounded; the balanced current mode (panel (e)) injected $+I$ at A and $-I$ at B (net zero), yielding a pure-Neumann system for the extracellular potential ϕ_{e}, resolved by attaching a constant nullspace; the balanced current with reference ground mode (panel (f)) added a Dirichlet ground at C to make the solution unique without altering the injected currents.

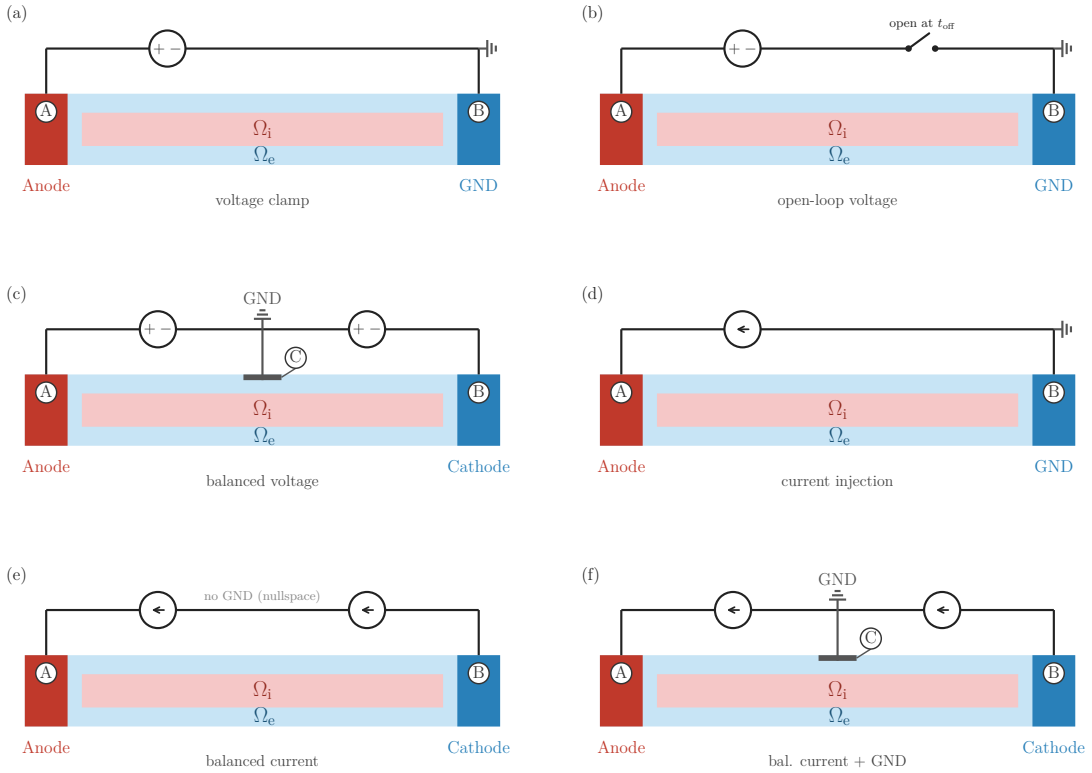


Figure 4: Six extracellular stimulation modes. Panels (a)–(f) correspond to the voltage-clamp, open-loop voltage, balanced voltage, current injection, balanced current, and balanced current with reference ground modes, respectively. Ω_{e} (blue): extracellular bath; Ω_{i} (red): intracellular domain. Electrode A (red bar, left) is the anode; electrode B (blue bar, right) is the cathode or ground; electrode C (grey patch, top centre) serves as ground in modes (c) and (f). Voltage and current source symbols follow IEC conventions; the open switch in panel (b) indicates that electrode A is disconnected at $t = t_{\text{off}}$.

All ten myocytes activated in every mode. The action-potential wavefront initiated near electrode B (cell 10, rightmost) and propagated leftward to cell 1. Conduction velocity, computed as the reciprocal slope of a least-squares linear fit of activation time versus cell-centroid position over interior cells 2–9, spanned 21.60–21.69 cm/s across all six topologies. The maximum inter-mode activation-time deviation relative to the voltage-clamp mode was $\Delta t_{\max} \leq 0.020\,\text{ms}$, corresponding to one output sampling interval. Figure 5 shows the voltage traces and per-mode activation-time residuals. Voltage modes (voltage clamp, open-loop voltage, balanced voltage) activated tissue slightly before the per-cell mean (+0.001 to +0.006 ms); current modes (current injection, balanced current, balanced current with reference ground) arrived slightly

after (-0.020 ms).

The near-identical activation across six independently formulated circuit topologies confirms that the extracellular boundary-condition machinery in openCARP's EMI framework (Dirichlet grounding, Neumann current injection, nullspace regularisation, and mixed configurations) is correctly implemented. The systematic voltage-versus-current offset is consistent with the physical distinction between the two source classes: voltage clamping enforces the target potential exactly at the electrode surface, whereas the current estimate (23) neglects edge spreading and therefore delivers a slightly different realised field.

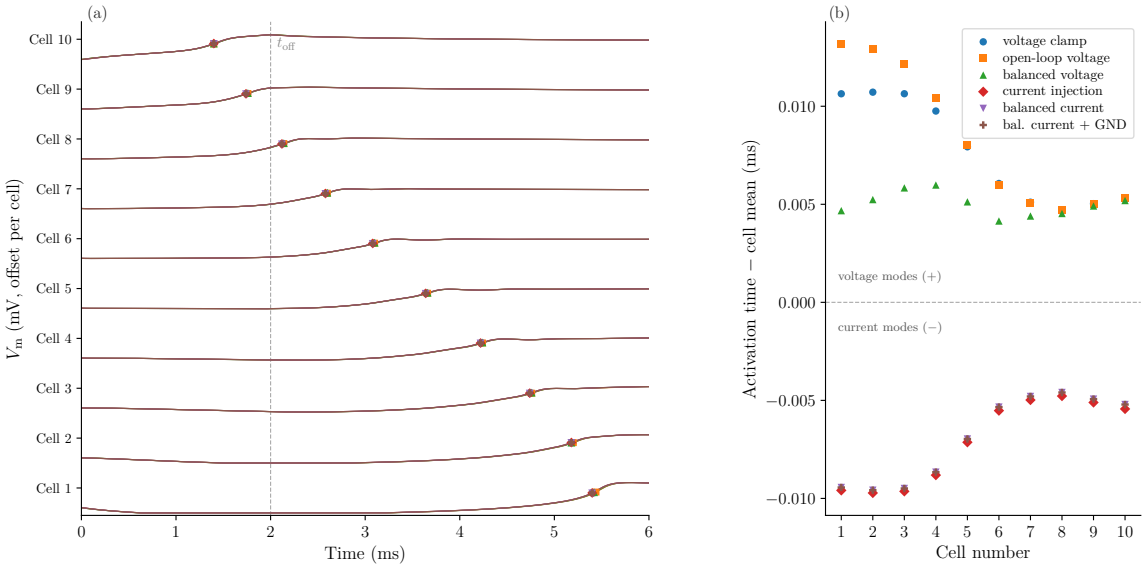


Figure 5: Transmembrane voltage traces and per-mode activation-time residuals for all six extracellular stimulation modes. **(a)** V_m traces for all ten myocytes, colour-coded by stimulation mode, all six modes overlaid and offset by 200 mV per cell for clarity. Cell 10 (nearest cathode B) activates first; the wavefront propagates leftward to cell 1. Per-mode activation markers (colour and shape, matching panel (b)) indicate the first crossing of $V_m = -20$ mV; the dashed vertical line marks the end of the 2 ms stimulus pulse. **(b)** Activation-time deviation of each mode from the per-cell mean across all six modes. Voltage modes (circles, squares, triangles) arrive slightly before the mean; current modes (diamonds, inverted triangles, plus signs) arrive slightly after.

5.1.2 Transmembrane Current Stimulation

As an alternative to extracellular field stimulation, a current density can be applied directly to the membrane interface of the target cell, bypassing the extracellular circuit entirely. A monophasic pulse of 150 µA/cm^2 and duration $t_{off} = 2$ ms was applied as a distributed inward flux to the membrane faces of the left half of cell 1; no electrode geometry or ground was required.

At the nominal resistance ($R_g = 0.0045$ kΩ cm^2), all ten cells activated and the wavefront propagated rightward from cell 1 to cell 10 through the gap junctions, yielding a conduction velocity of 26.83 cm/s (Fig. 6). To further characterise the influence of intercellular coupling, the gap junction resistance was varied over nearly three decades (0.001–0.500 kΩ cm^2, 20 logarithmically spaced values) while the stimulus was held fixed. Over the sweep, conduction velocity decreased monotonically from 49.06 cm/s at $R_g = 0.001$ kΩ cm^2 to 0.97 cm/s at $R_g = 0.300$ kΩ cm^2, a roughly 50-fold variation (Fig. 7). At $R_g = 0.500$ kΩ cm^2, the wave activated only three of ten cells; the sub-threshold electrotonic foot of the arrested front decayed back toward rest, confirming true conduction block. The block threshold lies between 0.300 and 0.500 kΩ cm^2.

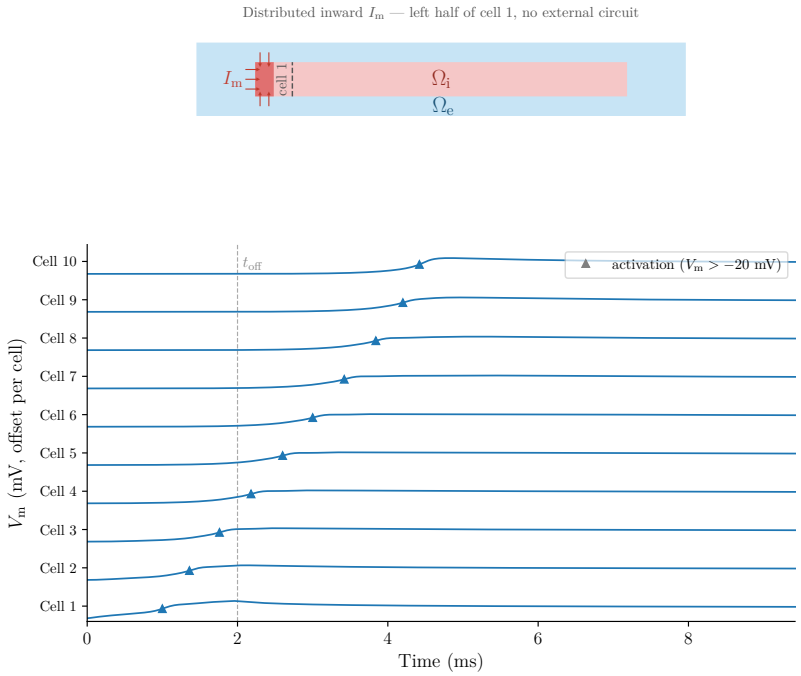


Figure 6: Transmembrane current stimulation. **(top)** Circuit schematic: a distributed inward current I_m is applied to the membrane faces of the left half of cell 1 (darker shading); no external electrodes or bath field are present. **(bottom)** V_m traces for all ten myocytes, offset by 250 mV per cell for clarity. Cell 1 activates first; the wavefront propagates rightward. The dashed vertical line marks the end of the 2 ms stimulus pulse.

The higher conduction velocity in Figure 6 relative to the extracellular modes (21.60–21.69 cm/s) reflects the absence of a bath-redistribution step: membrane depolarisation was driven directly, without current having to enter the cell through the extracellular compartment. In homogenised bidomain and monodomain formulations [9], gap junction properties are absorbed into a bulk intracellular conductivity tensor; the same propagating conduction velocities observed here can in principle be reproduced by retuning that tensor, but the gap junction resistance R_g in the EMI model is a condition applied at each individual intercalated disc interface [33], making it a directly interpretable microscale parameter. The existence of a sharp conduction-block threshold (above which the wavefront arrests and the downstream electrotonic foot decays sub-threshold) is a discrete, geometry-resolved phenomenon: it arises from the inability of the intercalated-disc interface to transfer sufficient current to bring the downstream cell to threshold, and is not captured by smooth bulk-conductivity formulations.

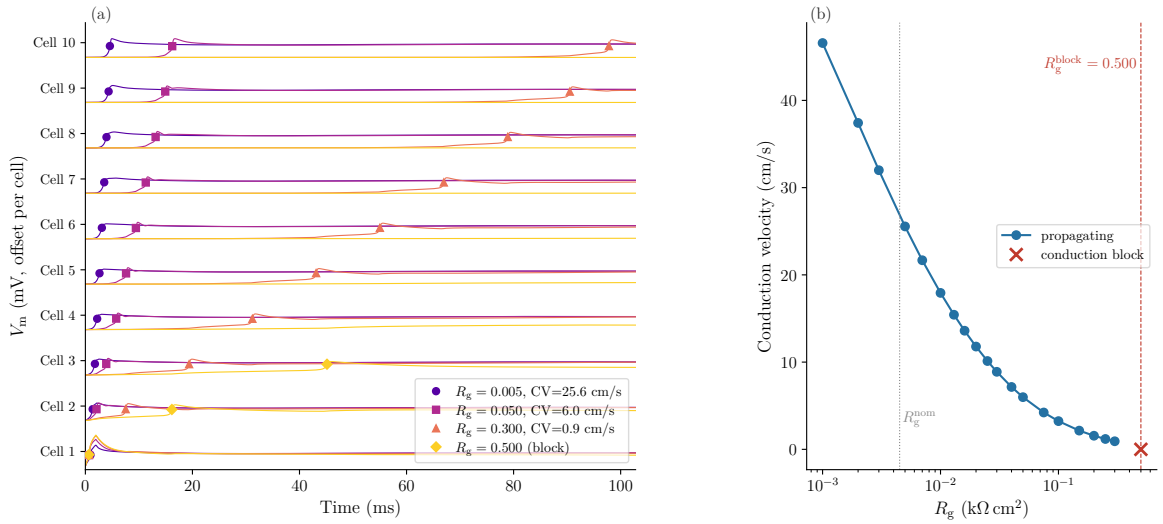


Figure 7: Gap junction resistance R_g sweep (transmembrane current stimulus, 150 µA/cm^2, $t_{off} = 2$ ms). **(a)** Transmembrane voltage traces for all ten myocytes, colour-coded by R_g, offset by 250 mV per cell, for four representative R_g values. Activation markers (same colour and shape as the trace, coded by R_g, with conduction velocity from cells 2–9 in the legend) indicate the first crossing of $V_m = -20$ mV. At $R_g = 0.500$ kΩ cm^2 the wavefront arrests after cell 3 and the downstream electrotonic foot decays sub-threshold. **(b)** CV as a function of R_g over the full sweep (0.001–0.500 kΩ cm^2). Filled circles: all ten cells activate (propagating); red cross: conduction block at $R_g = 0.500$ kΩ cm^2. The block threshold lies between 0.300 and 0.500 kΩ cm^2 (dashed red line). The dotted grey line marks the nominal resistance ($R_g^{nom} = 0.0045$ kΩ cm^2) used in the extracellular stimulation experiments (Section 5.1.1).

5.2 Spatiotemporal Dynamics, Multiple Ionic Models and Electrograms

We demonstrate wave propagation around a scar region modeled with multiple ionic models, including spatiotemporal dynamics and electrogram (EGM) extraction. For this purpose, we created a quasi-2D tissue geometry using the `anni` code [29], which is available on Zenodo [31]. Its dimensions are $(x, y, z) \approx (1.5 \times 0.08 \times 1.5)$ mm^3, surrounded by extracellular space; see Figure 8. We coarsened the input mesh to reduce runtime while preserving the membrane surface sufficiently, resulting in an average edge length of (8.70 ± 4.01) µm. During preprocessing, this tagged input mesh is converted to the decoupled EMI volume mesh and the unique-face surface mesh used for membrane quantities. In the center of the mesh, a circular non-conductive scar core with radius 0.25 mm is modeled by assigning the passive Plonsey model [28] to membrane faces in that region. The scar core is surrounded by a concentric border-zone ring of thickness 0.2 mm. In this region, we consider a mixture of surviving excitable myocytes using the Courtemanche et al. model [10], non-excitable myocytes using the Plonsey model, and fibroblasts whose membrane dynamics are represented by the MacCannell model [21]. These three cell models are assigned randomly to the unique-face membrane facets of individual myocytes in the border zone, with equal proportions for each cell type. The scar is surrounded by tissue under healthy conditions, again using the Courtemanche model. To initiate wave propagation, we applied a current density stimulus with an amplitude of $I_{stim} = 400$ µA/cm^2 for 1 ms at $\Gamma_{stim} = \{\vec{x} \in \Gamma_m : x < 150\,\mu m\}$, see Fig. 9. We evolved the system in time for $T = 3$ ms and applied a Dirichlet boundary condition located at a corner, $\vec{x}^D = (\max x, \max y, \max z) \in \Omega_e$, $\phi_e(\vec{x}^D, t) = 0\ \forall t \in [0, T]$. In Fig. 8, we show that, besides the possibility to represent spatially extended electrodes, EGMs can simply be taken as $\phi_e(\vec{x}_{elec}, t)$ for any point electrode position $\vec{x}_{elec} \in \Omega_e$.

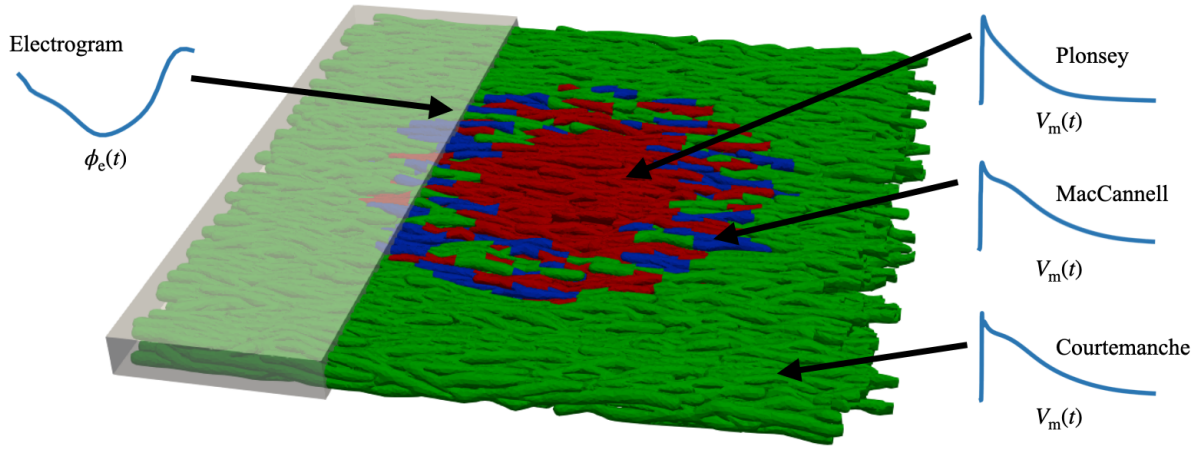


Figure 8: Simulation setup for the heterogeneous quasi-2D tissue example. Green cells use the Courtemanche model under healthy conditions, red cells use the passive Plonsey model, and blue cells use the MacCannell fibroblast model. The transparent box denotes the surrounding extracellular space Ω_e, shown only partially. For each ionic model, an example transmembrane-voltage trace V_m is shown over time on the membrane surface. The traces are similar because the local response is strongly influenced by neighboring excitable Courtemanche cells. An example electrogram is also shown; such signals can be extracted from any point $\vec{x} \in \Omega_e$.

Figure 8 also shows the spatiotemporal dynamics of the excitation wave through V_m on the unique-face membrane surface. This example demonstrates the use of multiple ionic models in the same setup, the extraction of EGMs from extracellular bath DOFs, and the detailed resolution of V_m in microstructurally heterogeneous electrophysiological regions enabled by the EMI model. Since the scope of this section is to provide an overview of the available functionality, these findings are presented for demonstration purposes and are not analyzed further.

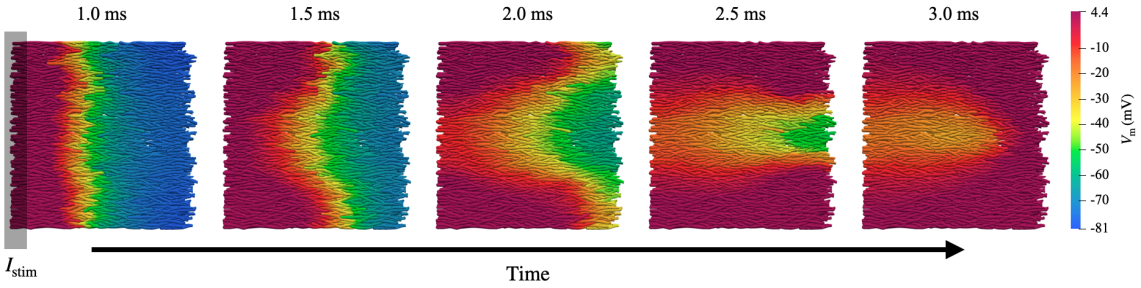


Figure 9: Spatiotemporal dynamics of a wave propagating around a circular scar region. Only the unique-face membrane surface is shown, with V_m color-coded on the surface.

5.2.1 Performance Considerations for Different Ionic Models

Impact of Ionic Model Complexity: We evaluate the computational cost of different membrane models by comparing the runtime of a simplified two-variable Aliev–Panfilov model [1] with higher-dimensional ionic models. Additional membrane models considered in the study include Luo–Rudy 91 [20], Grandi [13], O'Hara–Rudy [24], and Tomek [32]. As expected, the cost of the face-based ionic update on the unique-face surface mesh increases with the number of gating variables and the complexity of the membrane model. With the exception of the Grandi model, the ionic cell computations on the cell membranes make a negligible contribution to the overall simulation runtime.

Table 4: Impact of ionic models on the computation time for the $\mathcal{T}_2$ mesh over 200 time steps on a dual-socket AMD Epyc Turin 9755 system with 256 cores per node. The experimental setup is otherwise the same as in Section 5.3.1.

Ionic model	itr	EMI init (s)	EMI compute (s)	ionic init (s)	ionic compute (s)
AlievPanfilov	1430	38.47	587.45	0.11	0.06
Grandi	1086	39.06	459.99	0.08	19.40
LuoRudy91	923	39.31	384.08	0.09	0.28
OHara	1007	38.61	417.98	0.30	2.47
Tomek	876	39.27	368.34	0.24	3.96

The EMI and ionic timings are reported as exclusive timings: the ionic initialization and ionic compute timers are subtracted from the total EMI timers before output. Therefore, `EMI init` contains the one-time EMI setup excluding ionic-model initialization, including distributed EMI mesh processing, interface reconstruction, DoF decoupling and indexing, construction of transfer operators and coupling matrices, parabolic solver setup, stimulus setup, and initial membrane-voltage projection. `EMI compute` contains the time spent per time step outside the ionic update, including stimulus handling, transfer of ionic currents between surface layouts, scattering to the volume right-hand side, the parabolic solve, and reconstruction of the membrane voltage. The `ionic init` timer measures initialization of the face-based ionic model data structures, including region masks, ionic model instances, and state variables on the unique-face surface representation. The `ionic compute` timer measures the ionic update itself, i.e. ionic-current evaluation and state-variable updates on the unique-face surface mesh. Transfers between the ionic surface representation and the volumetric EMI system are counted in `EMI compute`, not in `ionic compute`.

5.3 Parallel Performance and HPC Scalability

The primary objective of the EMI implementation in openCARP is to enable cell-resolved cardiac simulations at scale. To assess the parallel performance of the framework, we report strong- and weak-scaling studies on a modern HPC system. Unless otherwise stated, the experiments in this subsection were performed on CPU-only resources using the PETSc backend.

Solver comparison with the PETSc backend: Table 5 compares the performance of PETSc-GAMG with PETSc-Hypre (BoomerAMG) on mesh $\mathcal{T}_2$. Both solvers are used within a CG framework with identical convergence tolerances (`ksp_rtol 1e-6`, `ksp_atol 1e-8`). For Hypre, we employ the BoomerAMG preconditioner (`-pc_type hypre -pc_hypre_type boomeramg`), while GAMG is employed in openCARP as the default PETSc configuration. For BoomerAMG we employed as additional parameter `-pc_hypre_boomeramg_strong_threshold 0.5`, while all other parameters were left at their default values.

A first observation is that BoomerAMG achieves a significantly lower iteration count (approximately 6 iterations per solve) compared to GAMG (about 54–56 iterations). This indicates that the Hypre hierarchy provides a more accurate approximation of the inverse operator, leading to faster convergence in terms of Krylov iterations.

However, this reduction in iteration count does not translate into improved time-to-solution. On the contrary, BoomerAMG is consistently slower than GAMG across all tested configurations, in some cases by a substantial margin. For instance, at 144 ranks the `EMI compute` time with Hypre is more than 2.5× larger than with GAMG, and although the gap narrows at higher core counts, GAMG remains more efficient overall.

This behavior suggests that the per-iteration cost of BoomerAMG is significantly higher than

that of GAMG on the considered CPU architecture. The additional setup complexity, denser coarse-grid operators, and more expensive smoothing and interpolation steps in BoomerAMG likely outweigh the benefit of reduced iteration counts. Moreover, we observed that several Hypre runs approach the time limit or fail to complete within the allocated time, further highlighting its higher computational cost in this setting.

Overall, while BoomerAMG exhibits superior convergence properties in terms of iteration count, PETSc-GAMG provides a better balance between convergence and per-iteration efficiency, resulting in a lower overall runtime for the EMI problem at hand.

Table 5: Comparison between PETSc-GAMG (GAMG) and PETSc-Hypre (BAMG) solvers for mesh $\mathcal{T}_2$.

	PETSc-GAMG (GAMG)		**PETSc-Hypre (BAMG)**	
#ranks	itr	EMI compute (s)	itr	EMI compute (s)
144	54.47	1669.74	5.87	4509.69
288	53.92	1481.32	5.83	2759.72
576	54.35	394.40	5.95	1326.84
1152	55.58	317.67	5.92	333.45

5.3.1 Strong Scaling Study with the PETSc Backend

We assess the strong-scaling behavior of the distributed EMI implementation on the reference meshes $\mathcal{T}_2$, $\mathcal{T}_5$, and $\mathcal{T}_7$ by increasing the number of MPI ranks at fixed problem size. All runs are performed with the PETSc backend using a conjugate-gradient (CG) solver preconditioned with GAMG, and a relative residual tolerance consistent with the production settings used throughout the manuscript. The same assembly path, solver configuration, and physical parameters are retained across all runs; only the number of ranks changes.

Table 7 reports the iteration count, end-to-end timings, and a phase breakdown into EMI initialization (`EMI init`), EMI parabolic compute (`EMI compute`), and ionic updates. We observe that the iteration count remains essentially constant across all configurations, including the largest mesh $\mathcal{T}_7$, indicating that the algebraic multigrid preconditioner is robust with respect to the parallel decomposition. This stability is a key prerequisite for strong scalability, as it prevents iteration-count inflation from masking parallel speedup.

The dominant contribution to the time-to-solution is the parabolic solve. For mesh $\mathcal{T}_2$, the `EMI compute` phase decreases from 1669.74 s at 144 ranks to 317.67 s at 1152 ranks, corresponding to a speedup of approximately 5.3× for an 8× increase in resources. A similar trend is observed for $\mathcal{T}_5$, where the compute time reduces from 2900.49 s at 576 ranks to 578.45 s at 4608 ranks. For the largest mesh $\mathcal{T}_7$, the compute time decreases from 4201.74 s at 1152 ranks to 923.53 s at 9216 ranks, corresponding to a speedup of approximately 4.5×. While still favorable, this scaling is less efficient than for the smaller meshes, and a clear saturation is observed at the highest core counts (from 4608 to 9216 ranks), indicating the onset of strong-scaling limits where communication and synchronization costs become dominant.

In contrast, the initialization phase (`EMI init`) exhibits weaker scalability and, in some cases, increasing cost at higher ranks. This behavior is particularly evident for $\mathcal{T}_7$, where the initialization time grows up to 790.87 s at 9216 ranks, becoming a substantial fraction of the total runtime. This is expected, as initialization involves mesh partitioning, surface extraction, and global indexing operations that are less amenable to parallel scaling and incur additional communication overhead. A similar trend can be observed in the ionic initialization, which increases noticeably at large core counts due to the communication required by face-based data structures.

The ionic updates remain negligible in all configurations, confirming that the overall performance is dominated by the linear solver. The excellent scaling of the parabolic compute phase can be attributed to the domain-aware partitioning strategy, which preserves cellular locality and minimizes MPI communication across membrane interfaces. This leads to a favorable balance between computation and communication, allowing the solver to efficiently exploit increased parallelism up to several thousand cores.

Finally, we note that for smaller core counts (e.g., 144 to 288 ranks on $\mathcal{T}_2$), the speedup is suboptimal and slight stagnation is observed. This regime reflects the transition from communication-dominated to computation-dominated execution, where the local problem size per rank is still sufficiently large to limit the benefits of parallelism. As the number of ranks increases, the solver enters a more favorable scaling regime; however, for the largest mesh $\mathcal{T}_7$, a second transition is observed at extreme concurrency, where communication and global operations again limit scalability.

Table 6: Extended mesh data for the scalability experiments. The column #vertices refers to the input mesh, #dofs to the decoupled EMI volume mesh, and #elems_surface to the unique-face surface mesh. Here, m denotes million and b denotes billion; entries marked "pending" require extraction from the final simulation logs before submission.

mesh	#vertices	#dofs	#elems_volume	#elems_surface	#intra	#extra	domain size
$\mathcal{T}_1$	3.5m	pending	21m	pending	531	pending	0.12 mm^3
$\mathcal{T}_2$	29m	37m	174m	30m	5641	8693	1 mm^3
$\mathcal{T}_3$	59m	76m	350m	66m	12411	16721	2 mm^3
$\mathcal{T}_4$	88m	114m	524m	101m	19063	24668	3 mm^3
$\mathcal{T}_5$	113m	146m	669m	132m	25023	32325	4 mm^3
$\mathcal{T}_6$	146m	190m	868m	169m	32143	40344	5 mm^3
$\mathcal{T}_7$	232m	302m	1.3b	272m	51578	62627	8 mm^3
$\mathcal{T}_8$	290m	pending	1.7b	pending	64893	77932	10 mm^3

Table 7: Strong scaling results for Meshes $\mathcal{T}_2$, $\mathcal{T}_5$ and $\mathcal{T}_7$.

mesh	#ranks	itr	EMI init (s)	EMI compute (s)	ionic init(s)	ionic compute(s)
$\mathcal{T}_2$	144	54.47	40.87	1669.74	1.35	0.33
$\mathcal{T}_2$	288	53.92	28.76	1481.32	2.90	0.10
$\mathcal{T}_2$	576	54.35	30.07	394.40	6.48	0.05
$\mathcal{T}_2$	1152	55.58	48.33	317.67	13.54	0.02
$\mathcal{T}_5$	576	55.02	99.37	2900.49	57.32	0.27
$\mathcal{T}_5$	1152	56.31	99.95	1882.73	13.07	0.76
$\mathcal{T}_5$	2304	56.20	136.57	792.06	29.26	0.34
$\mathcal{T}_5$	4608	56.72	254.05	578.45	58.68	0.03
$\mathcal{T}_7$	1152	64.99	175.57	4201.74	16.41	0.33
$\mathcal{T}_7$	2304	63.58	204.23	2000.55	29.45	0.12
$\mathcal{T}_7$	4608	61.55	396.06	1005.29	62.76	0.05
$\mathcal{T}_7$	9216	64.28	790.87	923.53	465.01	0.02

5.3.2 Lego-mesh weak-scaling benchmark with the Ginkgo backend

To exercise the distributed EMI implementation in a setting decoupled from the geometric and topological complexity of the reference meshes $\mathcal{T}_1, \ldots, \mathcal{T}_6$, we use a synthetic structured benchmark which we call the *lego mesh* (the `emigrid` configuration distributed with our implementation). The

lego mesh is generated by `gmsh` from a parameterized geometry that places an $n_x \times n_y \times n_z$ array of cuboidal myocytes embedded in extracellular tissue, connected through three intercalated-disk segments (one per axis). The default cell footprint is $100\,\mu\text{m} \times 40\,\mu\text{m} \times 40\,\mu\text{m}$ with intercalated-disk thickness $5\,\mu\text{m}$ and width $10\,\mu\text{m}$; the characteristic mesh length is bounded between $2\,\mu\text{m}$ and $5\,\mu\text{m}$. Each myocyte and the surrounding extracellular space carry a unique tag, so that increasing (n_x, n_y, n_z) produces meshes with a controlled, predictable number of EMI interfaces and a near-constant per-cell discretization cost. The lego mesh is therefore well suited to weak-scaling studies: increasing the rank count by replicating cells along one or more axes grows the global problem at an essentially constant per-rank work load.

Table 8 reports the discrete-problem statistics of the four lego configurations used in the present weak-scaling study. The number of volumetric elements in the derived EMI mesh, the number of unique and two-sided membrane faces, and the number of degrees of freedom of the parabolic system all grow linearly with the cell count $n_x n_y n_z$, confirming that the per-rank work load is preserved across the four configurations.

Table 8: Discrete problem sizes of the lego meshes used in the weak-scaling study. #cells is the number of intracellular myocytes (each carries one intracellular and one extracellular tag), #elem the number of volumetric tetrahedra of the decoupled EMI mesh, $\#\Gamma$ the number of unique membrane faces, $\#\Gamma^{\pm}$ the number of decoupled membrane faces (counting both sides), and #DOFs the dimension of the parabolic system.

grid	#cells	#elem	$\#\Gamma$	$\#\Gamma^{\pm}$	#DOFs
$4 \times 4 \times 4$	64	529 511	88 076	176 152	139 177
$8 \times 4 \times 4$	128	1 061 804	176 798	353 596	277 706
$8 \times 8 \times 4$	256	2 123 938	353 484	706 968	548 814
$8 \times 8 \times 8$	512	4 247 815	707 926	1 415 852	1 085 449

Weak-scaling results on Alps GH200 GPUs. We use the lego configurations of Table 8 to assess the weak-scaling behavior of the Ginkgo backend on the GPU partition of CSCS Alps. Each MPI rank is bound to one NVIDIA Grace Hopper GH200 GPU. The intracellular regions carry an Aliev-Panfilov membrane model, and a Plonsey gap junction model with $R_m = 0.0045\,\Omega\,\text{m}^2$ is assigned to the decoupled membrane interfaces; the volumetric conductivities are $\sigma_e = 2.0$ and $\sigma_i = 0.4\,\text{S/m}$ for the extracellular and intracellular compartments, respectively. A single intracellular potential stimulus (closed-loop ϕ_i, type 10) of amplitude 20 mV and duration 1.0 ms is applied at $t = 0$ at the corner myocyte (tag 1); the resulting Dirichlet anchor is retained for the remainder of the run as the ground reference of the EMI parabolic system. The simulation is integrated with a uniform time step of $\Delta t = 10\,\mu\text{s}$ up to $T_{\text{end}} = 3\,\text{ms}$, yielding 300 parabolic solves and 300 ionic updates per run. The parabolic EMI system is solved with conjugate gradients, preconditioned with a one-level additive Schwarz preconditioner [5] whose rank-local solver is an SPD incomplete sparse approximate inverse (ISAI, sparsity power 1). The convergence criterion is an absolute residual norm of 10^{-6} relative to the right-hand-side norm, with a cap of 600 iterations per solve. The same Ginkgo backend, the same JSON solver description, and the same EMI assembly path are used across all four runs; only the mesh size and the rank count change.

Table 9 summarizes the measured iteration count, end-to-end wall-clock time, and a phase breakdown into EMI initialization (`EMI init`: mesh decoupling, surface extraction, indexing, and assembly), EMI parabolic compute (`EMI compute`: time-stepping cost dominated by the parabolic solve), and ionic compute (`Ionics compute`: face-based LIMPET evaluation). Figure 10 reports the weak-scaling efficiency relative to the single-GPU baseline, and Figure 11 reports the corresponding decomposition of the physics-loop time.

Two observations are worth noting. First, the average iteration count grows from 204 at $p = 1$ to 440 at $p = 8$ as the global problem grows from roughly $1.4 \cdot 10^5$ to $1.1 \cdot 10^6$ degrees of freedom,

and the maximum-per-solve count rises from 298 to 561, the latter still below the 600-iteration cap. The growth follows from the combined h-dependence of the underlying CG convergence rate and the additive-Schwarz coupling: as the rank count and the problem size grow, the local ISAI factor approximates a smaller fraction of the global inverse, so a larger number of CG iterations is needed to drive the residual below 10^{-6} times the right-hand-side norm. A more aggressive preconditioning strategy (for example, a multigrid local solver in the Schwarz block) is the natural next step to keep the iteration count under control on still larger problems. Second, the time-to-solution is dominated by the parabolic compute, which grows from 19.96 s at $p = 1$ to 60.77 s at $p = 8$, while the EMI initialization phase contributes only 4.41 to 11.71 s and the ionic update is negligible at all rank counts. The resulting weak-scaling efficiency, defined as $\eta_p = T_1/T_p$ on the end-to-end wall-clock time, is 0.70, 0.44, and 0.37 at $p = 2$, 4, 8. The bulk of the loss is attributable to the increasing iteration count, which inflates both the number of CG operator applications and the number of cross-rank inner-product reductions; the per-iteration GPU cost itself remains close to constant. The larger lego configurations to be reported in subsequent runs, together with a stronger Schwarz local solver, will allow the per-iteration weak-scaling behavior to be isolated from the iteration-count drift seen here.

Table 9: Weak-scaling results for Ginkgo CG with a one-level additive-Schwarz ISAI (SPD, sparsity power 1) preconditioner on the lego meshes of Table 8, on NVIDIA GH200 GPUs of CSCS Alps. The iteration count is averaged over all parabolic solves of the simulated time horizon. The total time reports the end-to-end wall-clock duration of the run, and $\eta_p = T_1/T_p$ is the weak-scaling efficiency relative to the single-GPU baseline.

grid	#GPUs	avg. itr	EMI init (s)	EMI compute (s)	Ionics compute (s)	total (s)	η_p
$4 \times 4 \times 4$	1	203.9	4.41	19.96	0.13	29.58	1.00
$8 \times 4 \times 4$	2	299.9	5.12	31.58	0.14	42.19	0.70
$8 \times 8 \times 4$	4	414.3	9.75	50.99	0.22	66.70	0.44
$8 \times 8 \times 8$	8	439.6	11.71	60.77	0.23	80.30	0.37

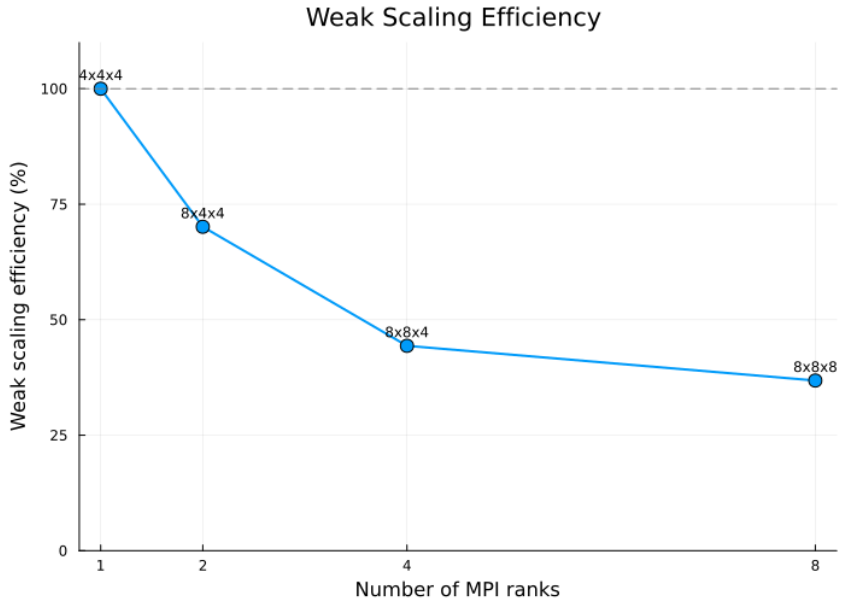


Figure 10: Weak-scaling efficiency $\eta_p = T_1/T_p$ for Ginkgo CG with a one-level additive-Schwarz ISAI preconditioner on the lego meshes of Table 8, on NVIDIA GH200 GPUs of CSCS Alps.

Mesh	Nodes	#ranks	Total [s]	Parab. solve [s]	Setup			
					Mesh [s]	Solver [s]	Output [s]	Other [s]
$\mathcal{T}_7$	16	1152	4626.2	4198.3	81.0	62.9	67.6	216.4
$\mathcal{T}_7$	32	2304	2446.1	1998.6	94.1	73.4	31.5	248.5
$\mathcal{T}_7$	64	4608	1913.4	1004.8	137.5	199.8	61.1	510.2
$\mathcal{T}_7$	128	9216	2755.4	923.2	414.4	204.8	289.5	923.5

Table 10: Breakdown of runtime (in seconds) into different parts, including mesh, solver and output setup, as well as the parabolic solver, which accounts for the main part of the computation.

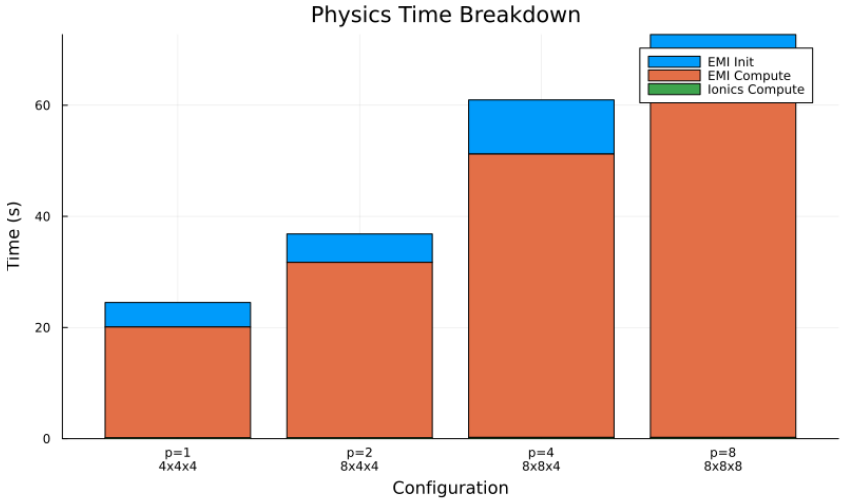


Figure 11: Decomposition of the physics-loop time for the same Ginkgo runs on the lego meshes of Table 8: EMI init, EMI compute, and Ionics compute.

5.3.3 Initialization Bottleneck and Timing Analysis

Table 10 reports a breakdown of the runtime into the main simulation phases for the strong scaling study on the $\mathcal{T}_7$ mesh from Section 5.3.1. Besides the parabolic solver, which constitutes the main compute phase, we identify three setup phases: mesh, solver and output setup. The remaining time reported in the "Other" column corresponds to portions of the setup phase that have not been identified explicitly. The reported measurements are obtained using the Caliper library [4] and correspond to a maximum runtime among all MPI processes for each phase.

In summary, these results suggest that construction of the derived EMI volume mesh, surface meshes, solver setup, and output setup become increasingly costly as the number of MPI processes increases. The "Other" column also grows with process count, indicating additional setup work that has not yet been identified and instrumented separately.

Because the input mesh is not already in the decoupled EMI representation, initialization includes substantial preprocessing before the first time step: construction of the derived EMI volume mesh, reconstruction of surface meshes and retained face links, canonical and PETSc numbering, operator construction, and output setup. The timing breakdown is therefore reported not only to characterize solver cost, but also to identify which parts of this preprocessing pipeline should be optimized in future large-scale runs.

To place this setup overhead in the context of complete simulations, Table 11 reports end-to-end wall-clock measurements for the reference meshes $\mathcal{T}_1, \ldots, \mathcal{T}_6$. For each benchmark, we record the simulated time horizon, the temporal resolution, the resulting number of time steps, and the

split between setup and time-stepping cost. This complements the phase-wise scaling results with a directly interpretable runtime reference for full EMI simulations and clarifies the regime in which the one-time initialization cost becomes negligible relative to the sustained throughput of the solver and membrane updates.

Table 11: End-to-end runtime summary for the reference EMI simulations on meshes $\mathcal{T}_1, \ldots, \mathcal{T}_6$. For each benchmark, the table reports the simulated time horizon, temporal resolution, number of time steps, setup cost, time-stepping cost, and total wall-clock time of the complete run.

mesh	#ranks	T_{final} (ms)	Δt (μs)	#steps	setup (s)	time stepping (s)	total (s)
$\mathcal{T}_2$	1152	10	5	-	232.1	2098.67	2121.12
$\mathcal{T}_3$	1152	10	5	-	386.96	3710.05	4129.97
$\mathcal{T}_4$	1152	10	5	-	542.60	3239.39	3821.94
$\mathcal{T}_5$	1152	16	5	-	747.64	4961.88	5837.59
$\mathcal{T}_6$	1152	10	5	-	951.62	4801.75	5801.62
$\mathcal{T}_7$	1152	16	5	-	1497.12	10664.71	12253.21

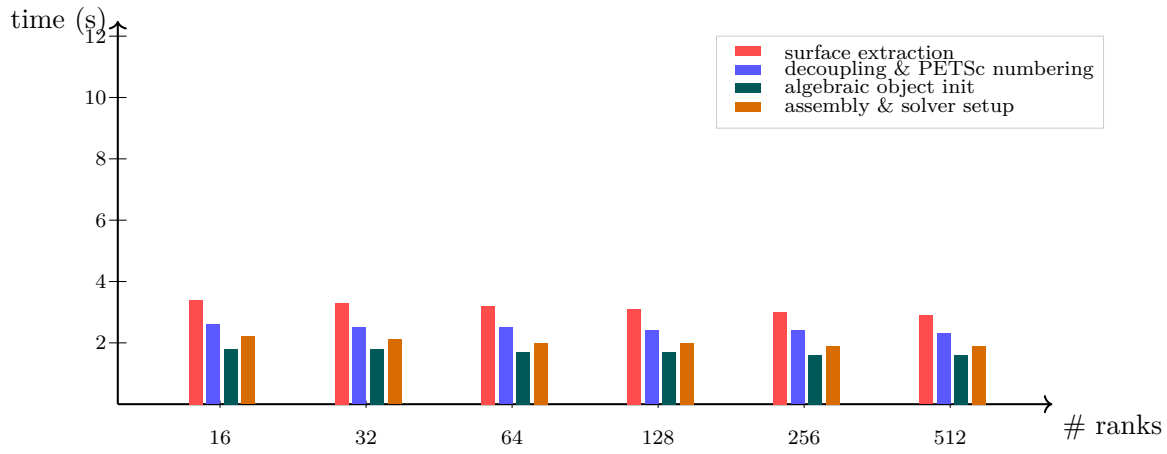


Figure 12: Template for the initialization-time breakdown on mesh $\mathcal{T}_6$ as a function of MPI ranks. The bars summarize four setup phases: surface extraction, interface decoupling with PETSc numbering, algebraic object initialization, and matrix assembly with solver setup. Current bar heights are placeholders and should be replaced by measured timings for $\mathcal{T}_6$.

6 Conclusion and Outlook

In this work, we presented a scalable distributed-memory computational framework for the Extracellular–Membrane–Intracellular (EMI) model within the openCARP ecosystem. By constructing a derived topologically decoupled EMI volume mesh and retaining the links between matching membrane faces, the framework provides a consistent pipeline from input-mesh preprocessing and interface reconstruction to operator assembly, membrane-model evaluation, canonical output, and large-scale linear system solution.

Our numerical results show that the framework achieves good performance in both weak- and strong-scaling studies for the main compute-intensive phases. The combination of domain-aware partitioning, a decoupled EMI volume mesh, and unique-face/two-sided surface representations allows EMI simulations to scale to systems with more than 10^8 degrees of freedom. Face-centered piecewise-constant membrane evaluation resolves ambiguities at complex junctions, while integration with the LIMPET library provides direct access to established ionic models for large-

scale studies. Comparisons of PETSc- and Ginkgo-based solver configurations further illustrate the flexibility of the implementation across solver backends and computational architectures.

The present work establishes the methodological and implementation basis for large-scale EMI simulations in openCARP and opens several directions for future research. One natural extension is the use of second-order finite-element discretizations together with more advanced time-integration strategies. Another important direction is the further development of preprocessing, solver, output, and partitioning strategies that reduce communication overhead and improve scalability on both CPU and GPU architectures. By providing the EMI model in a production-grade simulation environment, this work supports future cellular-resolved studies of tissue-scale conduction, heterogeneity, and arrhythmia mechanisms.

Acknowledgments

This work was supported by the MICROCARD-2 project (project ID 101172576). The project is supported by the EuroHPC Joint Undertaking and its members (including top-up funding by ANR, BMBF, and Ministero dello sviluppo economico). Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or EuroHPC. Neither the European Union nor EuroHPC can be held responsible for them.

References

[1] R. R. Aliev and A. V. Panfilov. A simple two-variable model of cardiac excitation. *Chaos, Solitons & Fractals*, 7(3):293–301, 1996.

[2] H. Anzt, T. Cojean, G. Flegar, F. Göbel, T. Grützmacher, P. Nayak, T. Ribizel, Y. M. Tsai, and E. S. Quintana-Ortí. Ginkgo: A modern linear operator algebra framework for high performance computing. *ACM Transactions on Mathematical Software*, 48(1):2:1–2:33, Feb 2022.

[3] N. Berre, M. E. Rognes, and A. Massing. Cut finite element discretizations of cell-by-cell EMI electrophysiology models. *arXiv preprint*, 2023.

[4] D. Boehme, T. Gamblin, D. Beckingsale, P.-T. Bremer, A. Gimenez, M. LeGendre, O. Pearce, and M. Schulz. Caliper: Performance introspection for HPC software stacks. In *SC '16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2016.

[5] X.-C. Cai and M. Sarkis. A restricted additive schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing*, 21(2):792–797, 1999.

[6] E. Centofanti, N. M. M. Huynh, L. F. Pavarino, and S. 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*, 442:118001, 2025.

[7] F. Chegini, A. Froehly, N. M. M. Huynh, L. F. Pavarino, M. Potse, S. Scacchi, and M. Weiser. Efficient numerical methods for simulating cardiac electrophysiology with cellular resolution. In *10th International Conference on Computational Methods for Coupled Problems in Science and Engineering*, 2023.

[8] C. Clayton. Stiffness at small strain: research and practice. *Géotechnique*, 61(1):5–37, 2011.

[9] P. Colli Franzone, L. F. Pavarino, and S. Scacchi. *Mathematical Cardiac Electrophysiology*. Springer, 2014.

[10] M. Courtemanche, R. J. Ramirez, and S. Nattel. Ionic mechanisms underlying human atrial action potential properties: insights from a mathematical model. *American Journal of Physiology – Heart and Circulatory Physiology*, 275(1):H301–H321, 1998.

[11] G. R. de Souza, R. Krause, and S. Pezzuto. Boundary integral formulation of the cell-by-cell model of cardiac electrophysiology. *Engineering Analysis with Boundary Elements*, 158:239–251, 2024.

[12] F. Göbel, N. M. M. Huynh, F. Chegini, L. F. Pavarino, M. Weiser, S. Scacchi, and H. Anzt. A BDDC preconditioner for the cardiac EMI model in three dimensions. *SIAM Journal on Scientific Computing*, 48(2):A646–A667, 2026.

[13] E. Grandi, F. S. Pasqualini, and D. M. Bers. A novel computational model of the human ventricular action potential and Ca transient. *Journal of Molecular and Cellular Cardiology*, 48(1):112–121, 2010.

[14] V. E. Henson and U. M. Yang. Boomeramg: A parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41(1):155–177, 2002.

[15] M. Houillon, J. Sánchez, M. A. F. Gsell, et al. Facilitating reproducible in silico experiments with opencarp: a step toward FAIR and open science in cardiac electrophysiology. *Frontiers in Physiology*, 2022.

[16] N. M. Huynh, F. Chegini, L. F. Pavarino, M. Weiser, and S. Scacchi. Convergence analysis of BDDC preconditioners for hybrid DG discretizations of the cardiac cell-by-cell model. *SIAM Journal on Scientific Computing*, 45(6):A2836–A2857, 2023.

[17] K. H. Jæger, K. G. Hustad, X. Cai, and A. Tveito. Efficient numerical solution of the EMI model representing the extracellular space (E), cell membrane (M) and intracellular space (I) of a collection of cardiac cells. *Frontiers in Physics*, 8:579461, 2021.

[18] K. H. Jæger and A. Tveito. Re-introducing the cell: the extracellular-membrane-intracellular (EMI) model. In *Differential Equations for Studies in Computational Electrophysiology*, pages 107–117. Springer, Cham, 2023.

[19] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.

[20] C. H. Luo and Y. Rudy. A model of the ventricular cardiac action potential. depolarization, repolarization, and their interaction. *Circulation Research*, 68(6):1501–1526, 1991.

[21] K. A. MacCannell, H. Bazzazi, L. Chilton, Y. Shibukawa, R. B. Clark, and W. R. Giles. A mathematical model of electrotonic interactions between ventricular myocytes and fibroblasts. *Biophysical Journal*, 92(11):4121–4132, 2007.

[22] R. Mills, M. Adams, S. Balay, et al. PETSc/TAO developments for early exascale systems. *arXiv preprint*, 2024.

[23] L. Mottet, C. Prigent, F. Frangiamore, et al. Challenges in meshing tissue at the micro-scale for cardiac modeling. In *Cardiac Physiome Workshop*, 2024.

[24] T. O'Hara, L. Virág, A. Varró, and Y. Rudy. Simulation of the undiseased human cardiac ventricular action potential: model formulation and experimental validation. *PLoS Computational Biology*, 7(5):e1002061, 2011.

[25] openCARP consortium et al. opencarp (v19.0). Karlsruhe Institute of Technology (KIT), 2026.

[26] F. Pellegrini and J. Roman. SCOTCH: A software package for static mapping by dual recursive bipartitioning of process and architecture graphs. In *High-Performance Computing and Networking*, volume 1067 of *Lecture Notes in Computer Science*, pages 493–498. Springer, 1996.

[27] G. Plank, A. Loewe, A. Neic, et al. The opencarp simulation environment for cardiac electrophysiology. *Computer Methods and Programs in Biomedicine*, 208:106223, 2021.

[28] R. Plonsey and R. C. Barr. *Bioelectricity: A Quantitative Approach*. Springer Science & Business Media, 2007.

[29] M. Potse, L. Cirrottola, and A. Froehly. A practical algorithm to build geometric models of cardiac muscle structure. In *ECCOMAS 2022 – The 8th European Congress on Computational Methods in Applied Sciences and Engineering*, Oslo, Norway, June 2022. https://hal.science/hal-03936963.

[30] J. W. Ruge and K. Stüben. Algebraic multigrid. In *Multigrid Methods*, pages 73–130. SIAM, 1987.

[31] J. Steyer, M. Potse, and G. Lourenço. A cell-resolved cardiac tissue mesh for the opencarp EMI example generated with anni, 2026.

[32] J. Tomek, A. Bueno-Orovio, E. Passini, X. Zhou, A. Mincholé, O. Britton, C. Bartolucci, S. Severi, A. Shrier, L. Virág, A. Varró, and B. Rodriguez. Development, calibration, and validation of a novel human ventricular myocyte model in health, disease, and drug block. *eLife*, 8:e48890, 2019.

[33] A. Tveito, K. H. Jæger, M. Kuchta, K.-A. Mardal, and M. E. Rognes. A cell-based framework for numerical modeling of electrical conduction in cardiac tissue. *Frontiers in Physics*, 5:48, 2017.

[34] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56:179–196, 1996.

[35] E. J. Vigmond, M. Hughes, G. Plank, and L. J. Leon. Computational tools for modeling electrical activity in cardiac tissue. *Journal of electrocardiology*, 36:69–74, 2003.

[36] M. Weiser and F. Chegini. Adaptive multirate integration of cardiac electrophysiology with spectral deferred correction methods. In *CMBE 2022: 7th International Conference on Computational and Mathematical Biomedical Engineering*, pages 528–531, 2022.