jh220041 Co-PI's: Kengo Nakajima (University of Tokyo), Xing Cai (Simula Research Laboratory, Norway)

High resolution simulation of cardiac electrophysiology on realistic whole-heart geometries

Purpose & Organization: International Project under Japan-Norway Collaboration

The intricate electrical activities in the heart are vital. Despite the progresses in mathematical modeling and supercomputing applicable to cardiac electrophysiology, further advances are needed in the realism of the mathematical models, together with simulation resolutions that can help reproduce the electrophysiological fidelity. Thus, simulators of electrophysiology must effectively use modern supercomputers to do "in-silico" experiments with very short turn-around times. Moreover, the simulators must handle realistic geometries and physiological properties, instead of only using simplified geometries and regular computational meshes. The main purpose of this project is to consolidate the existing expertise of the project partners on large-scale simulations using realistic geometries and unstructured meshes, while further improving the performance with more robust numerical algorithms, optimised parallel programming and use of non-x86 architecture (A64FX processors) and accelerator (Nvidia GPUs).

Information Technology Center, The University of Tokyo
- Kengo Nakajima (HPC, Algorithms), Co-PI
- Akihiro Ida (Algorithms) (⇒JAMSTEC)
- Toshihiro Hanawa (HPC)
- Masatoshi Kawai (HPC, Algorithms)
- Tetsuya Hoshino (HPC)
- Yohei Miki (HPC)
- Masaharu Matsumoto (HPC) (⇒Fukushima University)

NVIDIA Japan
- Akira Naruse (HPC)

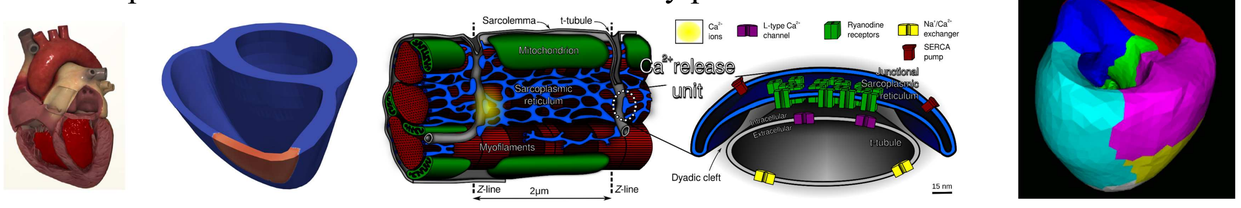
Simula Research Laboratory, Norway
- Xing Cai (Simulations, Modeling, HPC), Co-PI (Norway)
- Glenn Terje Lines (Modeling) (Norway)
- Johannes Langguth (HPC) (Germany)
- Jonas van den Brink (Simulations) (Netherlands)
- Kristian Gregorius Hustad (HPC) (Norway)
- Hermenegild Arevalo (Simulations) (USA)
- James Trotter (HPC) (Norway)
- Aadarsh Bussooa (HPC) (Mauritius)
- María Hernández Mesa (Simulations) (Spain)
- Lena Myklebust (Simulations) (Norway)
- Lisa Pankewitz (Simulations) (Germany)

3D Simulator for Cardiac Electrophysiology

- Mathematical model: a 3D nonlinear reaction-diffusion equation

$$\frac{\partial V_m}{\partial t} = \frac{-I_{\text{ion}}}{C_m} + \nabla \cdot (\mathbf{D} \nabla V_m)$$

- Operator splitting results in a "PDE" part and an "ODE" part
  - ✓PDE part: 3D diffusion equation for V_m (membrane potential) with an inhomogeneous and anisotropic conductivity tensor D
  - ✓ODE part: a system of nonlinear ODEs to model transmembrane ionic current I_{ion}
- Unstructured tetrahedral mesh used to represent the heart geometry
  - ✓ Mesh partitioning used for distributed-memory (MPI) parallelization
  - ✓ OpenMP threads used for shared-memory parallelization



- The PDE part: 3D diffusion equation with variable coefficient
  - ✓ Fully explicit temporal discretization (Forward Euler scheme)
  - ✓ In space, the diffusion equation is discretized by a cell-centered FVM: 5pt. Stencil
  - ✓ During each time step, the PDE work translates to a sparse matrix-vector multiply (SpMV), where the vector contains the membrane potential values at all the computational cell centers
  - ✓ Unstructured mesh partitioning & MPI parallelization of the SpMV
- The ODE part: a system of nonlinear ordinary different equations
  - ✓ The ODE system (up to 39 state variables) applies to each computational cell
  - ✓ Explicit time integration (Forward Euler or Generalized Rush-Larsen scheme)
  - ✓ During each time step, the ODE work is to advance the ODE system separately on each computational cell
  - ✓ The ODE work is embarrassingly parallel, follows the mesh partitioning for work distribution
- Strategies of performance enhancement in the JHPCN project
  - ✓ Code optimiation, parallelization overhead reduction, more advanced numerical algorithm

Research Plan: FY.2022

Development of Stable Algorithm for solving PDE combined with ODE (Cai, Trotter, Hustad, Lines) (OBCX, Odyssey)

In FY.2021, we developed an implicit solver of the 3D diffusion equation used in the overall mathematical model of cardiac electrophysiology. The implicit version (*Implicit Simulator*) solves a sparse linear system per time step. Although more computational work is needed (where the explicit version (*Explicit Simulator*) requires only one sparse matrix-vector multiply: SpMV), the implicit solver is more numerically stable and thus allows much larger time steps. In FY.2022, we will further investigate the implicit solver's stability and accuracy, while ensuring its performance on the A64FX processors (using related knowledge) and also porting it to using multiple Nvidia GPUs. Here, integration of a parallel AMG preconditioner (see below) will be vital. For the ODE part of the simulator, we will also investigate the use of more advanced numerical strategies than the forward Euler scheme, for higher accuracy and the possibility of adopting larger time steps.

AMG Preconditioning for Implicit Solver (Xing, Trotter, Kawai, Ida, Matsumoto) (OBCX, Odyssey)

In FY.2021, we confirmed the effect of an AMG preconditioned CG(AMG-CG) method on the implicit solver developed by Simula. Because the target size of realistic simulations is huge, it is necessary to parallelize AMG in FY.2022 for solving such problems efficiently. The challenges in parallelizing the AMG preconditioner are the construction of a coarse grid correction, and a Gauss-Seidel smoother. We will use a multi-coloring technique to deal with the parallelization of coarse grid collection and GS smoother. In addition, we will apply low-precision floating-points such as FP32 and FP16 to speed up AMG.

Further Optimization on OBCX & Odyssey (Nakajima, Hoshino, Hanawa, Bussooa) (OBCX, Odyssey)

We have been focusing on optimization of the codes on OFP with Intel Xeon Phi. In FY.2021, we started to use Wisteria/BDEC-01(Odyssey) with A64FX, and we will focus on optimization of the codes (both Explicit and Implicit) using Odyssey in FY.2022. Generally, SIMD optimization on OFP will work well on Odyssey. The *Implicit Simulator* is based on finite-volume method (FVM) with cell-centered tetrahedral meshes, where up to 16 nonzero off-diagonal components for each row in the sparse coefficient matrices, and it adopts ELLPACK matrix storage, which is proved to extract high performance of A64FX. In FY.2022, more sophisticated methods for SIMD optimization, such as SELL-C-σ will be evaluated. In FY.2021, we conducted preliminary studies for communication-computation overlapping (CC-Overlapping) for both implicit and explicit methods. While CC-Overlapping has been mainly applied to explicit time-marching and SpMV procedures, we are developing new methods for implicit methods, especially for iterative linear solvers with sophisticated preconditioning methods, such as IC/ILU and AMG. We will continue to develop such methods in FY.2022, and implement them in the Implicit Simulator. While our codes are based on FP64, we conducted preliminary studies for utilization of FP32 and FP16 in FY.2021, and it was effective for reducing computation time with keeping accuracy. In FY.2022, we will evaluate the effects of FP32 and FP16 on both of Explicit and Implicit Simulators. These optimizations are also applied to the codes on OBCX.

Porting to GPU Cluster (Aquarius) (Cai, Langguth, Miki, Naruse, Hoshino) (Aquarius)

Our initial motivation for this project was porting the codes to OFP, and now we are switching to Odyssey. Although we have been focusing on general purpose multicore/manycore CPU's, the Explicit Simulator has been ported to NVIDA's GPU's using CUDA. Current version of compilers (NVFORTRAN, NVC, NVC++) in the NVIDIA HPC Software Development Kit (SDK) support various types of capabilities, such as target offloading without explicit directives. This type of capability is very helpful for porting codes with complicated procedures, such as the Implicit Simulator. In FY.2022, we will port both the Explicit and Implicit Simulators to Aquarius using NVIDIA HPC SDK, and evaluate the performance. We also compare the performance of the codes using NVIDIA HPC SDK with that of codes written in CUDA and in OpenACC. This comparison will be done by benchmark codes.

Large-scale Simulations using OBCX & Odyssey (Cai, Brink, Avevalo, Hernandez, Myklebust, Pankewitz) (OBCX, Odyssey)

We have recently prepared a set of unstructured computational mesh that model the human heart. The largest mesh has 1.47 billion computational elements. Simulations using this set of realistic meshes will be carried out in FY.2022 on OBCX and Odyssey using the optimized code.

Odyssey
- 7,680x A64FX
- 25.9 PFLOPS Peak
- 20th in 59th TOP500 (June, 2022)

Aquarius
- 360x NVIDIA A100
- 90x Intel Icelake
- 7.2 PFLOPS Peak
- 115th in 59th TOP500 (June, 2022)

Contact: Kengo Nakajima (nakajima@cc.u-toyo.ac.jp)
Information Technology Center (ITC), The University of Tokyo