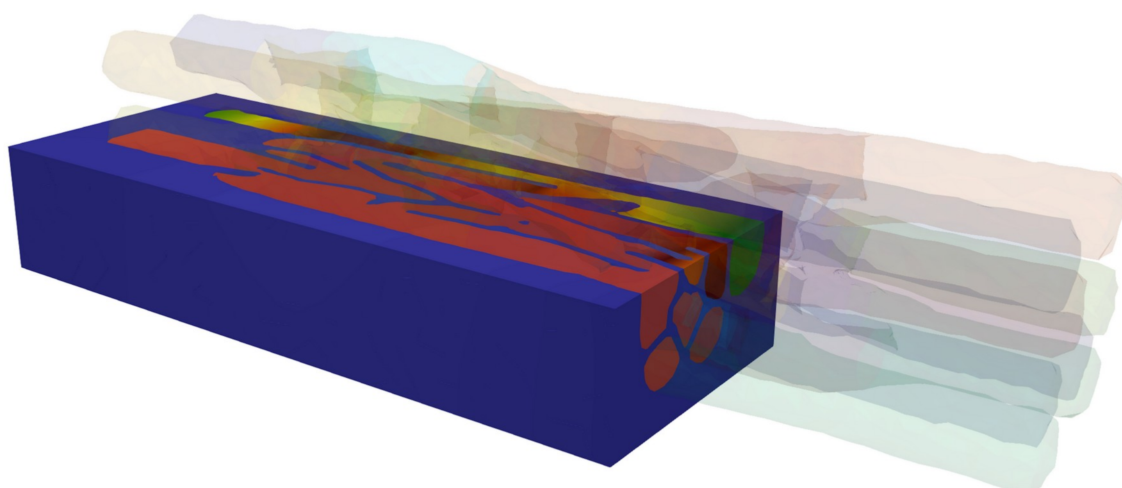
Adaptive higher-order time integration and BDDC preconditioning; EMI in openCARP

F. Chegini, M. Weiser



MICROCARD

04 .07.2023

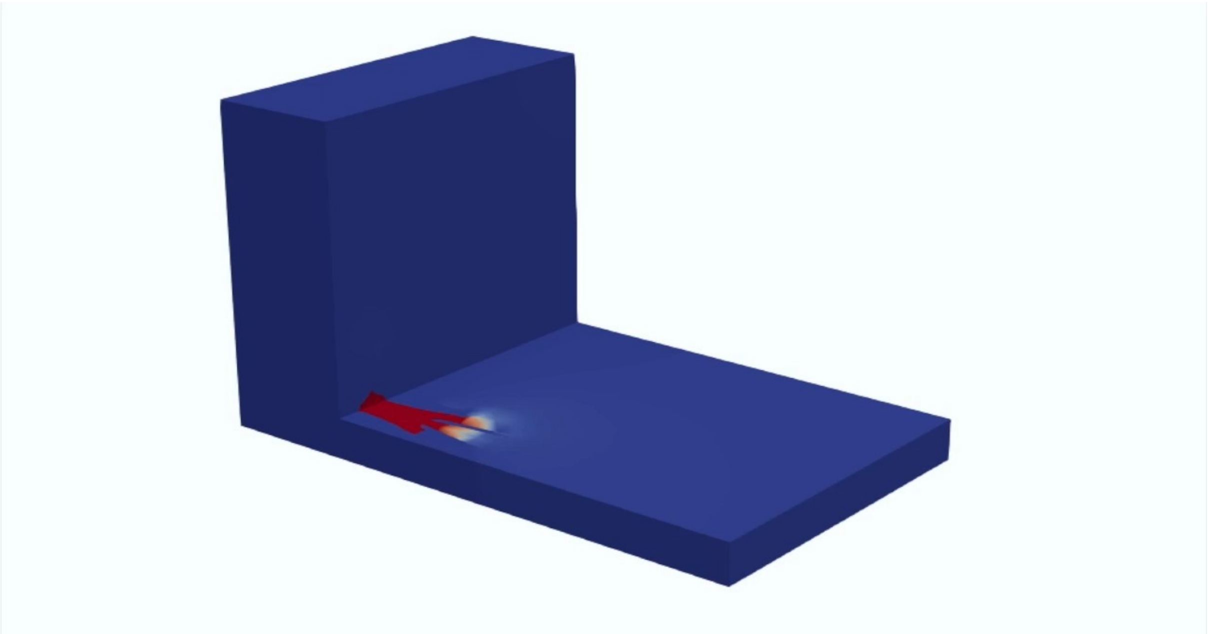
SPONSORED BY THE
Federal Ministry of Education and Research

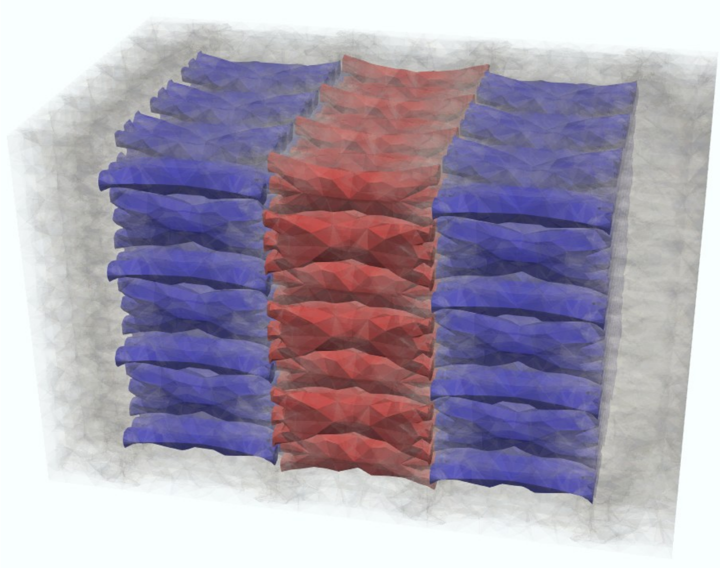
EuroHPC Joint Undertaking

The project leading to this application has received funding from the European High-Performance Computing Joint Undertaking Joint Undertaking (JU) under grant agreement No 955495. The JU receives support from the European Union's Horizon 2020 research and innovation programme and France, Italy, Germany, Austria, Norway, Switzerland

Outline

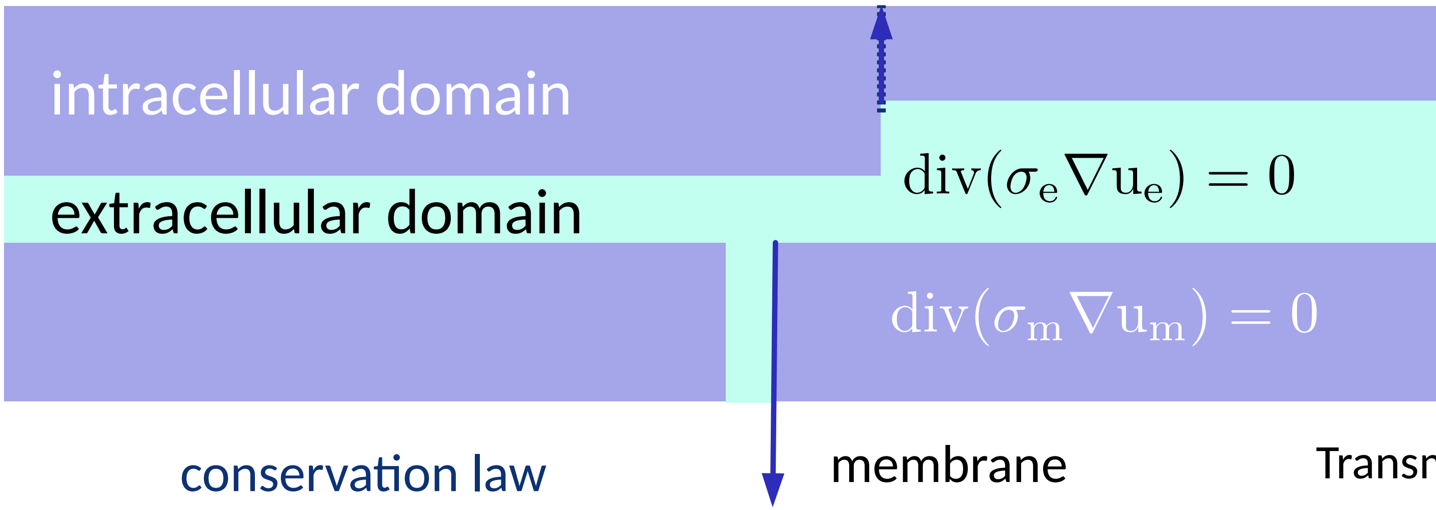
- EMI formulation
- An efficient higher order time integration method
- Sub-matrices for BDDC preconditioner
- EMI implementation in openCarp





EMI model

gap junction $$-\sigma_{\mathrm{m}} \nabla u_{\mathrm{m}} \cdot n = \sigma_{\mathrm{m}} \nabla u_{\mathrm{m}} \cdot n$$



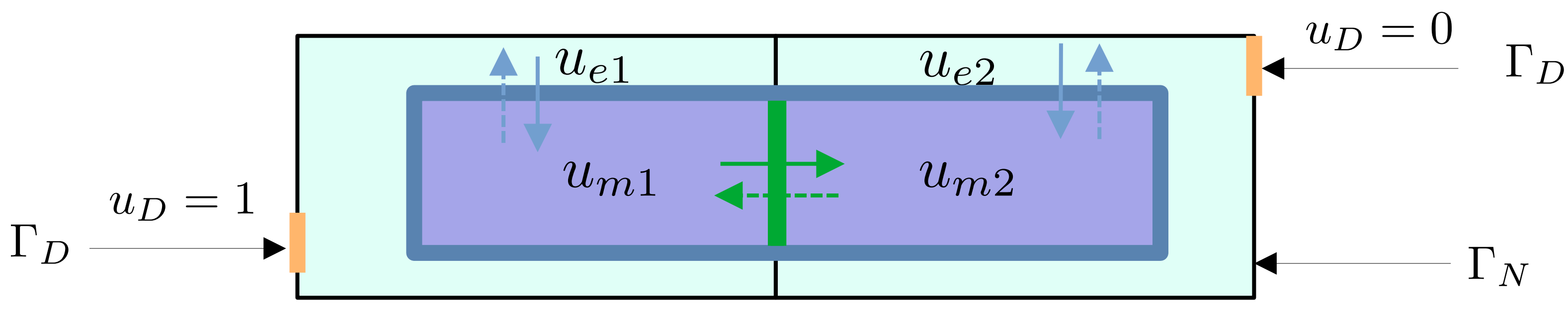
conservation law

$$-\sigma_{\mathrm{m}} \nabla u_{\mathrm{m}} \cdot n = \sigma_{\mathrm{e}} \nabla u_{\mathrm{e}} \cdot n$$
$$= c_m \partial v + I^{\mathrm{ion}}(v, w)$$

Transmembrane voltage

$$v = u_{\mathrm{m}} - u_{\mathrm{e}}$$

EMI model

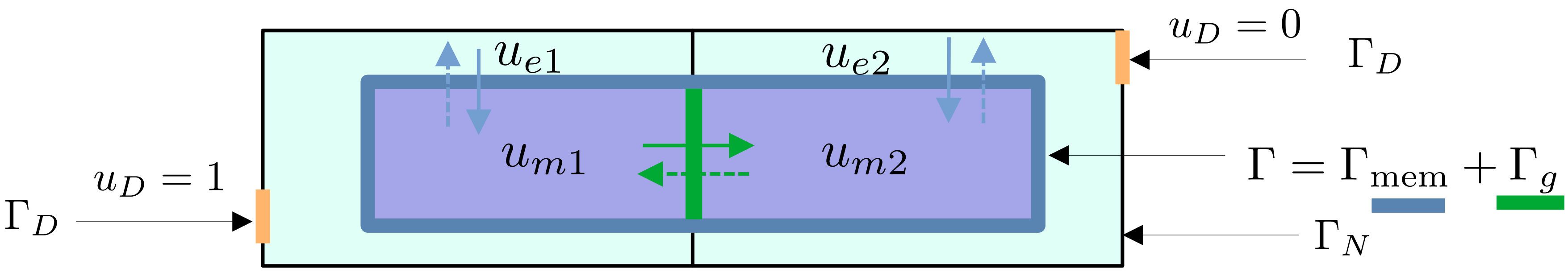


$$\Gamma_N + \Gamma_D = \Gamma_e$$

$$\sigma_{\mathrm{e}} \nabla u_e \cdot n_e + \gamma(u_e - u_D) = 0 \quad \text{on} \quad \Gamma_e$$

$$\gamma = \begin{cases} \gg 1 & \text{on} \quad \Gamma_D \\ 0 & \text{on} \quad \Gamma_N \end{cases}$$

EMI model



$$u_e \in H^1(\Omega_e), \qquad \text{where} \quad e = 1, \cdots, s_e$$

$$u_m \in H^1(\Omega_m), \qquad \text{where} \quad m = 1, \cdots, s_m$$

Number of subdomain in extra or myocytes

$$\mathrm{div}(\sigma_e \nabla u_e) = 0 \quad \text{in} \quad \Omega_e$$

$$\mathrm{div}(\sigma_m \nabla u_m) = 0 \quad \text{in} \quad \Omega_m$$

$$-\sigma_m \nabla u_m \cdot n = \sigma_e \nabla u_e \cdot n = c_m \partial v + I^{\text{ion}}(v, w)$$

$$\partial_t w = f(v, w)$$

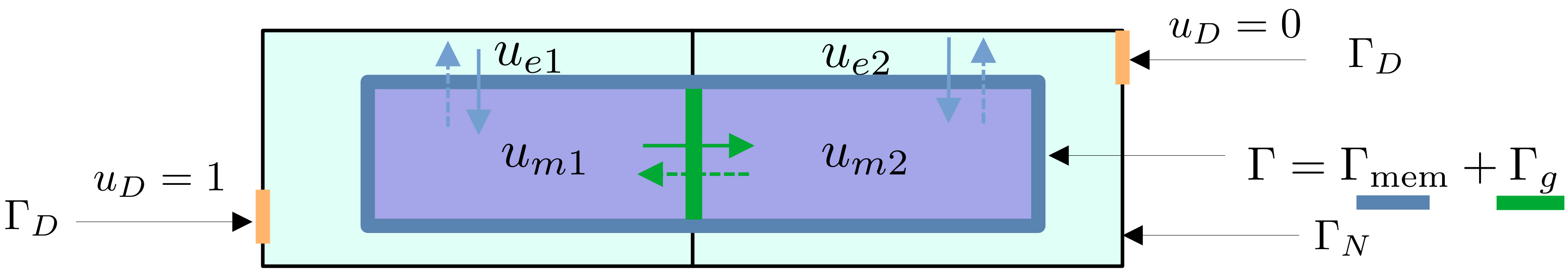
$$v = u_m - u_e$$

$$\Gamma_N + \Gamma_D = \Gamma_e$$

$$\sigma_e \nabla u_e \cdot n_e + \gamma(u_e - u_D) = 0 \quad \text{on} \quad \Gamma_e$$

$$\gamma = \begin{cases} \gg 1 & \text{on} \quad \Gamma_D \\ 0 & \text{on} \quad \Gamma_N \end{cases}$$

EMI model

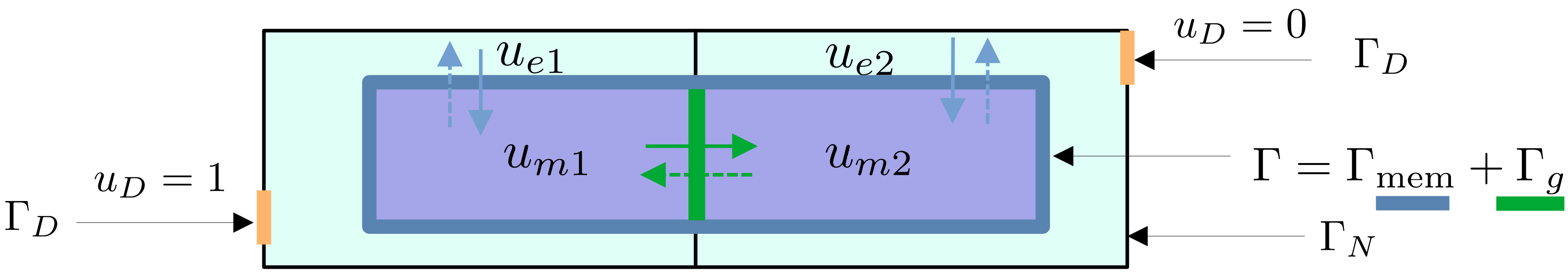


Variational Formulation

$$\text{on}\quad \Gamma_{ij} = \overline{\Omega_i} \cap \overline{\Omega_j} \quad \begin{cases} v_{ij} = u_i - u_j = -v_{ji} \\ n_i^T \sigma_{\text{i}} \nabla u_i = -I_{ij}(v_{ij}) \quad \text{incoming current density into} \quad \Omega_i \\ I_{ij} = C\dot{v}_{ij} + I_{ij}^{\text{ion}}(v_{ij}, w) \end{cases}$$

$$I_{ij} = \begin{cases} C\dot{v}_{ij} + I_{ij}^{\text{m}}(v_{ij}, w) & \text{on} \quad \Gamma_{\text{mem}} \\ C\dot{v}_{ij} + I_{ij}^{\text{g}}(v_{ij}, 0) & \text{on} \quad \Gamma_g \end{cases}$$

EMI Discretization



Variational Formulation

$$\int_\Omega \nabla \cdot \sigma_\mathrm{i} \nabla u_i \phi_i dx = 0$$

$$0 = -\int_{\Omega_i} \nabla \phi_i^T \sigma_\mathrm{i} \nabla u_i \, dx + \int_{\partial\Omega_i} n_i^T \sigma_\mathrm{i} \nabla u_i \phi_i \, ds$$

$$= -\underbrace{\int_{\Omega_i} \nabla \phi_i^T \sigma_\mathrm{i} \nabla u_i \, dx}_{} + \sum_{j \neq i} \int_{\Gamma_{ij}} -(C\dot{v}_{ij} + I_{ij}^{ion}(v_{ij}))\phi_i \, ds.$$

on $\Gamma_{ij} = \overline{\Omega_i} \cap \overline{\Omega_j}$

[Bécue, Potse, Coudière 2018, Jæger, Tveito 2021]

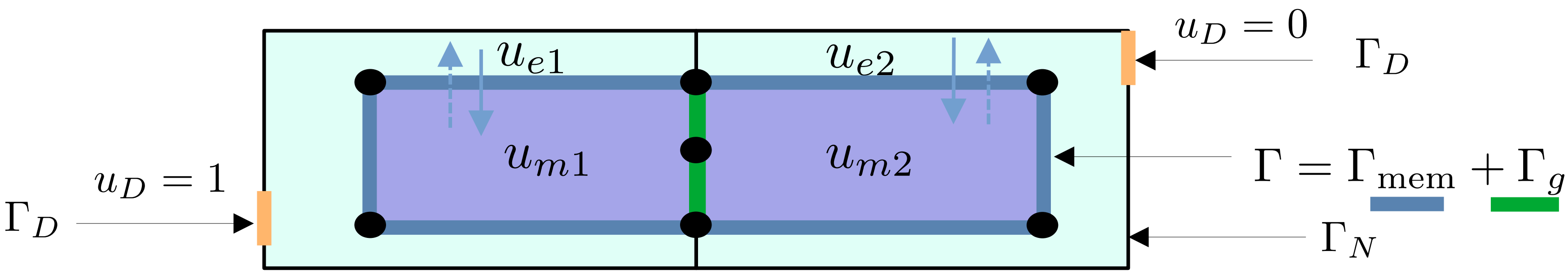
$v_{ij} = u_i - u_j$

In total

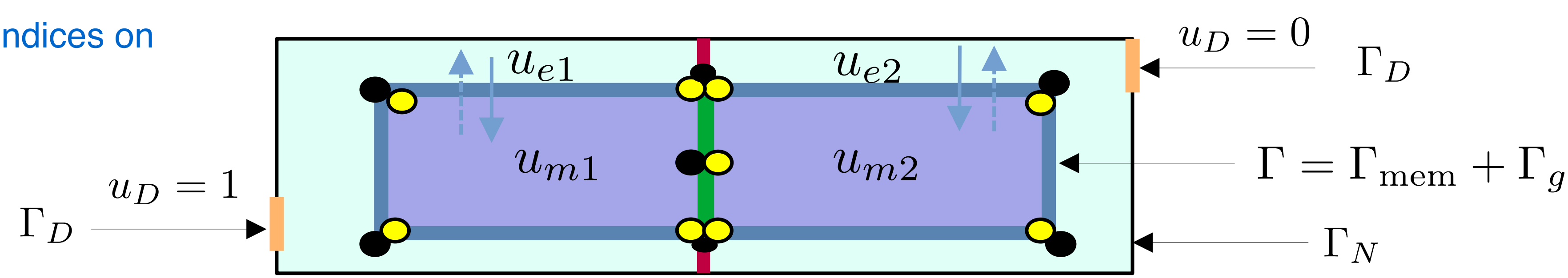
$$\int_0^T \left(\int_{\Omega_e} \sigma_\mathrm{e} \nabla u_e \cdot \nabla \phi_e dx + \int_{\Omega_m} \sigma_\mathrm{m} \nabla u_m \cdot \nabla \phi_m dx + \int_{\Gamma_{ij(i\neq j)}} (C\dot{v}_{ij} + I_{ij}^{\text{ion}}(v_{ij}, w_{ij}))\varphi dx \right) dt = 0$$

Discontinuity on the interfaces

original

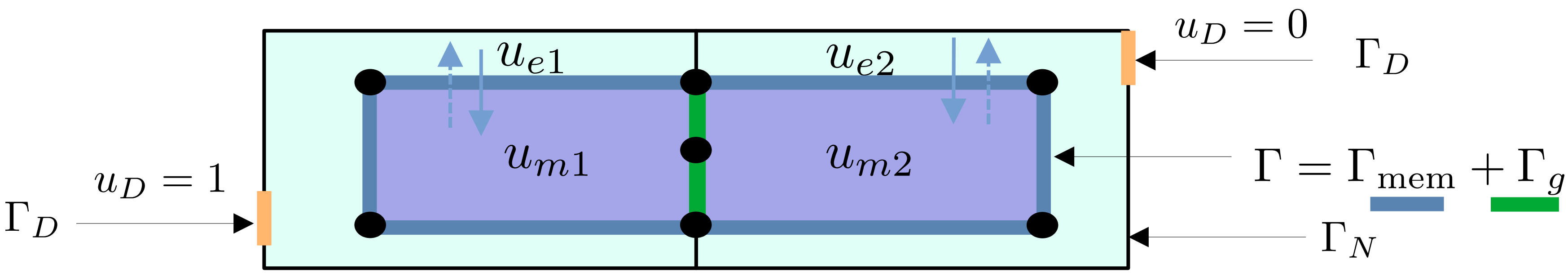


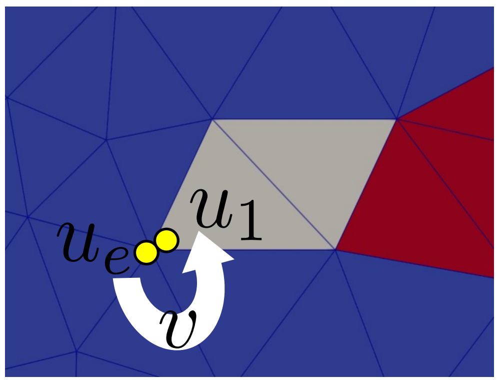
define dof indices on interfaces

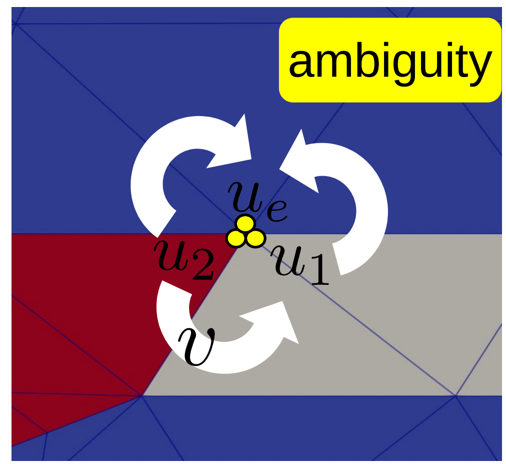


Discontinuity on the interfaces

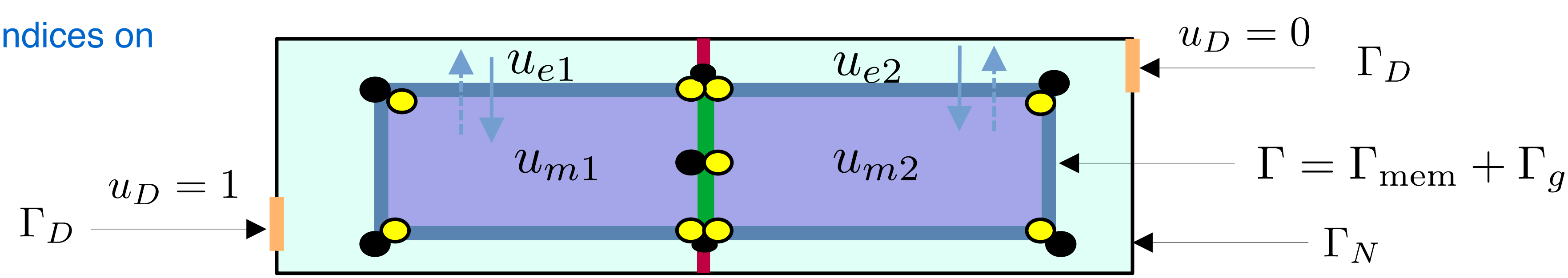
original





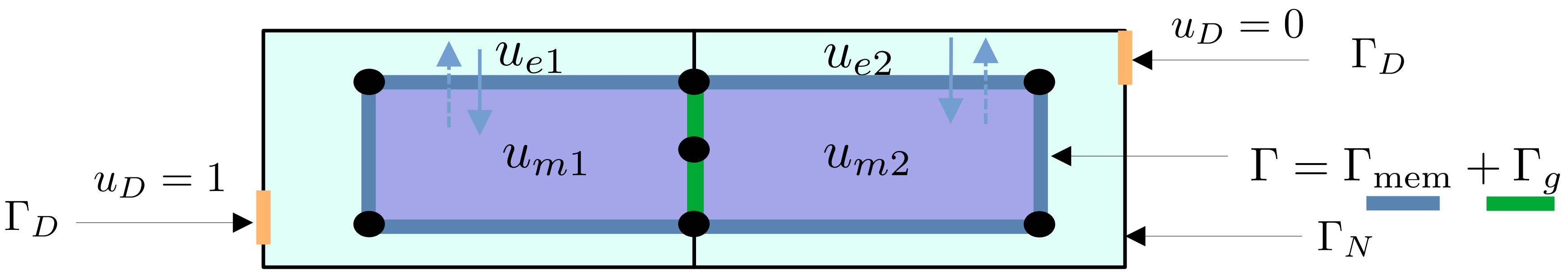


define dof indices on interfaces

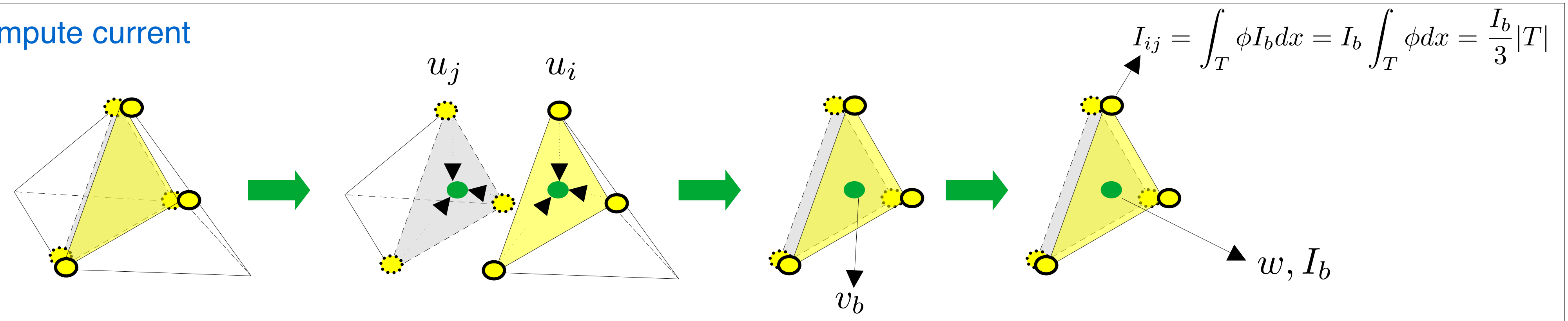


Current computation

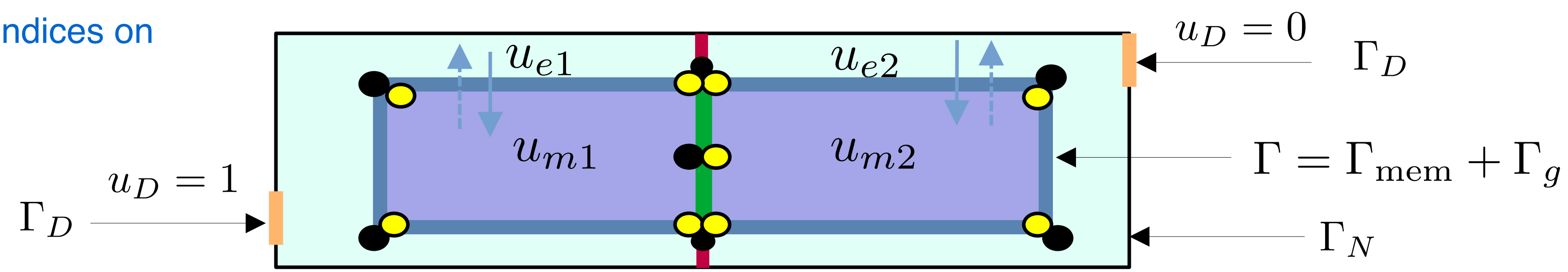
original



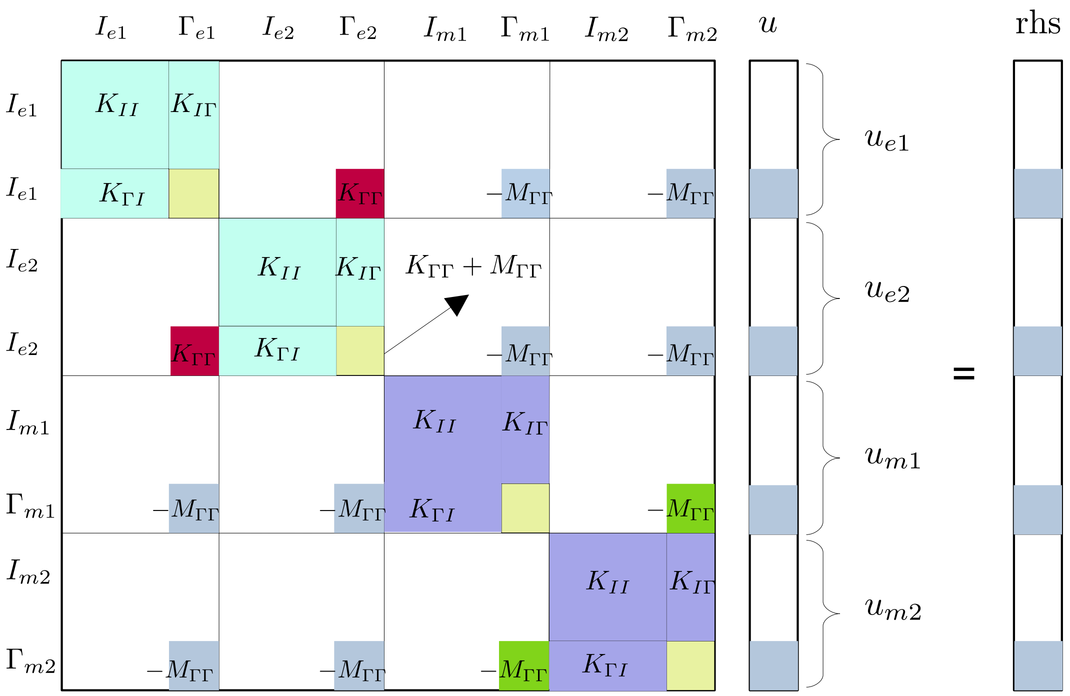
Compute current

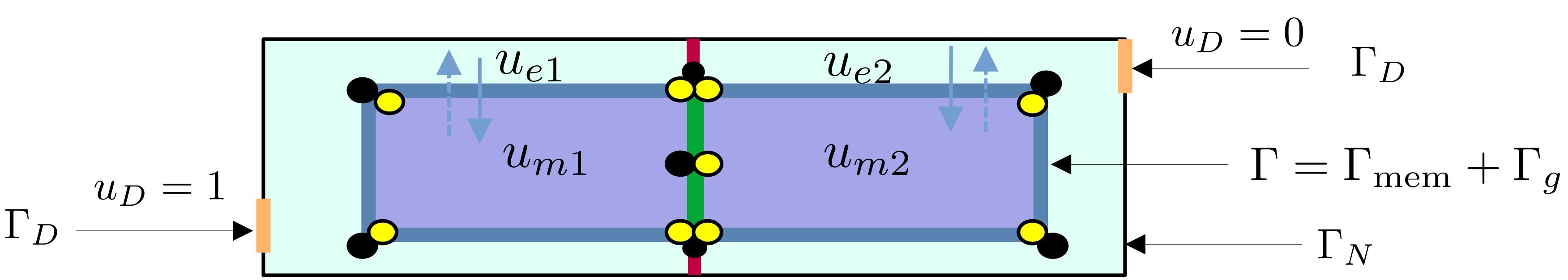


define dof indices on interfaces

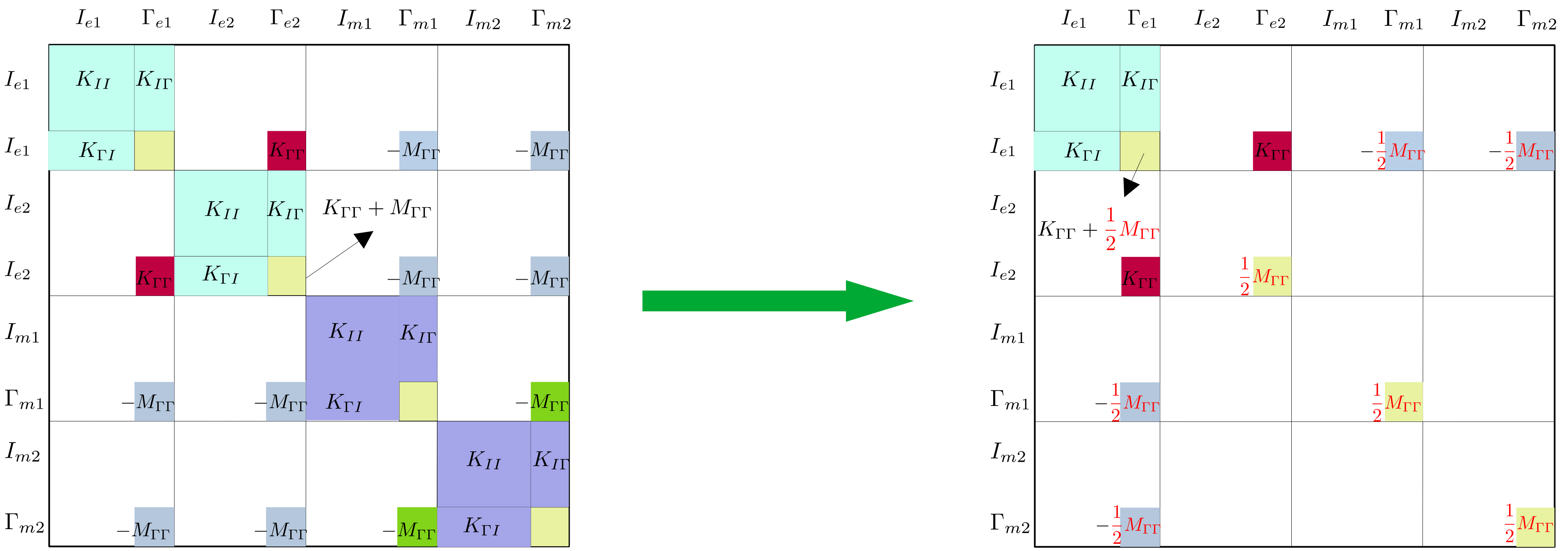


LHS Matrix & RHS

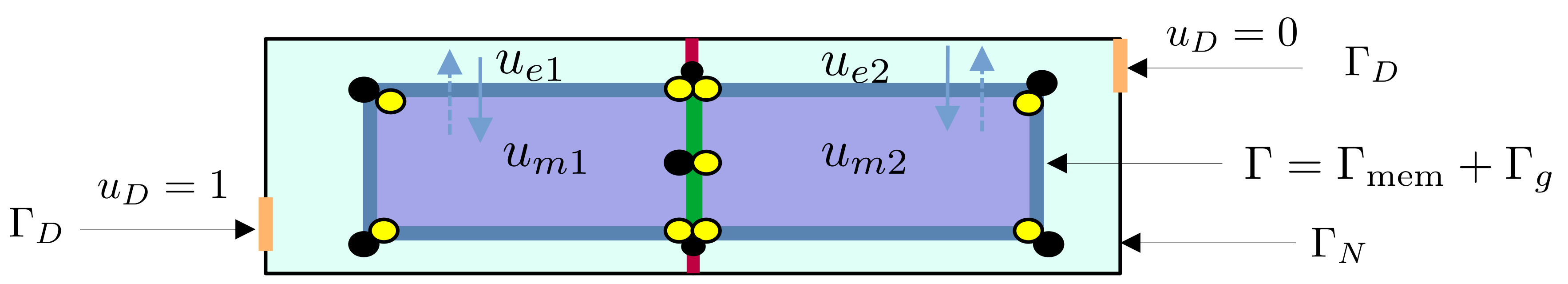




LHS Matrix & RHS



submatrix



EMI model

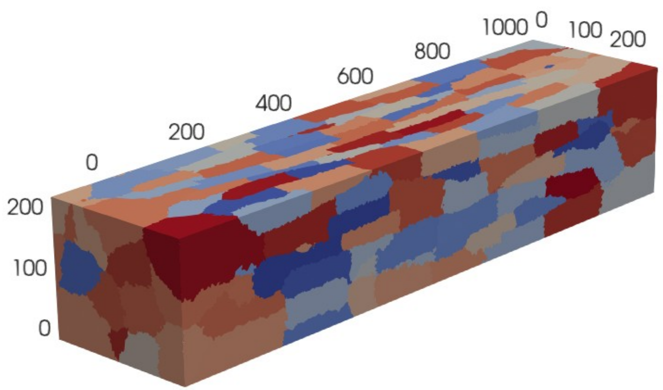
K_e K_m

Ionic model ★ Gap Junction ★

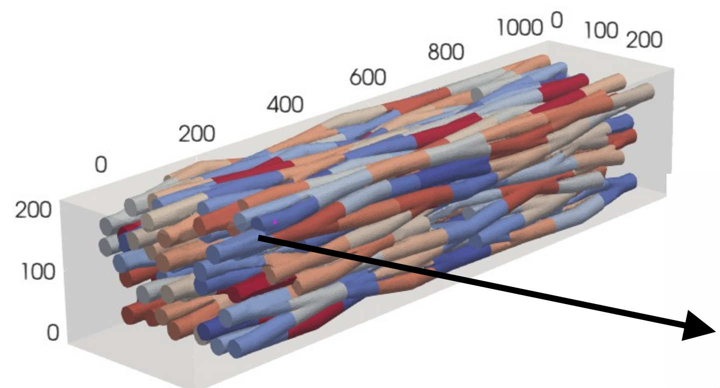
$$\int_0^T \left(\underbrace{\int_{\Omega_e} \sigma_\mathrm{e} \nabla u_e \cdot \nabla \phi_e dx}_{K_e} + \underbrace{\int_{\Omega_m} \sigma_\mathrm{m} \nabla u_m \cdot \nabla \phi_m dx}_{K_m} + \int_{\Gamma_{ij(i\neq j)}} (C\dot{v}_{ij} + I_{ij}^{\mathrm{ion}}(v_{ij}, w_{ij}))\varphi dx \right) dt = 0$$

Integration over elements

Integration over faces

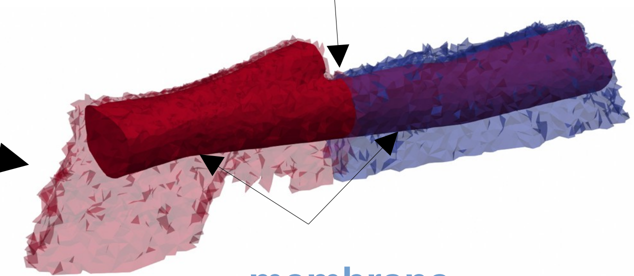


mesh



myocytes

$$I(v)$$



$$I(v, w)$$

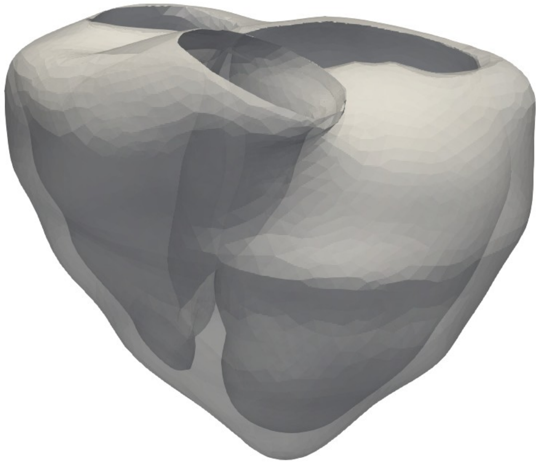
$$\dot{w} = f(v, w)$$

[Mark Potse, et al. Mmg]

ZUSE INSTITUTE BERLIN

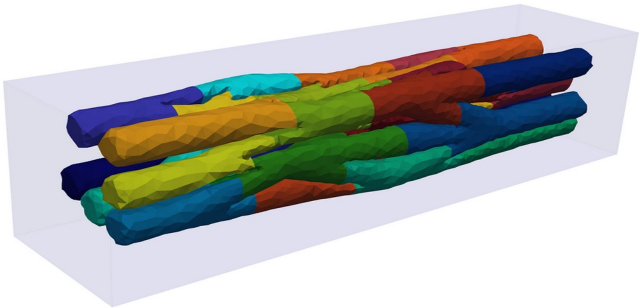
Efficient adaptivity for simulating cardiac electrophysiology with spectral deferred correction methods

Electrophysiological models



Bidomain model

mesh width:	~100µm	heart diameter:	10cm
time step:	~100µs	heart beat:	1s



EMI model

mesh width:	<10µm	heart diameter:	10cm
time step:	~10µs	heart beat:	1s

⇩

factor 10,000
larger than bidomain

Efficient adaptivity approach

Discretization

space: finite elements, finite differences, finite volumes
time: IMEX Euler/RK, Rush-Larsen

Challenge

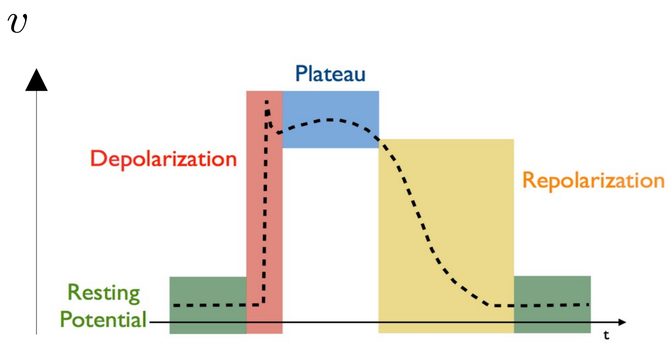
- spatially local structures → fine grids
- fast dynamics → small time steps

⇨ huge computational effort

Approach

- massive parallelization
- space-time discretization
- model selection & coupling
- mesh & time step adaptivity
- Error estimation, frequent mesh refinement & coarsening

⇨ Classical adaptivity fails to provide high efficiency

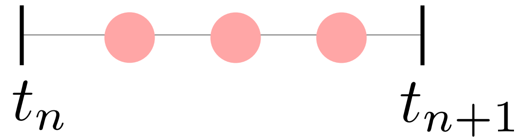


Wish list

- Higher convergence order
- Adaptive coarsening to the algebraic level
  - Avoid mesh modification & reassembly
  - Reduced the computational time

Spectral Deferred Correction Methods

Problem $$M\dot{z} = L(z)$$



time grid $$0 = \tau_0 < \tau_1 < \cdots < \tau_m = \tau$$

e.g., Radau, Gauss, Lobatto

approximate rhs $$\tilde{z} \in \mathbb{P}_m, \; z_i = z(\tau_i)$$

Integrate Picard equation
$$M(z_{i+1} - z_i) = \int_{\tau=\tau_i}^{\tau_{i+1}} L(z)\, d\tau, \quad i = 0, \dots, m-1$$
$$\approx \sum_{j=0}^{m} S_{ij} L(z_j).$$

Fixed point iteration
$$M(\delta z_{i+1}^k - \delta z_i^k) - \sum_{j=0}^{n} S_{ij} L'(z_j^k)\delta z_j^k$$

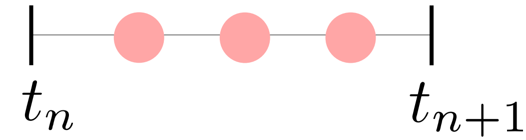
Collocation points $\tau_0, \dots, \tau_{m-1}$
$$= -M(z_{i+1}^k - z_i^k) + \underbrace{\sum_{j=0}^{n} S_{ij} L(z_j^k)}$$

high-order quadrature on a collocation time grid

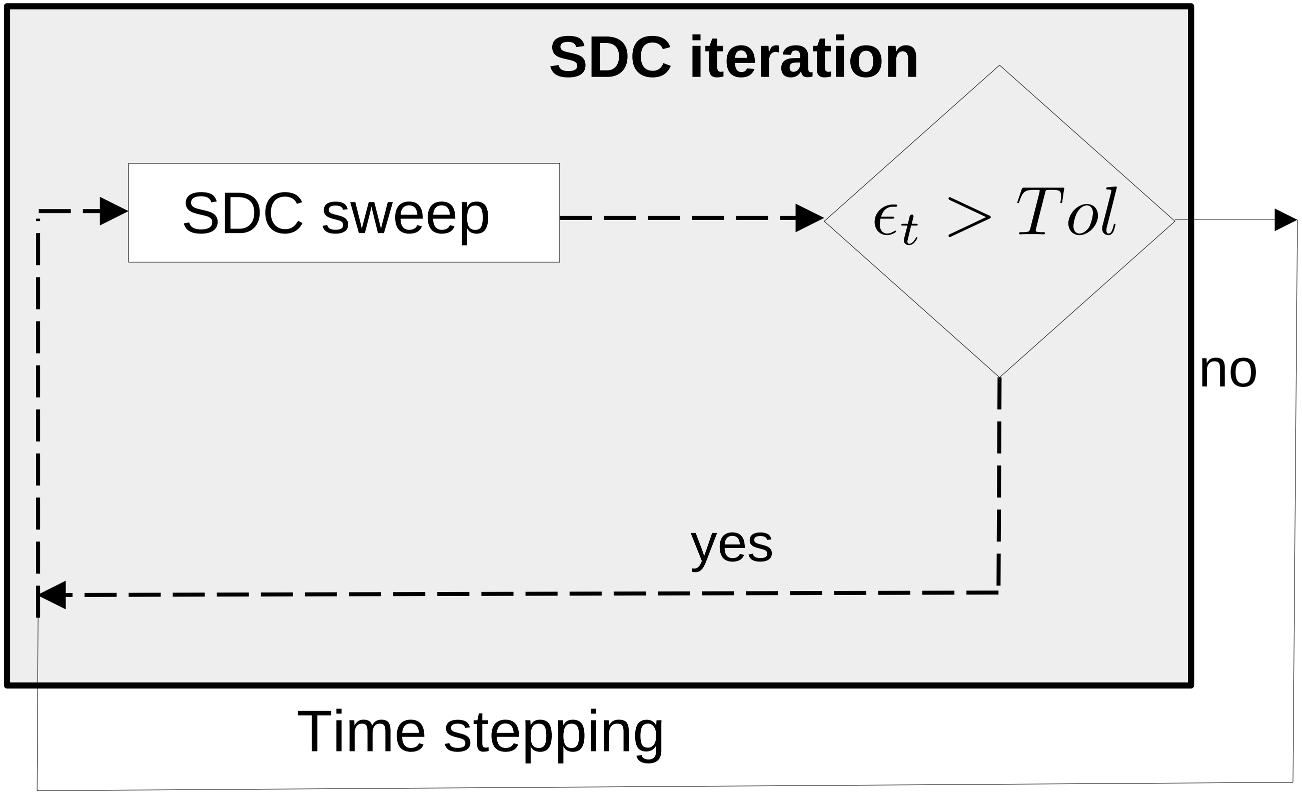
$$z^{k+1} = z^k + \delta z^k$$

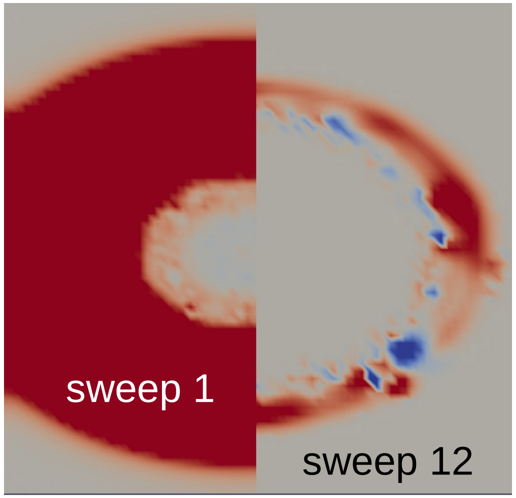
[Dutt, Greengard, Rokhlin 2000, Auzinger et al. 2003,Layton, Minion 2007, Weiser 2015, Speck, Ruprecht, Minion, Emmett, Krause 2016]

SDC iteration



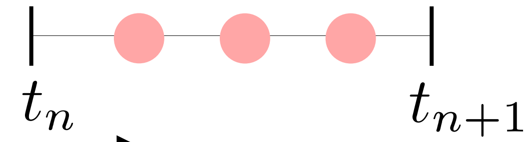
e.g., **Radau,** Gauss, Lobatto

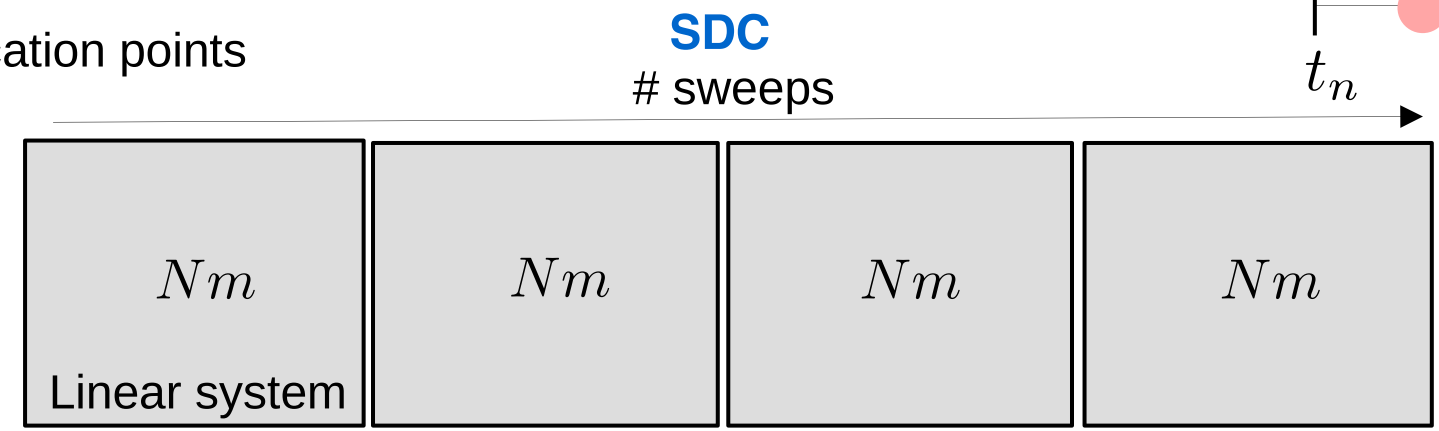




SDC Correction Structure

m Collocation points



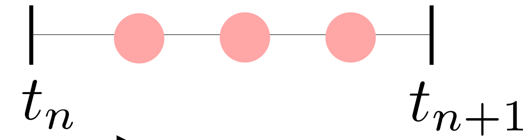


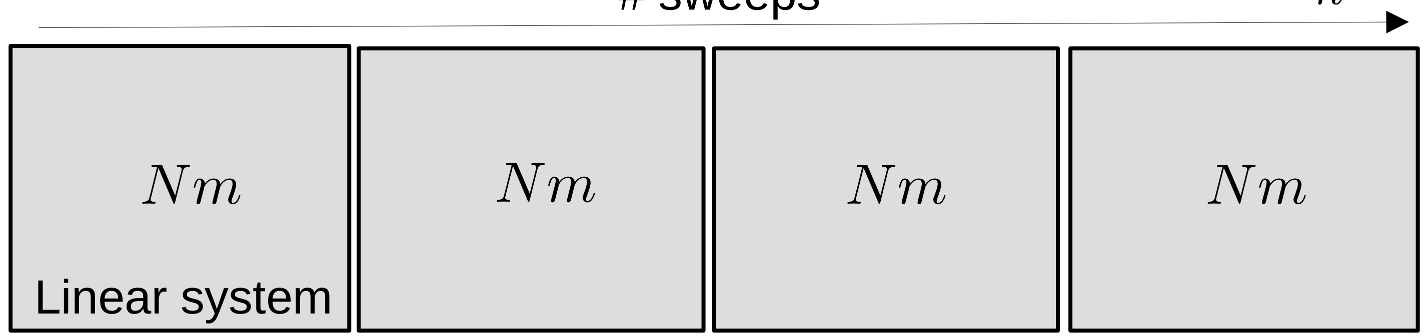
SDC Correction Structure

m Collocation points

SDC

\# sweeps





SDC + Algebraic adaptivity

Extracted Nested sub-matrices

Algebraic adaptivity

Higher order with
degree of freedom subset selection

Realization

- cheap submatrix extraction
- nested approximation spaces

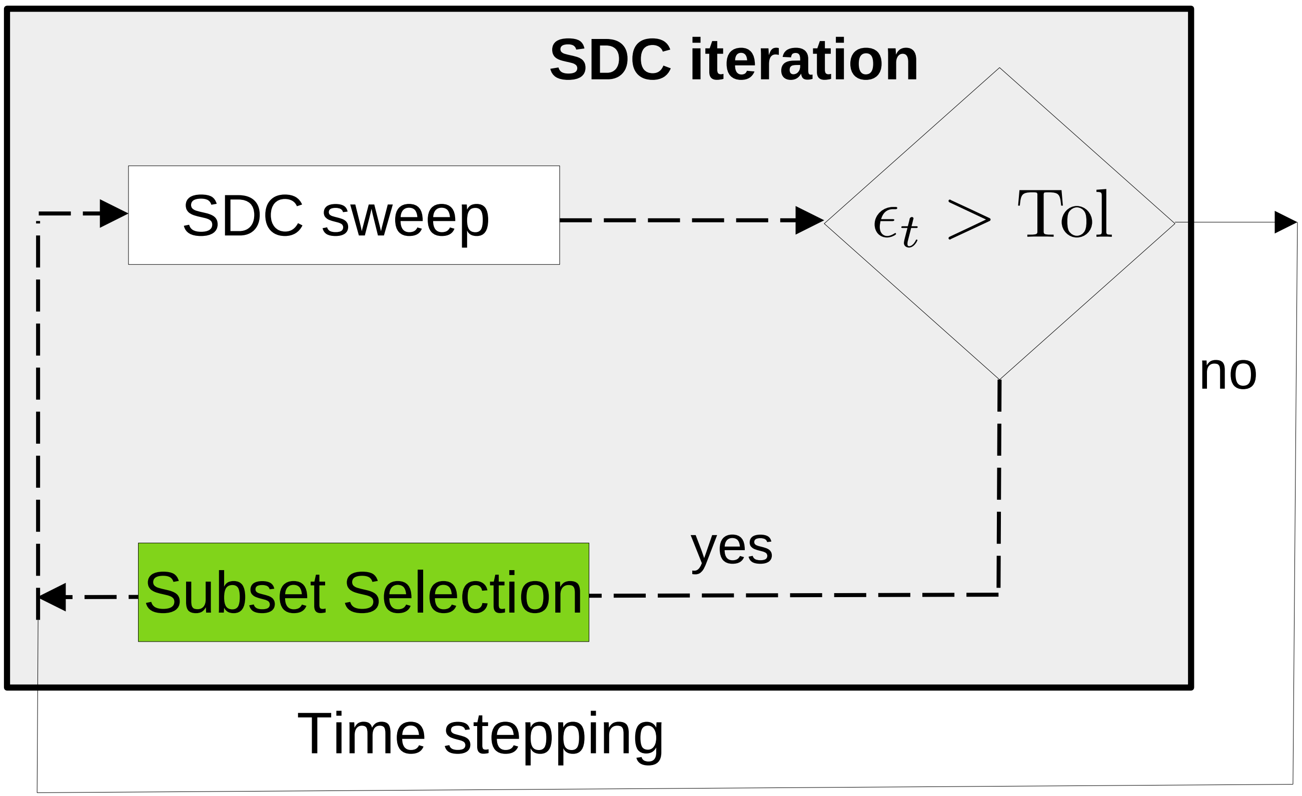
+ BDDC preconditioner

Applied nested sub-matrices for preconditioner

Multirate integration SDC

algebraic DOF subset selection during SDC sweeps

Spatial multirate: algebraic adaptivity



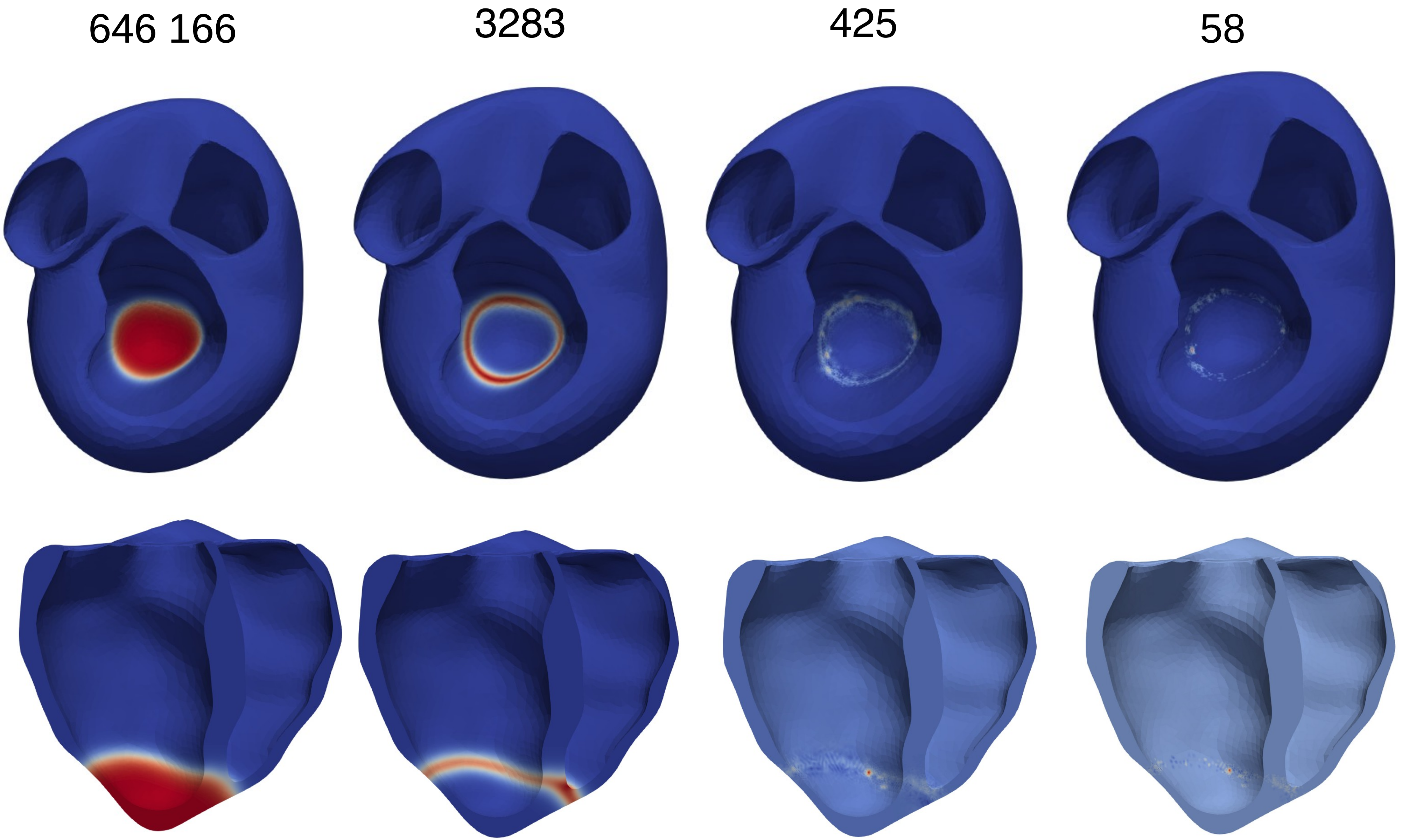
limit SDC improvements
to spatial regions
with significant dynamics
by selecting DoFs

impose homogeneous Dirichlet
b.c. for SDC corrections

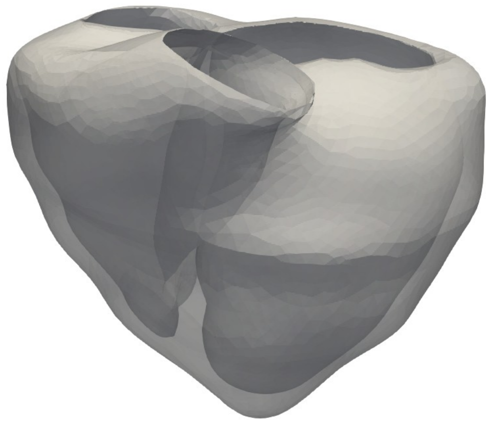
$$\tilde{\Omega}_k = \{x \in \Omega \mid |\delta u^k(x)| \geq T_{\mathrm{drop}}\}$$

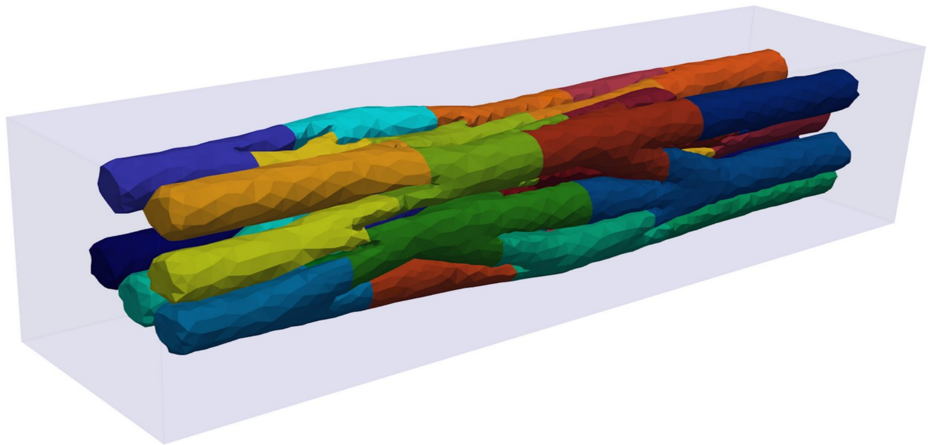
drop node if error estimate below drop tolerance

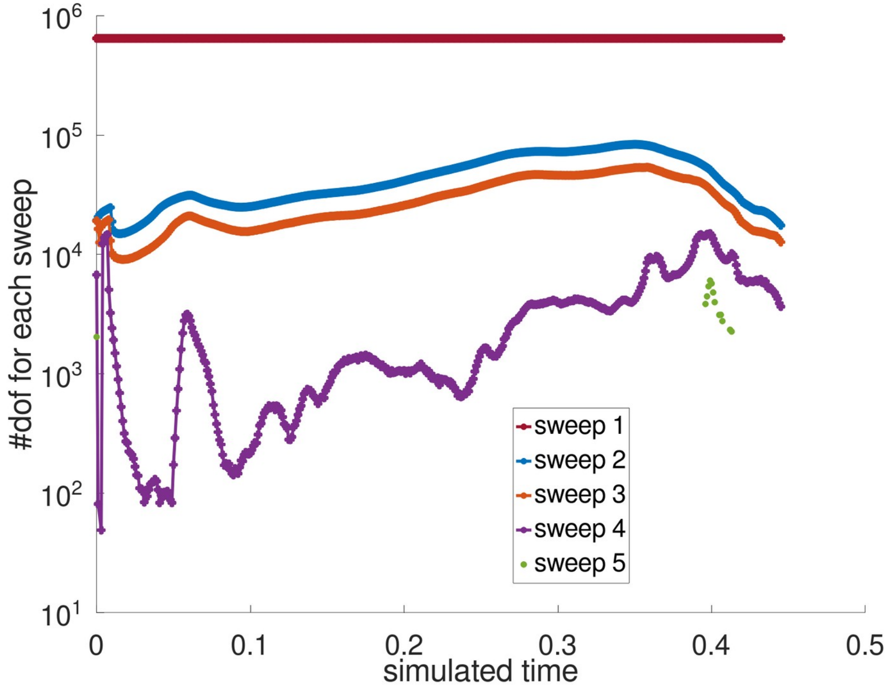
SDC update with Algebraic Adaptivity
monodomain model

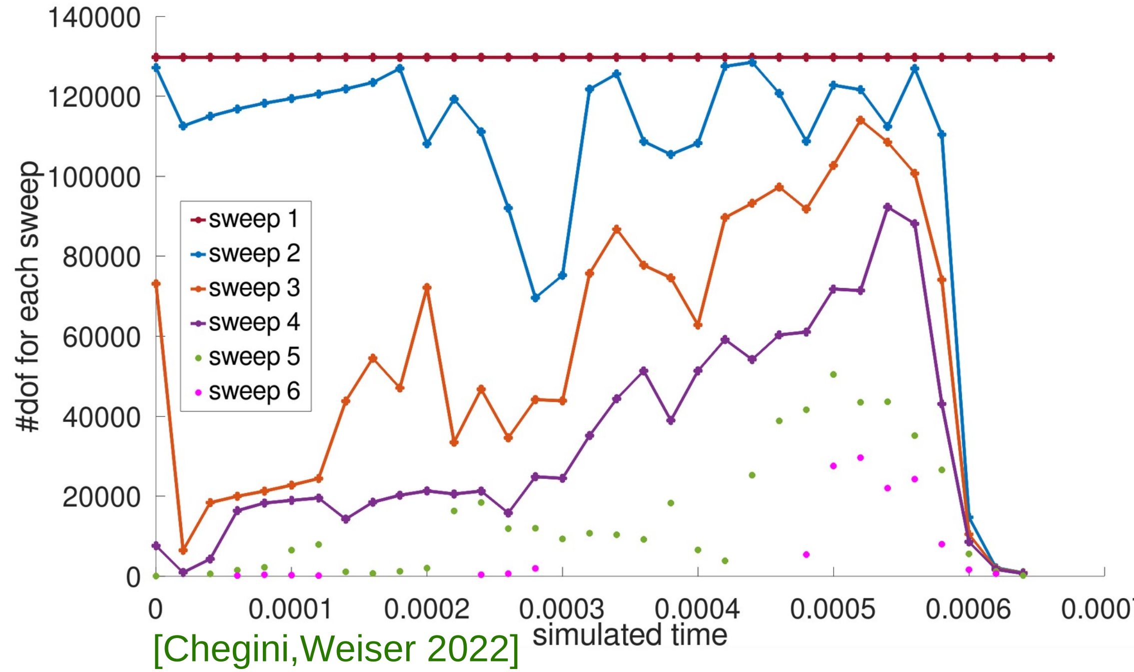


Number of active dofs in each sweep
Monodomain model & EMI model



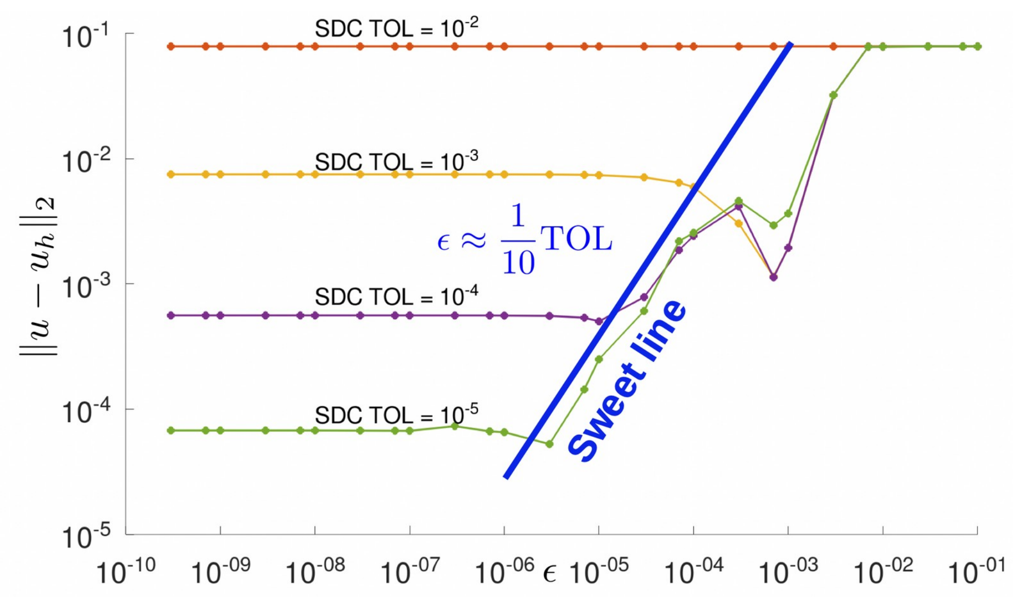




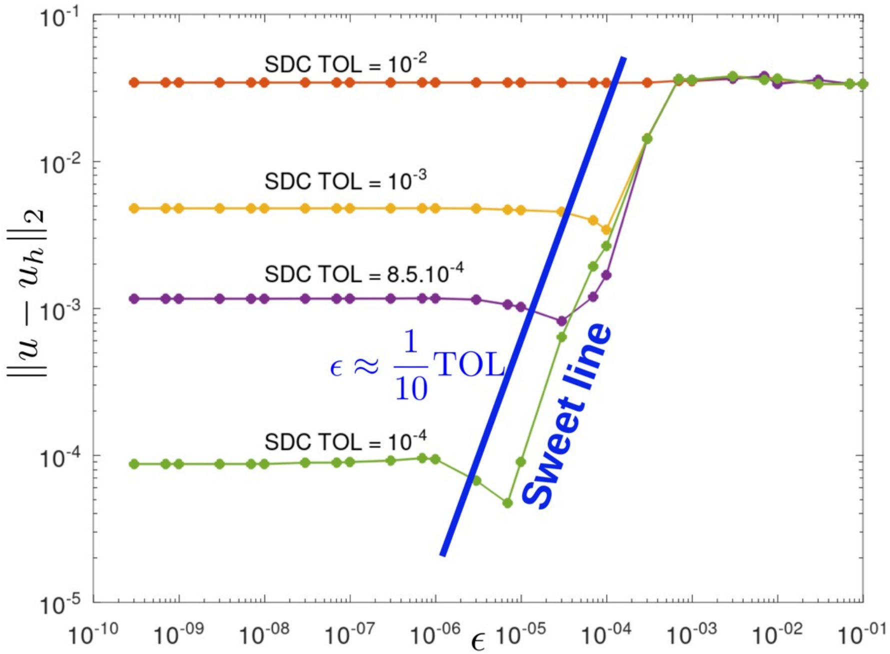


[Chegini,Weiser 2022]

Simulation error



Monodomain model



EMI model

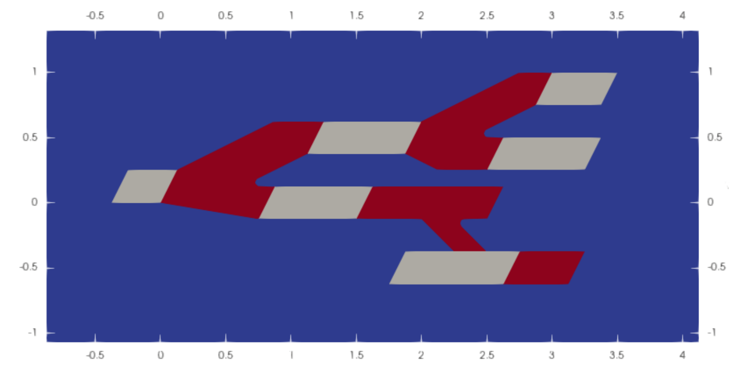
Computational speedup

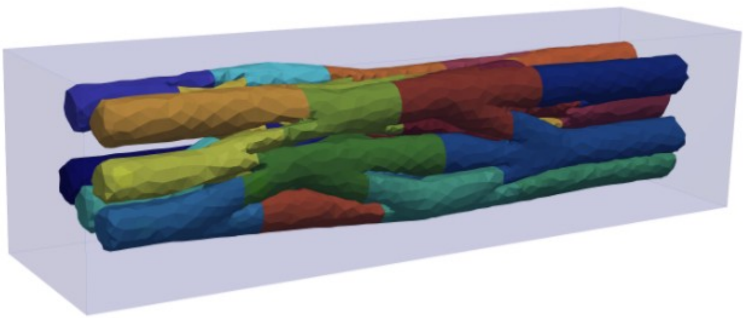
	monodomain	EMI
2D	2.13	3.336
3D	3.29	4.344

Using SDC w/o algebraic adaptivity

[Chegini,Steinke,Weiser 2022]

EMI domain





Conclusions

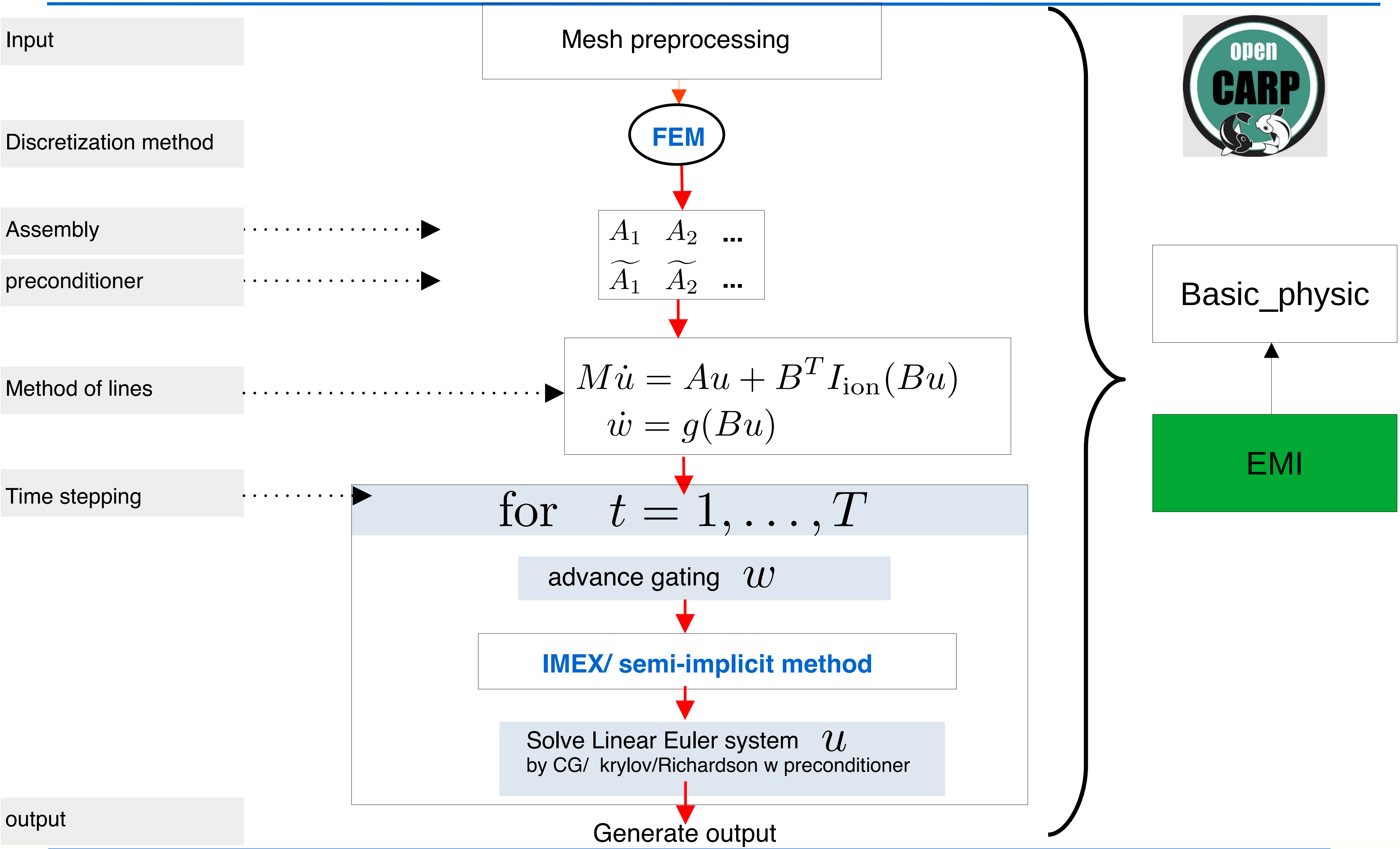
- a great match – multirate integration and SDC methods play exceedingly well together

- multirate SDC is a cheap and simple way to do adaptive cardiac simulations
  - cheap submatrix extraction

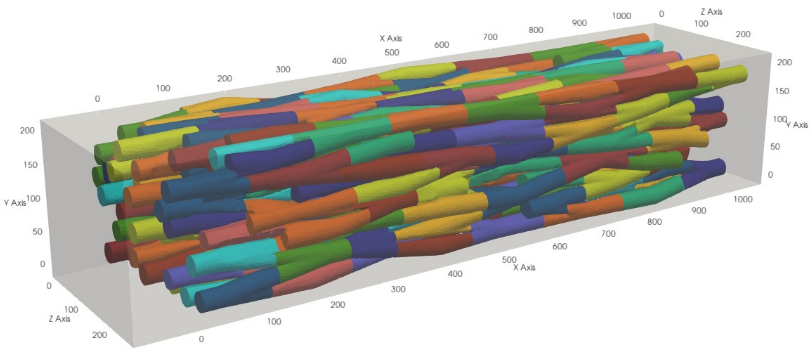
Notes:

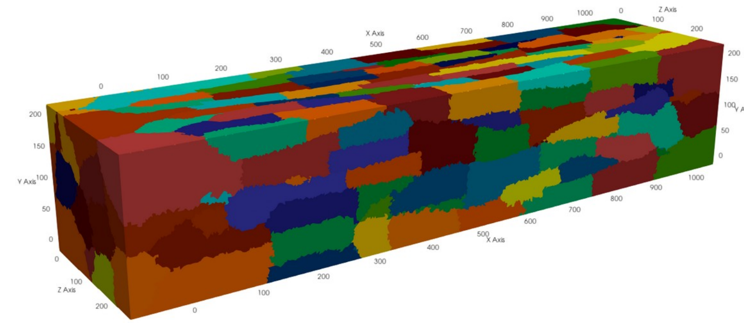
- adaptive selection of drop tolerance
- cheap restriction of preconditioners → preconditioners/factorizations need to be recomputed
  - a conjugate gradient method with block Jacobi preconditioner
  - Applied nested sub-matrices for BDDC precondioner
- load imbalance in distributed simulations of large scale problems
- reducing the energy consumption of large scale simulations.

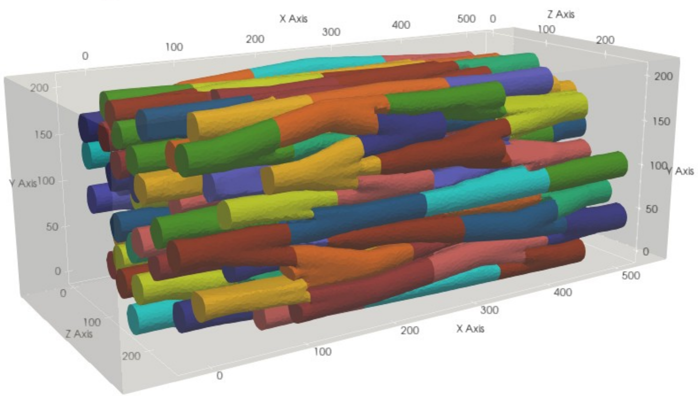
Top level steps

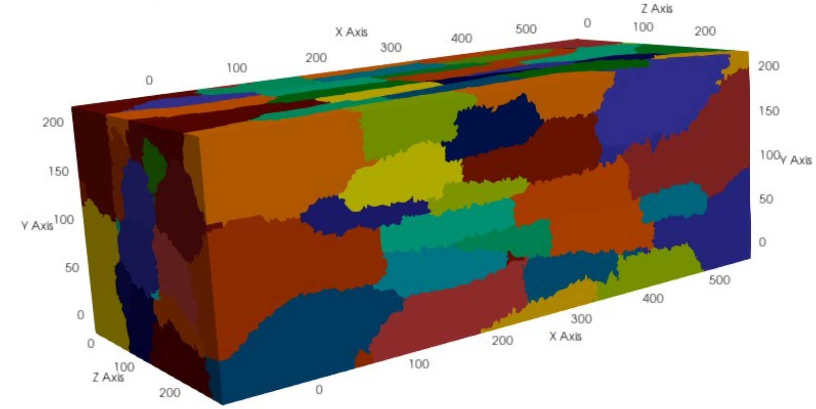


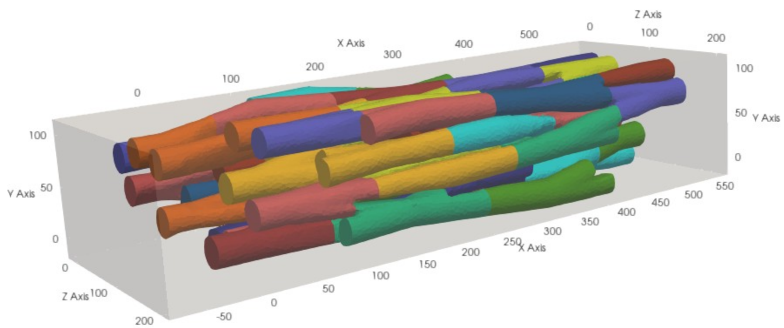
Mesh

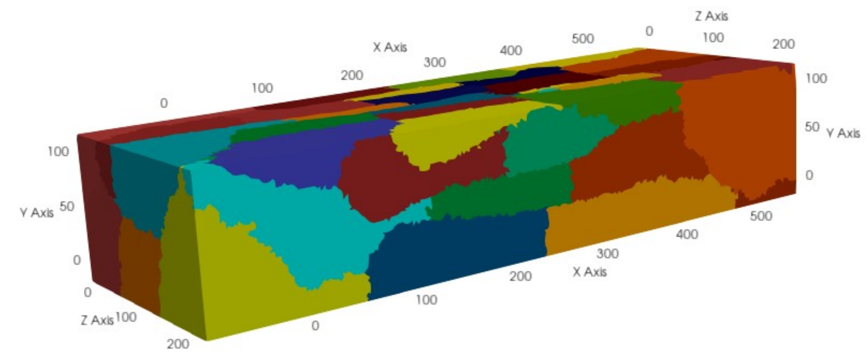

X Axis
Y Axis
Z Axis


X Axis
Y Axis
Z Axis

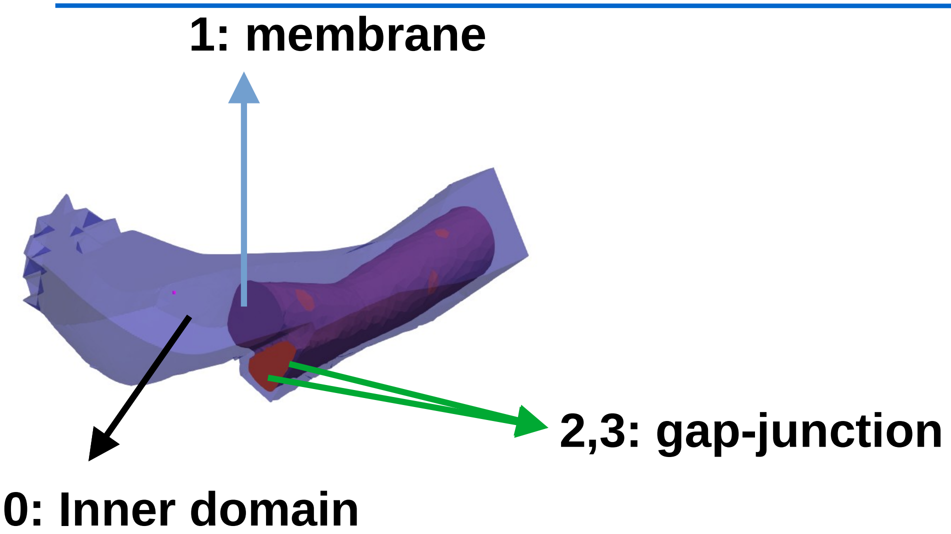

X Axis
Y Axis
Z Axis

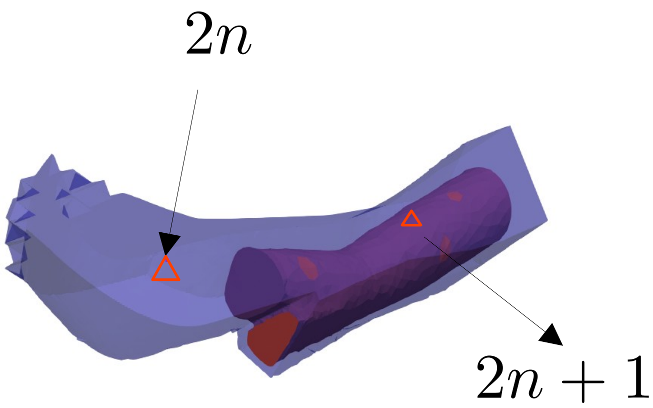

X Axis
Y Axis
Z Axis

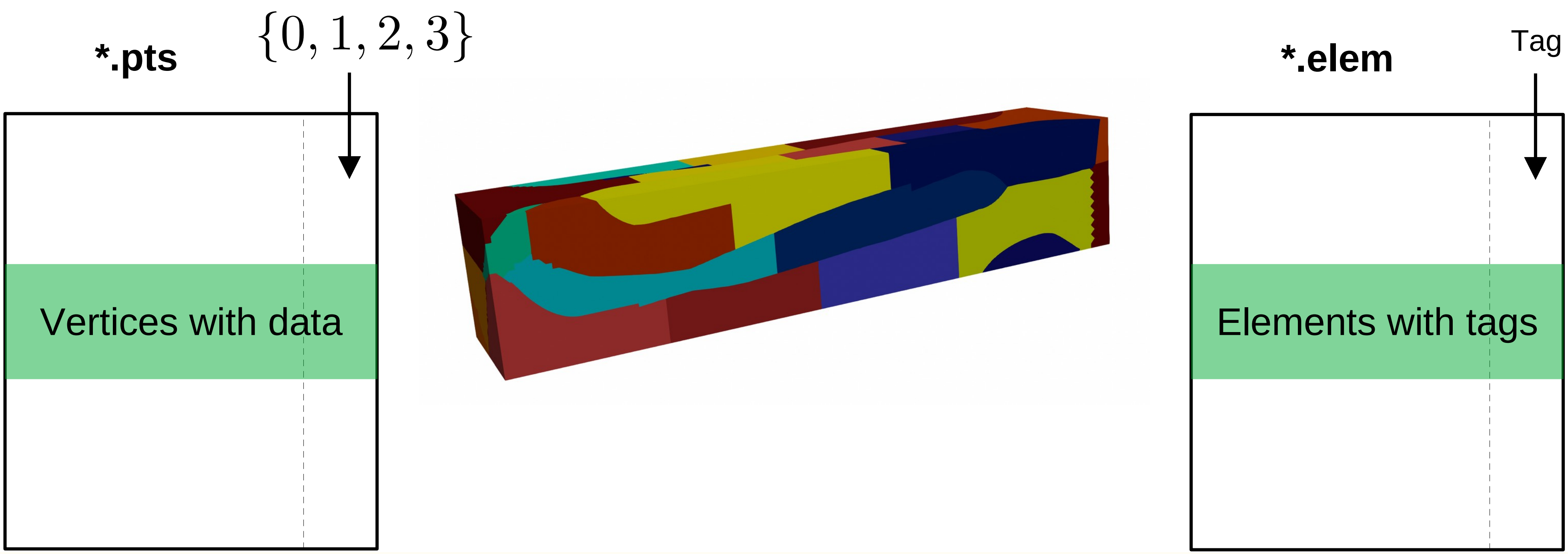

X Axis
Y Axis
Z Axis


X Axis
Y Axis
Z Axis

Mesh data in EMI

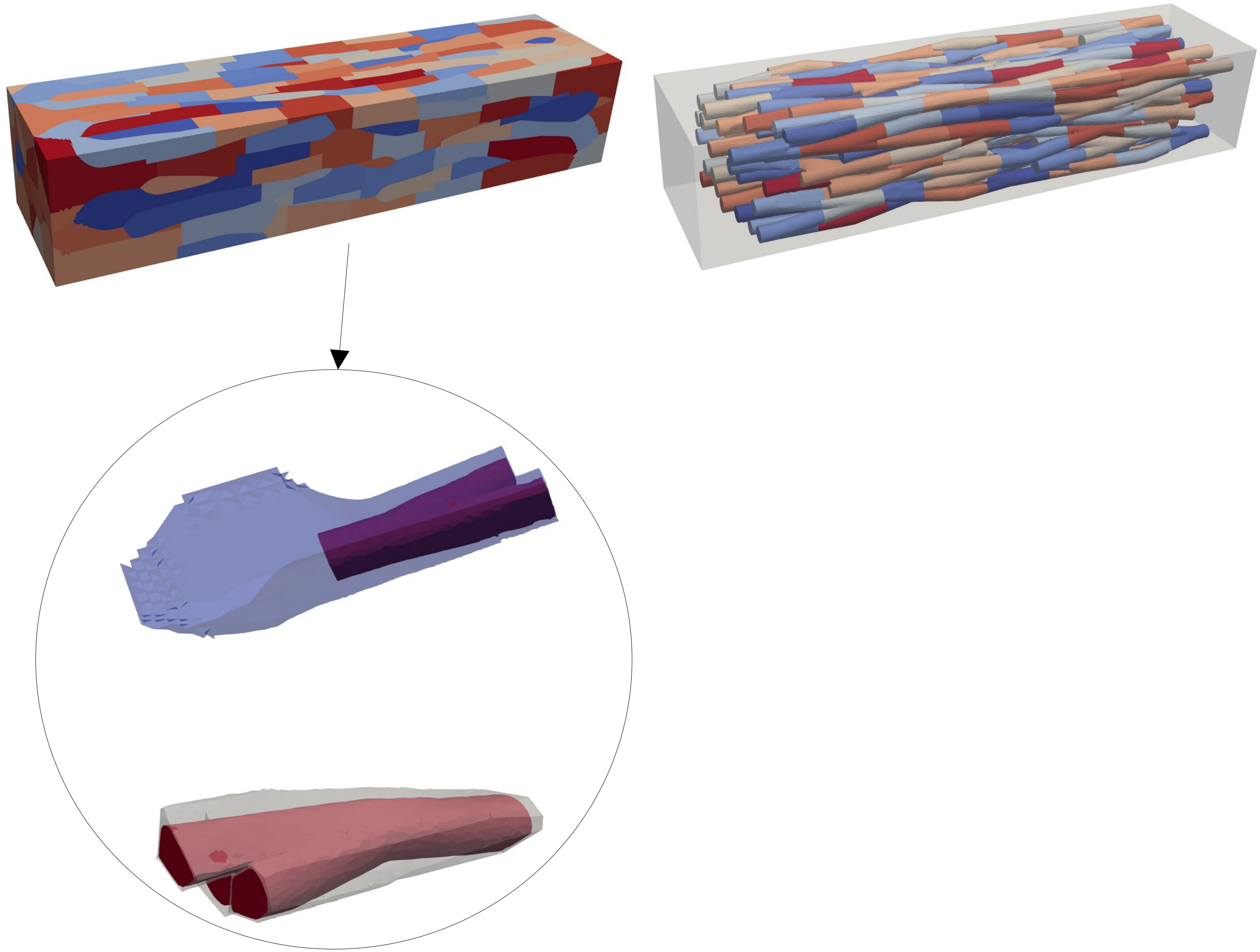






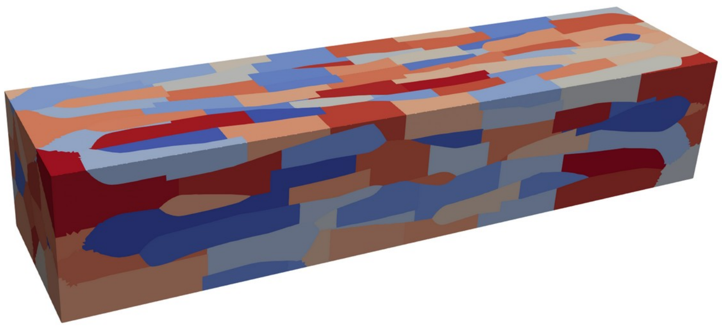
Mesh distribution

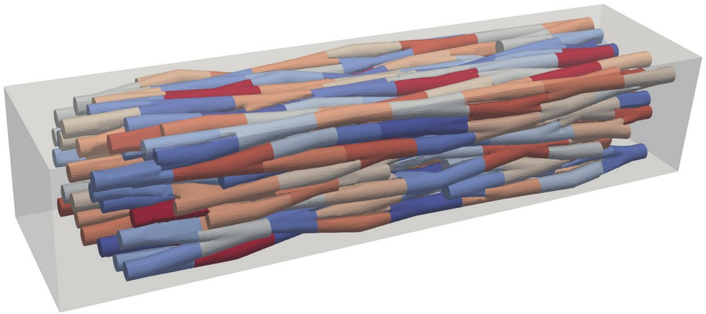
Mesh distribution



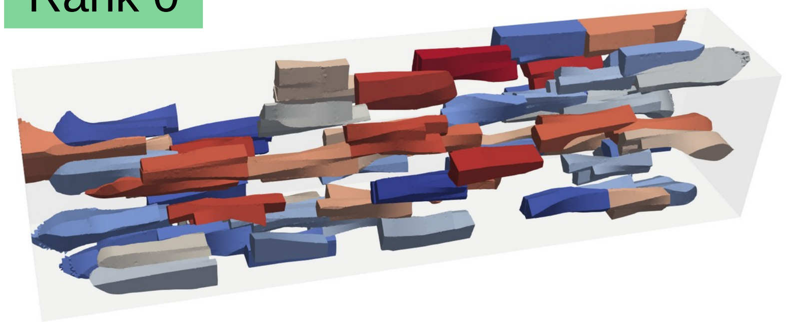
Mesh distribution

Mesh distribution

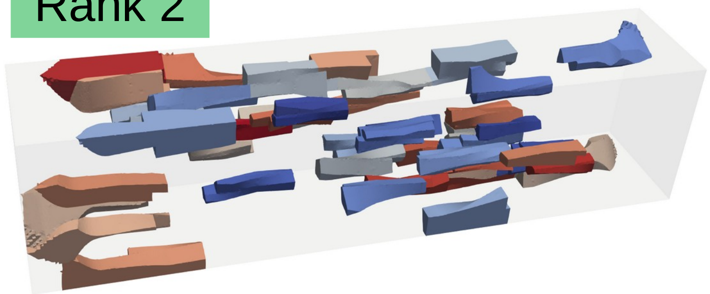




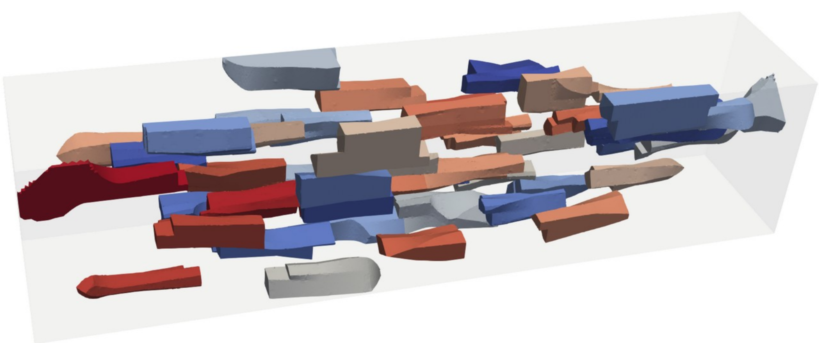
Rank 0



Rank 2

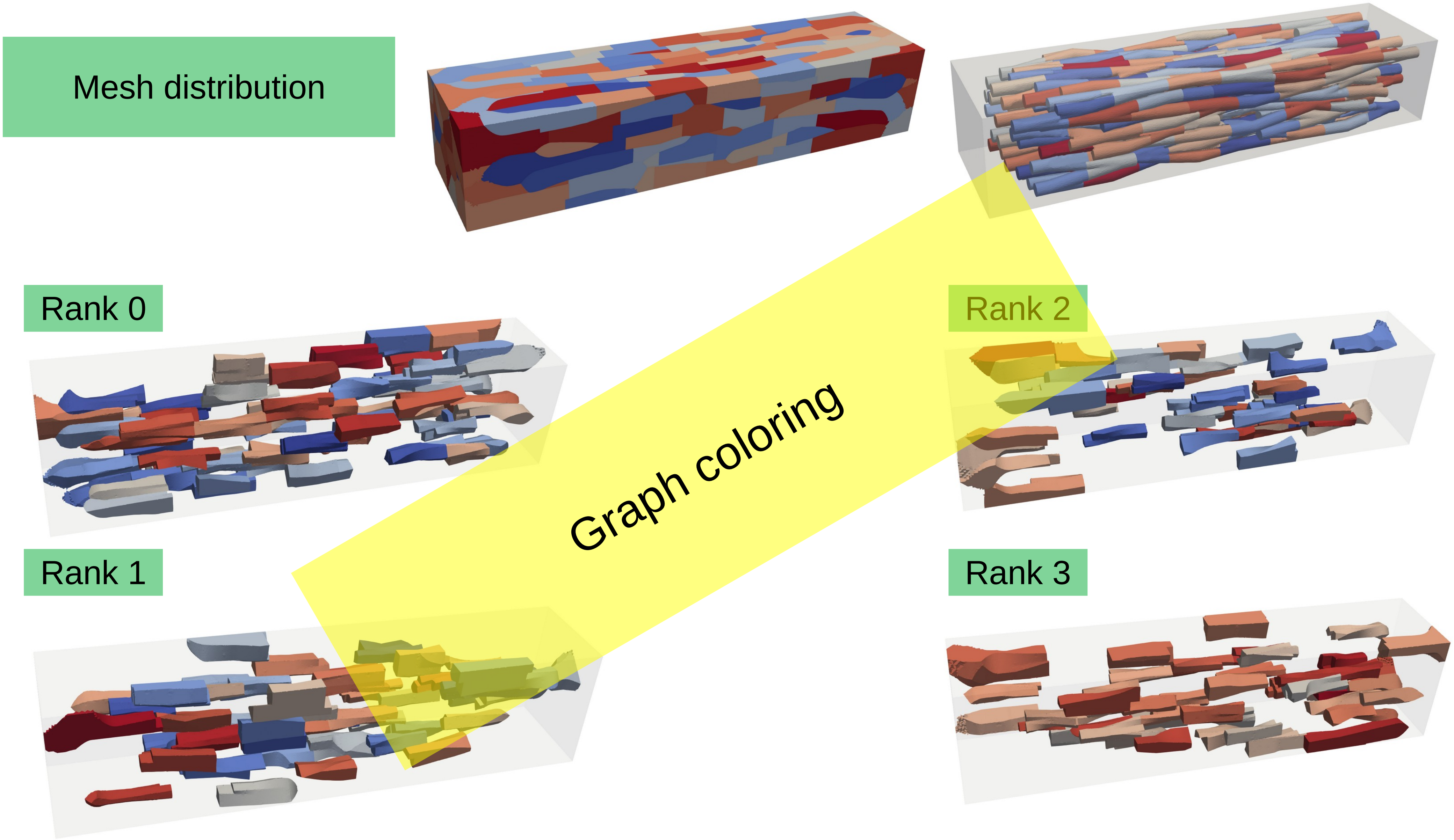


Rank 1



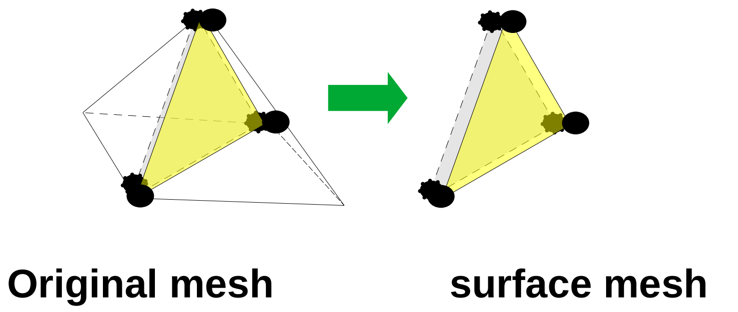
Rank 3

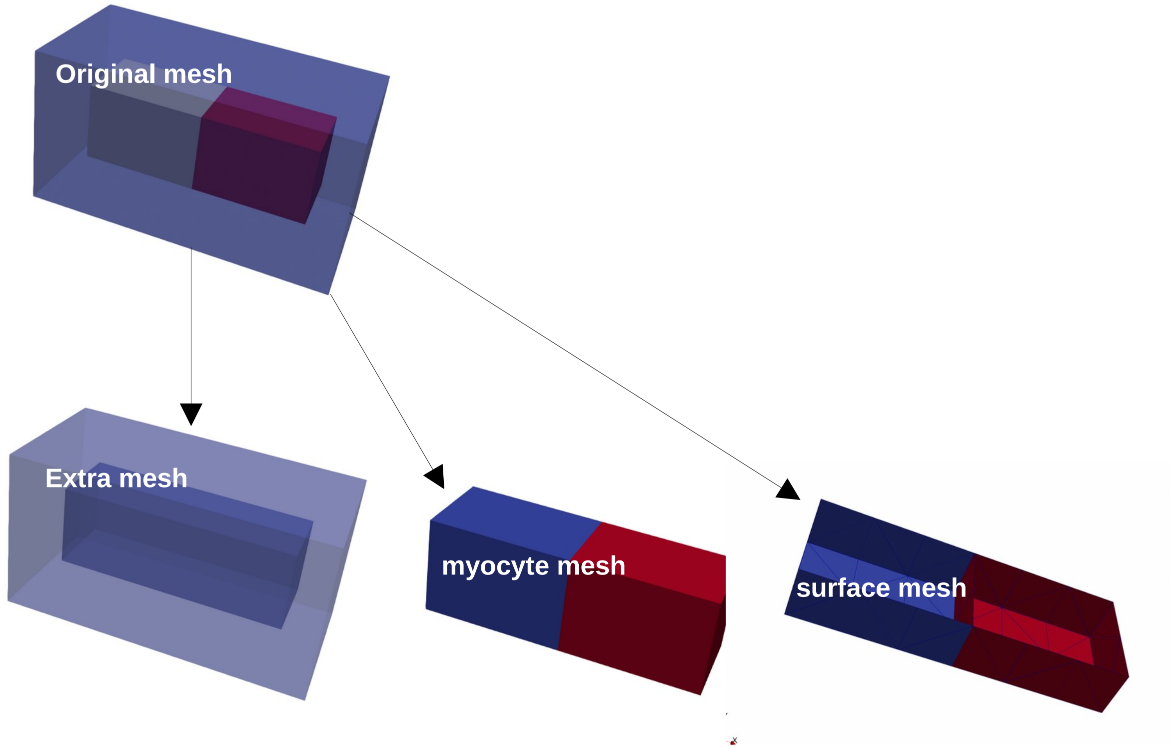
Mesh distribution



Mesh processing

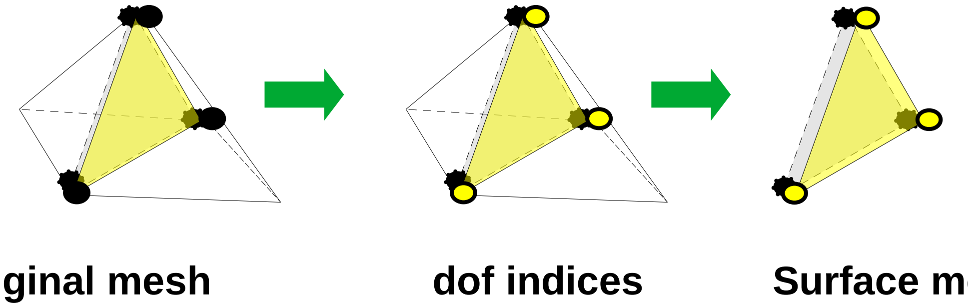
Extract interfaces





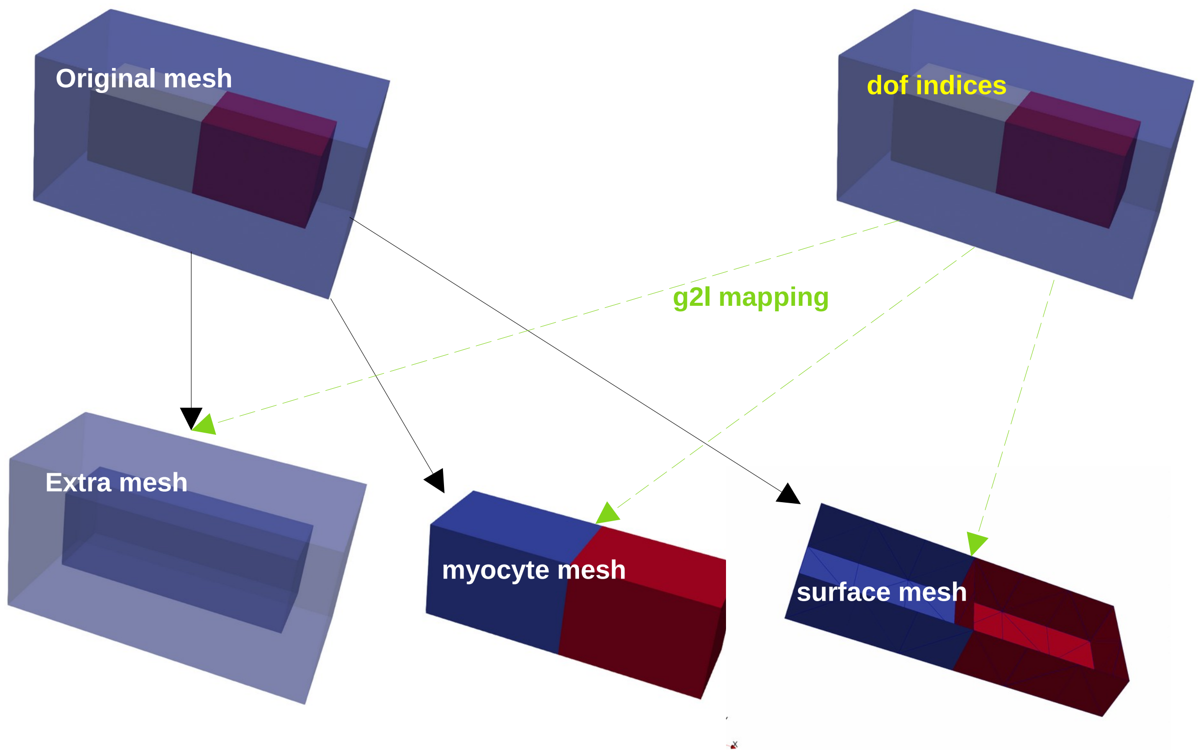
Mesh processing

Define dof indices

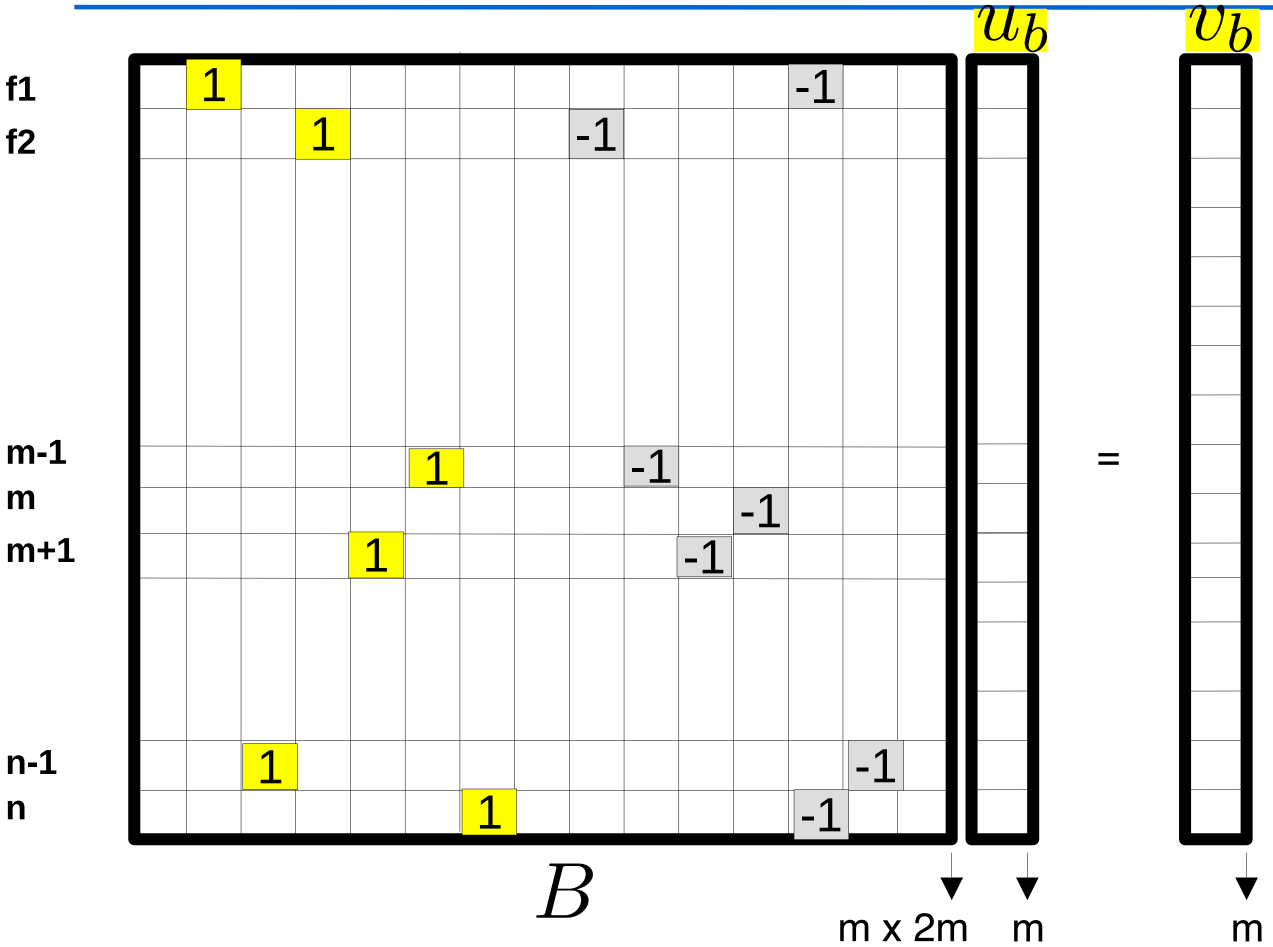


Extract submeshes

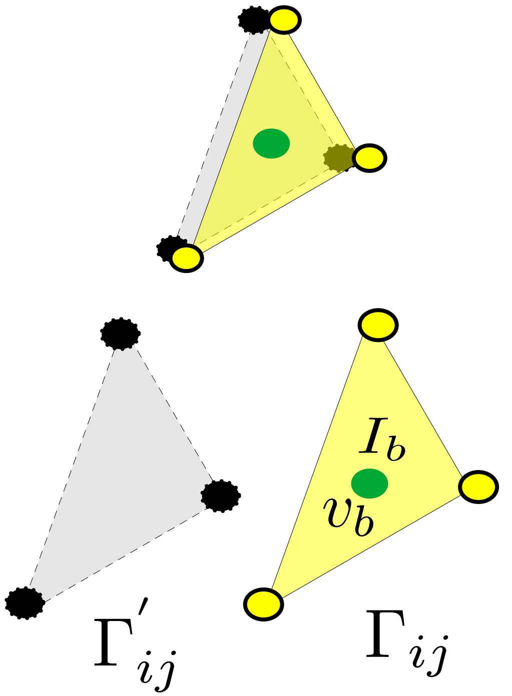
$$(\mathrm{Idx}_{\mathrm{old}}, \mathrm{tag}) \rightarrow (\mathrm{Idx}_{\mathrm{new}}, \mathrm{tag})$$



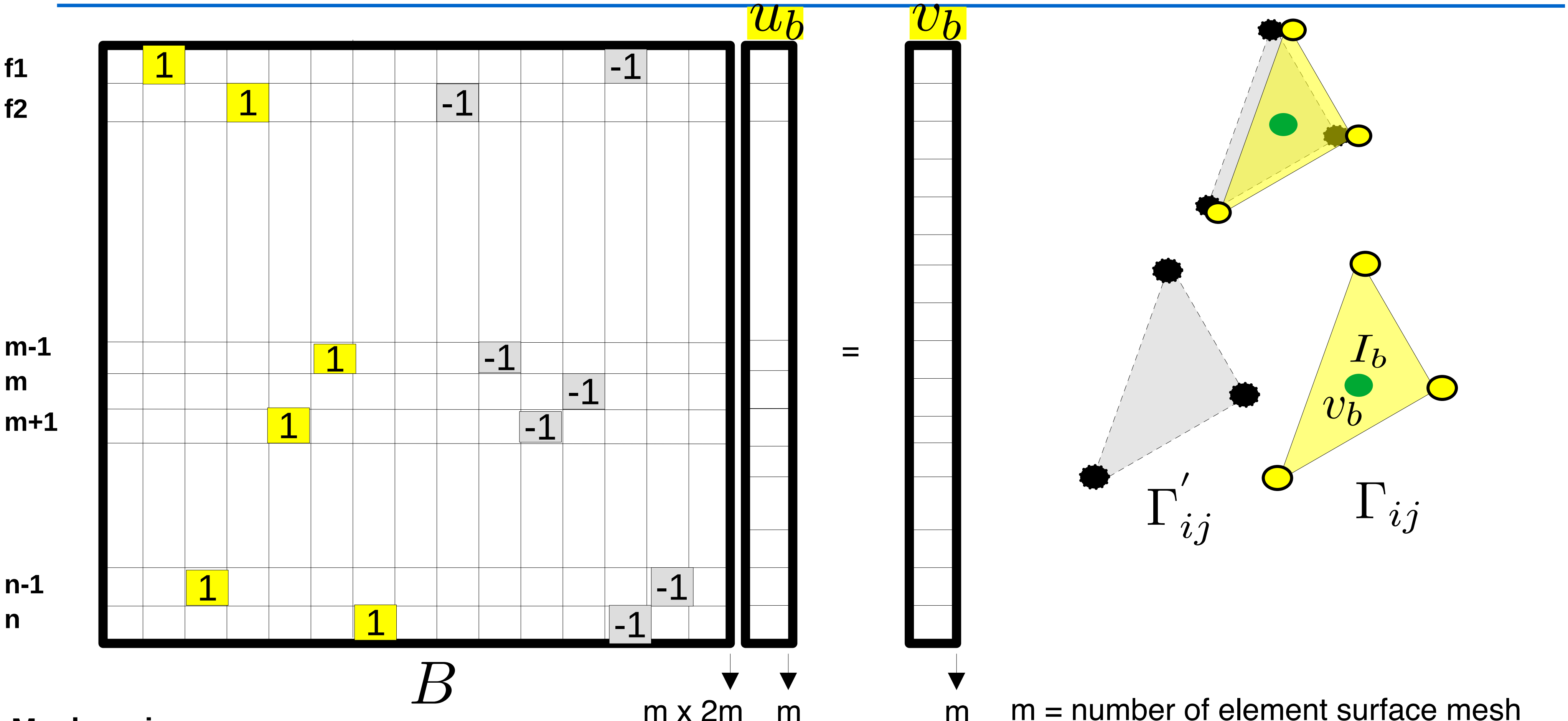
Operator B



m = number of element surface mesh



Operator B



Mask region

$$I_{ij}^{\text{ion}}(v_{ij}, w_{ij}) = \begin{cases} I_m(v_{ij}, w_{ij}) & \Gamma_m = \{\Gamma_{ij}, \Gamma'_{ij}\} \\ I_g(v_{ij}) = \dfrac{v_{ij}}{R_g}, \quad w_{ij} = 0 & \Gamma_g = \{\Gamma_{ij}, \Gamma'_{ij}\} \end{cases}$$

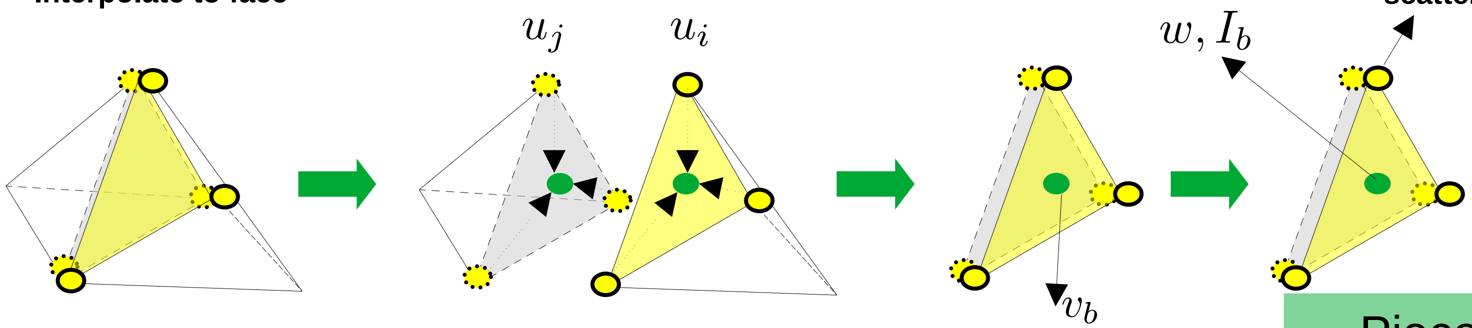
Time integration method

$$\int_0^T \left(\int_{\Omega_e} \sigma_e \nabla u_e \cdot \nabla \phi_e dx + \int_{\Omega_m} \sigma_m \nabla u_m \cdot \nabla \phi_m dx + \int_{\Gamma_{ij(i \neq j)}} (C(\dot{u}_i - \dot{u}_j) + I_{ij}^{\text{ion}}(u_i - u_j, w_{ij}))\varphi dx \right) dt = 0$$

$u_i, u_j \longrightarrow \dot{w} = f(v_b, w) \longrightarrow I_b(v_b, w) \longrightarrow I_{ij}(v_{ij}, w)$

Interpolate to face

scatter it to vertices



Piecewise constant

$M\dot{u} = L(u)$

Implicit Euler $\longrightarrow$ $M\dfrac{u_{k+1} - u_k}{\tau} = L(u_{k+1})$

Linearization point for Jacobian u_0

1st order Taylor exp $\longrightarrow$ $M\dfrac{u_{k+1} - u_k}{\tau} = L(u_k) + L^{'}(u_0)(u_{k+1} - u_k)$

$$(M - \tau L'(u0))\delta u_k = \tau L(u_k), \quad u_{k+1} = u_k + \delta u_k$$

$M_{\Gamma\Gamma}$ $K_e,$ K_m

Semi-implicit method

Status update

EMI in Kaskade

- SDC& BDDC
  - Algebraic BDDC in SDC
- BDDC Sub matrices
  - data for PETSc
  - data for Ginkgo
  - data for Kaskade
  - WP 3,4,5
- SDC
  + higher order Time integ.
  + algebraic adaptivity
  + ladder method
- EMI
  + semi-implicit method
  + more extra subdomains
  + Gating variable on faces
  + activation Time

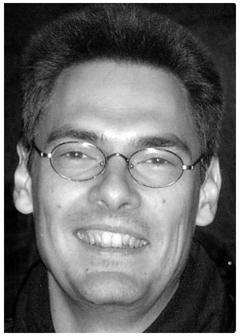
EMI in openCarp

- SDC& BDDC
- SDC
- BDDC Sub matrices
- EMI

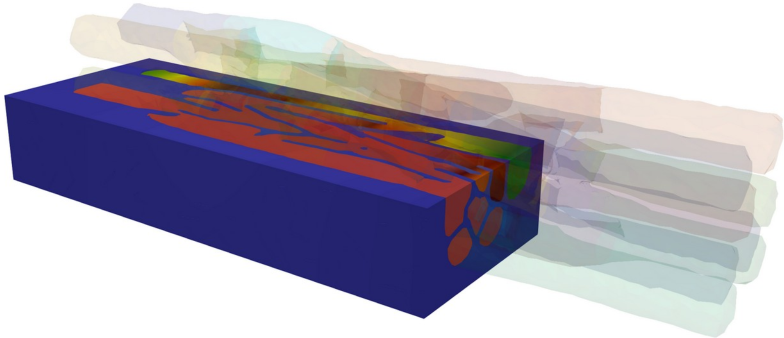
WP 1,3

EMI Mesh processing In openCarp

- Ionic class
  - Compute potential on barycentric coords
  - Compute current on barycentric coords.
  - Scatter current on the nodes
- semi-implicit
  - extra + myo + interface meshes
  - Assemble mass & stiffness matrices
  - Assemble rhs
- dof indices
  - new indices (only) on membrane and gap junction
  - operator B
  - (Idx(new), tag): Idx(old)
- Interface mesh
  + extract surface mesh with counter part
- Mesh distribution
  + mesh distribution
  + extra and myo (one rank)

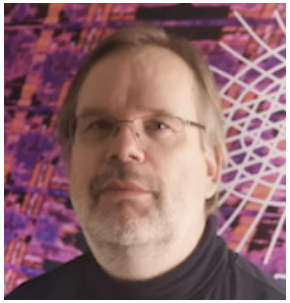


Martin Weiser(**Head of WP3**)



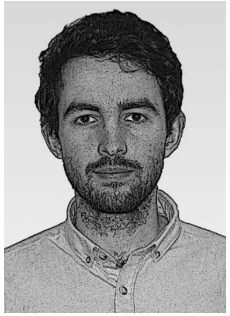
SPONSORED BY THE
Federal Ministry
of Education
and Research

EuroHPC
Joint Undertaking



Thomas Steinke(**WP3**) Aurel Neic(**WP1**)

Thank you for your attention



Fritz Göbel(**WP4**) Ngoc Mai Monica Huynh (**WP5**) Tomas Stary(**WP1**)

The project leading to this application has received funding from the European High-Performance Computing Joint Undertaking Joint Undertaking (JU) under grant agreement No 955495. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and France, Italy, Germany, Austria, Norway, Switzerland