

Défis de calcul pour comprendre le rythme cardiaque, les projets MICROCARD

Yves Coudière

Université de Bordeaux, Inria et Liryc, France



EuroHPC
Joint Undertaking



**Co-funded by
the European Union**

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.

This work was supported by the European High-Performance Computing Joint Undertaking EuroHPC under grant agreement No 955495 (MICROCARD) co-funded by the Horizon 2020 programme of the European Union (EU), the French National Research Agency ANR, the German Federal Ministry of Education and Research, the Italian ministry of economic development, the Swiss State Secretariat for Education, Research and Innovation, the Austrian Research Promotion Agency FFG, and the Research Council of Norway.

Cardiac muscle has a unique structure and electrophysiology

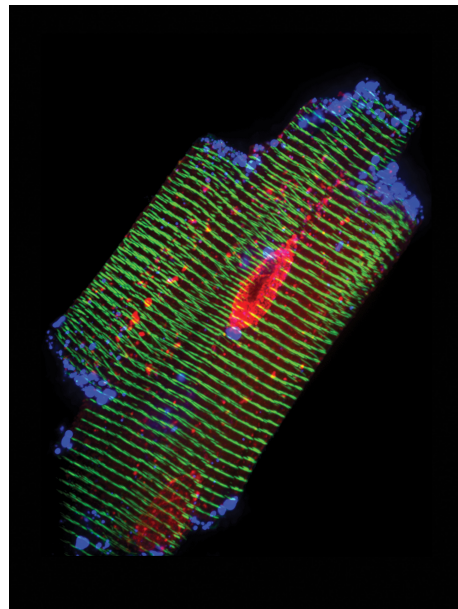
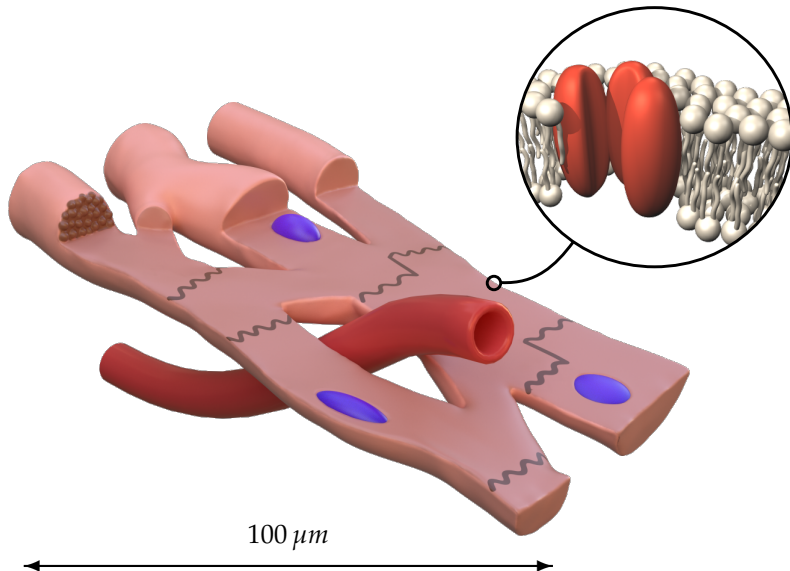
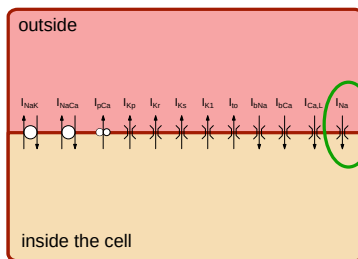
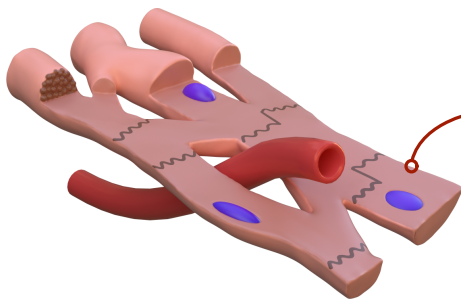


Illustration Dana Hamers Scientific Art.

Micrograph: Smyth et al. *Circulation Research*, 2012; 110: 978–989. blue: Connexin 43, green: α -actinin, red: filamentous actin.

Cardiac muscle has a unique structure and electrophysiology

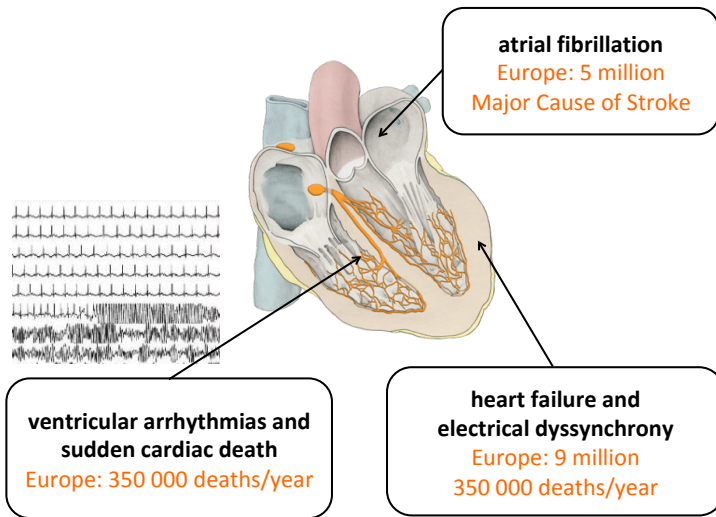
- contracts like a muscle
- transports signals like a nerve
- semi-autonomous
- specializations (e.g. in automaticity)



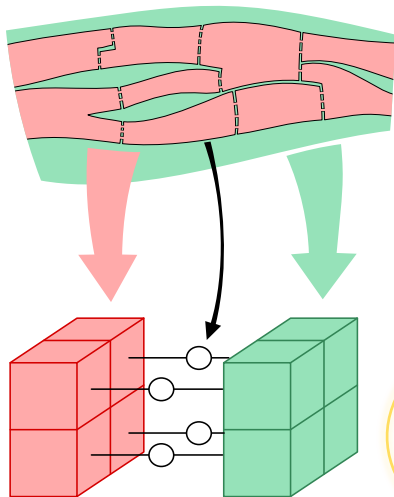
$$\begin{aligned}
 I_{Na} &= G_{Na} m^3 h j (V_m - E_{Na}) \\
 \partial_t m &= (m_\infty - m) / \tau_m \\
 \partial_t h &= (h_\infty - h) / \tau_h \\
 \partial_t j &= (j_\infty - j) / \tau_j \\
 m_\infty &= [1 + \exp((-56.86 - V_m) / 9.03)]^{-2} \\
 \tau_m &= \alpha_m \beta_m \\
 \alpha_m &= [1 + \exp((-60 - V_m) / 5)]^{-1} \\
 \beta_m &= 0.1 / [1 + \exp((V_m + 35) / 5)] \\
 &\quad + 0.1 / [1 + \exp((V_m - 50) / 200)] \\
 h_\infty &= [1 + \exp((V_m + 71.55) / 7.43)]^{-2} \\
 \tau_h &= (\alpha_h \beta_h)^{-1} \\
 \alpha_h &= \begin{cases} 0 & \text{if } V \geq -40 \text{ mV} \\ 0.057 \exp((-V_m + 80) / 6.8) & \text{otherwise} \end{cases} \\
 &\quad (10 \text{ more lines like that})
 \end{aligned}$$

Structural abnormalities play a role in almost all cardiac arrhythmia

- the heart contains 1. to 10×10^9 individual cells
- the network microstructure yields (rhythmic) synchronisation by fast electrical propagation



Almost all work until now relied on homogenization

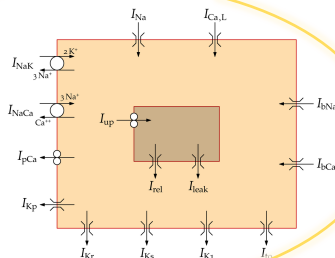


$$\begin{cases} \nabla \cdot (\mathbf{G}_i \nabla \phi_i) = I_m \\ \nabla \cdot (\mathbf{G}_e \nabla \phi_e) = -I_m \end{cases}$$

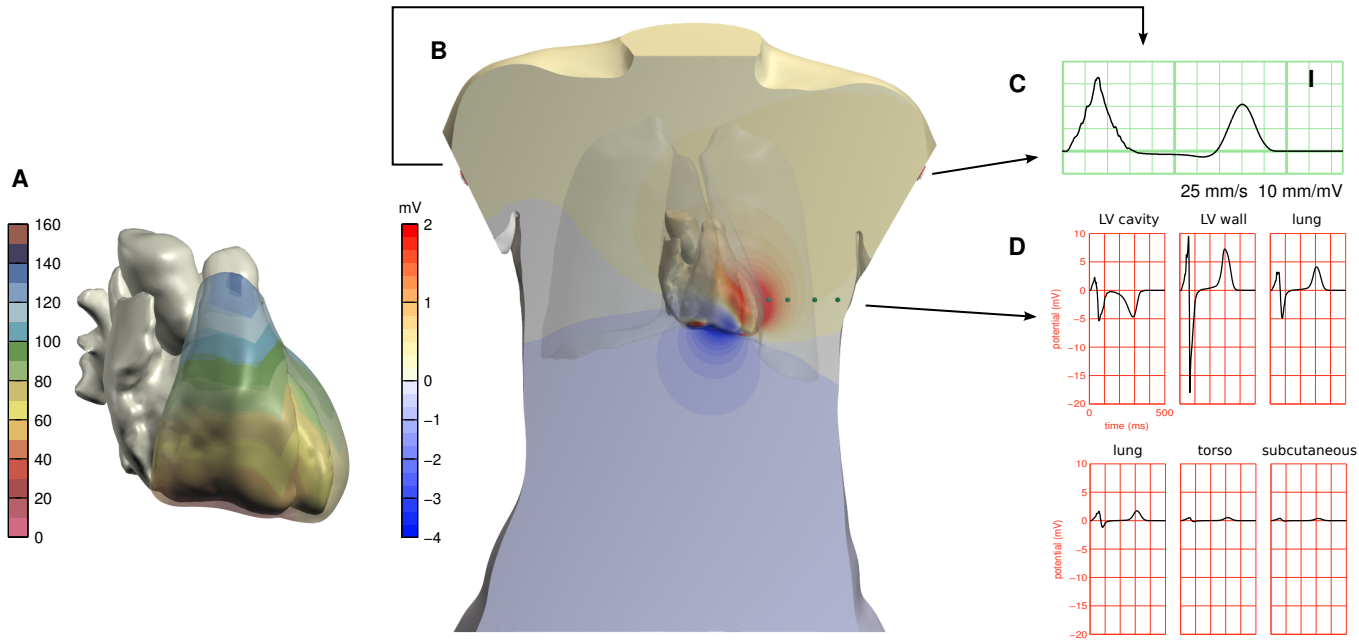
$$V_m = \phi_i - \phi_e$$

$$I_m = \beta C_m \frac{\partial V_m}{\partial t} + I_{\text{ion}} + I_s$$

membrane:
20-50 ODEs
per node

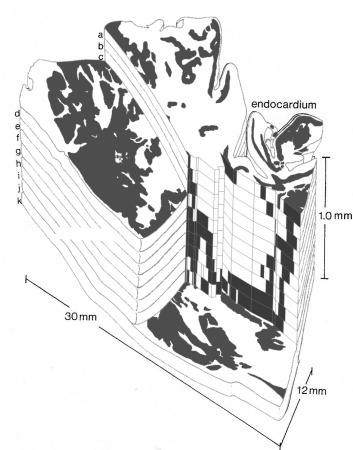


Homogenization allows EP modeling from cell to bedside

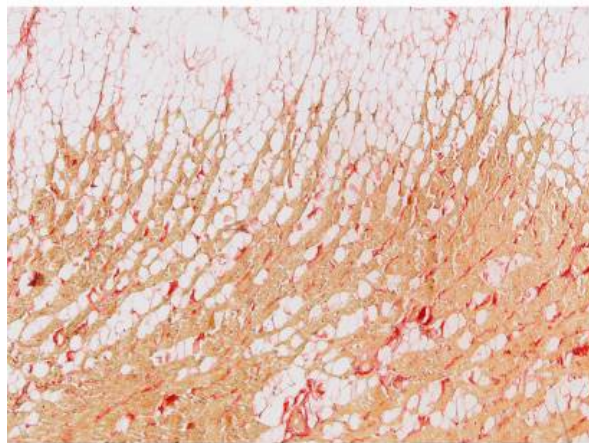


but homogeneous models are poor approximations for damaged tissue

- no notion of individual cells
- cannot represent damaged tissue or cellular heterogeneity
- damaged tissue is everywhere (ageing, infarction scars, cardiomyopathies, ...)

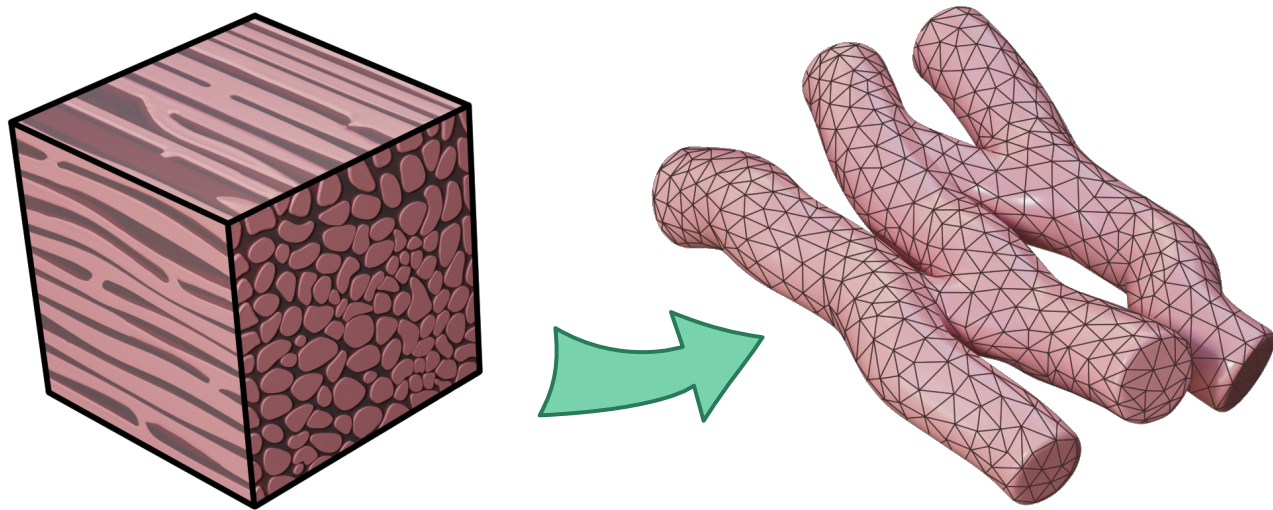


De Bakker et al. *JACC* 15:1594-1607 (1990)



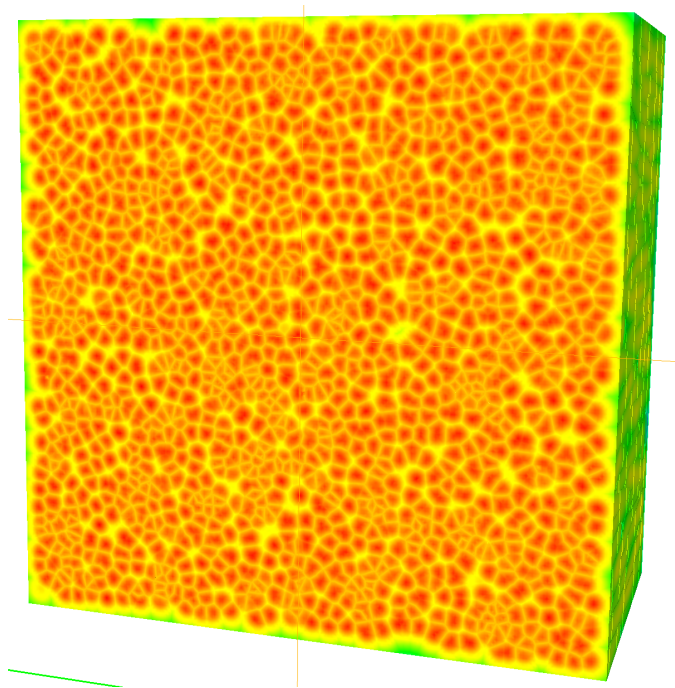
Hoogendijk et al. *Heart Rhythm* 7:238-248 (2010)

so we scale from $200\text{ }\mu\text{m}$ to $1\text{ }\mu\text{m}$

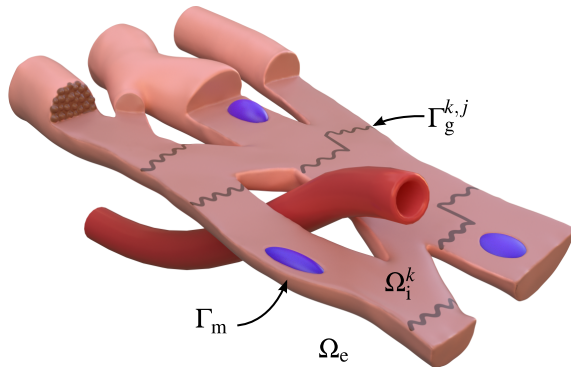


Consequences

1 mm³
7538 cells
271 million tetrahedra
15 GB storage



A more complicated model formulation



$$\left\{ \begin{array}{ll} \nabla \cdot (G_i^k \nabla \phi_i^k) = 0, & \text{on } \Omega_i^k, \\ \nabla \cdot (G_e \nabla \phi_e) = 0, & \text{on } \Omega_e, \\ V_m^k = \phi_i^k - \phi_e, & \text{on } \Gamma_m^k, \quad (\text{Transmem. voltages}) \\ -G_i^k \nabla \phi_i^k \cdot \mathbf{n} = G_e \nabla \phi_e \cdot \mathbf{n} = C_m \partial_t V_m^k + I_{\text{ion}}(V_m^k, \mathbf{y}), & \text{on } \Gamma_m^k, \quad (\text{Transmem. currents}) \\ -G_i^k \nabla \phi_i^k \cdot \mathbf{n} = G_i^j \nabla \phi_i^j \cdot \mathbf{n} = \kappa(\phi_i^k - \phi_i^j), & \text{on } \Gamma_g^{k,j}, \quad (\text{Gap junctions}) \\ \partial_t \mathbf{y} = \mathbf{F}(V_m, \mathbf{y}), & \text{on } \Gamma_m^k \end{array} \right.$$

Our problem

- equations that are hard to solve on a large scale
- very large computations will require exascale hardware
 - very high degree of parallelism
 - heterogeneous hardware
 - need resilience to hardware failures
 - need to mitigate energy consumption
- huge meshes

Solution

- An EU grant for a multidisciplinary collaborative project →



- Another EU grant for a multidisciplinary collaborative project →



- Currently planning to apply for a 3rd EU grant

The MICROCARD project



EuroHPC
Joint Undertaking



simula



université de BORDEAUX

Inria



“EuroHPC JU is a joint initiative between the EU, European countries and private partners to develop a World Class Supercomputing Ecosystem in Europe”

- funds development of hardware, libraries, and applications



EuroHPC
Joint Undertaking



Plusieurs disciplines académiques à faire travailler ensemble

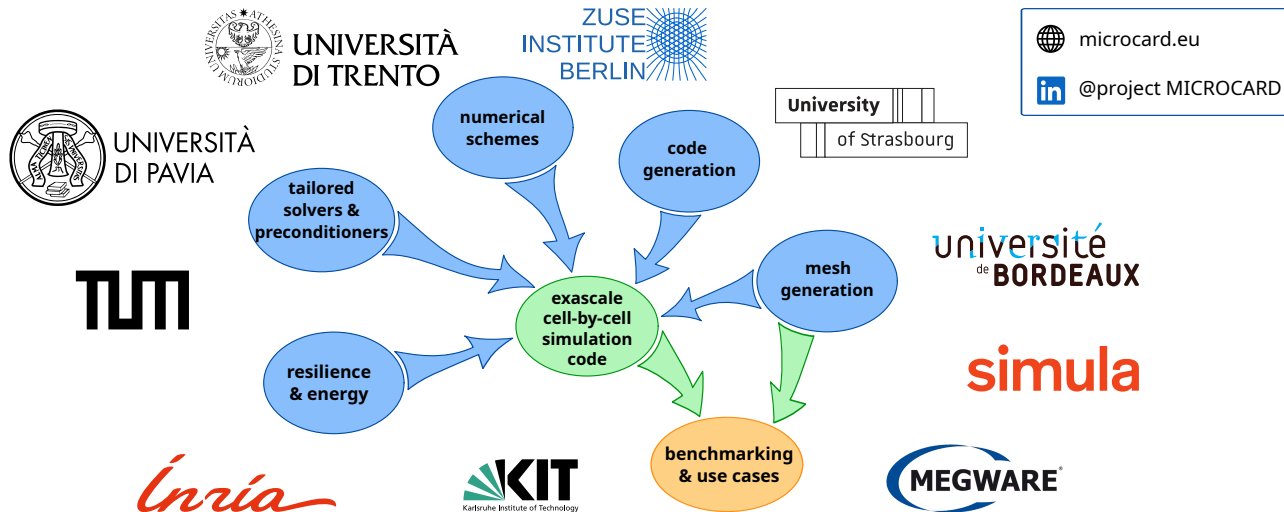
Construire un logiciel qui fonctionne de manière efficace sur des machines exascales et peut être utile à une communauté applicative (i.e. servir à mener des campagnes expérimentales en biologie)

- mathématique
- informatique
- ingénierie biomédicale

Contribuer à un logiciel open source communautaire



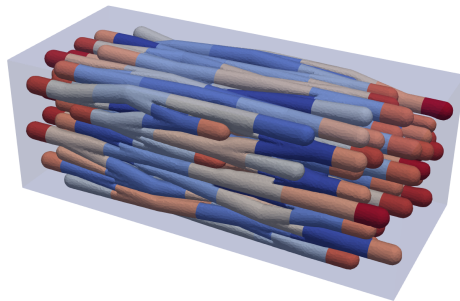
Project outline



The math problem

The domain is composed of $N + 1$ subdomains (0: ECM and $1 \leq k \leq N$: cells) with *complex interfaces*

- The problem has a solution [PhD P.-E. Bécue 2018]
- Error estimate for FV approximations [PhD Z. Chehade 2025]



There is a variational formulation for (ϕ_k) in $\prod_{k=0}^N H^1(\Omega_k)$,

$$\sum_{k=0}^N \int_{\Omega_k} \nabla \phi_k^T \sigma_k \nabla \psi_k dx + \sum_{k=0}^N \sum_{\ell \neq k} \int_{F_{k\ell}} \left(C_m \partial_t (\phi_k - \phi_\ell) + I_{k\ell}^{\text{ion}} (\phi_k - \phi_\ell, w_{k\ell}) \right) \psi_k ds = 0$$

coupled to ODEs on the interfaces $F_{k\ell}$, part of which are semi-linear

$$\partial_t w_{k\ell} = A_{k\ell} (\phi_k - \phi_\ell, w_{k\ell}) w_{k\ell} + B(w_{k\ell} \phi_k - \phi_\ell, w_{k\ell})$$

- time-dependent transmission conditions for 3D electrostatic eqs. $\rightarrow$ 2D evolution equation
- very diverse time scales in the ODEs, heterogeneous number of ODEs

Sketched time iteration (example with IMEX Euler)

$$(M + \Delta t K) (\Phi^{n+1} - \Phi^n) = -\Delta t \left(K\Phi^n + B_s I^{\text{ion}}(B\Phi^n, W^n) \right) \quad (1)$$

$$W^{n+1} - W^n = \Delta t \left(A(\Phi^n, W^n)W^n + B(\Phi^n, W^n) \right) \quad (2)$$

For P1 finite elements + collocation points, a time step amounts to

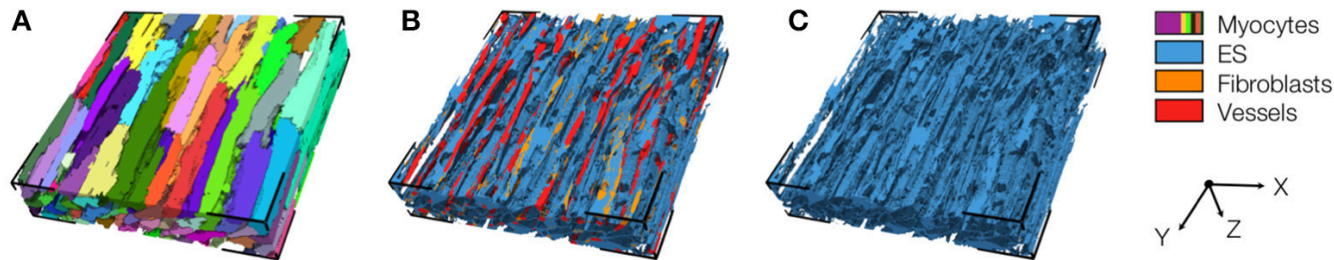
- eq. (2): compute complex ionic models on all membrane interfaces (collocation nodes, W piecewise constant on interfaces)
- eq. (1): solve the large sparse linear system

How to iterate for higher order FE approximation ?

- use spectral deferred correction methods (SDC)
- possibly exponential ones (ESDC)
- repeat the 1st order iteration several times, preserve the level of parallelism

Why do we need homemade 3D geometries ?

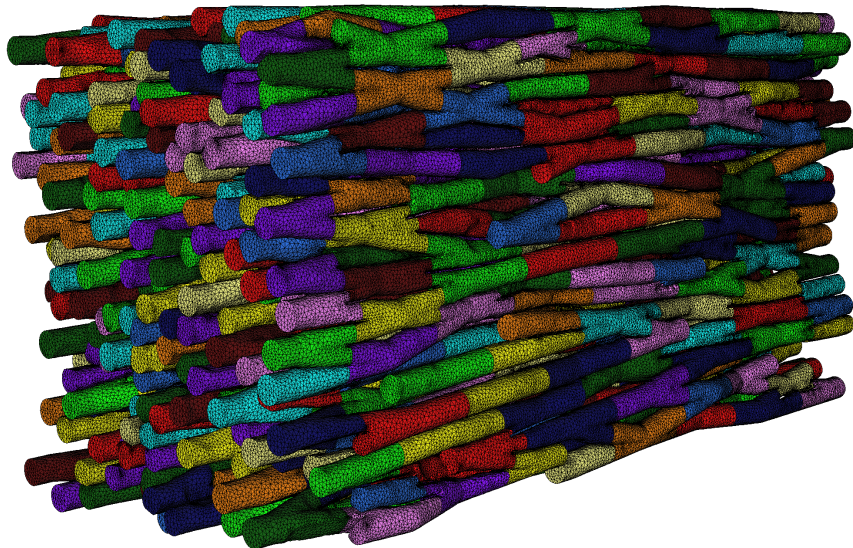
- Available biological images: confocal microscopy with fluorescent markers
- But there are challenges
 - confocal microscopy has limited depth
 - sample size limited to a few millimeters
 - interpretation and meshing is difficult



Greiner et al. Front. Physiol., 2018;9:239. doi:10.3389/fphys.2018.00239

Alternative: weave our own tissue

- can be much larger
- complexity can be constrained
- can also help to improve understanding of tissue structure



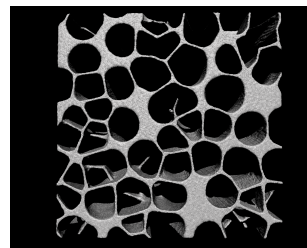
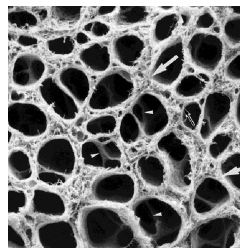
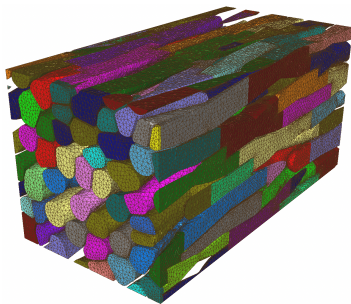
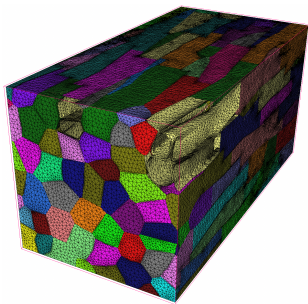
Method to construct a mesh of an artificial cell network

- create a random network of links, add centers, filter result to match known statistics
- build a first mesh, with subdomains containing the cells
- define an implicit level set function, remesh



- Many bugs or problems in Mmg, because of multimateriality. The geometry is especially complicated
- What can be done in parallel ?

Example



Mesh partitionning

- partition the mesh nodes, but
 - keep the cardiac cells intact ?
 - combine two constraints on load balance ?
 - * mesh nodes, i.e. solving the linear system
 - * interfaces nodes, i.e. computing the membrane and gap junctions currents
- very large size of the mesh → parallel partitionning



Code generation for ionic models

A DSL (Domain Specific Language – MathML) is used to describe the equations

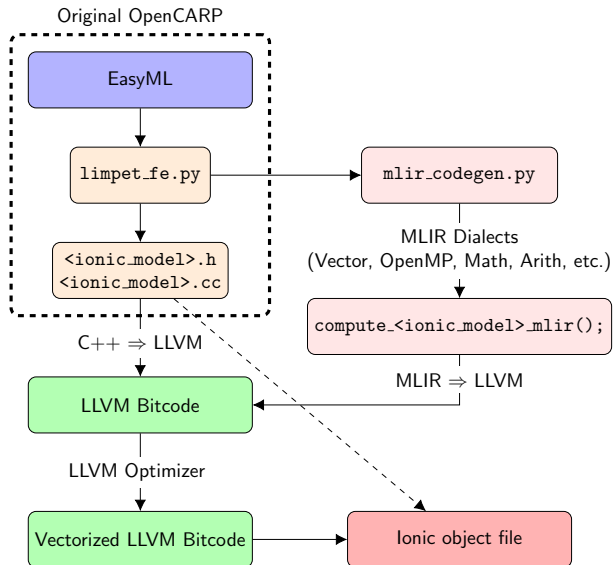
A Python script generates C++ code... How to improve performance?

- OpenMP parallelization is easy
- vectorization! Compilers can't do it, even with SIMD pragma, because function evaluations are too complex
- GPU ? Other accelerators, or architectures (FPGA, ARM, EPI, RiscV...)

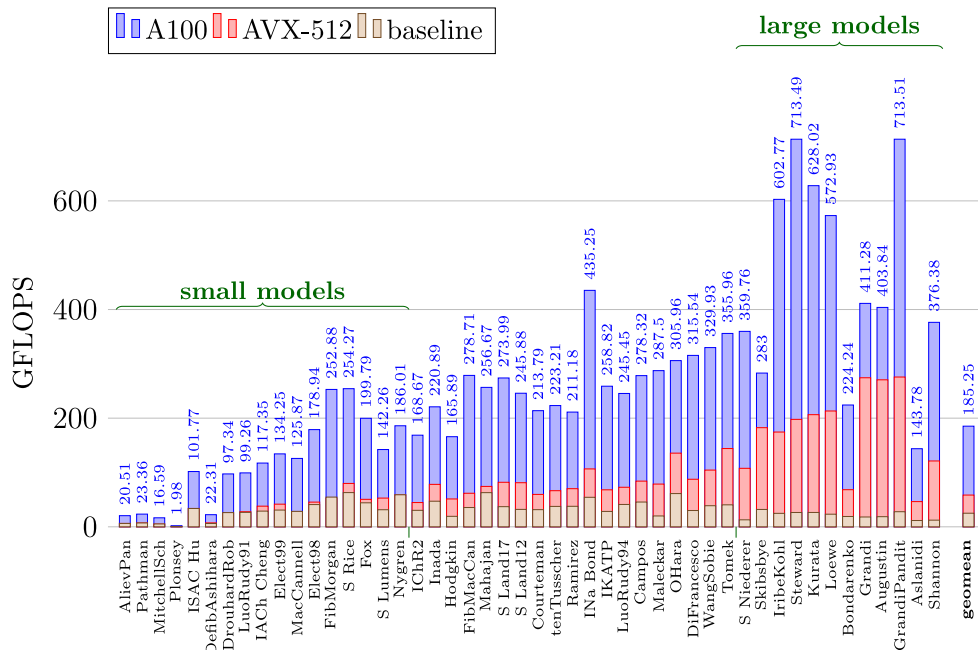
Solution: give some *vectorization tips* to the compiler

- leverage the existing infrastructure
- through MLIR (Multi-level Intermediate Representation) (mlir.llvm.org)

New compilation scheme



Results for the AVX-512 (Intel vector CPU) and A100 (Nvidia GPU) architectures



Ionic model

Active energy management

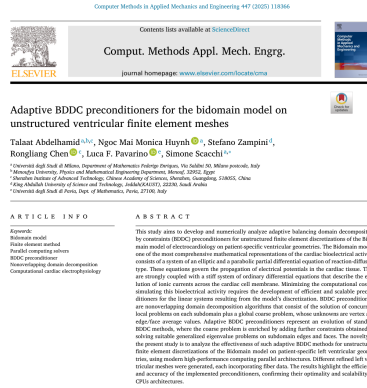
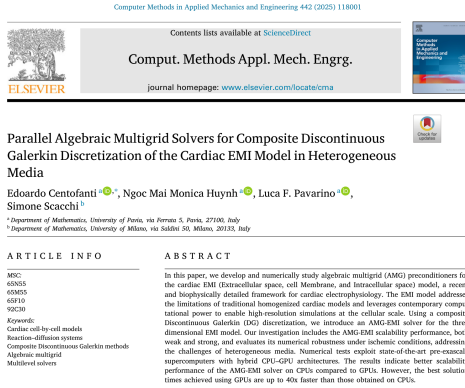
Problems

- application side: load imbalance due to partitionning, algorithmic adaptativeness, variable iteration numbers...
- system side: HW adaptation, runtime system-wide energy saving

Anticipate work imbalances, model performance, measure app and act

- lower frequency scaling
- lower parallelism
- change computational kernel, mixed precision, algorithmic variant...

Do we need dedicated preconditioning techniques ?



- AMG methods are preferred ones
 - Adaptive BDDC methods may perform better on larger test cases and parallel architectures
- adaptive BDDC for the EMI model ?
- are GPU implementation useful ?

Tests include pre-exascale EuroHPC machines, e.g. LEONARDO

And optimal solver implementation



BDDC PRECONDITIONING ON GPUS FOR CARDIAC SIMULATIONS

Fritz Göbel¹ Terry Cojman¹ [0000–0002–1560–921X] Hartwig Anzt^{2,1} [0000–0003–2177–952X]

October 22, 2024

ABSTRACT

In order to understand cardiac arrhythmia, computer models for electrophysiology are essential. In the EuroHPC MicroCARD project, we adapt the current models and leverage modern computing resources to model diseased hearts and their microstructure accurately. Towards this objective, we develop a portable, highly efficient, and performing BDDC preconditioner and solver implementation, demonstrating scalability with over 90% efficiency on up to 100 GPUs.

Keywords BDDC preconditioning · Sparse Linear Algebra · High Performance Computing · GPUs · Cardiac simulations

- Support and orchestration of GPUs in OpenCARP ? Memory transfer...
- Implementation of AMG and BDDC methods ?
- Support for GPU to solve PDE on unstructured grids ?

Thank you and see us on microcard.eu / LinkedIn



EuroHPC
Joint Undertaking



Questions

PDEs, unstructured meshes on exascale architectures ?

- adapt discretization methods
- adaptative order in space and time ? Other methods
- numerical analysis: approximation of complex functions (tabulation), mixed precision arithmetic ?

Consider energy to solution, not only time

- collaborate with informatics
- control adaptivity, order, precision, etc

Make sure that HPC is used thoughtfully

- Find “real life” challenges in other disciplines