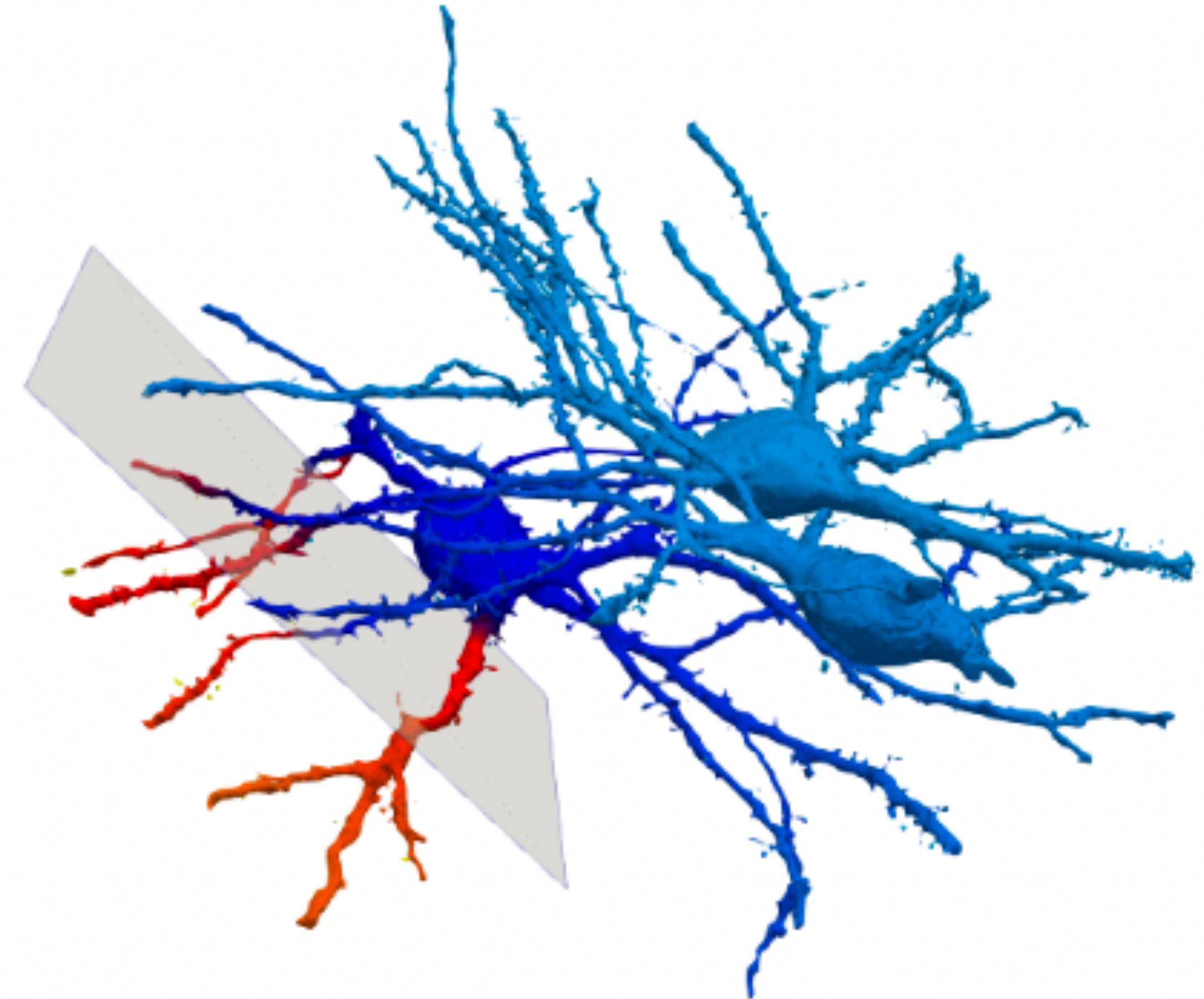


Cell-by-cell Modelling Tutorial

Pietro Benedusi

Università della Svizzera Italiana, Lugano, Switzerland

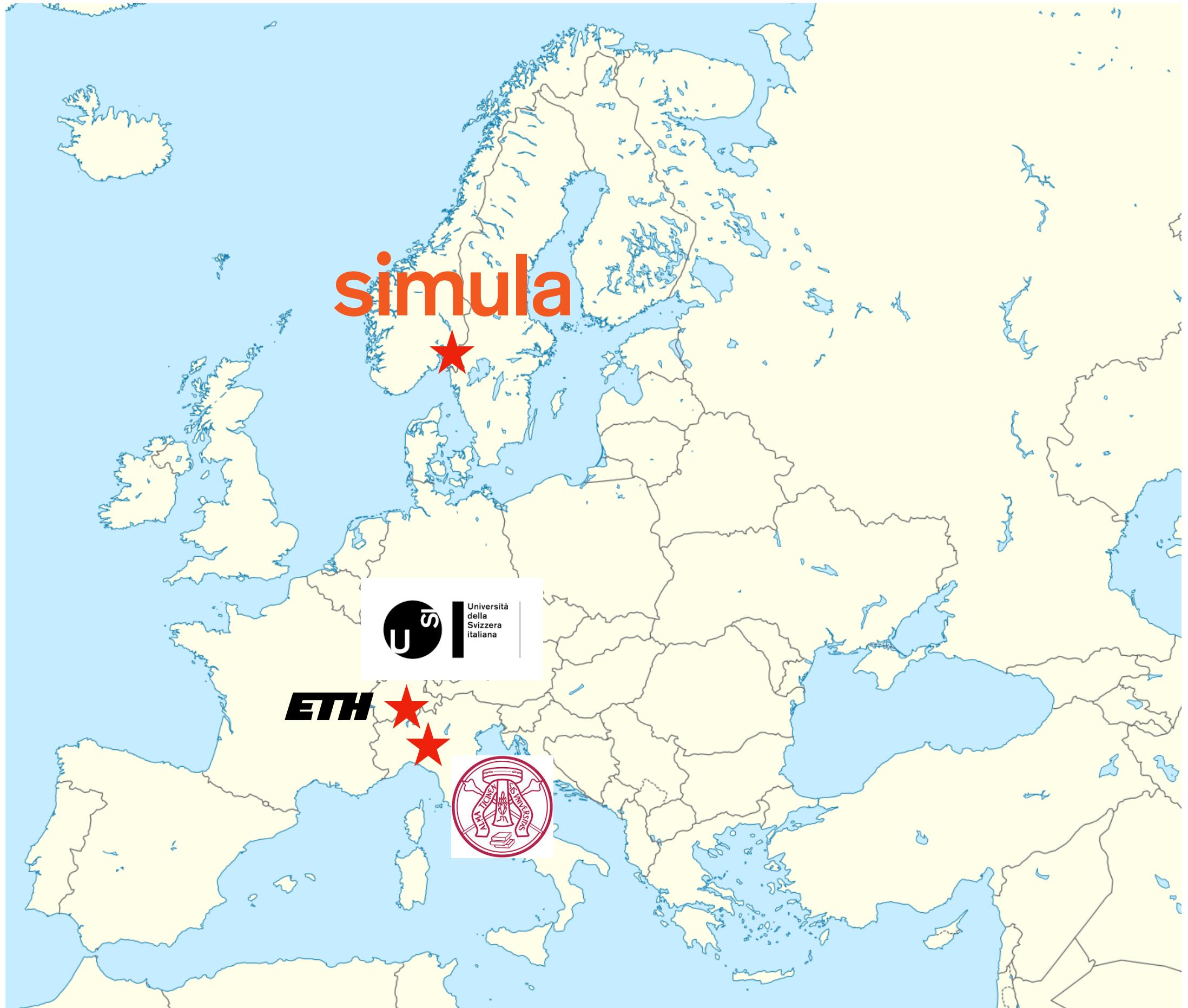
MICROCARD Summer Workshop, Oslo, June 26 2024



simula

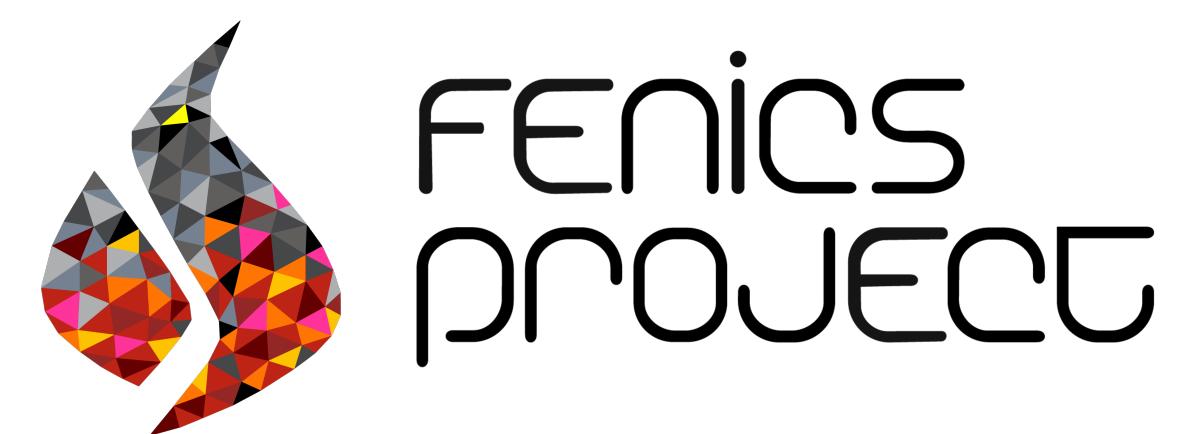


Università
della
Svizzera
italiana

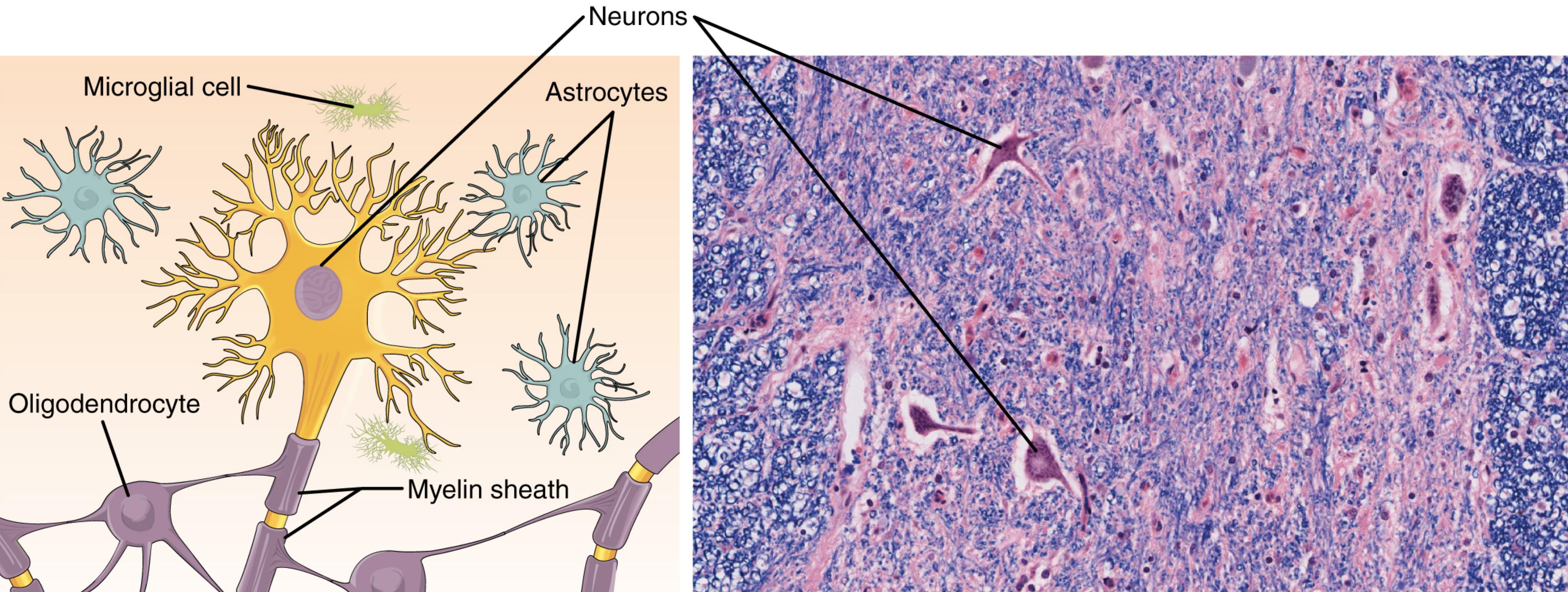


Program

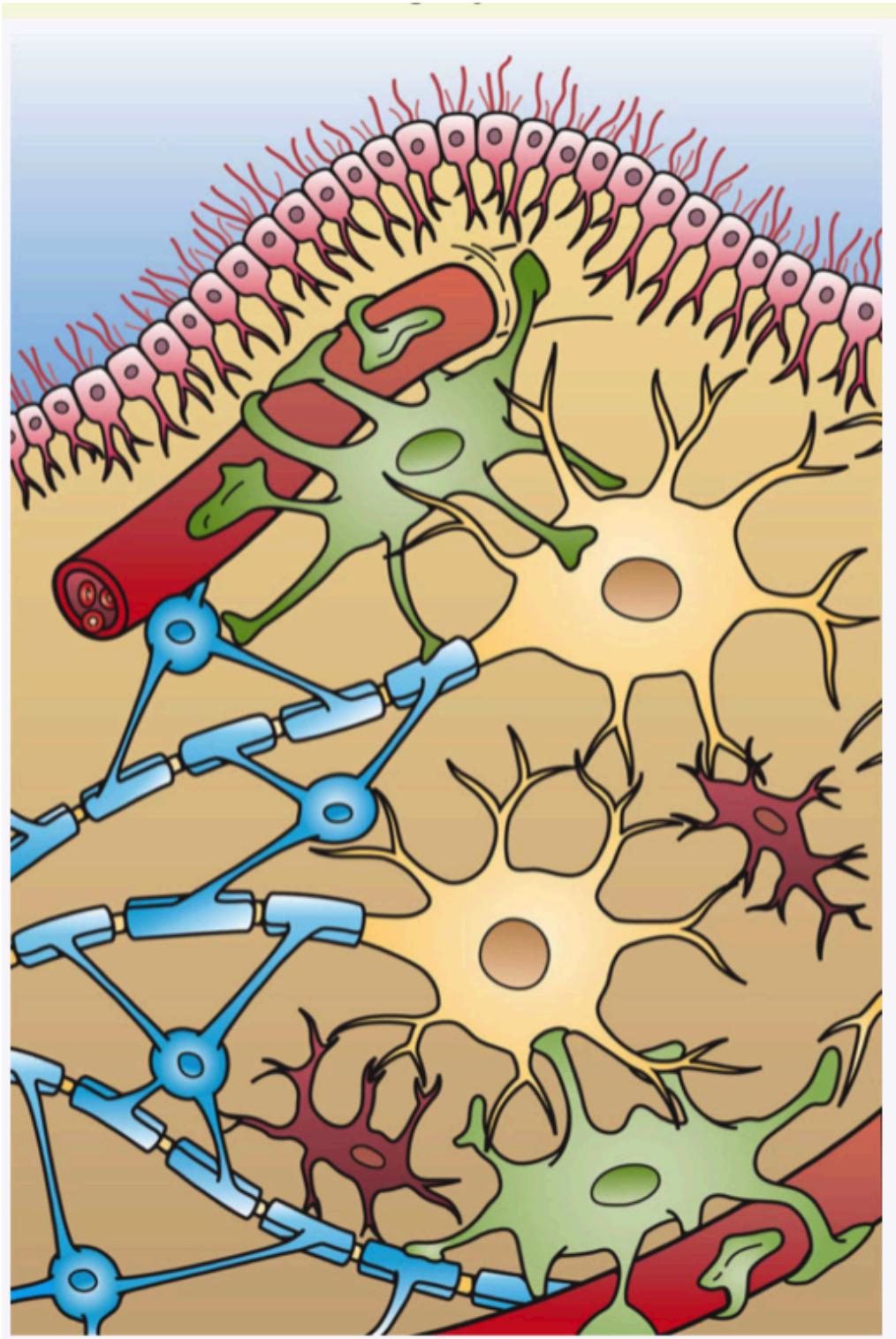
- Cell-by-cell modelling introduction (~ 1 hour)
 - A. PDEs modelling
 - B. Finite element method
 - C. Electrophysiology
- break (5 min)
- Hands-on exercises (~ 1 hour)
- break (5 min)
- Cell-by-cell coding (~ 1 hour)
 - Docker, FEniCS, Paraview



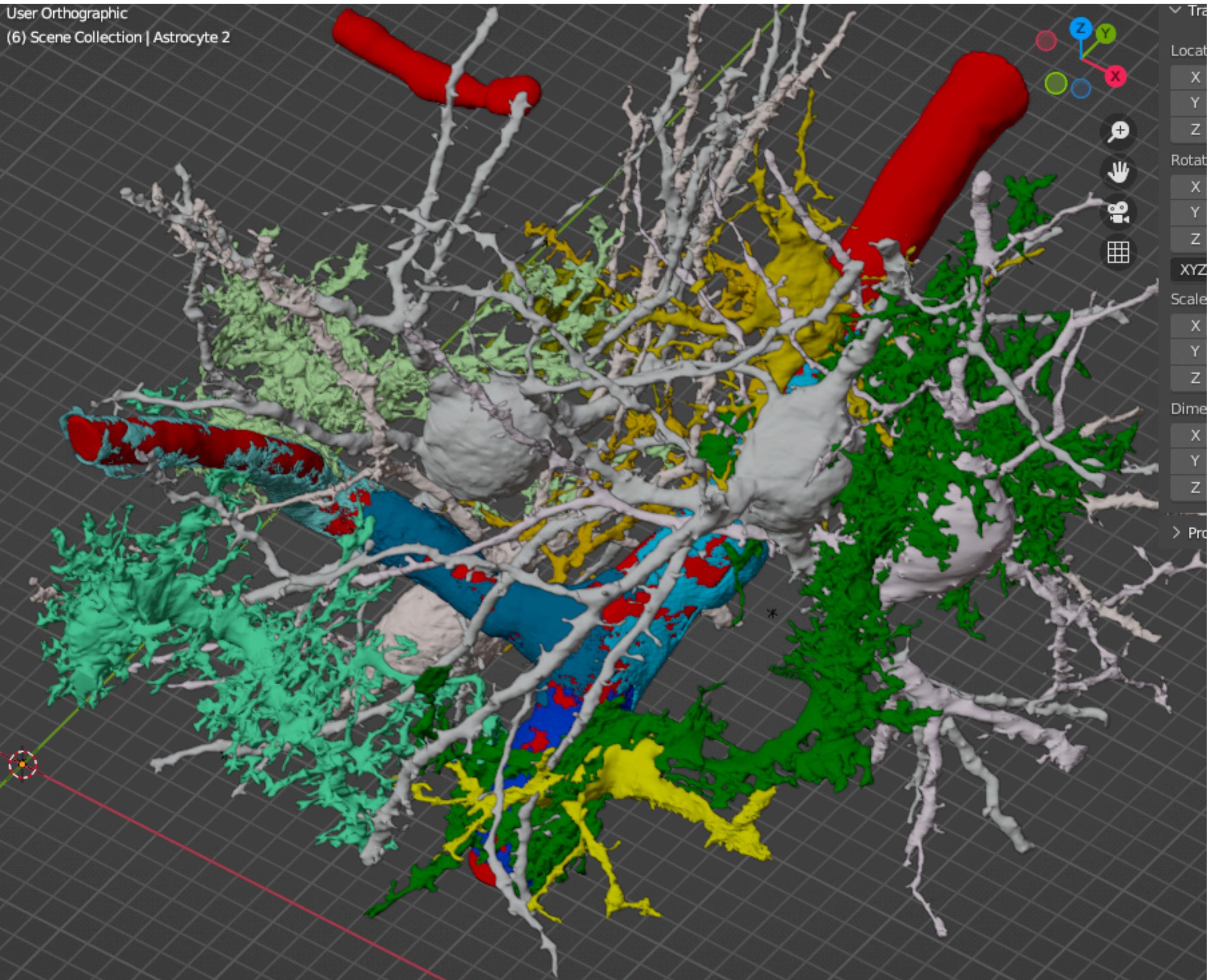
Computational electrophysiology



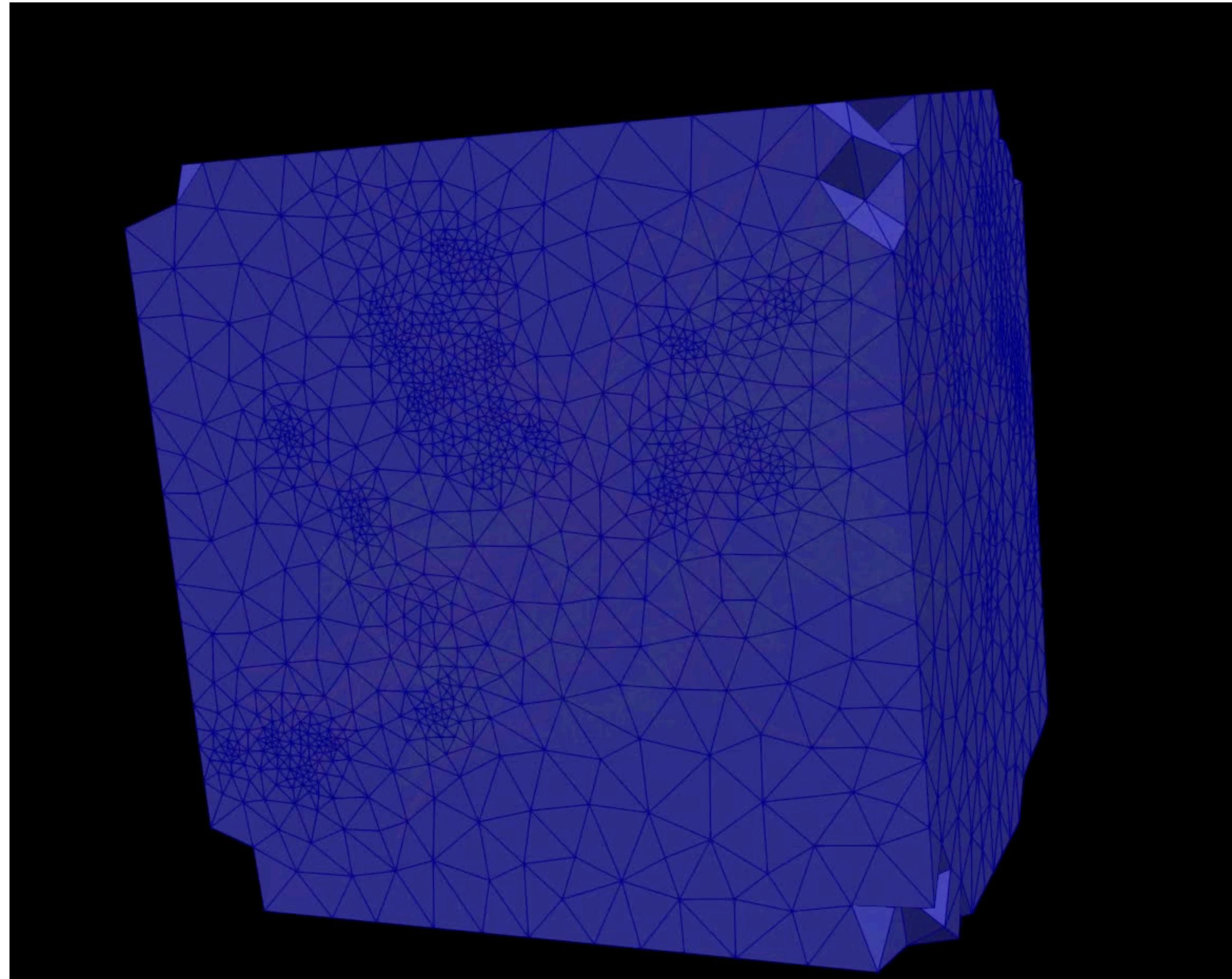
Motivation: dense 3D reconstructions



Murine cortex
[Calí 2016]

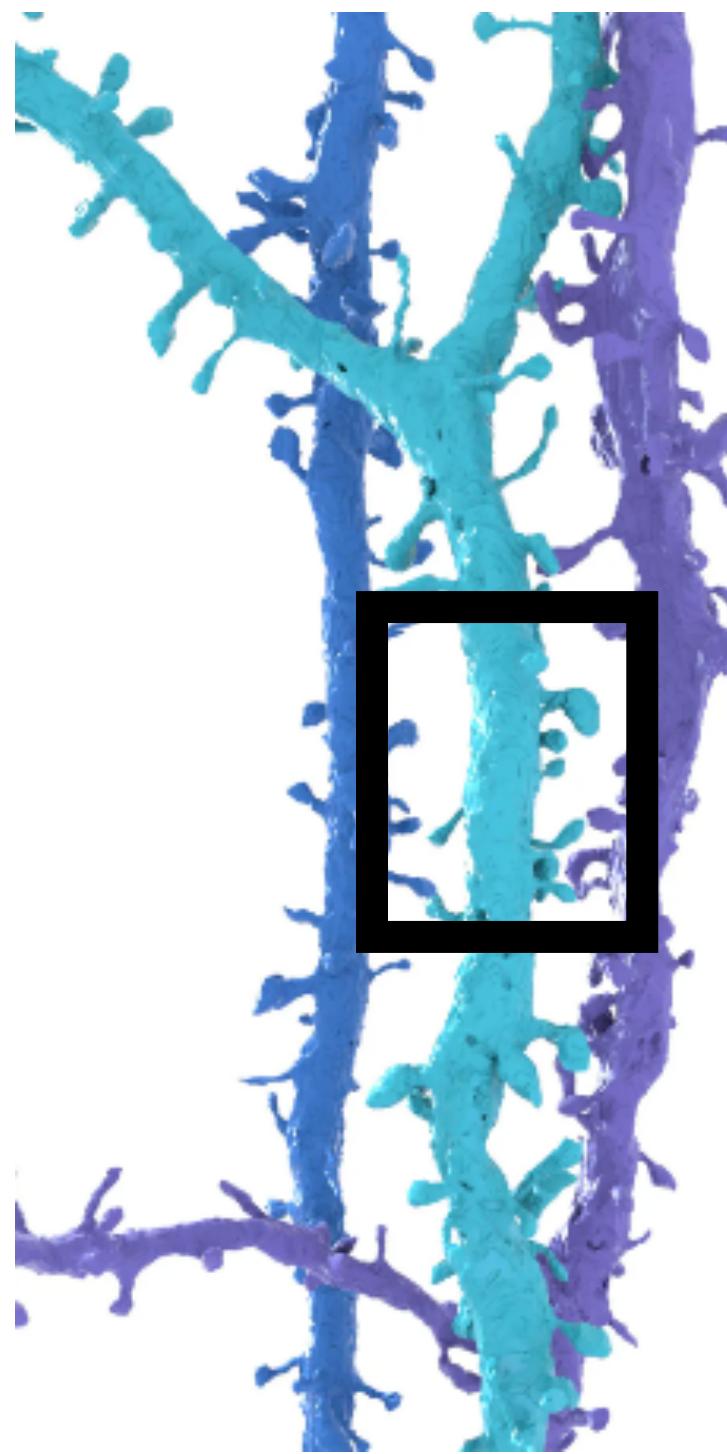


Motivation: dense 3D reconstructions

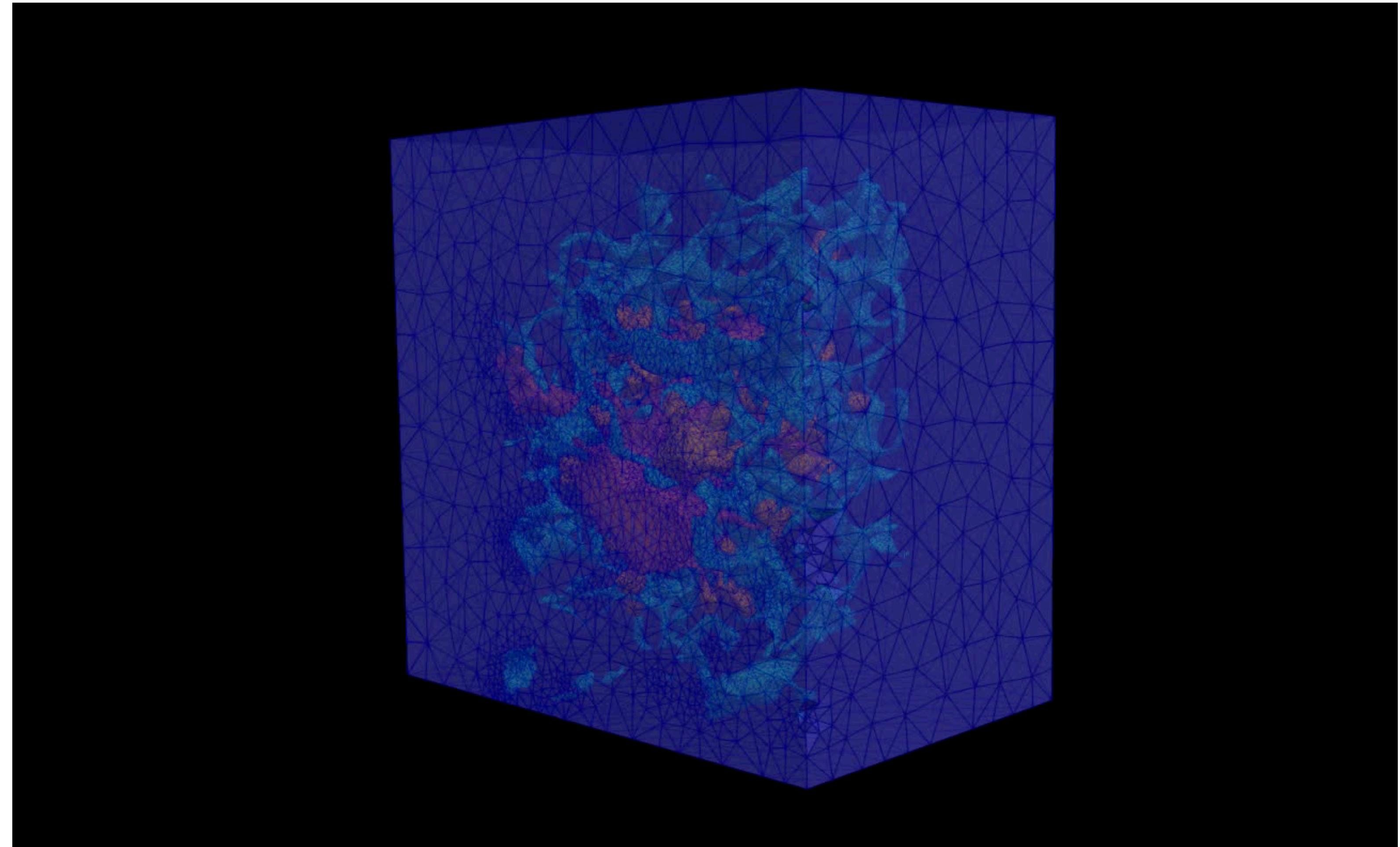


Murine cortex
[Calí 2016]

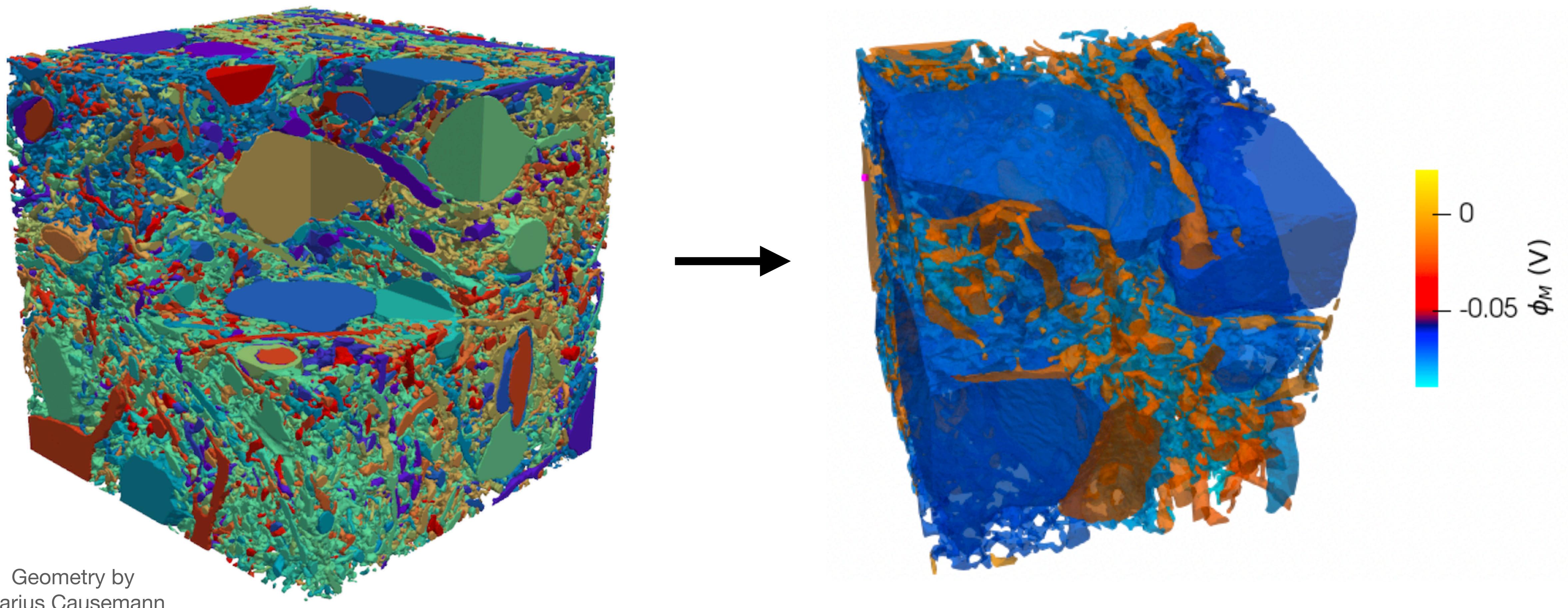
Motivation: dense 3D reconstructions



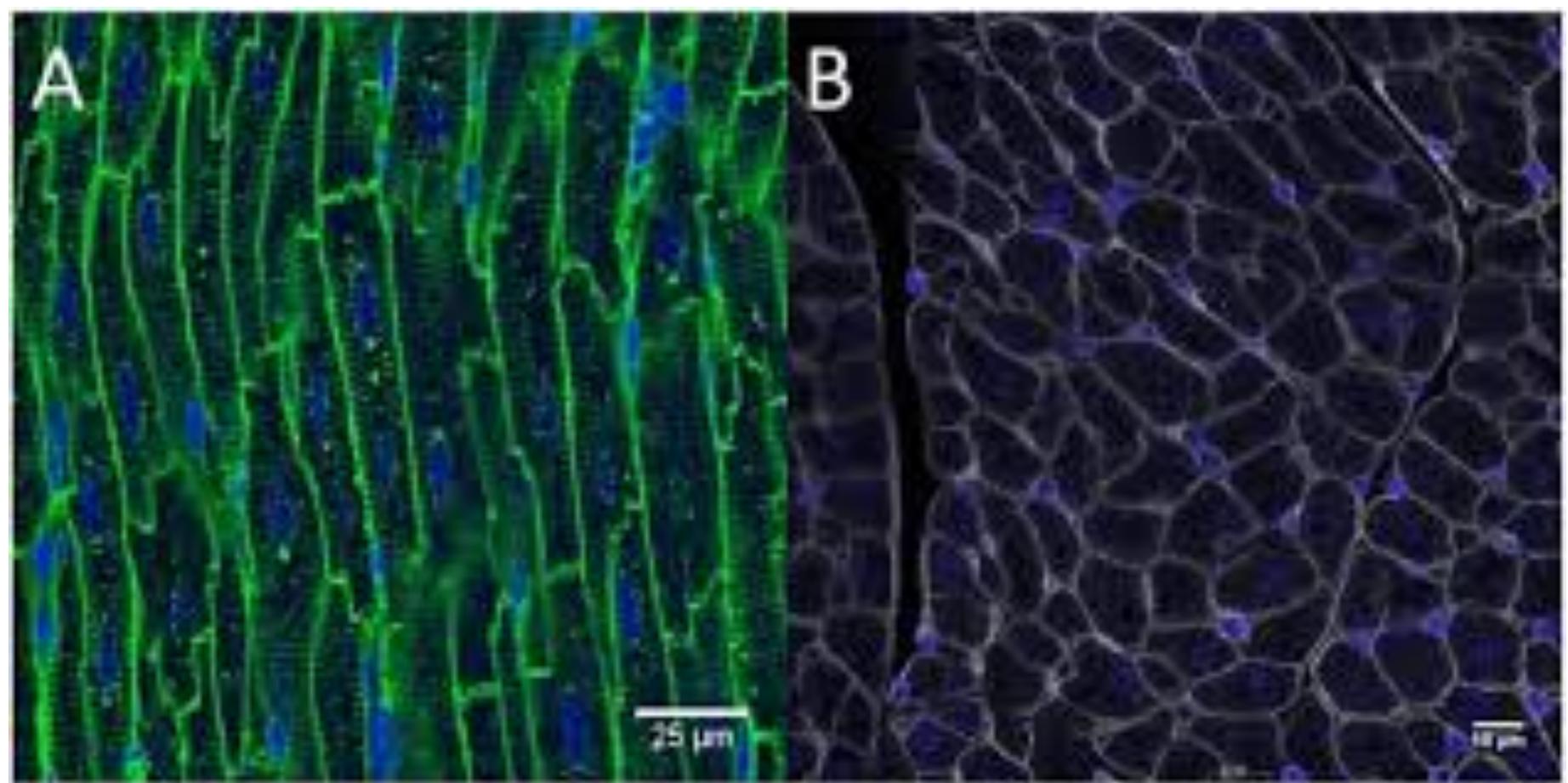
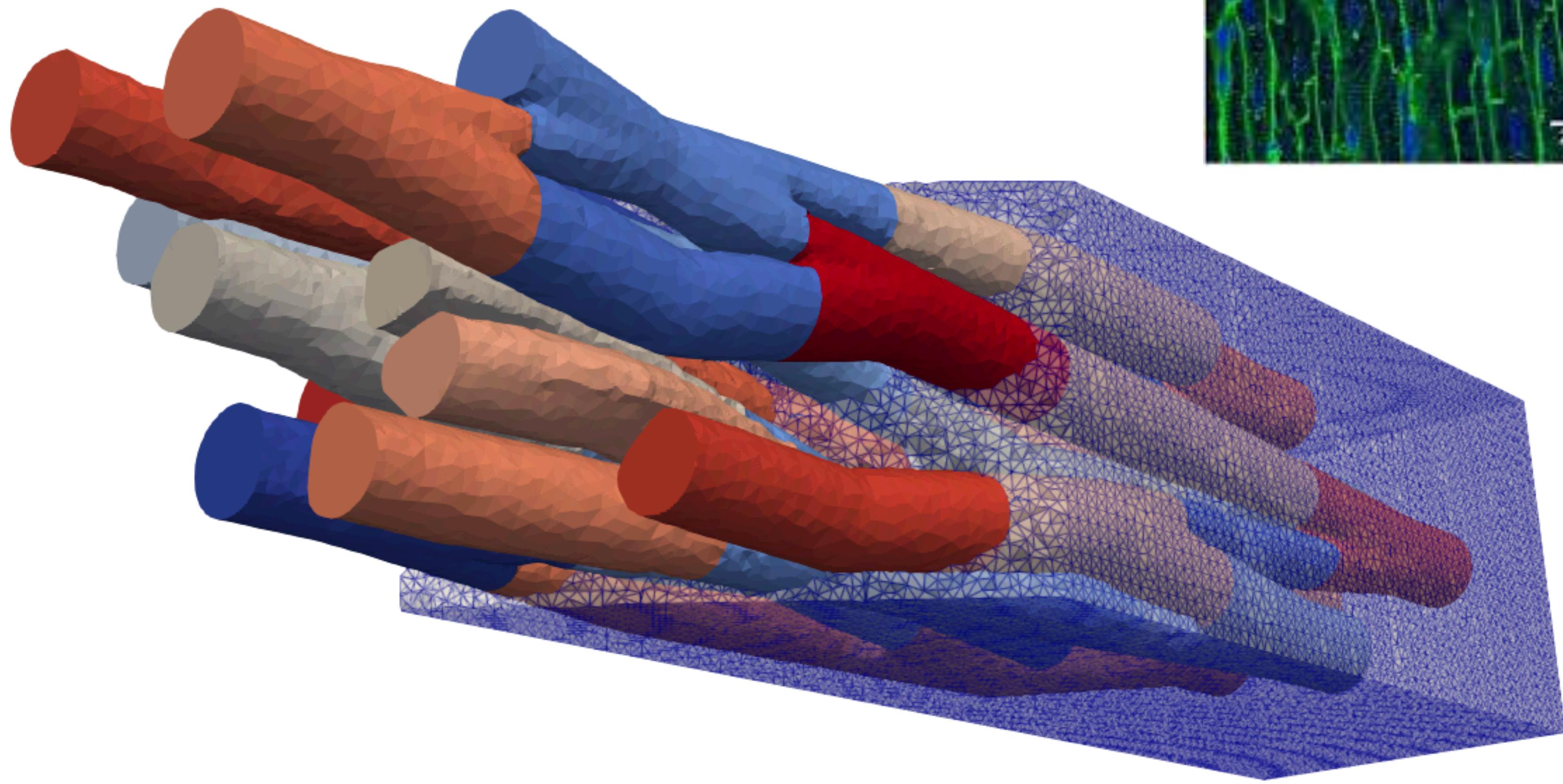
Dendritic segment
[Bell 2019]



Motivation: dense 3D reconstructions



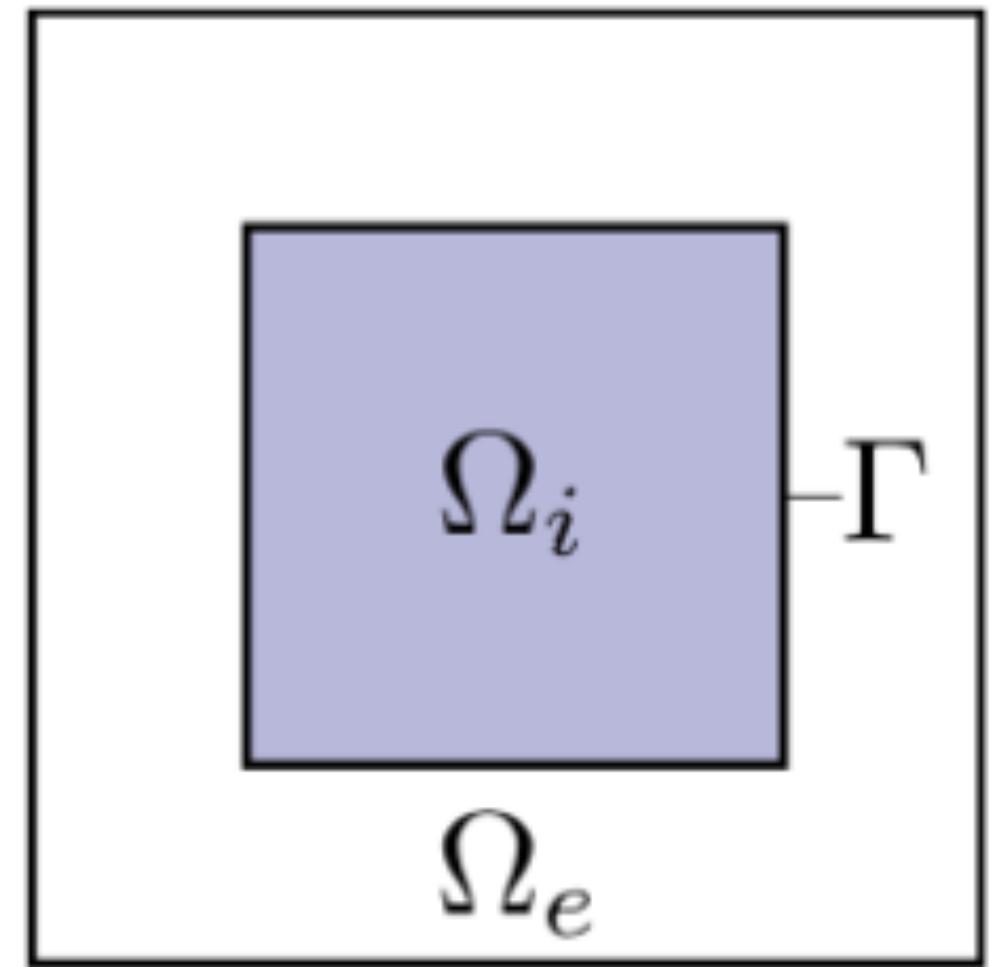
Motivation: cardiomyocytes



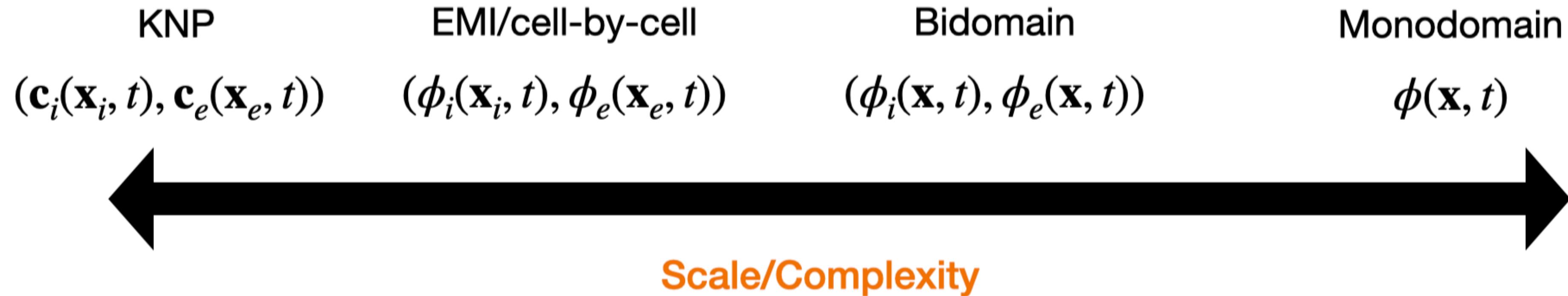
Electrophysiological models

Quantities of interest:

- Potential $\phi : \Omega \rightarrow \mathbb{R}$
- Ionic concentrations $c_1, c_2, \dots, c_{N_{\text{ion}}} : \Omega \rightarrow \mathbb{R}_{\geq 0}$



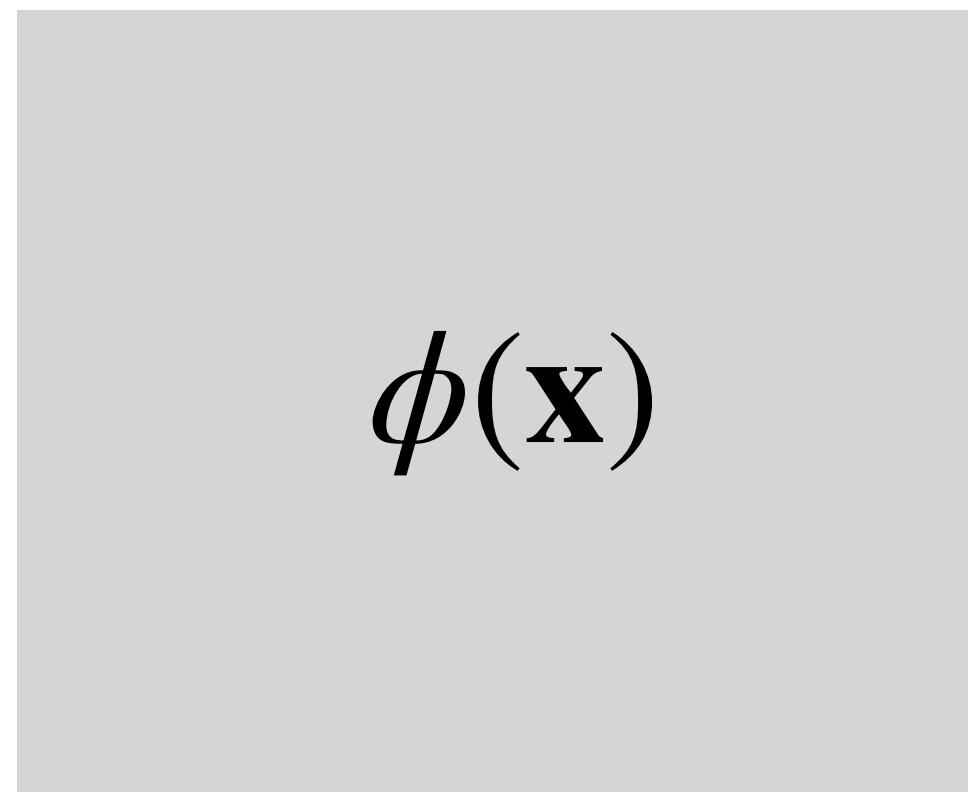
$$\mathbf{x} \in \Omega = \Omega_e \cup \Omega_i, \quad \mathbf{x}_i \in \Omega_i, \quad \mathbf{x}_e \in \Omega_e$$



EMI = Extra-Membrane-Intracellular
KNP = Kirkhoff-Nerst-Plank

Electrophysiological models: setup

Monodomain ($\sim \text{mm/cm}$)



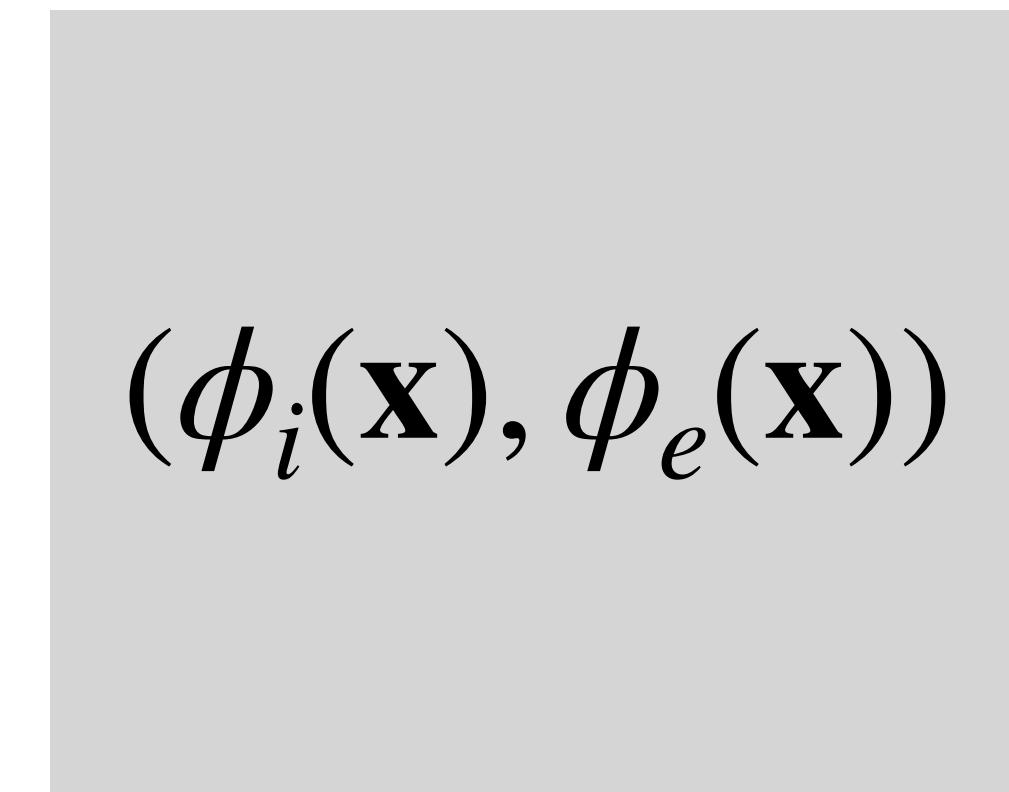
$$\mathbf{x} \in \Omega$$

In bi/monodomain properties that vary along the cell, or differences from cell to cell can not be included.

KNP or PNP-EMI

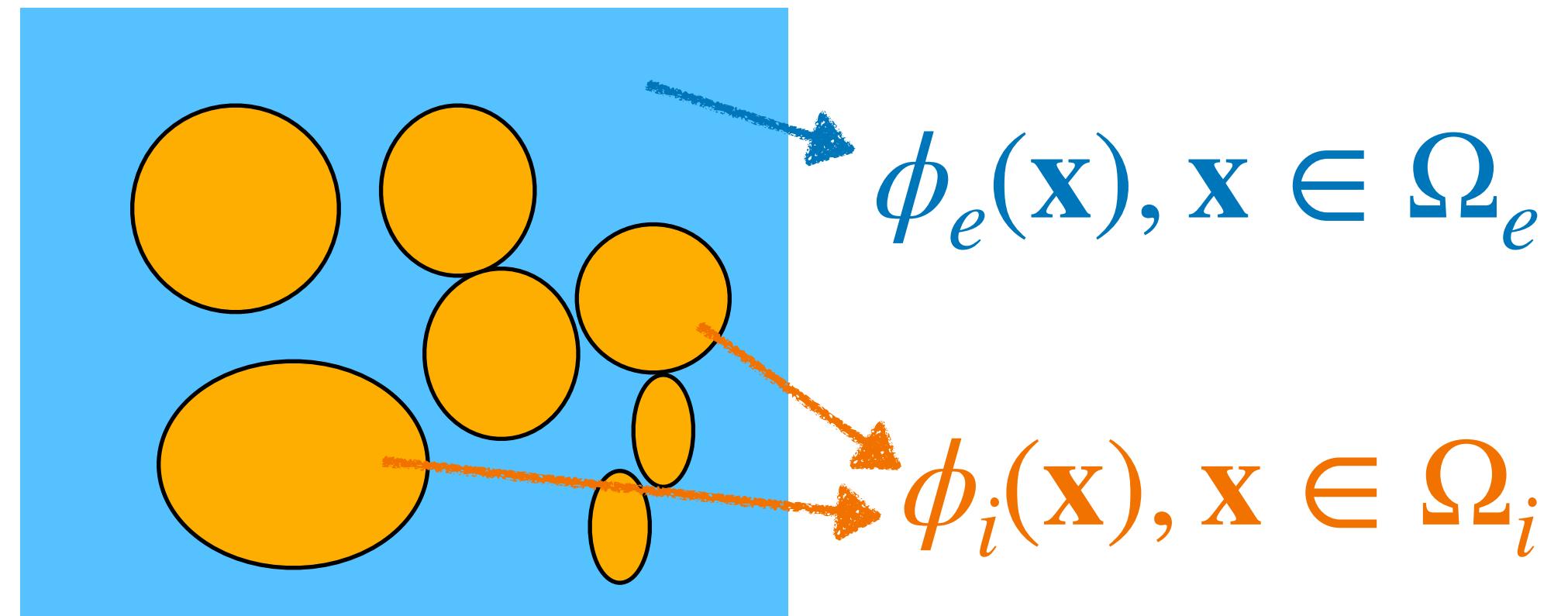
$$([\text{Na}^+]_e, [\text{K}^+]_e, \dots, \phi_e)$$
$$([\text{Na}^+]_i, [\text{K}^+]_i, \dots, \phi_i)$$

Bidomain ($\sim \text{mm/cm}$)



$$\mathbf{x} \in \Omega$$

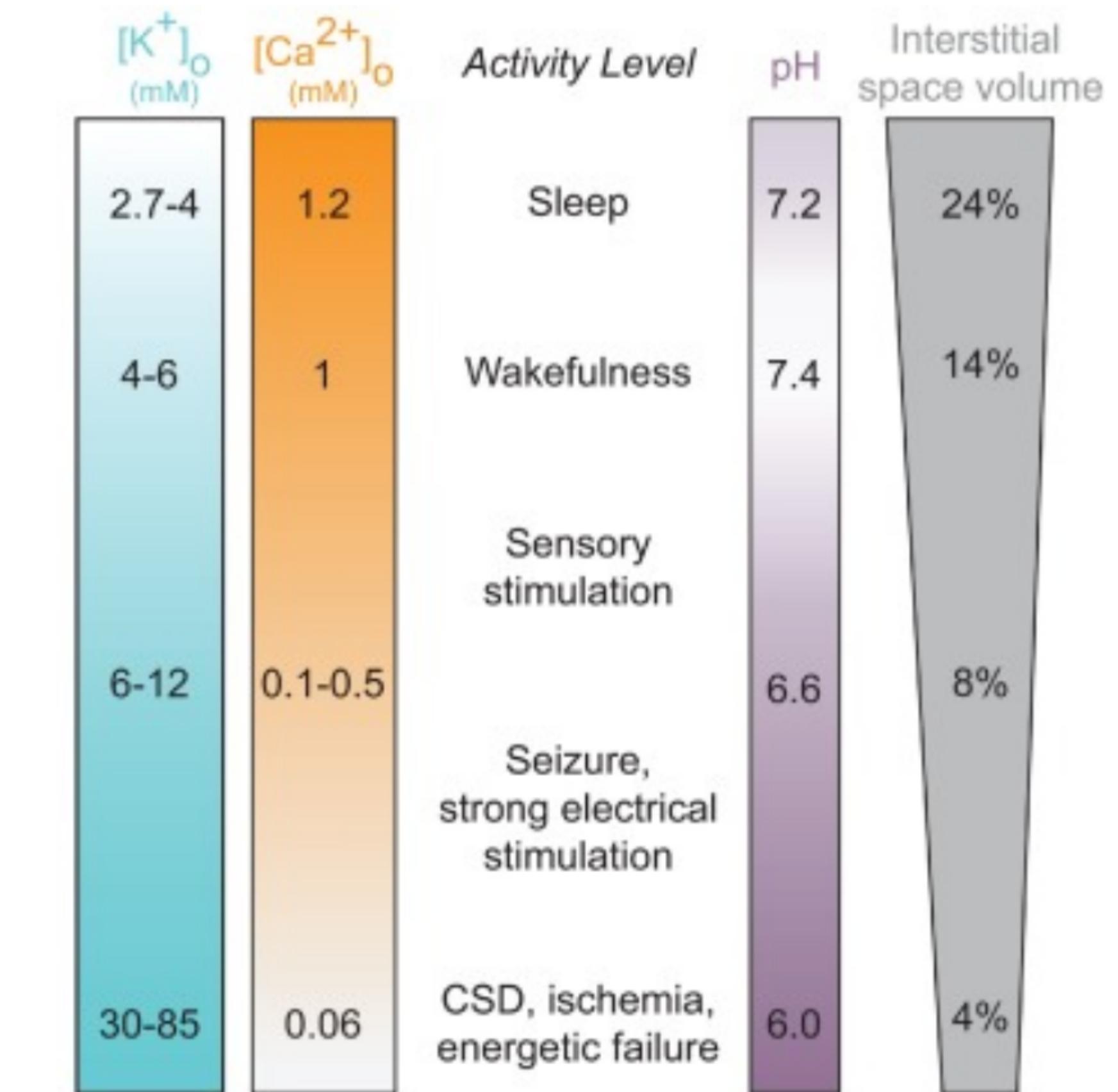
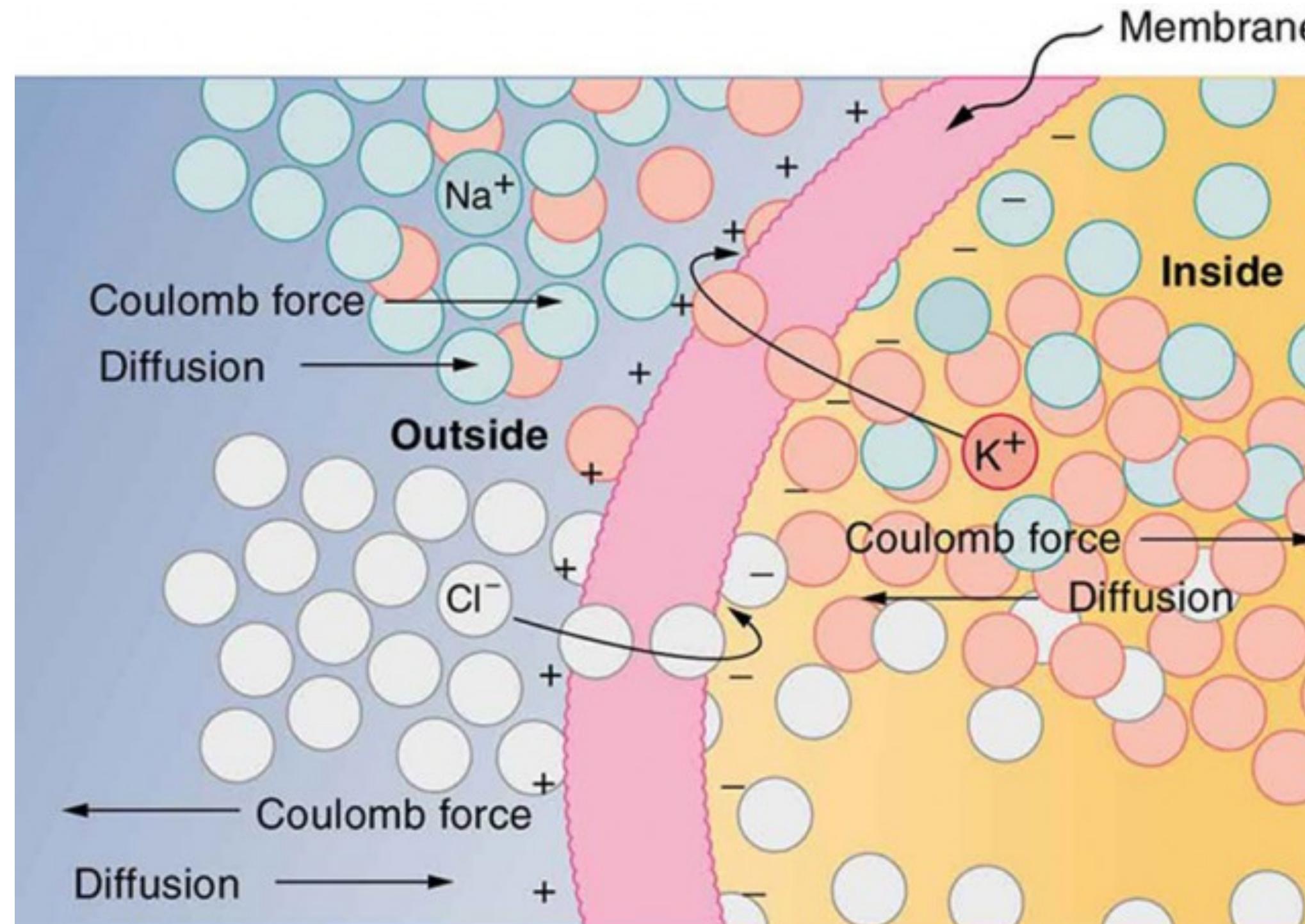
EMI/cell-by-cell (< mm, ~ 1000 cells in 0.1 mm^3)



$$\Omega = \Omega_i \cup \Omega_e \cup \Gamma$$

$$\Omega_i = \bigcup_{n=1}^N \Omega_n, \quad \Gamma = \bigcup_{n=1}^N \Gamma_n$$

Why consider ionic concentrations?



Let us consider only potentials for now...

[Rasmussen 2021]

Cell-by-cell framework (a.k.a. EMI): explicit cellular geometries

Find the intracellular and extracellular potentials $\phi_i = \phi_i(x, t)$ and $\phi_e = \phi_e(x, t)$, and the transmembrane current $I_M = I_M(x, t)$ s.t.:

$$-\nabla \cdot (\sigma_i \nabla \phi_i) = 0 \quad \text{in } \Omega_i, \quad (1)$$

$$-\nabla \cdot (\sigma_e \nabla \phi_e) = 0 \quad \text{in } \Omega_e, \quad (2)$$

$$\phi_M = \phi_i - \phi_e \quad \text{at } \Gamma, \quad (3)$$

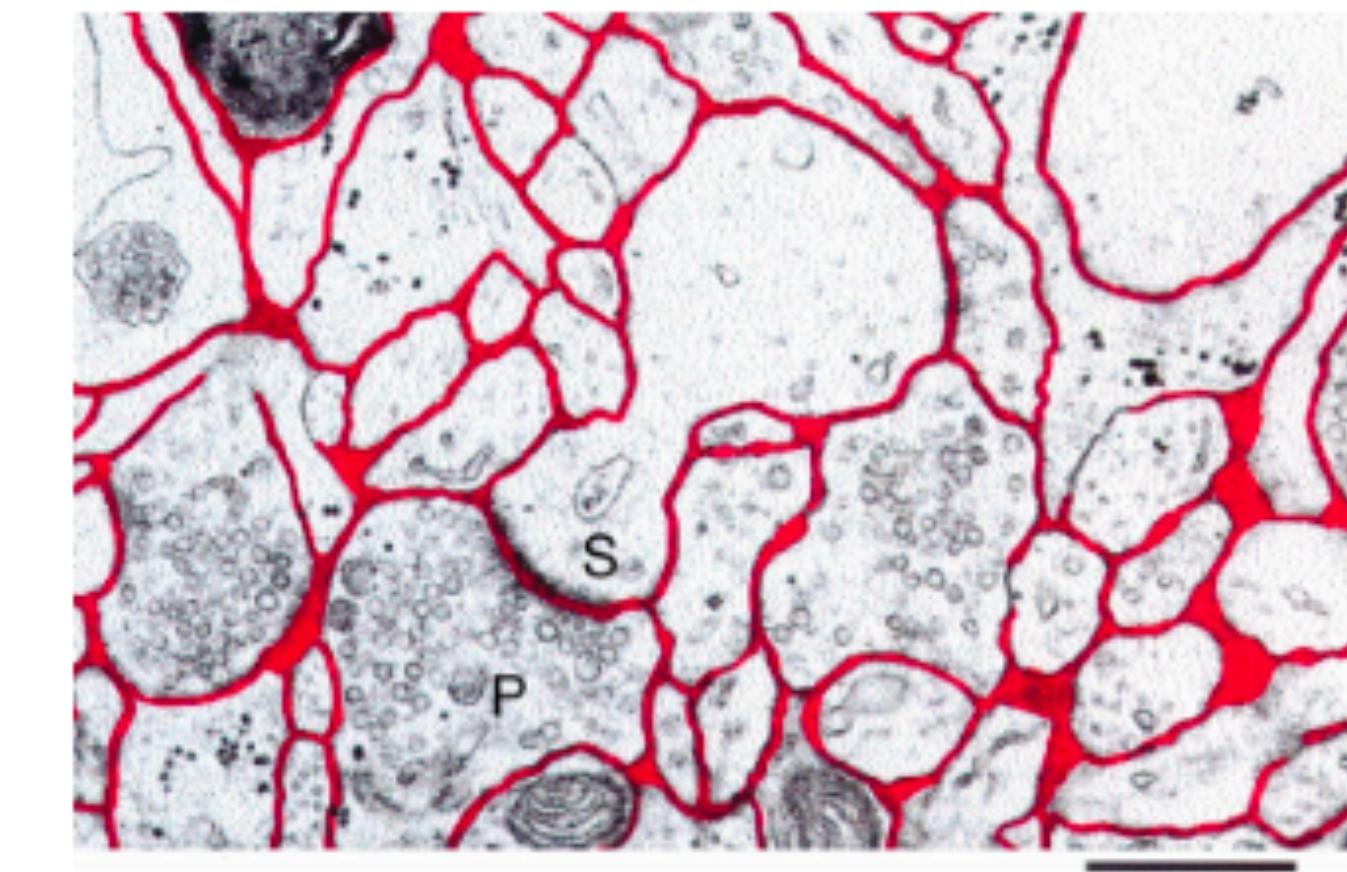
$$\sigma_e \nabla \phi_e \cdot n_e = -\sigma_i \nabla \phi_i \cdot n_i = I_M \quad \text{at } \Gamma, \quad (4)$$

$$\frac{\partial \phi_M}{\partial t} = \frac{1}{C_M} (I_M - I_{\text{ion}}) \quad \text{at } \Gamma. \quad (5)$$

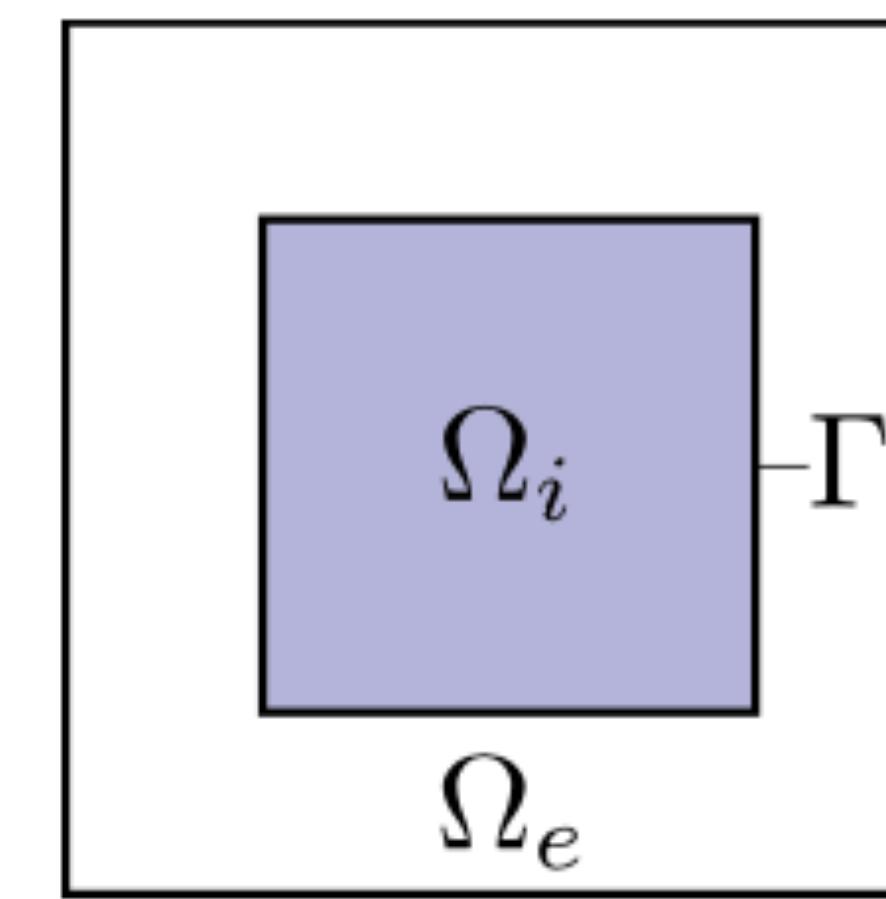
Ion concentrations are assumed to be constant in space and time – often an accurate approximation, but not always . . .

[Krassowska & Neu 1994],
Tveito et al. 2017

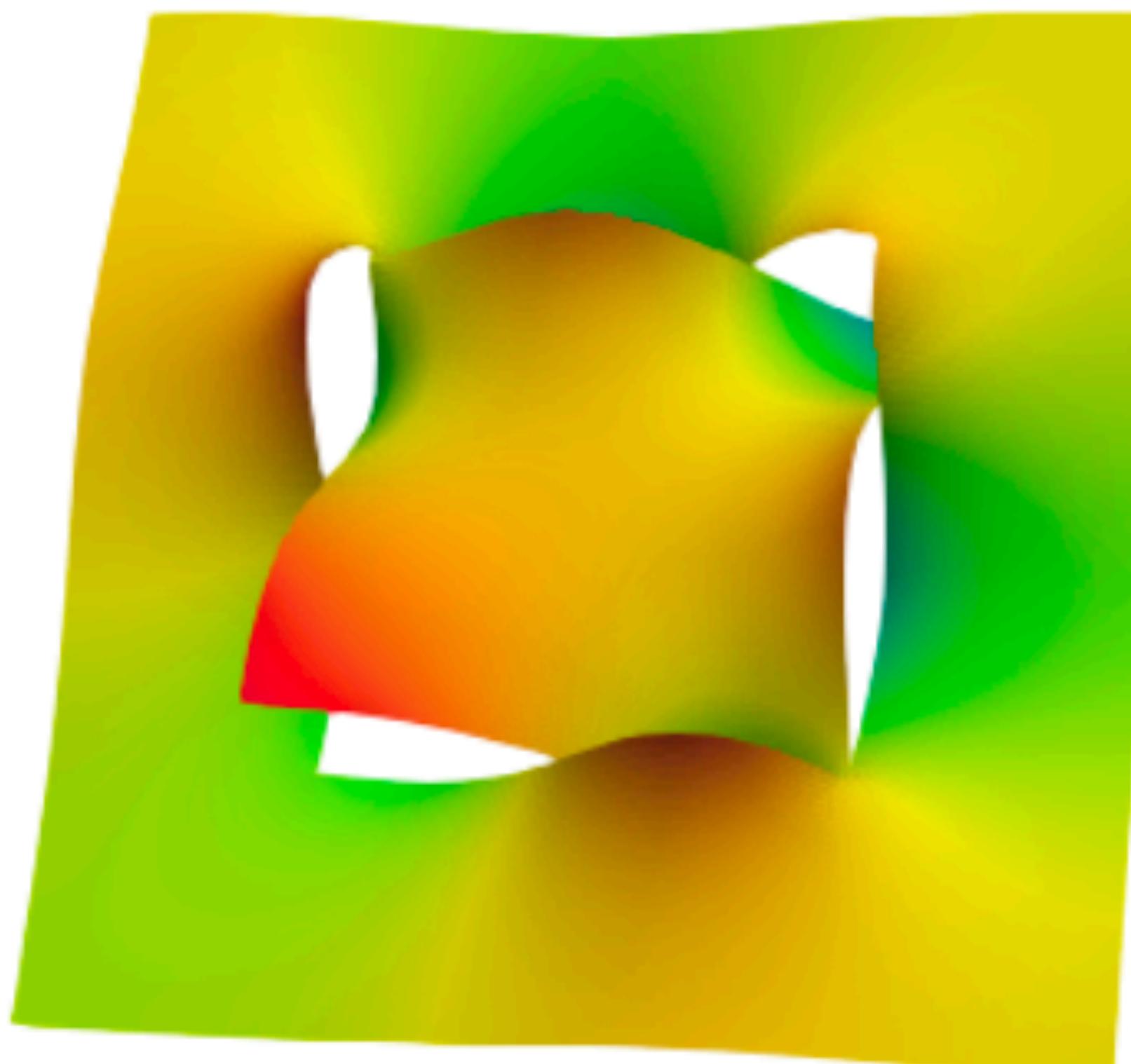
C_m	membrane capacitance
I_{ion}	sum of current density through membrane proteins
σ_i	intracellular conductivity
σ_e	extracellular conductivity



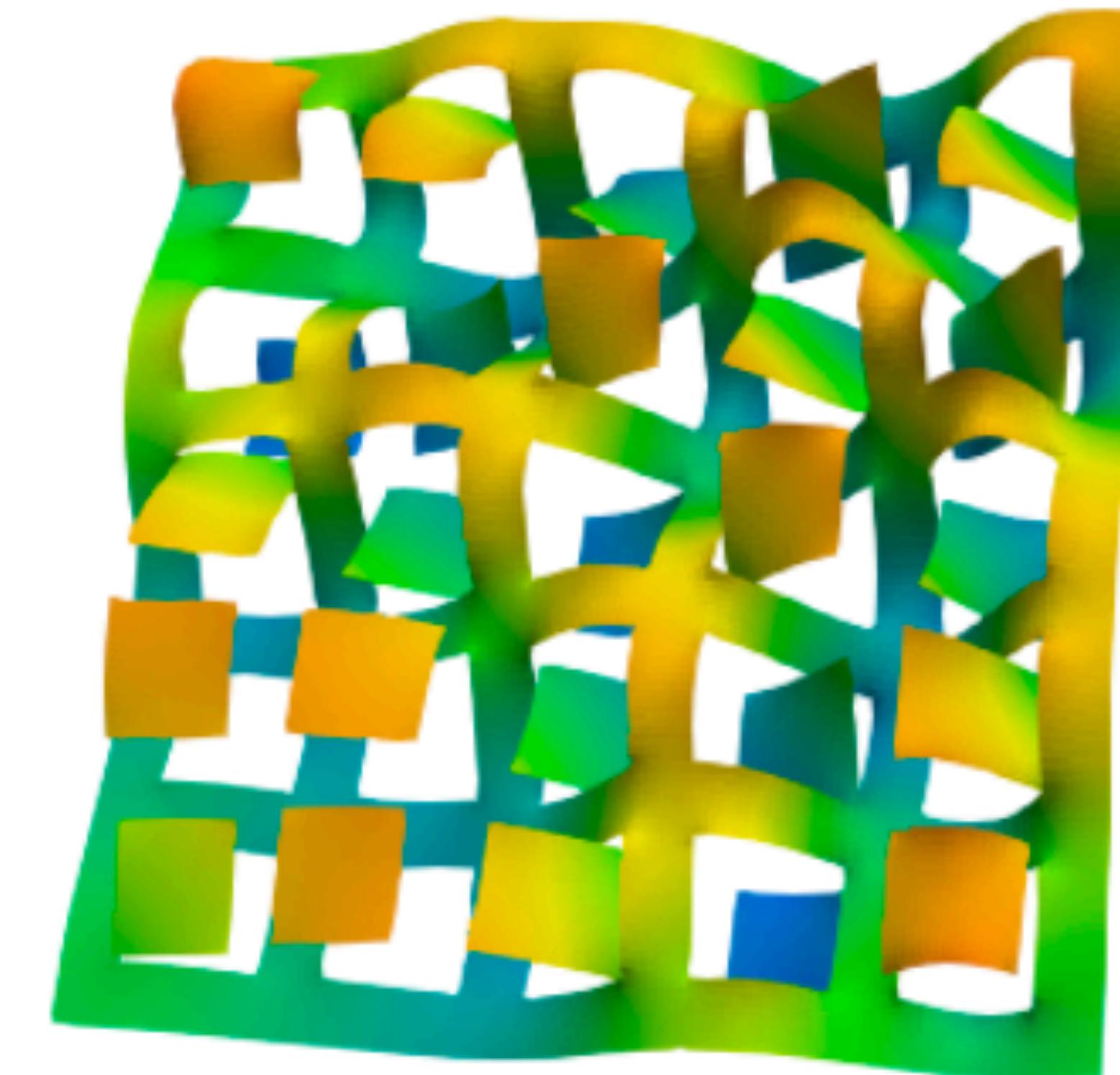
Rat cortex with ECS in red [Nicholson, 1998]



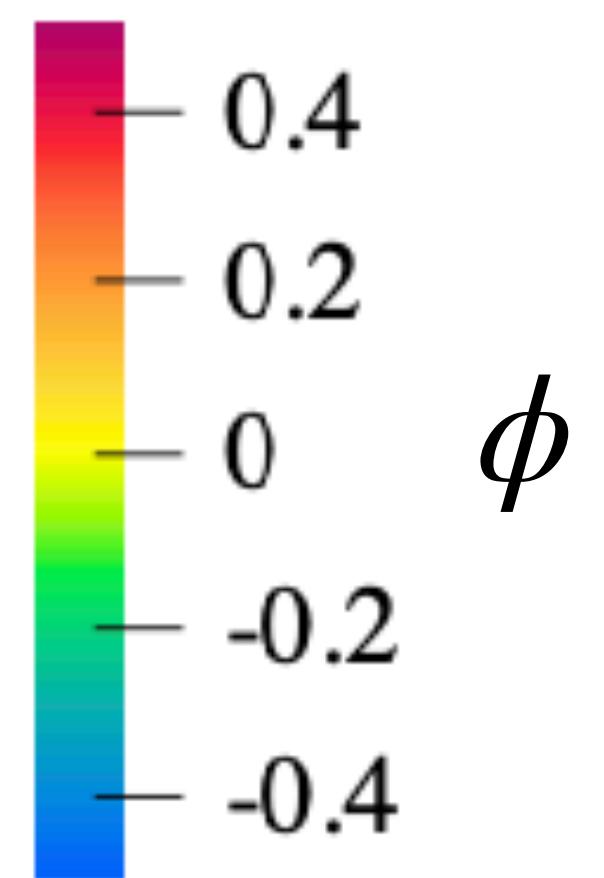
EMI 2D solutions with N cells



$N = 1$



$N = 25$



EMI equations derivation

The quasi-static approximation* Maxwell's equations, given an electric field \mathbf{E} a magneti field \mathbf{B} and a current density \mathbf{J} , reads:

$$\nabla \times \mathbf{E} = - \frac{\partial \mathbf{B}}{\partial t}, \quad \nabla \times \mathbf{B} = \mathbf{J} + \epsilon \frac{\partial \mathbf{E}}{\partial t}$$

$$\nabla \times \mathbf{E} = 0, \quad \nabla \times \mathbf{B} = \mathbf{J} = \sigma \mathbf{E} = - \sigma \nabla \phi$$

$$\nabla \times \mathbf{B} = - \sigma \nabla \phi$$

conductivity
(Ohm's law)

$$\nabla \cdot (\nabla \times \mathbf{B}) = - \nabla \cdot (\sigma \nabla \phi)$$

*Assuming that free unbalanced charges are instantly balanced. The assumption holds in the intracellular and extracellular spaces.

EMI equations derivation

$$\nabla \cdot (\sigma \nabla \phi) = 0$$



$$\nabla \cdot (\sigma_i \nabla \phi_i) = 0$$

$$\nabla \cdot (\sigma_e \nabla \phi_e) = 0$$

If we set conductivities σ_i, σ_e to be constant, we have two Poisson's equations:

$$\Delta \phi_i = 0, \quad \text{in } \Omega_i$$

$$\Delta \phi_e = 0, \quad \text{in } \Omega_e$$

EMI equations derivation: membrane terms

Given the total membrane current $I_M : \Gamma \rightarrow \mathbb{R}$:

$$I_M = I_{ion} + I_{cap} = I_{ion} + C_M \frac{\partial \phi_M}{\partial t},$$

with the ionic I_{ion} current given, e.g., by the Hodgkin–Huxley model.

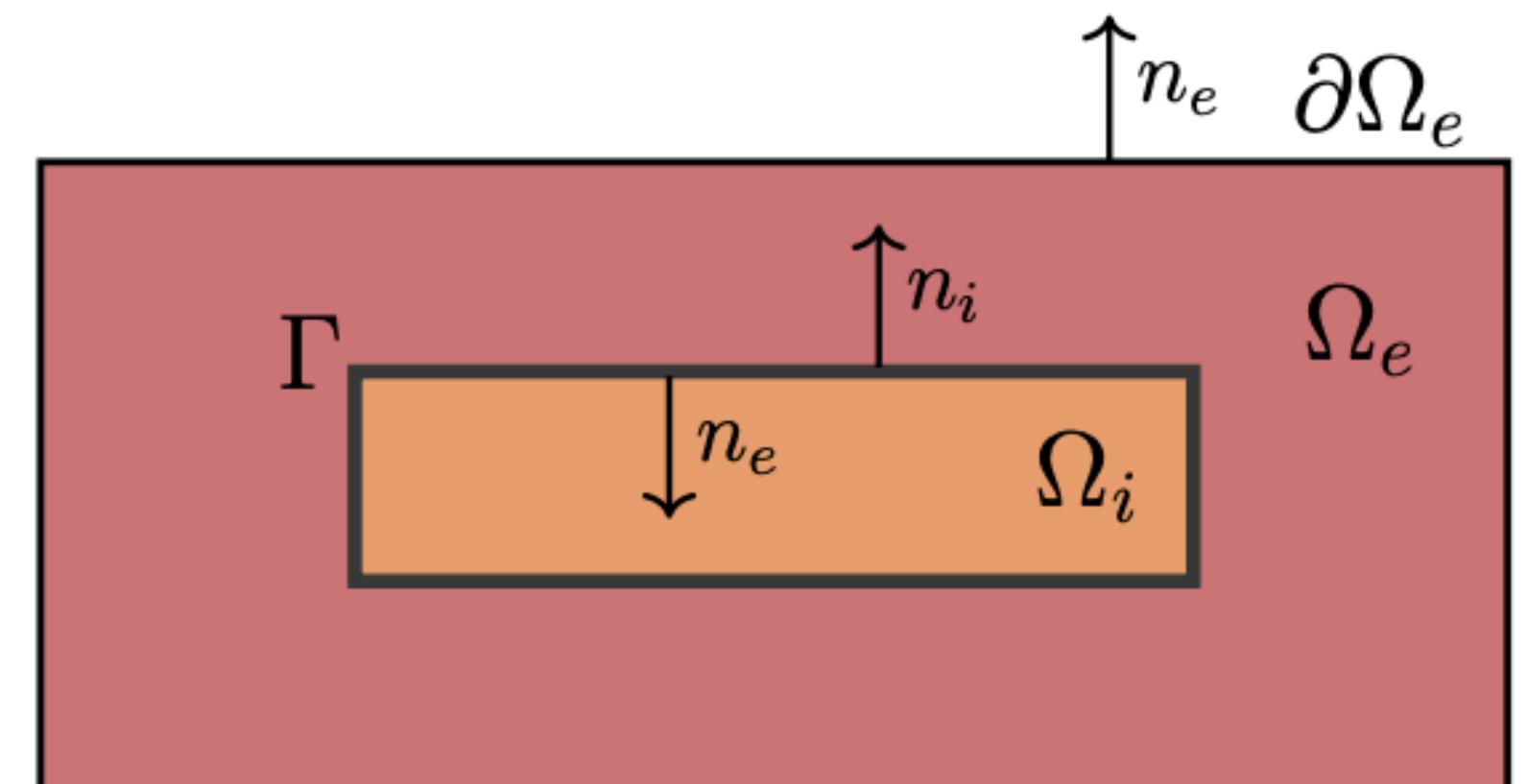
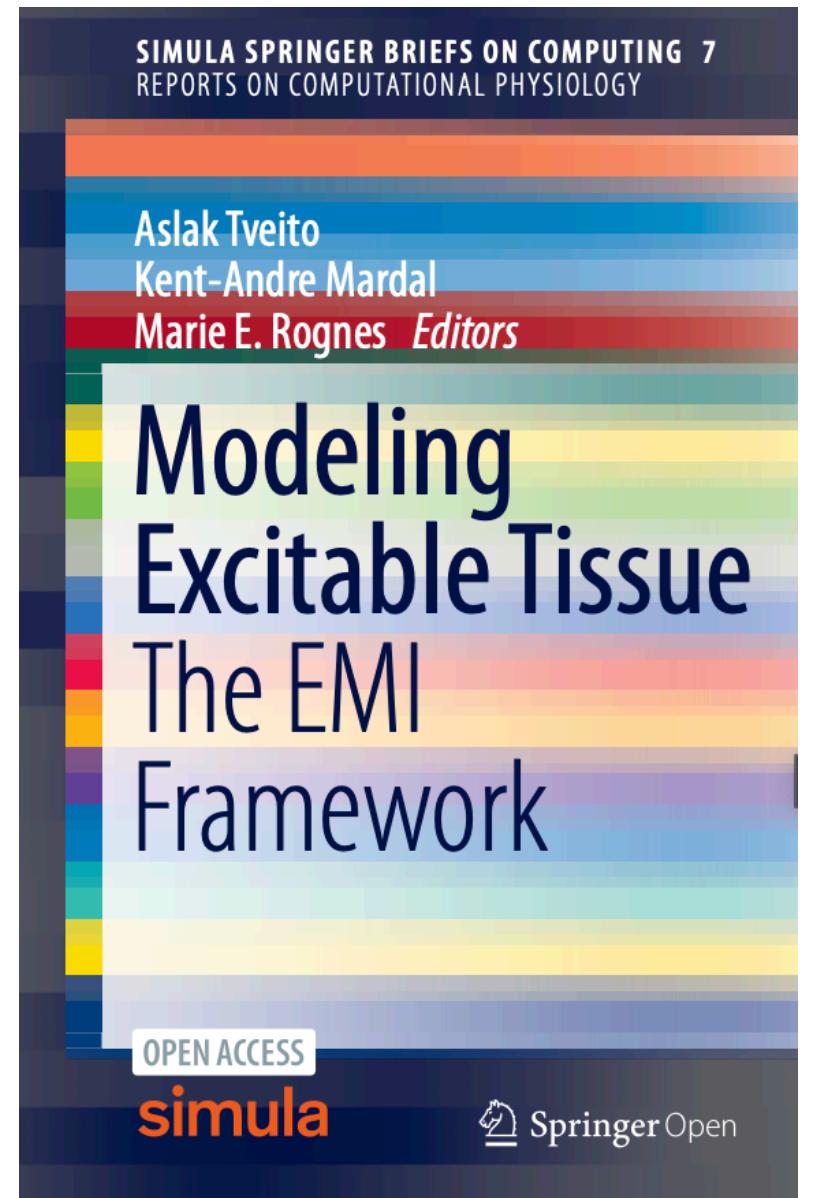
We also have for the outgoing vector \mathbf{n} in Γ :

$$\sigma_i \nabla \phi_i \cdot \mathbf{n} = I_m \quad \text{in } \Gamma,$$

$$\sigma_e \nabla \phi_e \cdot \mathbf{n} = I_m \quad \text{in } \Gamma.$$

Given the arbitrary choice $\mathbf{n} = \mathbf{n}_e = -\mathbf{n}_i$, we have in Γ

$$\sigma_e \nabla \phi_e \cdot \mathbf{n}_e = -\sigma_i \nabla \phi_i \cdot \mathbf{n}_i = I_m.$$



EMI model

$$\begin{aligned} -\nabla \cdot (\sigma_i \nabla \phi_i) &= 0 && \text{in } \Omega_i, \\ -\nabla \cdot (\sigma_e \nabla \phi_e) &= 0 && \text{in } \Omega_e, \\ \phi_M &= \phi_i - \phi_e && \text{at } \Gamma, \\ \sigma_e \nabla \phi_e \cdot \mathbf{n}_e &= -\sigma_i \nabla \phi_i \cdot \mathbf{n}_i = I_M && \text{at } \Gamma, \\ \frac{\partial \phi_M}{\partial t} &= \frac{1}{C_M} (I_M - I_{\text{ion}}) && \text{at } \Gamma. \end{aligned}$$

Is this differential problem well posed?

No, add zero Neumann boundary condition (natural): $\nabla \phi_e \cdot \mathbf{n}_e = 0$ in $\partial\Omega$ and an initial condition for the ODE. Now?

No, if (ϕ_e, ϕ_i) is a solution also $(\phi_e + C, \phi_i + C)$ is valid, given a constant C .

EMI model: enforce uniqueness

- Integral constrain

$$\int_{\Omega_e} \phi_e \, d\mathbf{x} = 0 \quad \xrightarrow{\text{red arrow}} \quad \text{Lagrange multiplier}$$

- Point-wise constrain:

$$\phi_e(\mathbf{x}_c) = 0, \quad \mathbf{x}_c \in \Omega_e$$

- Deal with the singularity on the discrete algebra level

$$A\mathbf{x} = \mathbf{b}, \quad \dim(\ker(A)) = 1, \quad \ker(A) = \{\mathbf{1}\}$$

KNP-EMI model: including ionic concentrations

For each ion species $k \in K$ and $r \in \{i, e\}$ find the *ion concentrations* $\mathbf{c}_r^k : \Omega_r \times (0, T] \rightarrow \mathbb{R}$ and the *potentials* $\phi_r : \Omega_r \times (0, T] \rightarrow \mathbb{R}$ such that:

$$\frac{\partial \mathbf{c}_r^k}{\partial t} + \nabla \cdot \mathbf{J}_r^k = 0 \quad \text{in } \Omega_r, \tag{6}$$

$$F \sum_k z^k \nabla \cdot \mathbf{J}_r^k = 0 \quad \text{in } \Omega_r, \tag{7}$$

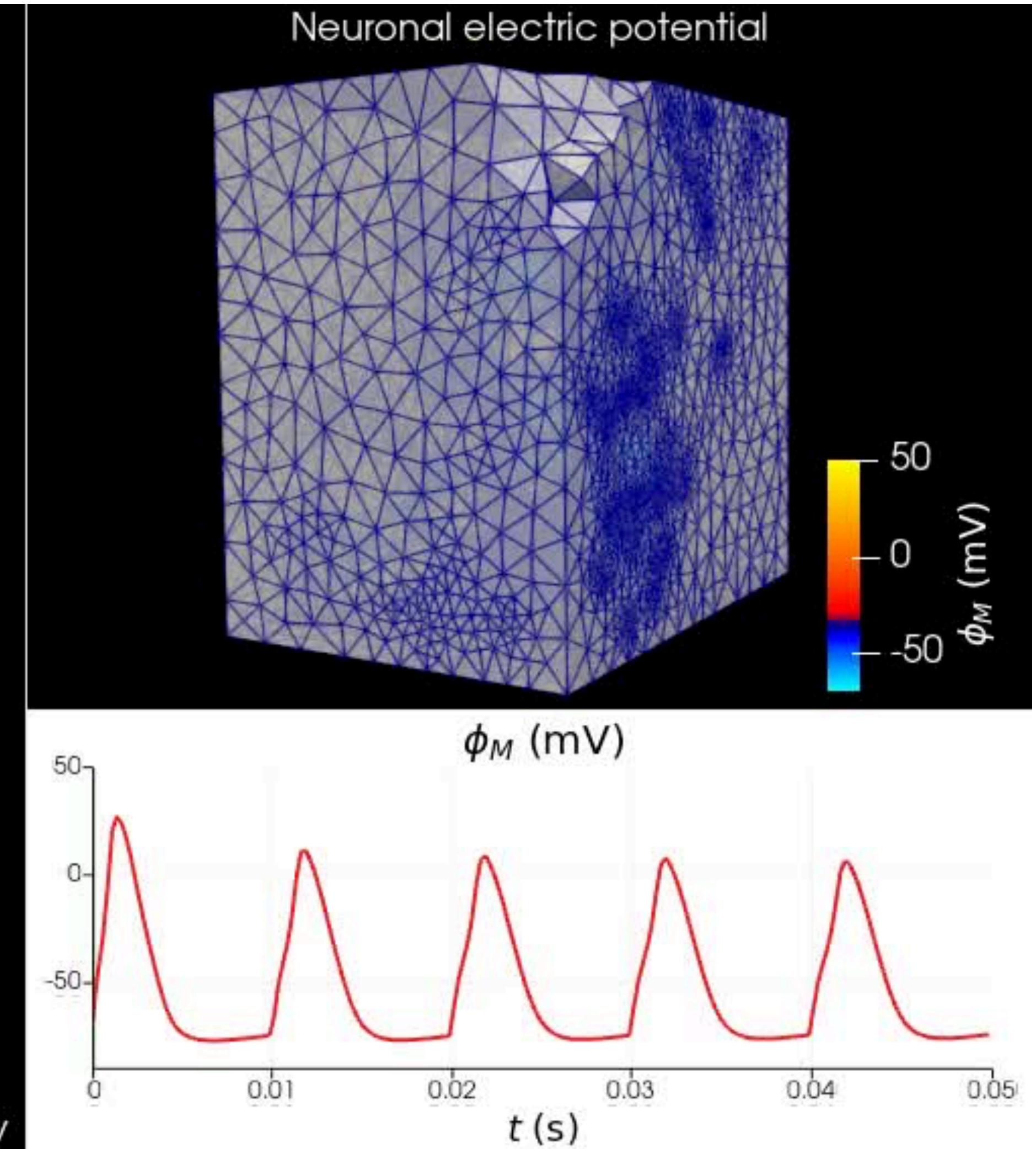
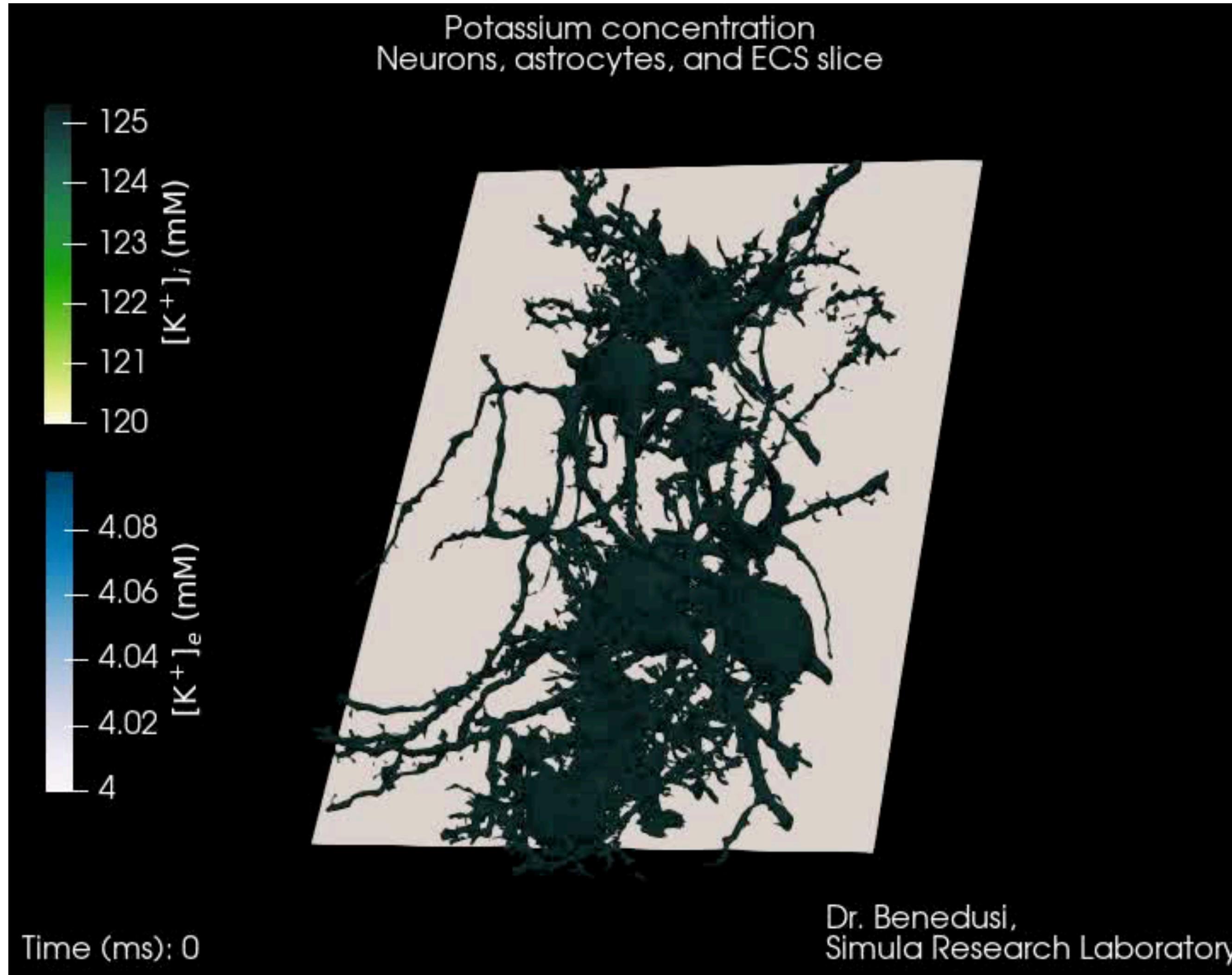
where the **ion flux densities** are given by:

$$\mathbf{J}_r^k = -D_r^k \nabla \mathbf{c}_r^k - D_r^k z^k \psi^{-1} \mathbf{c}_r^k \nabla \phi_r, \quad \text{in } \Omega_r. \tag{8}$$

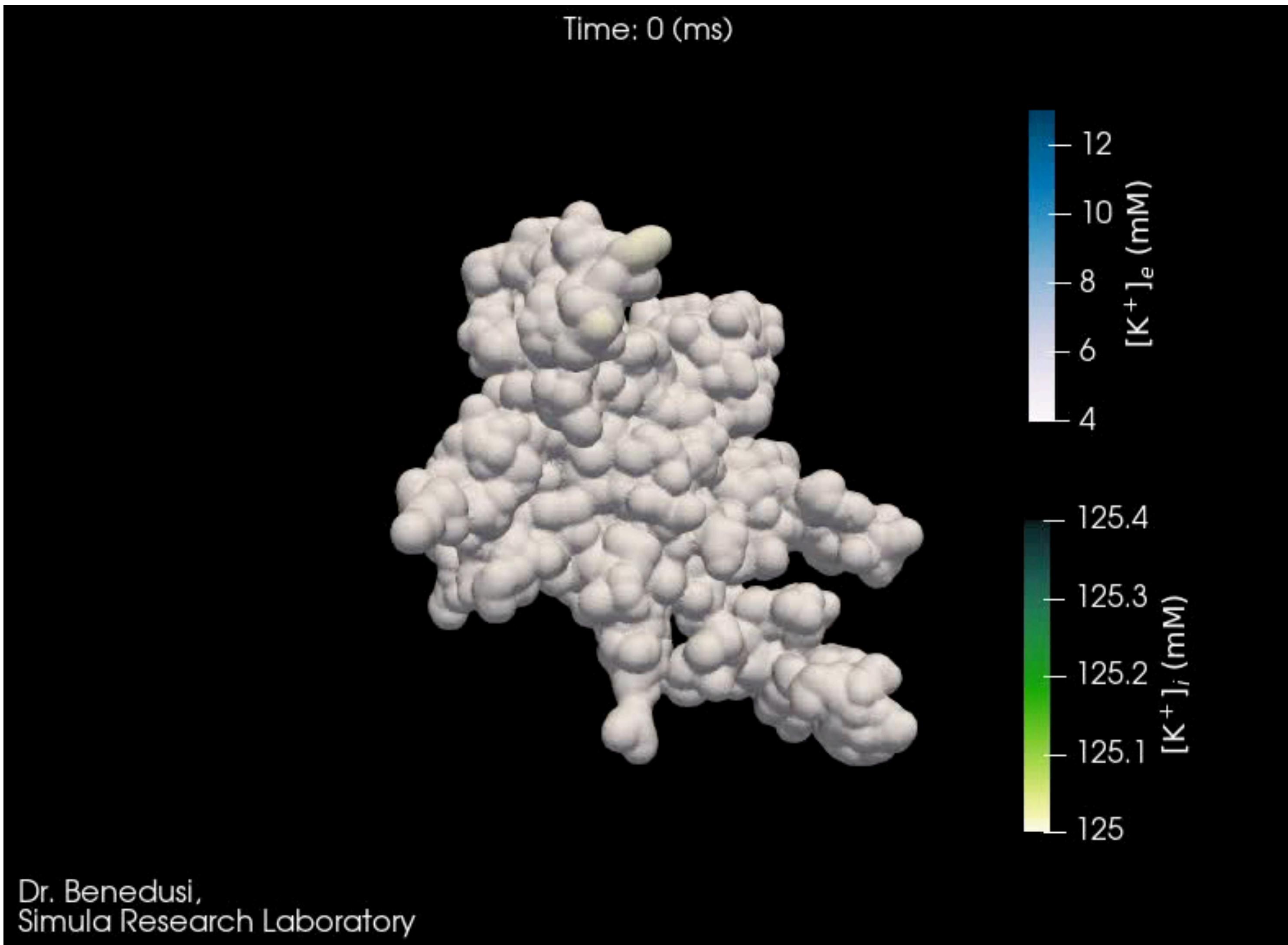
The system remains to be closed by appropriate initial conditions, boundary conditions, and importantly **interface conditions**.

$$I_M = F \sum_{k \in K} z_k \mathbf{J}_i^k \cdot \mathbf{n}_i = -F \sum_{k \in K} z_k \mathbf{J}_e^k \cdot \mathbf{n}_e = \sum_{k \in K} I_{ch}^k(\mathbf{x}, t) + C_m \partial_t \phi_M(\mathbf{x}, t), \quad \text{for } \mathbf{x} \in \Gamma, \tag{9}$$

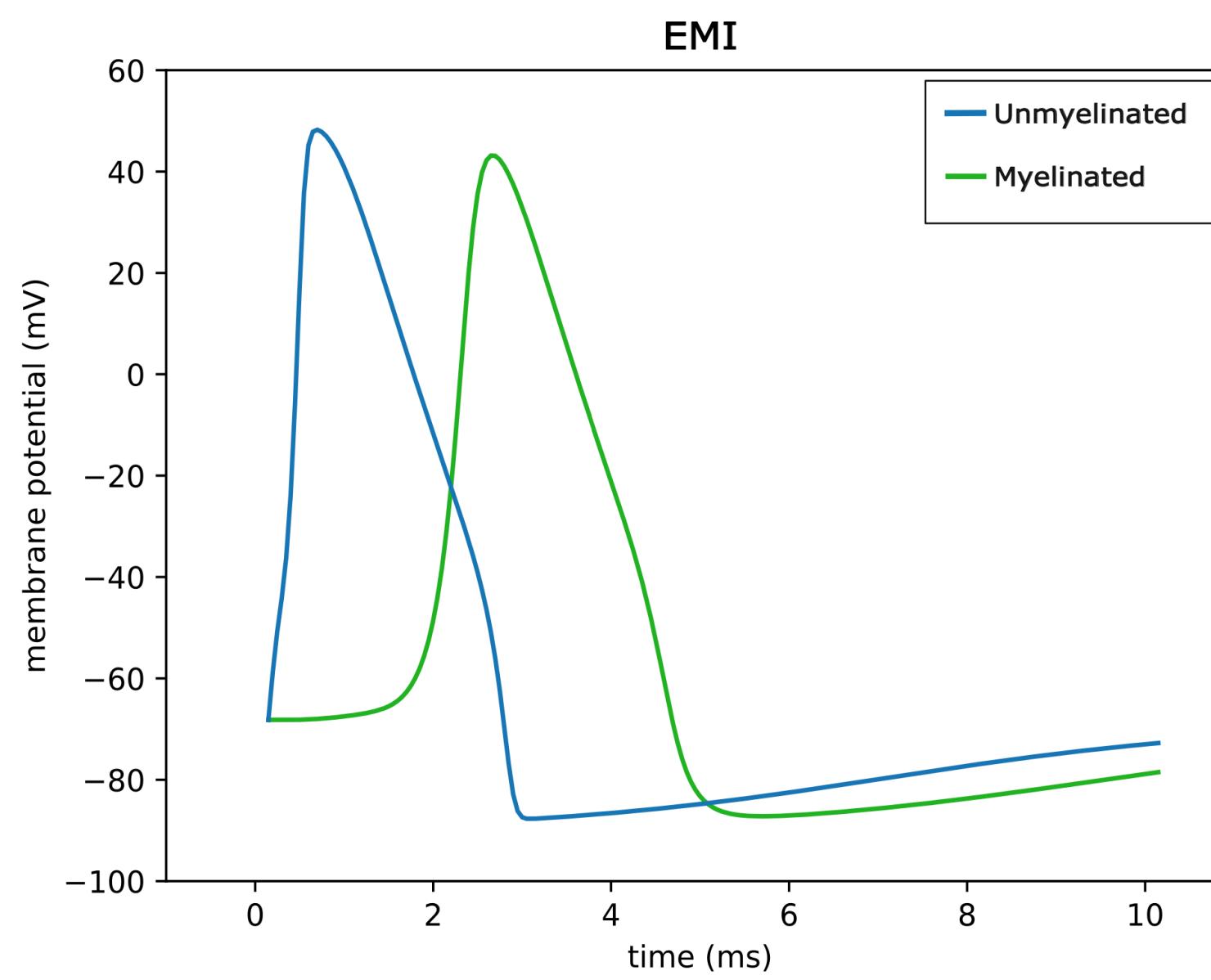
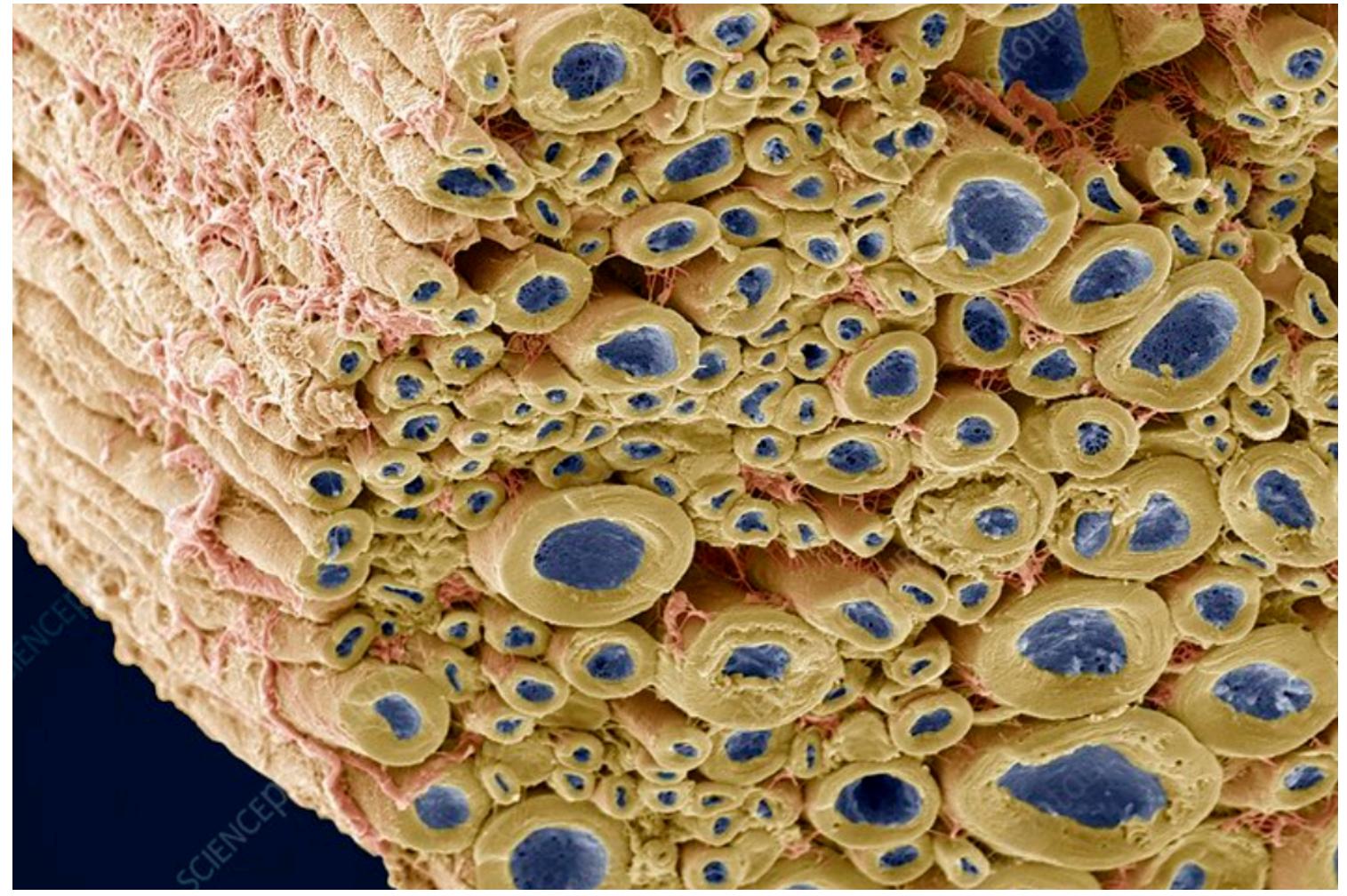
KNP-EMI model (10 M DoFs)



KNP-EMI model

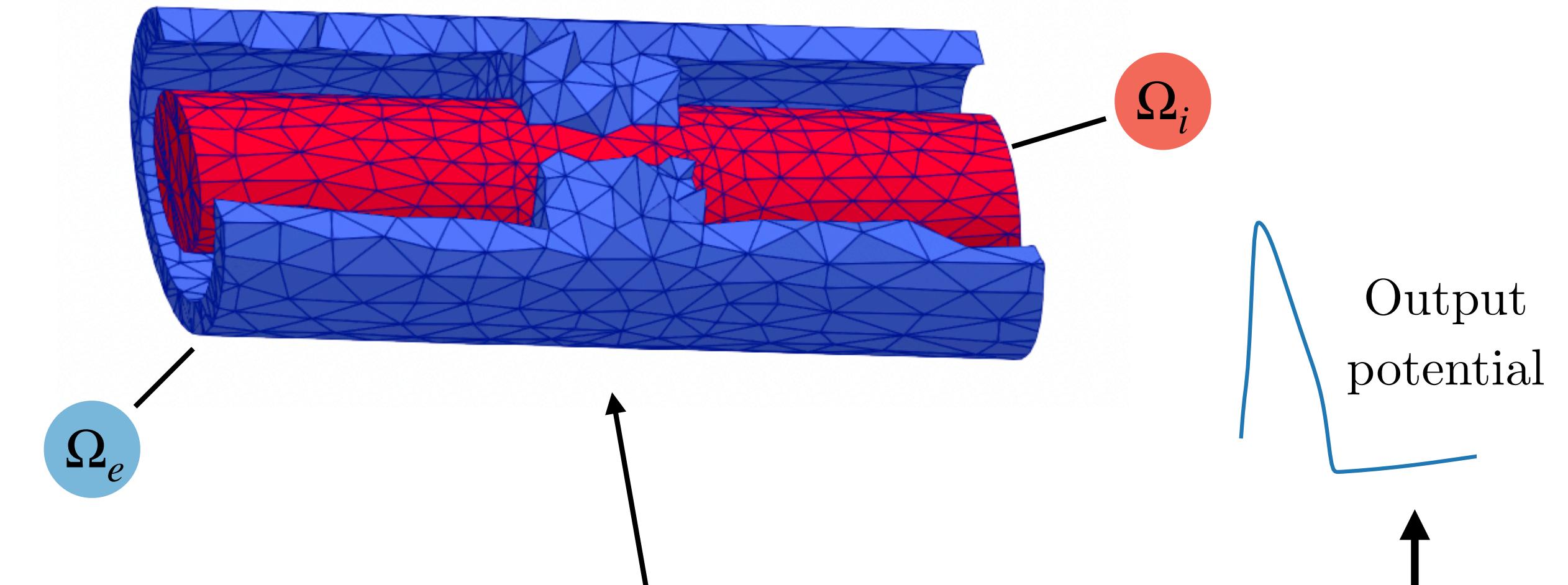


KNP-EMI model: myelination

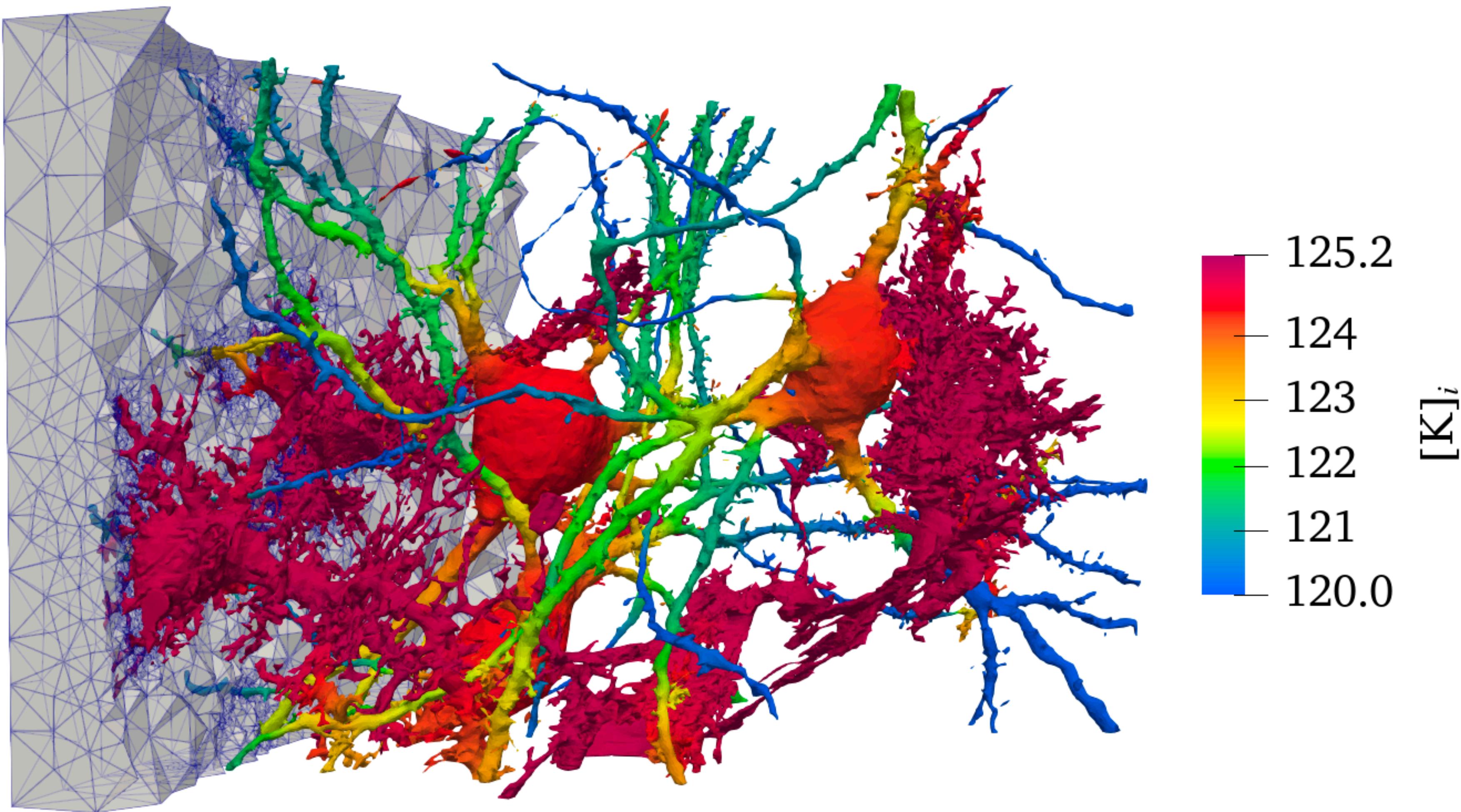


● Ranvier node

Input stimulus

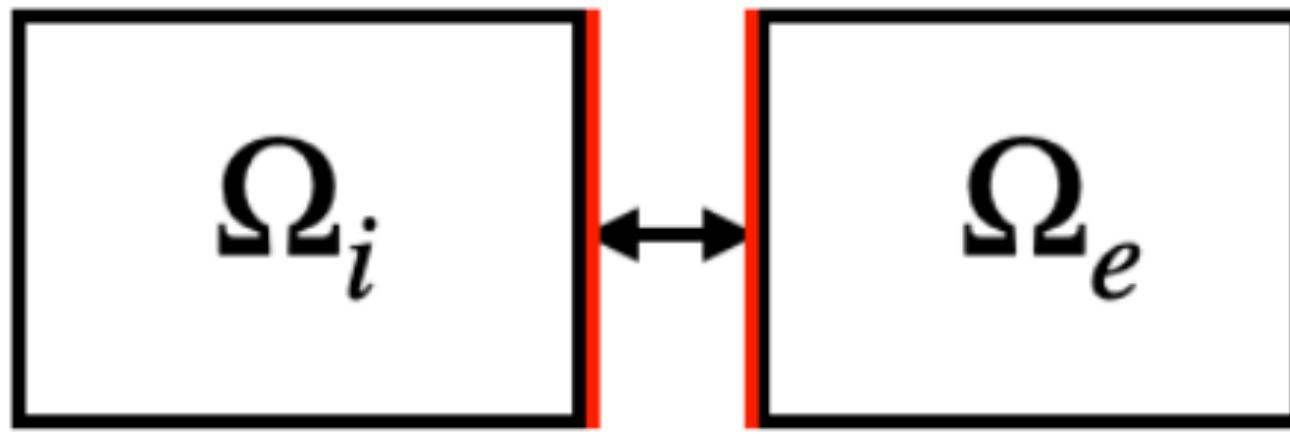


EMI modelling: numerics

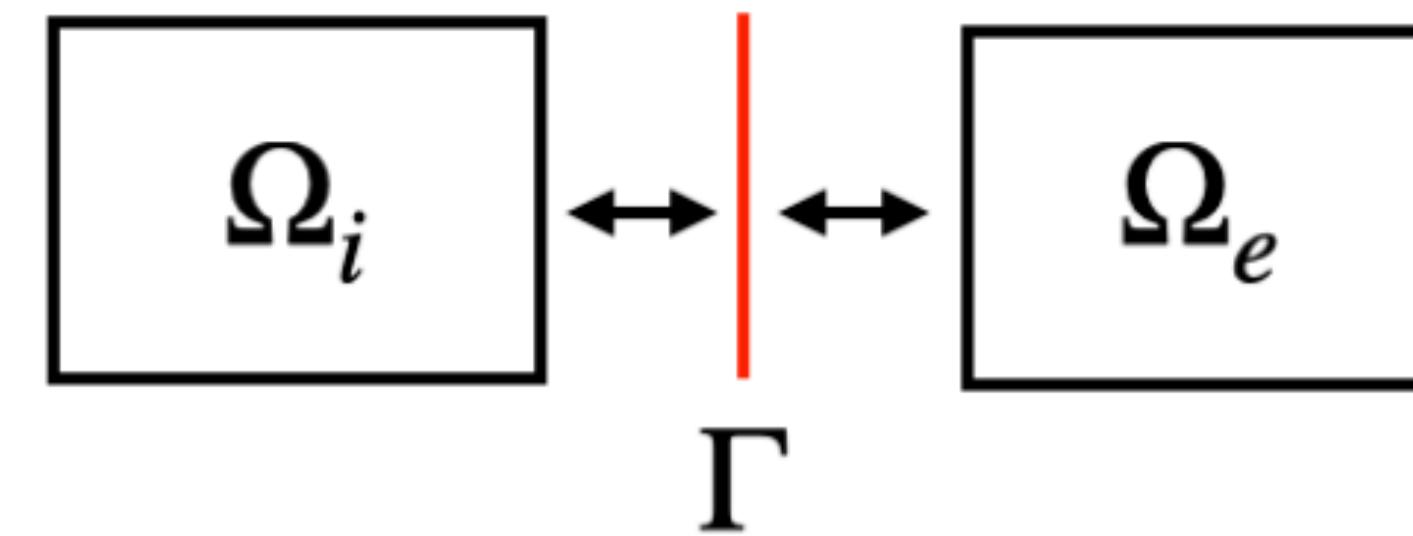


Mixed dimensional problem

Single-dimensional form



Multi-dimensional form



$$\begin{bmatrix} -\Delta_i & \textcolor{red}{T_{ie}} \\ \textcolor{red}{T_{ie}'} & -\Delta_e \end{bmatrix} \begin{bmatrix} \phi_i \\ \phi_e \end{bmatrix} = \begin{bmatrix} f_i \\ f_e \end{bmatrix}$$

$$\begin{bmatrix} -\Delta_i & 0 & \textcolor{red}{T_i} \\ 0 & -\Delta_e & \textcolor{red}{T_e} \\ \textcolor{red}{T_i}' & \textcolor{red}{T_e}' & -I \end{bmatrix} \begin{bmatrix} \phi_i \\ \phi_e \\ I_m \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ f \end{bmatrix}$$

Efficient and robust AMG-CG

Discretization: time

Given the time dependent membrane ODE in Γ :

$$\frac{\partial \phi_M}{\partial t} = \frac{1}{C_M}(I_M - I_{ion}),$$

and a discretization of the time axes $[0, T]$

$$0 = t_0 < t_1 < t_2 < \dots < t_N = T, \quad \text{with} \quad \Delta t = t_n - t_{n-1}, \quad n = 1, \dots, N$$

We consider a first order finite difference stencil for $\partial_t \phi_M$ and an implicit-explicit treatment of the currents. With notation $X^{[n]} = X(t_n)$:

$$\frac{\phi_M^{[n]} - \phi_M^{[n-1]}}{\Delta t} = \frac{1}{C_M}(I_M^{[n]} - I_{ion}^{[n-1]}),$$

Discretization: time

Given:

$$\frac{\phi_M^{[n]} - \phi_M^{[n-1]}}{\Delta t} = \frac{1}{C_M} (I_M^{[n]} - I_{ion}^{[n-1]}),$$

we can write an expression for the total membrane current at time step t_n

$$I_M^{[n]} = \frac{C_M}{\Delta t} \phi_M^{[n]} - \frac{C_M}{\Delta t} \phi_M^{[n-1]} + I_{ion}^{[n-1]},$$

$$I_M^{[n]} = \frac{C_M}{\Delta t} \phi_M^{[n]} - f^{[n-1]}, \quad \text{or simply} \quad I_M = \frac{C_M}{\Delta t} \phi_M - f_{\text{prev}}$$

Weak form

Given the bulk equation in Ω_e (similarly for Ω_i)

$$\nabla \cdot (\sigma_e \nabla \phi_e) = 0,$$

we first consider its variational (or weak) formulation. Given a sufficient regular Hilbert space V_e , and test functions $v_e \in V_e(\Omega_e)$, we multiply the strong form by v_e and integrate over Ω_e :

$$\int_{\Omega_e} \nabla \cdot (\sigma_e \nabla \phi_e) v_e \, d\mathbf{x} = 0.$$

Integrating by parts yields to

$$\int_{\Omega_e} \sigma_e \nabla \phi_e \nabla v_e \, d\mathbf{x} - \int_{\partial\Omega_e} \sigma_e \nabla \phi_e \cdot \mathbf{n}_e v_e \, d\mathbf{x} = 0,$$

for $\partial\Omega_e = \Gamma \cup \partial\Omega$, we have

$$\int_{\Omega_e} \sigma_e \nabla \phi_e \nabla v_e \, d\mathbf{x} - \int_{\Gamma} \sigma_e \nabla \phi_e \cdot \mathbf{n}_e v_e \, d\mathbf{x} - \int_{\partial\Omega} \sigma_e \nabla \phi_e \cdot \mathbf{n}_e v_e \, d\mathbf{x} = 0.$$

Weak form

$$\int_{\Omega_e} \sigma_e \nabla \phi_e \nabla v_e \, d\mathbf{x} - \int_{\Gamma} \sigma_e \nabla \phi_e \cdot \mathbf{n}_e v_e \, d\mathbf{x} - \int_{\partial\Omega} \sigma_e \nabla \phi_e \cdot \mathbf{n}_e v_e \, d\mathbf{x} = 0 \quad (1)$$

We recall the boundary and membrane conditions:

$$\nabla \phi_e \cdot \mathbf{n}_e = 0 \quad \text{in } \partial\Omega ,$$

$$\sigma_e \nabla \phi_e \cdot \mathbf{n}_e = I_m, \quad \text{in } \Gamma$$

Equation (1) simply becomes:

$$\int_{\Omega_e} \sigma_e \nabla \phi_e \nabla v_e \, d\mathbf{x} - \int_{\Gamma} I_m v_e \, d\mathbf{x} = 0$$

Weak form

Given:

$$\int_{\Omega_e} \sigma_e \nabla \phi_e \nabla v_e \, d\mathbf{x} - \int_{\Gamma} I_M v_e \, d\mathbf{x} = 0$$

Recalling $I_M = C_M \Delta t^{-1} \phi_M - f_{\text{prev}}$, we finally have

$$\int_{\Omega_e} \sigma_e \nabla \phi_e \nabla v_e \, d\mathbf{x} - \frac{C_M}{\Delta t} \int_{\Gamma} \phi_M v_e \, d\mathbf{x} = - \frac{C_M}{\Delta t} \int_{\Gamma} f_{\text{prev}} v_e \, d\mathbf{x}.$$

Since $\phi_M = \phi_i - \phi_e$, we have

$$\int_{\Omega_e} \sigma_e \nabla \phi_e \nabla v_e \, d\mathbf{x} + \frac{C_M}{\Delta t} \int_{\Gamma} \phi_e v_e \, d\mathbf{x} - \frac{C_M}{\Delta t} \int_{\Gamma} \phi_i v_e \, d\mathbf{x} = - \frac{C_M}{\Delta t} \int_{\Gamma} f_{\text{prev}} v_e \, d\mathbf{x}$$

Variational problem

Given appropriate Hilbert spaces $V_e(\Omega_e), V_e(\Omega_i)$, find $\phi_i \in V_i, \phi_e \in V_e$ satisfying

$$\int_{\Omega_e} \sigma_e \nabla \phi_e \nabla v_e \, d\mathbf{x} + \frac{C_M}{\Delta t} \int_{\Gamma} \phi_e v_e \, d\mathbf{x} - \frac{C_M}{\Delta t} \int_{\Gamma} \phi_i v_e \, d\mathbf{x} = - \frac{C_M}{\Delta t} \int_{\Gamma} f_{\text{prev}} v_e \, d\mathbf{x},$$

$$\int_{\Omega_i} \sigma_i \nabla \phi_i \nabla v_i \, d\mathbf{x} + \frac{C_M}{\Delta t} \int_{\Gamma} \phi_i v_i \, d\mathbf{x} - \frac{C_M}{\Delta t} \int_{\Gamma} \phi_e v_i \, d\mathbf{x} = \frac{C_M}{\Delta t} \int_{\Gamma} f_{\text{prev}} v_i \, d\mathbf{x}$$

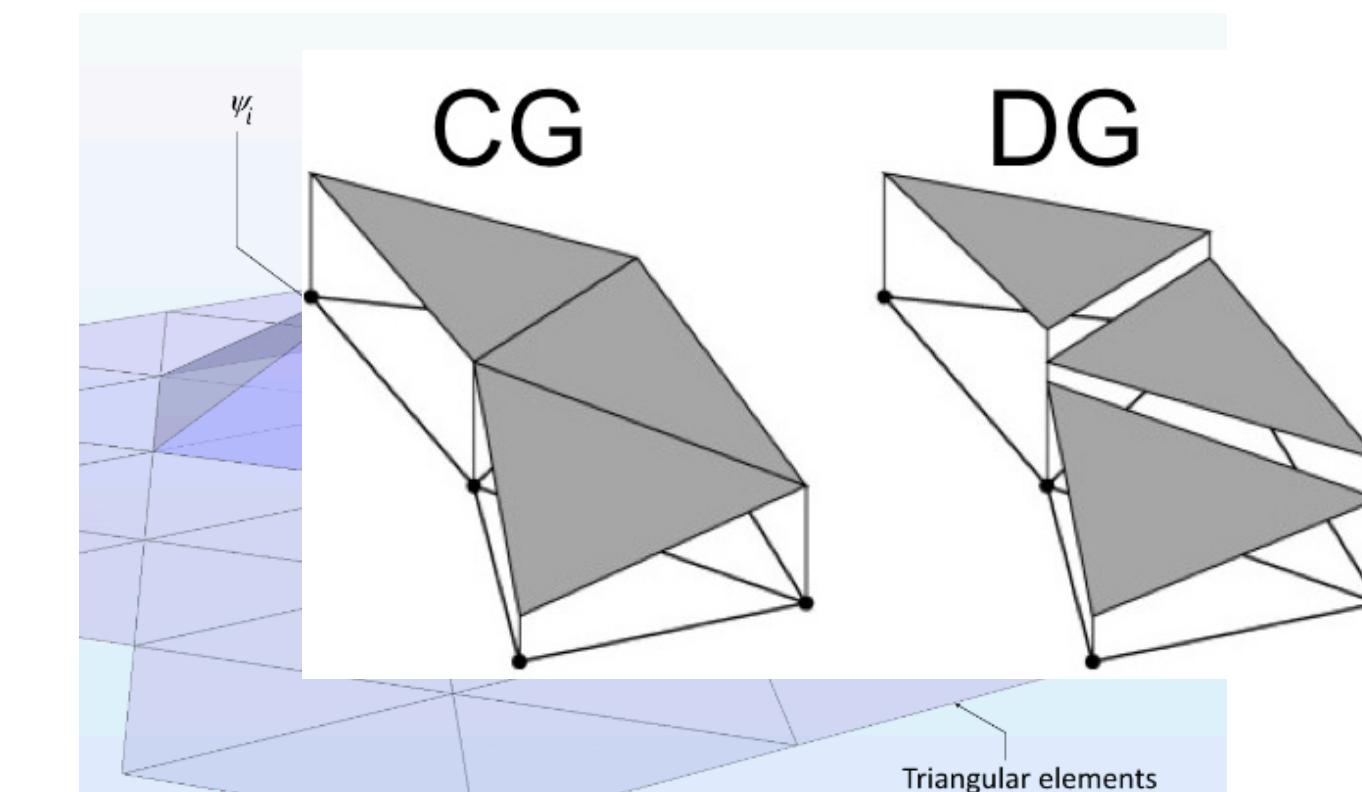
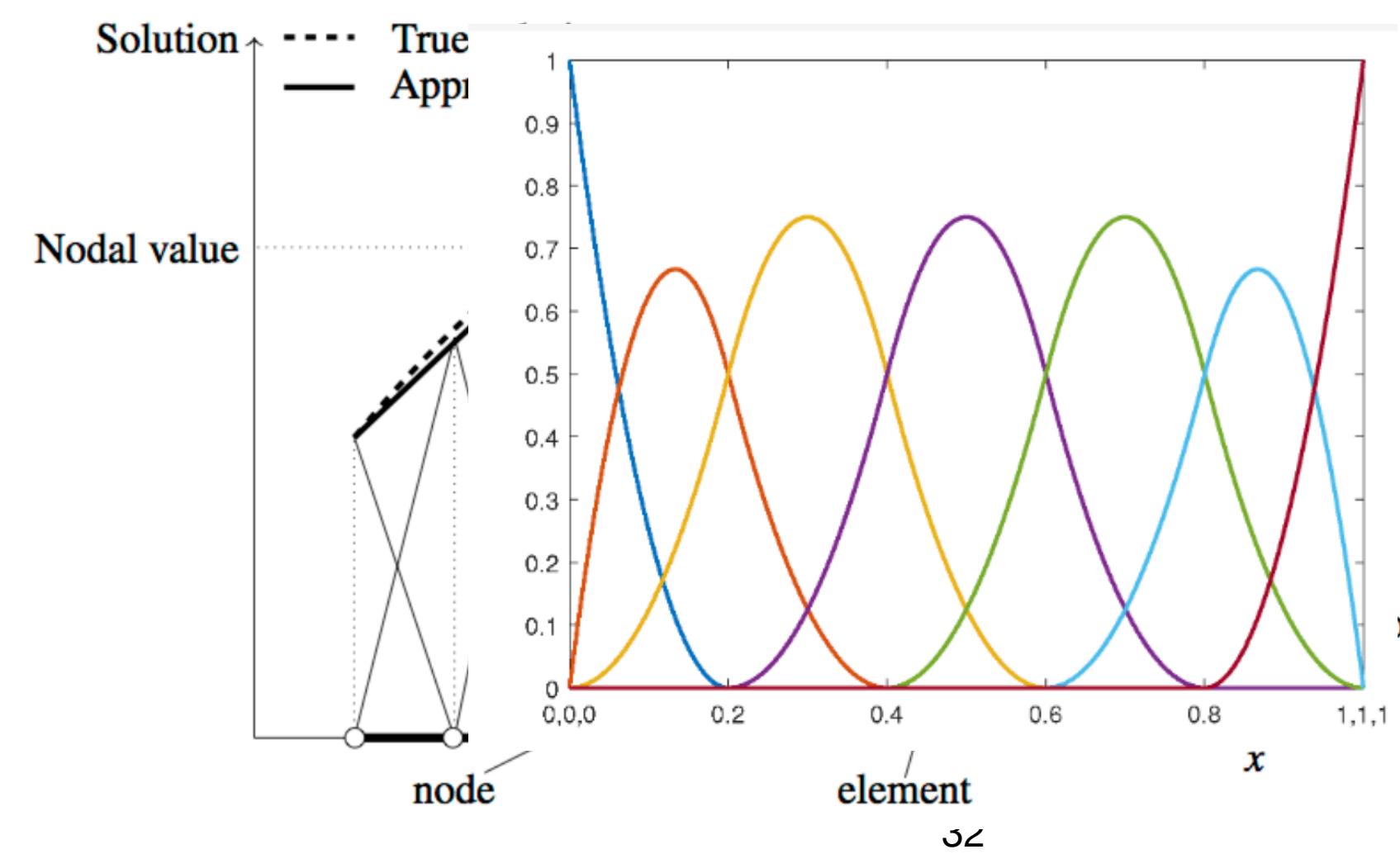
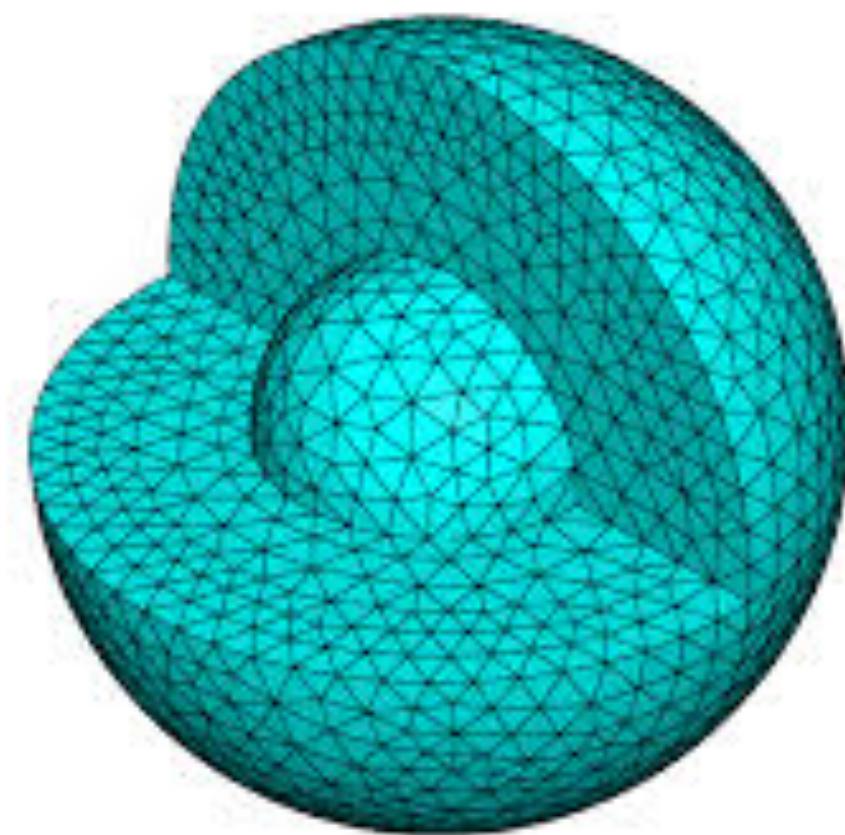
for all test functions $v_e \in V_e, v_i \in V_i$.

Finite Element spatial discretisation

The infinite dimensional spaces V_i, V_e are substituted by finite dimensional ones, e.g. with dimension N_h . For example, given arbitrary $V_h \subseteq V$

$$V_h = \text{span}\{\psi_n(\mathbf{x})\}_{n=1}^{N_h} \quad \text{and} \quad \phi \approx \phi_h = \sum_{n=1}^{N_h} \phi_n \psi_n(\mathbf{x}).$$

The discrete problem is now to find the vector of coefficients $[\phi_1, \phi_2, \dots, \phi_{N_h}]$, chosen V_h . How to choose V_h ? E.g. piecewise polynomials of order $p \in \{1, 2, 3, \dots\}$ and regularity $k \in \{-1, 0, 1, 2, 3, \dots, p - 1\}$, i.e. $\psi_j \in C^k(\Omega)$.



Finite Element discrete problem

Given appropriate finite element spaces $V_{e,h}(\Omega_e)$, find $\phi_e \in V_{e,h}$ satisfying

$$\int_{\Omega_e} \sigma_e \nabla \phi_e \nabla v_e \, d\mathbf{x} + \frac{C_M}{\Delta t} \int_{\Gamma} \phi_e v_e \, d\mathbf{x} - \frac{C_M}{\Delta t} \int_{\Gamma} \phi_i v_e \, d\mathbf{x} = - \frac{C_M}{\Delta t} \int_{\Gamma} f_{\text{prev}} v_e \, d\mathbf{x},$$

for each test functions $v_e \in \{\psi_{e,1}, \psi_{e,2}, \dots, \psi_{e,N_h}\}$. Algebraically:

$$(S + M_{\Gamma} - C_{\Gamma})\mathbf{x} = \mathbf{b},$$

with $\mathbf{x} = [\phi_1, \phi_2, \dots, \phi_{N_h}] \in \mathbb{R}^{N_h}$, or simply

$$A\mathbf{x} = \mathbf{b},$$

with $A \in \mathbb{R}^{N_h \times N_h}$, symmetric positive definite (SPD).

Linear solvers

Given the linear system

$$A\mathbf{x} = \mathbf{b},$$

with $A \in \mathbb{R}^{n \times n}$, there are two main solving approaches:

- **Direct solvers:**

LU decomposition, $O(n^3)$, flexible, high precision and fill-in (memory heavy)
Feasible for small/dense systems.

- **Iterative solvers:**

State-of-art for large sparse linear systems: **preconditioned Krylov** methods, examples of popular Krylov methods:

- Conjugate Gradient (CG), for SPD operators
- MinRES, for symmetric operators
- GMRES, BiCGStab

Complexity: $O(m \cdot n)$, with $m \in \mathbb{N}$ the number of iterations.

Linear solvers: Krylov method

Top Ten Algorithms in Science (Dongarra and Sullivan, 2000)

- ① Metropolis Algorithm (Monte Carlo method)
 - ② Simplex Method for Linear Programming
 - ③ Krylov Subspace Iteration Methods
 - ④ The Decompositional Approach to Matrix Computations
 - ⑤ The Fortran Optimizing Compiler
 - ⑥ QR Algorithm for Computing Eigenvalues
 - ⑦ Quicksort Algorithm for Sorting
 - ⑧ Fast Fourier Transform
 - ⑨ Integer Relation Detection Algorithm
 - ⑩ Fast Multipole Method
- Red: Algorithms within the exclusive domain of NLA research.
 - Blue: Algorithms strongly (though not exclusively) connected to NLA research.

Linear solvers: Krylov method

Given the linear system $Ax = b$, with $A \in \mathbb{R}^{n \times n}$, we construct a Krylov subspace:

$$K_m(A, b) = \text{span}(b, Ab, A^2b, \dots, A^{m-1}b).$$

We define the minimal polynomial $q(t)$: the (monic) polynomial of minimal degree m such that

$$q(A) = 0,$$

and by the Cayley–Hamilton $m \leq n$. By definition, we have that

$$q(A) = \alpha_0I + \alpha_1A + \alpha_2A^2 + \dots + \alpha_mA^m = 0,$$

for some real numbers $\alpha_0, \alpha_1, \dots, \alpha_m$. Therefore:

$$\alpha_0A^{-1} + \alpha_1I + \alpha_2A^1 + \dots + \alpha_mA^{m-1} = 0,$$

$$A^{-1} = -\alpha_0^{-1} \sum_{j=0}^{m-1} \alpha_j A^j.$$

Linear solvers: Krylov method

For $A\mathbf{x} = \mathbf{b}$, with $A \in \mathbb{R}^{n \times n}$, and

$$K_m(A, \mathbf{b}) = \text{span}(\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, \dots, A^{m-1}\mathbf{b}),$$

given

$$A^{-1} = -\alpha_0^{-1} \sum_{j=0}^{m-1} \alpha_j A^j,$$

we have that

$$\mathbf{x} = A^{-1}\mathbf{b} \in K_m(A, \mathbf{b}).$$

In practice Krylov methods work well since:

$$m_* \ll m \ll n,$$

with m_* consider finite arithmetic.

[Ipsen, Meyer, The idea behind Krylov methods. *The American mathematical monthly*, 1998]

Linear solvers: preconditioning

Given the linear system

$$A\mathbf{x} = \mathbf{b},$$

with $A \in \mathbb{R}^{n \times n}$, we want to find a non-singular operator $P \in \mathbb{R}^{n \times n}$, which is a good and cheap approximation of A in the sense that:

$$\kappa(P^{-1}A) \approx 1,$$

with P^{-1} being cheap to compute and solve the equivalent system

$$P^{-1}A\mathbf{x} = P^{-1}\mathbf{b}.$$

Linear solvers: preconditioning

Given the linear system $A\mathbf{x} = \mathbf{b}$, and the equivalent one:

$$P^{-1}A\mathbf{x} = P^{-1}\mathbf{b}, \quad \text{minimize } \kappa(P^{-1}A) + \text{cost}(P^{-1})$$

$$\min \kappa(P^{-1}A)$$

$$P = A$$

$$P = \text{ILU}(A)$$

$$P = \text{tri}(A) \\ (\text{Gauss-Seidel})$$

$$P = \text{diag}(A) \\ (\text{Jacobi})$$

$$\min \text{cost}(P^{-1})$$

$$P = I$$

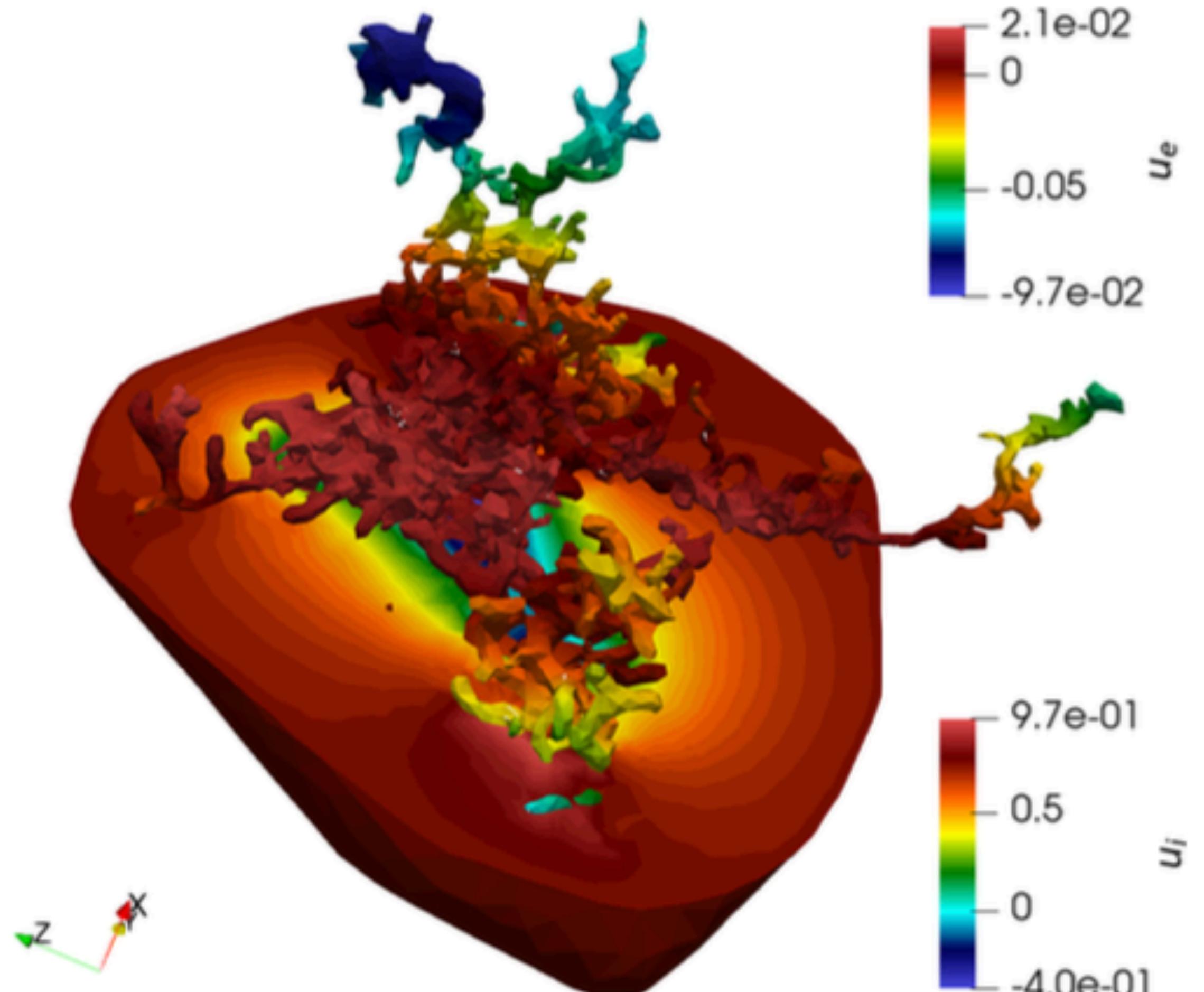
Or, let P be a coarse approximation (or a hierarchy of coarse approximations) of A : **Multigrid (AMG)**
Or an approximation of A using a simplified physical model (**physics-based P**).

EMI solution

Runtime (s), per time step.

Number of iterations m in square brackets:

Assembly	8.4
Direct	16
CG	6.3 [725]
PCG(Jacobi)	3.0 [337]
PCG(SOR)	2.6 [129]
PCG(ILU)	2.2 [98]
PCG(AMG)	1.8 [12]



Preconditioning: robustness

Preconditioned Krylov complexity: $\underline{O}(m \cdot n)$, with $m \in \mathbb{N}$ the number of iterations.

If $m \propto n$ we obtain the following complexity: $\underline{O}(n^2)$.

In practice, this is unfeasible for large problem!

For standard FEM discretizations,
for $h \rightarrow 0, n \rightarrow \infty$,
the number of CG iterations grows as

$m = \log n$ leading to $O(n \log n)$ complexity.

Can we do better?

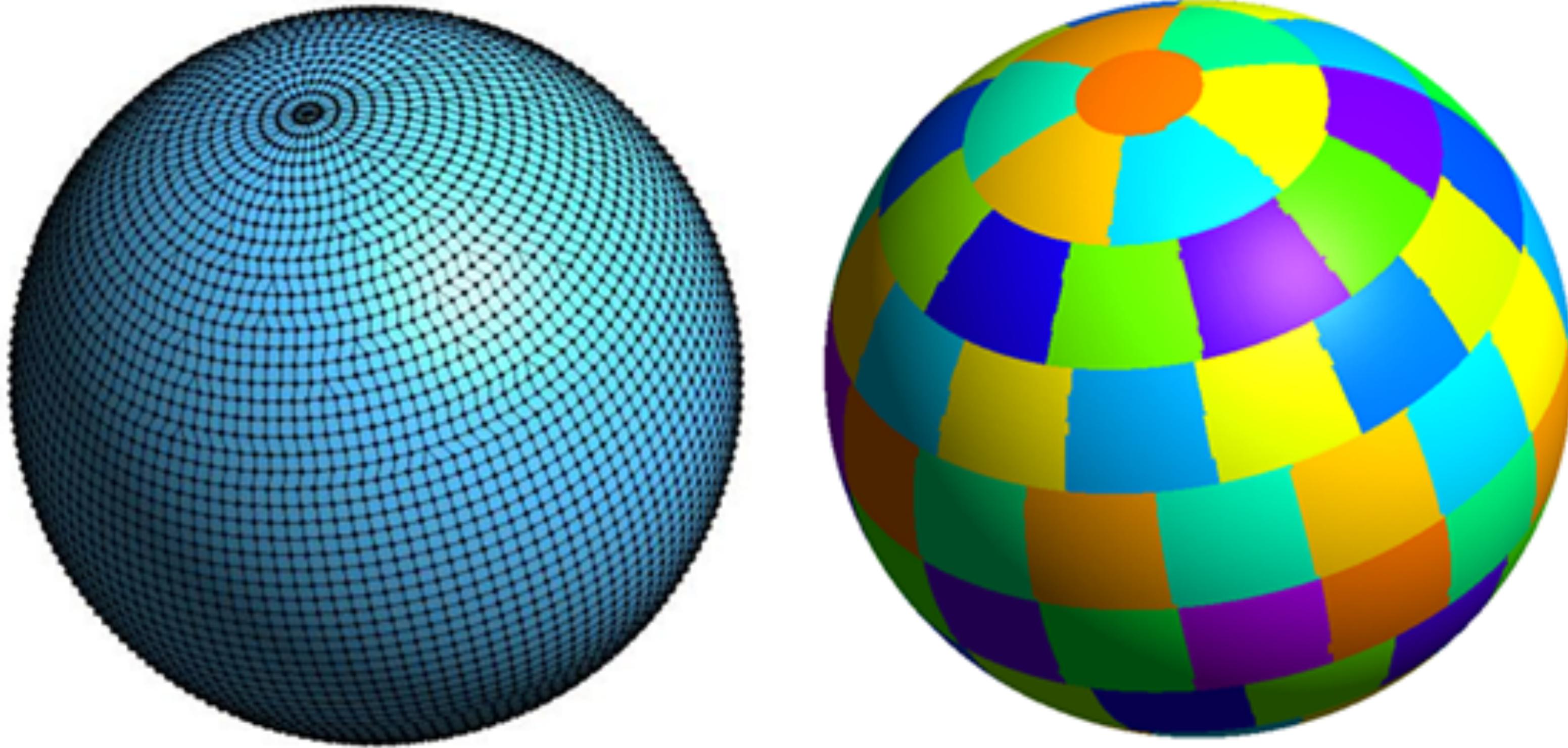
Yes, multigrid achieves h independency!

$m = O(1)$

(N, p)	(32, 1)	(64, 1)	(128, 1)	(256, 1)	(512, 1)
n	1153	4353	16,897	66,561	264,193
$\tau = 1$	5	5	5	6	5
$\tau = 10^{-1}$	5	5	6	6	6
$\tau = 10^{-2}$	5	5	5	6	7
$\tau = 10^{-3}$	5	5	5	6	6

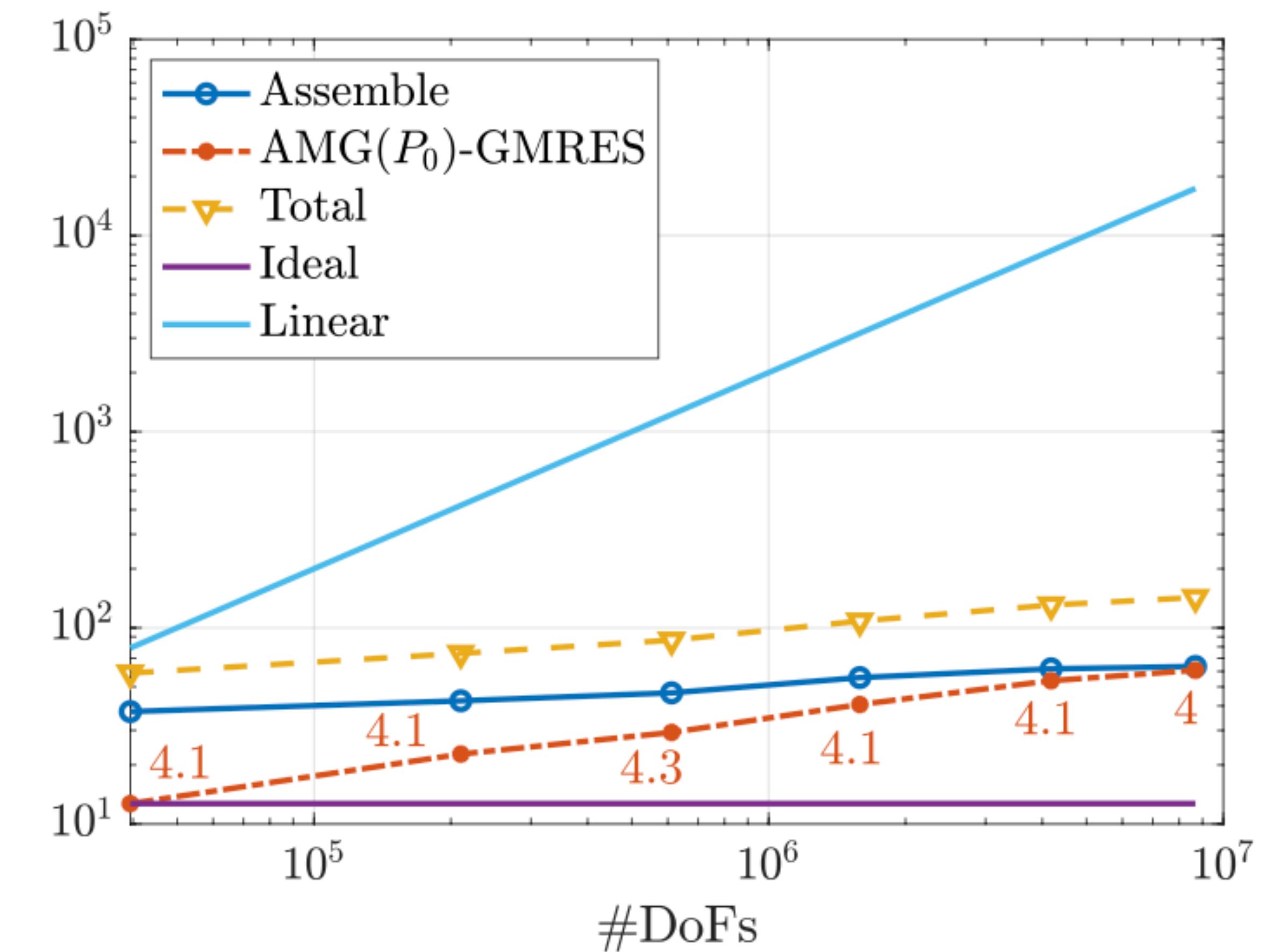
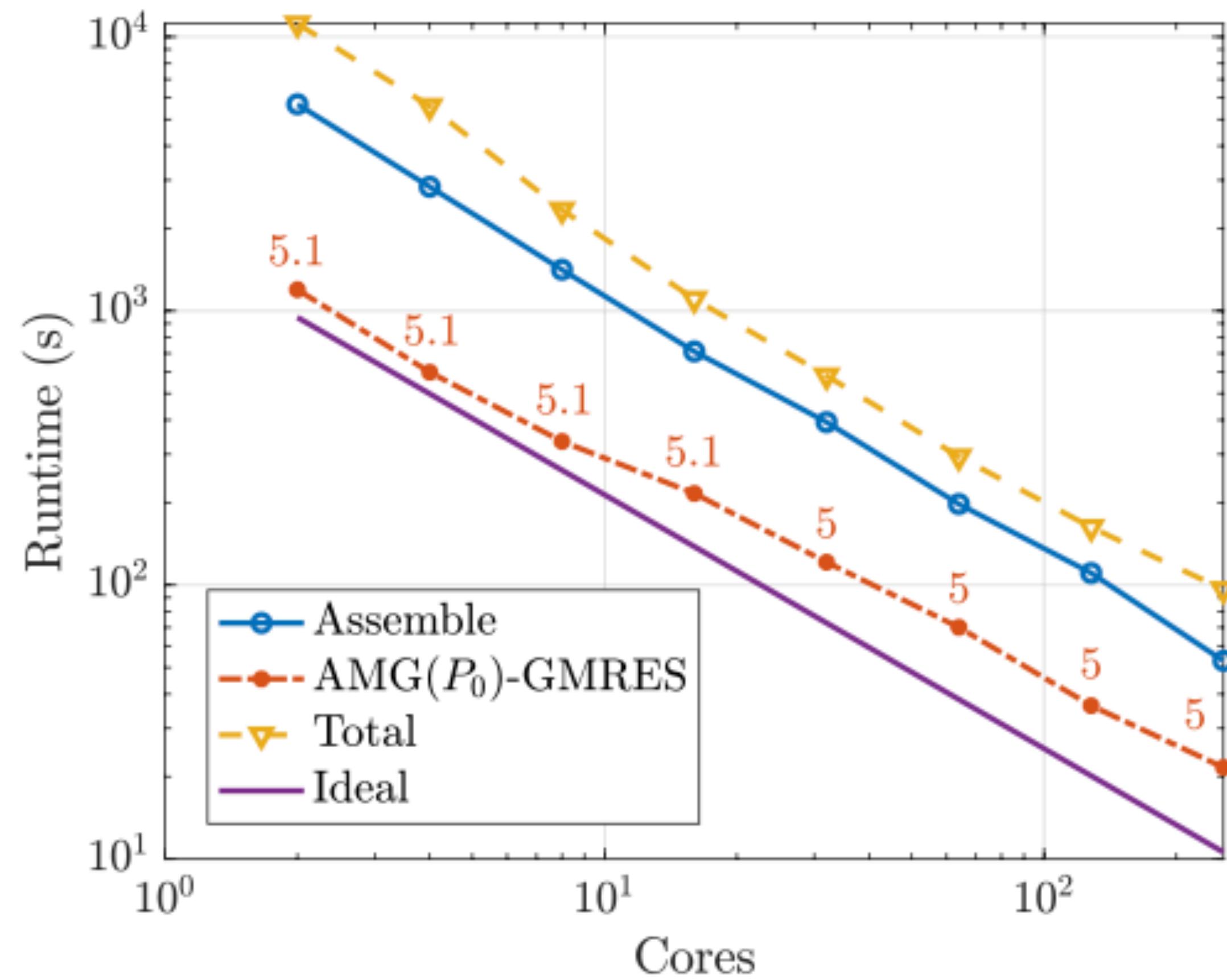
(N, p)	(16, 2)	(32, 2)	(64, 2)	(128, 2)	(256, 2)
$\tau = 1$	5	5	5	6	5
$\tau = 10^{-1}$	5	6	5	6	6
$\tau = 10^{-2}$	4	5	5	5	6
$\tau = 10^{-3}$	6	5	5	6	6

Preconditioning: domain decomposition



[Additive Schwarz or block Jacobi]

Preconditioning: parallel scaling



Hands-on exercises

Exercises and experiments:
L15(EMI)/EMI.ipynb

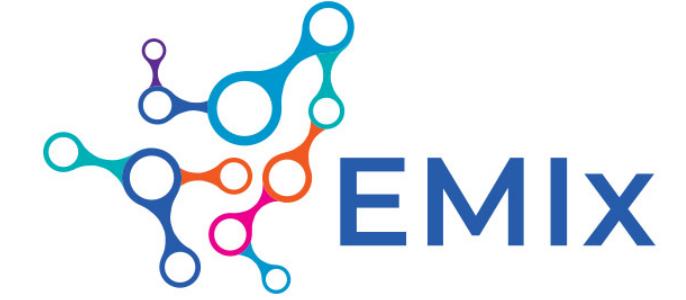


EMIx library examples:
<https://github.com/pietrobe/EMIx>



- follow installation instructions
- try run examples/EMI_example.py
- measure parallel speedup (with mpirun -N)
- change geometry or ionic modelling
(advanced)





Hands-on exercises: EMI_example.py

```
EMI_example.py      x

from pathlib import Path
from EMIx import *
from dolfin import *

# Na stimulus definition
def g_Na_stim(t):
    return Expression('40*exp(-fmod(t,0.01)/0.002)', t=t, degree=4)

if __name__=='__main__':

    # create EMI problem and ionic model
    problem = EMI_problem('config.yml')

    # add HH ionic model
    problem.add_ionic_model("HH", stim_fun=g_Na_stim)

    # solve with just .png output
    solver = EMI_solver(problem, save_xdmf_files=True, save_png_files=True)
    solver.solve()
```

Hands-on exercises: linear solver

EMIx/src/EMIx/EMI/EMI_solver.py

```
424
425      # solvers parameters
426      direct_solver      = False
427      ksp_rtol           = 1e-6
428      ksp_type            = 'cg'
429      pc_type             = 'hypre'
430      max_amg_its         = 1
431      pc_block_Jacobi    = False
432      norm_type           = 'preconditioned'
433      nonzero_init_guess = True
434      verbose              = False
```



Google “KSPtype” and “PCtype” with a (long) list of available solvers and preconditioners:

<https://petsc.org/release/manualpages/KSP/KSPType/>

<https://petsc.org/release/manualpages/PC/PCType/>

Hands-on exercises: config.yml

```
config.yml          x

# Numerical parameters
time_steps : 100
dt : 0.00002
mesh_conversion_factor : 1.0e-6

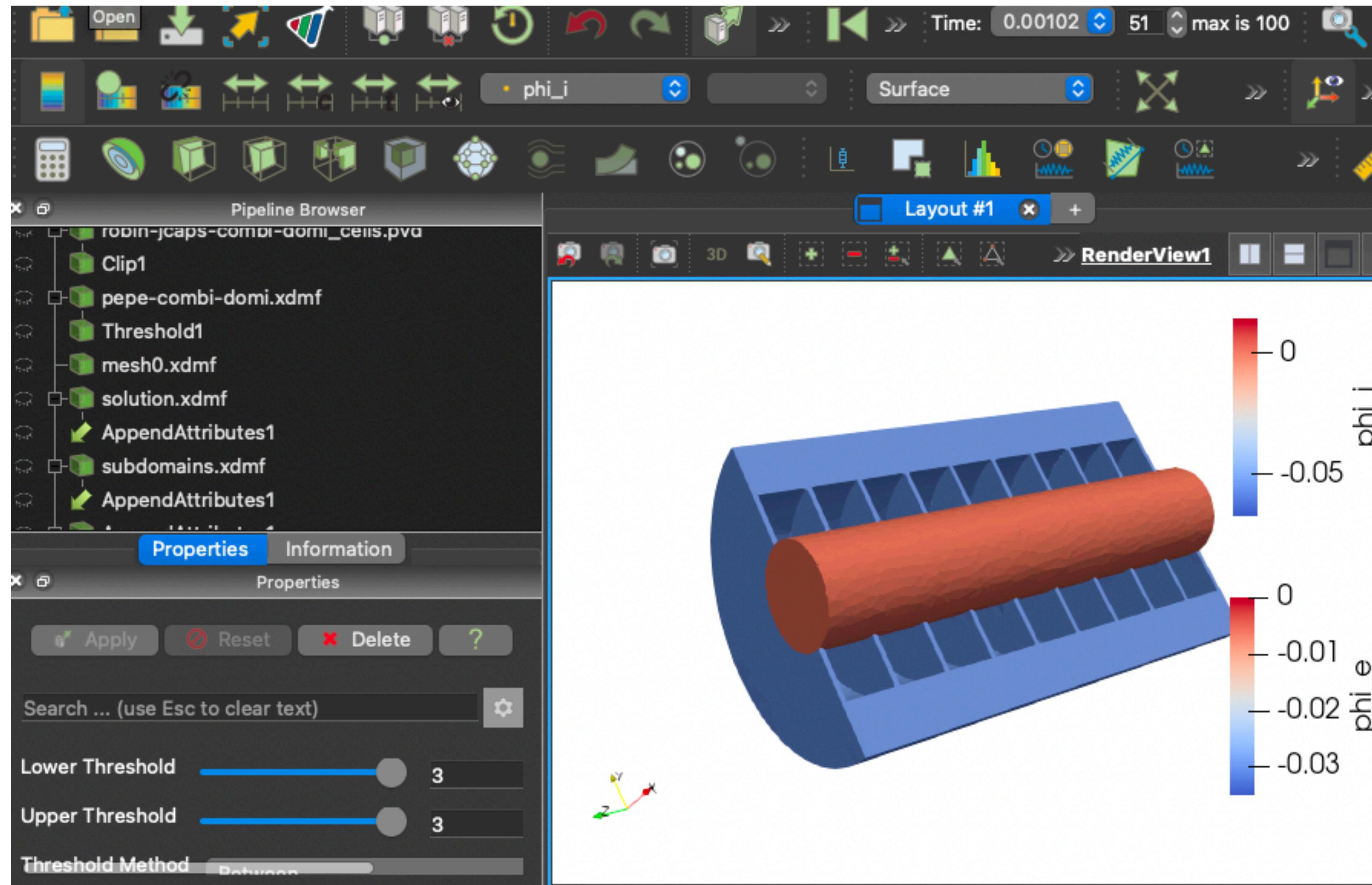
input_dir           : "../data/myelin/"
cell_tag_file      : "mesh0.xdmf"
facet_tag_file     : "facets0.xdmf"
intra_restriction_dir : "interior_restriction0 rtc.xdmf"
extra_restriction_dir : "exterior_restriction0 rtc.xdmf"

ecs_tags : 1
ics_tags : 3
membrane_tags: 2
```

Additional parameters to play with:

C_M = 0.01
sigma_i = 1.0
sigma_e = 1.0
fem_order = 1

Hands-on exercises: visualise with Paraview



Output:

- solution.xdmf
- subdomains.xdmf

Filters -> AppendAttributes

Filters -> Threshold

Filters -> Clip