MICROCARD: modeling the heart cell by cell

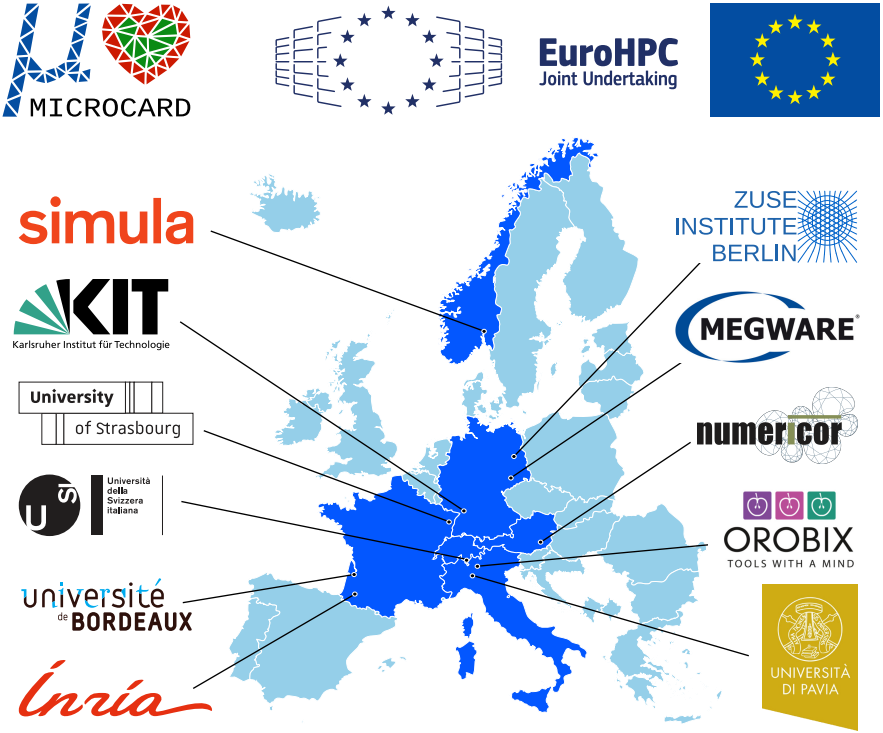
Mark Potse

coordinator, EuroHPC MICROCARD project
IHU Liryc, Université de Bordeaux, France
Inria Bordeaux, France

EuroHPC
Joint Undertaking

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.

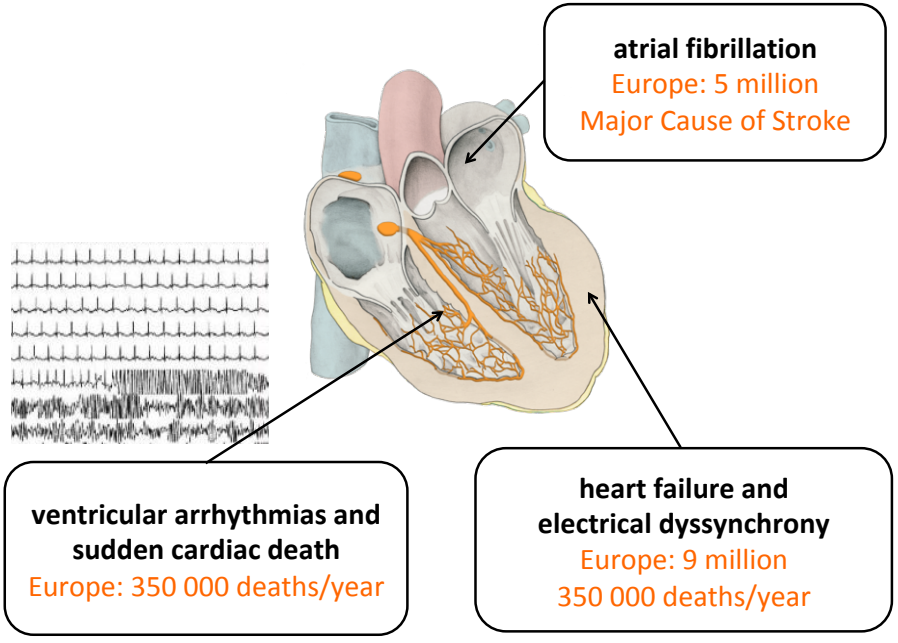
The MICROCARD project



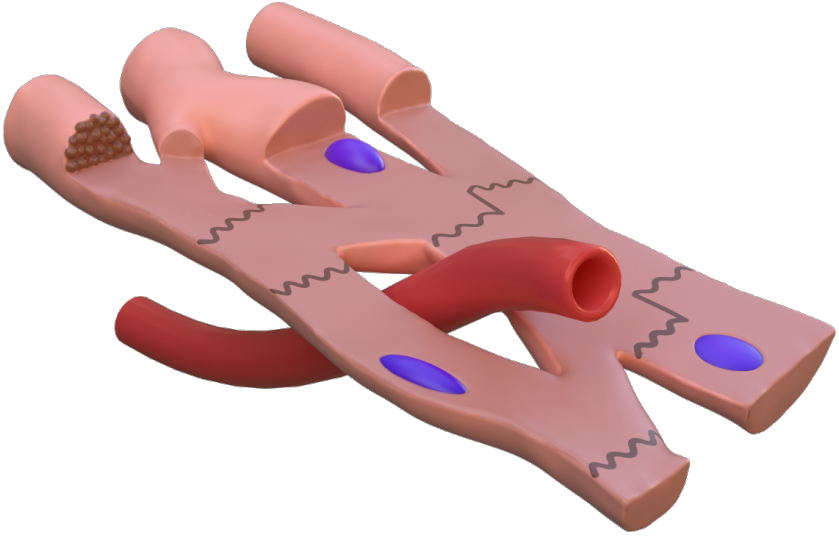
This project has received funding from the European High-Performance Computing Joint Undertaking EuroHPC (JU) under grant agreement No 955495. The JU receives support from the European Union's Horizon 2020 research and innovation programme and France, Italy, Germany, Austria, Norway, and Switzerland.

Our challenge

- cardiac disease is the #1 cause of death in Europe and half of these deaths are caused by electrical malfunctions
- structural muscle damage is crucial in most of these



Cardiac muscle has a unique structure and electrophysiology



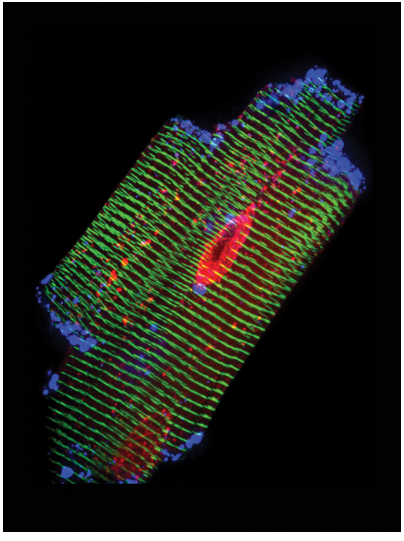
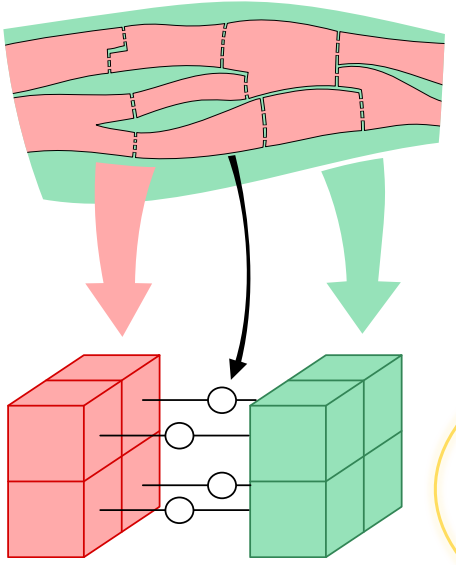


illustration Dana Hamers Scientific Art

The present: bidomain model

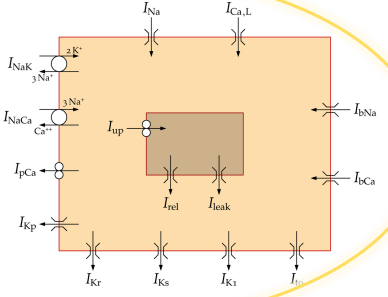


$$\begin{cases} \nabla \cdot (\mathbf{G}_\mathrm{i} \nabla \phi_\mathrm{i}) = I_\mathrm{m} \\ \nabla \cdot (\mathbf{G}_\mathrm{e} \nabla \phi_\mathrm{e}) = -I_\mathrm{m} \end{cases}$$

$$V_\mathrm{m} = \phi_\mathrm{i} - \phi_\mathrm{e}$$

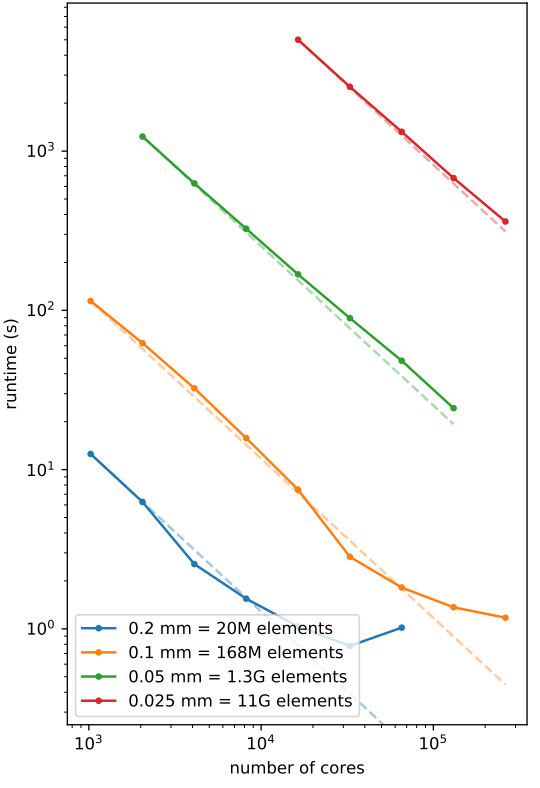
$$I_\mathrm{m} = \beta C_\mathrm{m} \frac{\partial V_\mathrm{m}}{\partial t} + I_\mathrm{ion} + I_\mathrm{s}$$

membrane:
20-50 ODEs
per node



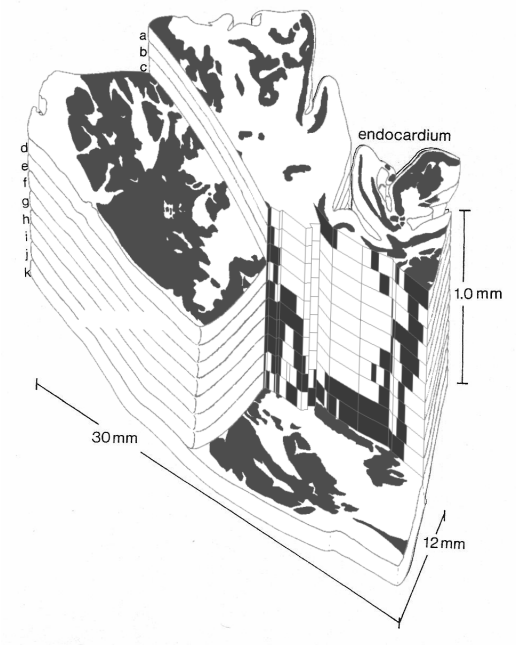
Bidomain models do well on current supercomputers

- demonstrated scaling upto 260 k cores

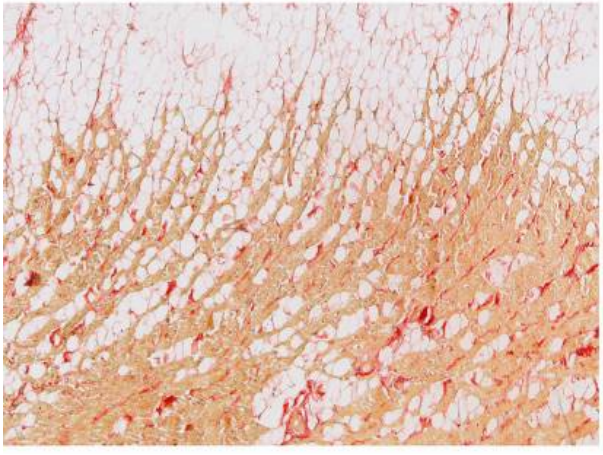


Limitations of the bidomain model

- no notion of individual cells
- cannot represent damaged tissue
- 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)

Example

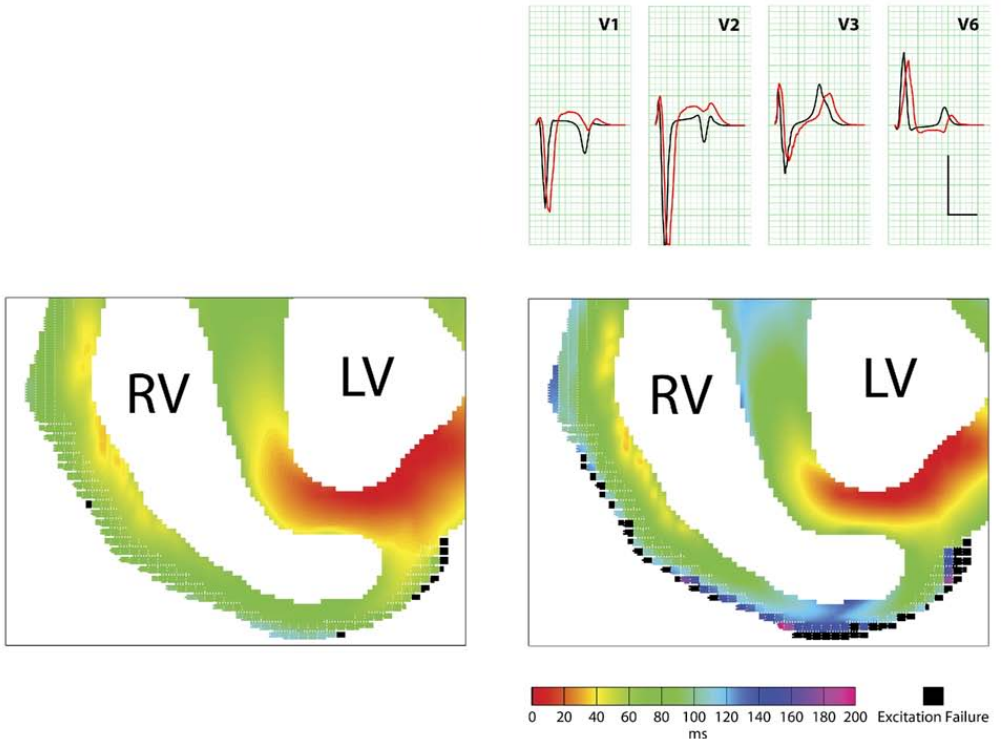
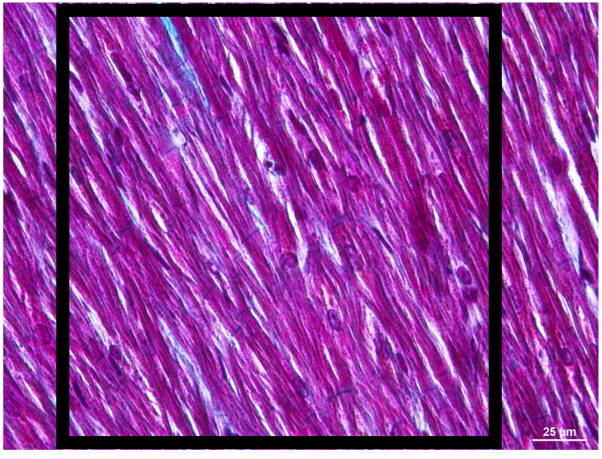


figure adapted from Hoogendijk et al. *Heart Rhythm* 7:238–248 (2010)

Solution: Build a model in which individual cells are discretized

before

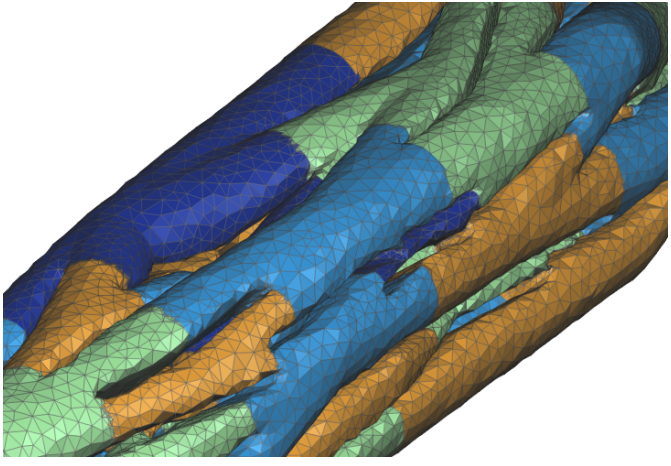
about 200 cells per model element



200 µm

after

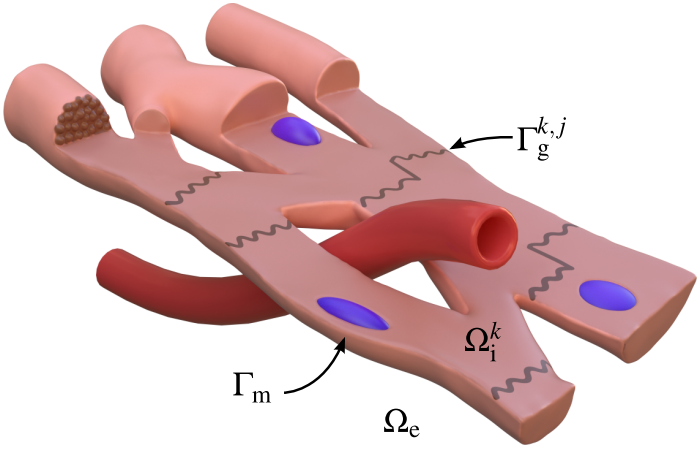
about 1000 model elements per cell



- the MICROCARD project (`microcard.eu`) develops code that can run such models on exascale supercomputers

MICROCARD

A more complicated model formulation



$$
\begin{cases}
\nabla \cdot (G_{\mathrm{i}}^{k} \nabla \phi_{\mathrm{i}}^{k}) = 0, & \text{on } \Omega_{\mathrm{i}}^{k}, \\
\nabla \cdot (G_{\mathrm{e}} \nabla \phi_{\mathrm{e}}) = 0, & \text{on } \Omega_{\mathrm{e}}, \\
V_m^k = \phi_i^k - \phi_e, & \text{on } \Gamma_{\mathrm{m}}^{k}, \quad \text{(Transmem. voltages)} \\
-G_{\mathrm{i}}^{k} \nabla \phi_{\mathrm{i}}^{k} \cdot \mathbf{n} = G_{\mathrm{e}} \nabla \phi_{\mathrm{e}} \cdot \mathbf{n} = C_{\mathrm{m}} \partial_t V_{\mathrm{m}}^{k} + I_{\mathrm{ion}}(V_{\mathrm{m}}^{k}, \mathbf{y}), & \text{on } \Gamma_{\mathrm{m}}^{k}, \quad \text{(Transmem. currents)} \\
-G_{\mathrm{i}}^{k} \nabla \phi_{\mathrm{i}}^{k} \cdot \mathbf{n} = G_{\mathrm{i}}^{j} \nabla \phi_{\mathrm{i}}^{j} \cdot \mathbf{n} = \kappa(\phi_{\mathrm{i}}^{k} - \phi_{\mathrm{i}}^{j}), & \text{on } \Gamma_{\mathrm{g}}^{k,j}, \quad \text{(Gap junctions)} \\
\partial_t \mathbf{y} = \mathbf{F}(V_{\mathrm{m}}, \mathbf{y}), & \text{on } \Gamma_{\mathrm{m}}^{k}
\end{cases}
$$

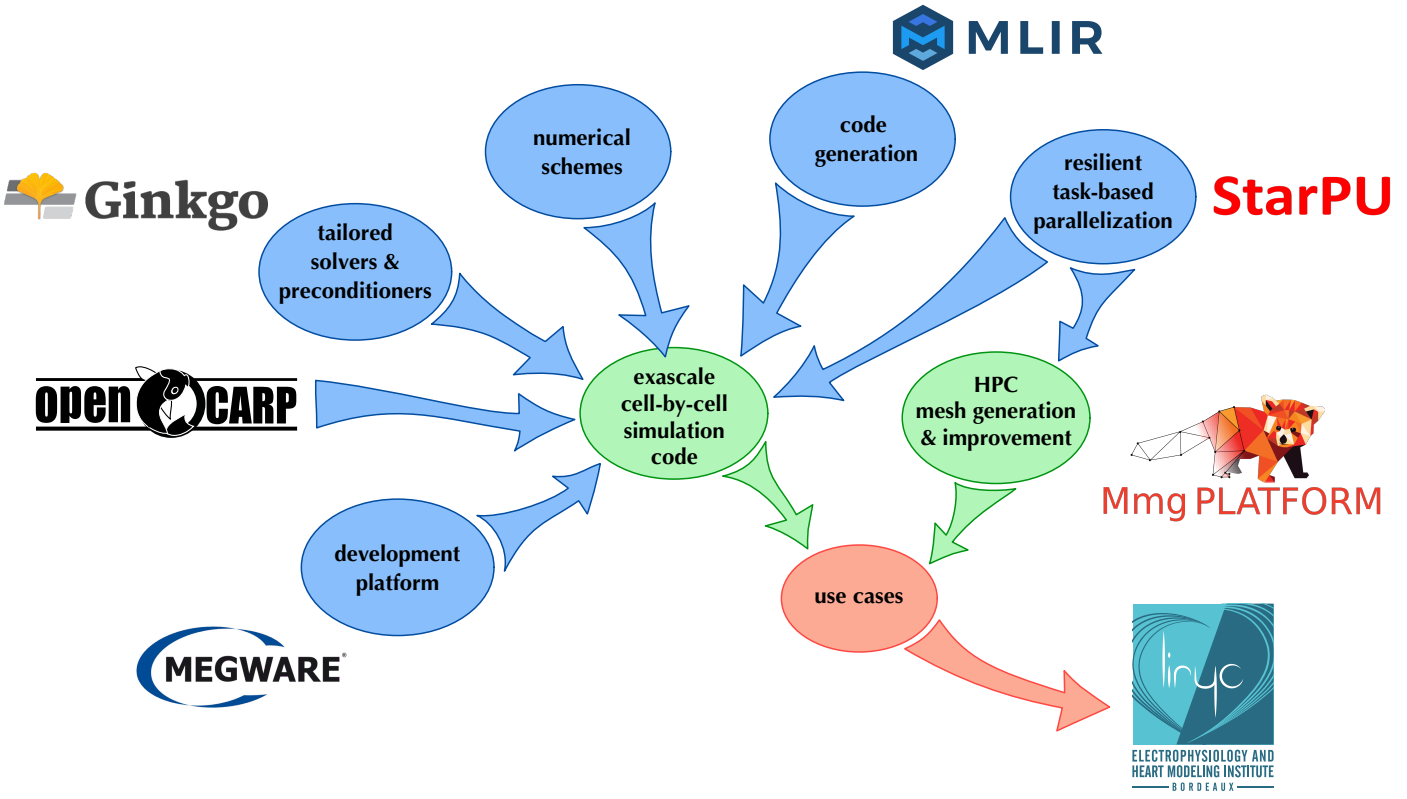
Our problem

- equations that are hard to solve on a large scale (numerical science)
- very large computations will require future exascale hardware
- these machines will be hard to program efficiently (HPC informatics)
- we want realistic physiology (biomedical engineering)
- we want realistic anatomy (image segmentation)
- we will need much more powerful meshing software (informatics)

Objectives in detail

- **cell-by-cell**
- **10,000 times larger** than current routine
- numerical schemes suitable for **exascale** parallelism
- automated translation of high-level model descriptions into optimized, **energy-efficient** system code for **heterogeneous hardware**
- **resilience to hardware failures** and an energy-aware task placement strategy
- application and testing with **real-life use cases**
- construction of **geometrical models** (from imaging data and synthetic)
- develop **parallel segmentation and (re)meshing softwarea**, necessary to create our meshes

Project outline



First steps: integrating the Ginkgo library

Open cardiac electrophysiology simulator for in-silico experiments.
Supports shared memory parallelization via OpenMP, and multinode parallelization through MPI.

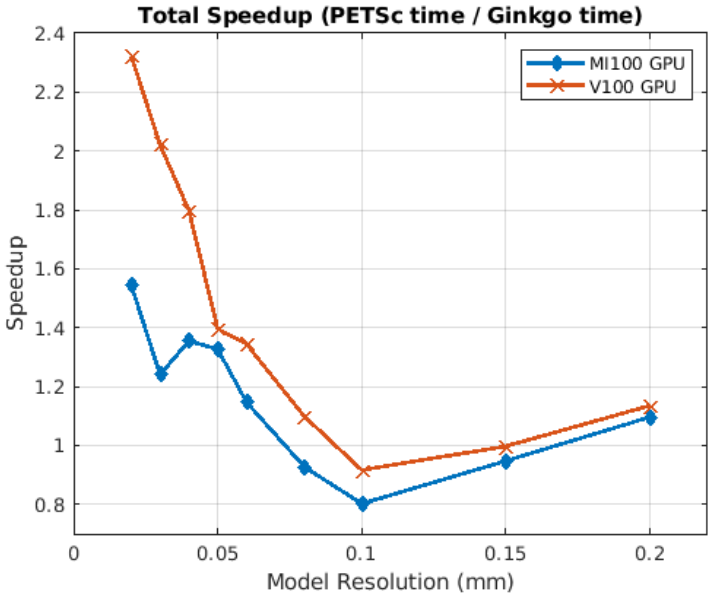
GPU-focused high performance linear algebra library for manycore systems, with a focus on solving sparse linear systems.

- New numerical back end (Ginkgo) added to the existing one (PETSc)
- Ginkgo brings implementations for NVIDIA, AMD and Intel GPUs
- Choice of the back end to use (PETSc or Ginkgo) at runtime
- Preconditioners and solvers customizable using configuration files.

work of Terry Cojean, Fritz Goebel, and Marie Houillon (Karlsruhe Institute of Technology) presented at the Fifth ISC Workshop on HPC Applications in Precision Medecine, Hamburg, June 2022

GPU performance with Ginkgo

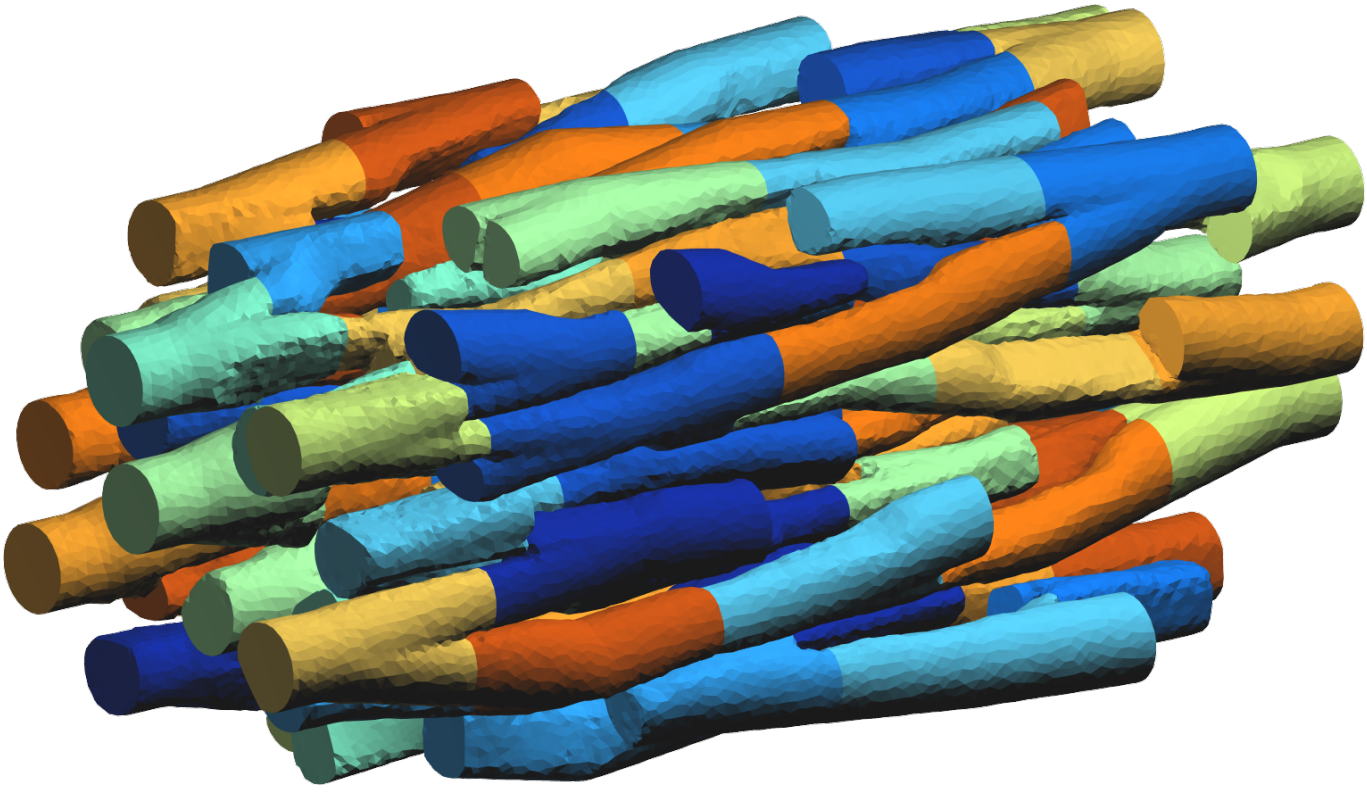
Test case: bidomain simulation on rectangular tissue block, using Drouhard-Roberge ionic model



- Good portability over AMD MI100 GPU, NVIDIA V100 GPU and CPU parallelization.
- Better performance on GPU than CPU when problem is large enough.

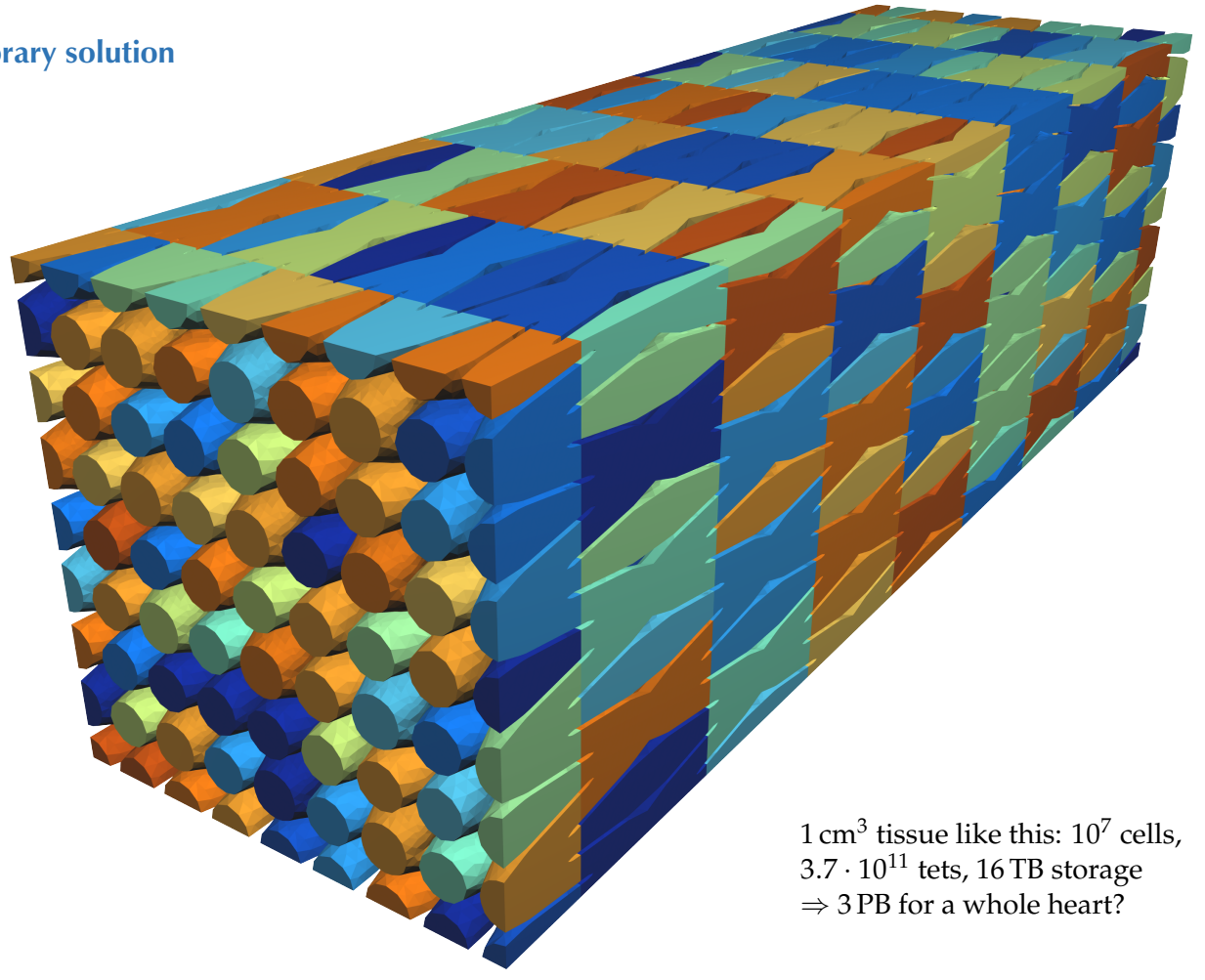
Resolution (mm)	Number of tetrahedra
0.02	7 840 000
0.1	62 720
0.2	7 840

Results: mesh creation



work of Mark Potse, Luca Cirrottola, and Algiane Froehly (University of Bordeaux and Inria, France) presented at ECCOMAS 2022

temporary solution



$1\,\mathrm{cm}^3$ tissue like this: 10^7 cells,
$3.7 \cdot 10^{11}$ tets, 16 TB storage
$\Rightarrow$ 3 PB for a whole heart?

Results: exploiting vector capabilities of CPUs

Transmembrane currents are governed by large systems of nonlinear ODEs. Integration requires extensive evaluation of math functions, but this part of the code is embarrassingly parallel.

- modern Intel CPUs can operate on multiple data simultaneously (SIMD)
- but compilers are difficult to coerce into using this
- they even undo manual vectorization of code

our strategy

- model equations are written in a special-purpose language (EasyML)
- we developed a program that compiles these into MLIR
- using explicit vector operations and vector data
- thus we force the compiler to produce vectorized machine code

results

- upto a factor 15 speedup depending on model complexity

work of Tiago Trevisan Jost, Arun Thangamani, Vincent Loechner, Stéphane Genaud, and Bérenger Bramas (ICube lab, University of Strasbourg and Inria, France) to be presented at PACT 2022

Combining uncore frequency and dynamic power capping to save power

- Computational power increase
  - Frontier 1.1 EFlop/s
- Large power consumption
  - Frontier 21.1 MW
  - $\sim$ 12000 households
- DoE limit on power budget for exascale machines ($20 - 30\ MW$)
  - Techniques to reduce power consumption (DVFS, UFS, ...)
  - Power capping

work of Amina Guermouche (Inria STORM team) presented at IEEE IPDPS 2022

Combining uncore frequency and dynamic power capping to save power

- Designed tool: DUFP
  - Adapt the power cap to the application needs
  - Handle a user-defined tolerated slowdown
  - Save power without increasing energy consumption.
- Results:
  - Up to 5.56 % power savings with less than 1 % performance loss (0.85 % slowdown)
  - Up to 8.76 % power savings with less than 5 % performance loss (2.32 % slowdown)
  - Best energy savings with 0 % tolerated slowdown
  - Best power savings with no energy loss with 10 % tolerated slowdown

Amina Guermouche, `https://hal.archives-ouvertes.fr/hal-03563120/`

Final words

- follow us on **microcard.eu** / LinkedIn / Twitter
- project workshop on 6 and 7 July 2022 at Liryc + online
- 2 open positions in Bordeaux on mesh adaptation
- 1 open position in Karlsruhe on code integration