jh180024

Physiologically realistic study of subcellular calcium dynamics with nanometer resolution

Kengo Nakajima（The University of Tokyo）

Abstract
This project aims to combine advanced mathematical modeling and very large-scale numerical simulations for studying subcellular calcium dynamics, which are of vital importance for the function of the heart. By incorporating realistic geometries of the calcium release units (CRUs) and their microanatomical structures, the numerical experimentation that is enabled by this joint Japan-Norway JHPCN project will help to improve the realism of state-of-the-art mathematical models of subcellular calcium signaling, allowing detailed studies of the impact of disease-driven structural changes on the excitation-contraction coupling in the heart. At the same time, the associated challenge of computational capacity and efficiency will be addressed by algorithmic improvements and optimization of the parallel simulator on various coding levels. During the first year of this JHPCN project, a large number of small and medium scale numerical experiments (for investigating the behavior of both a single CRU and multiple CRUs) have been carried out on the Oakforest-PACS system. The obtained results include a quantitative understanding of the impact of sarcoplasmic reticulum load on the process of calcium induced calcium release, validating the mathematical model adopted. Moreover, the parallel simulator has undergone a substantial restructuring and improvement process, including reduction of memory footprint and traffic, optimization of MPI communication and mixed OpenMP-MPI programming. The enhanced simulator will form the basis for further code optimization and tuning, with the main future goal of running very large-scale simulations of subcellular calcium dynamics using a large number of compute nodes on the Oakforest-PACS system, during the second year of this project.

1. Basic Information

(1) Collaborating JHPCN Centers

Information Technology Center, The University of Tokyo

(2) Research Areas

- ■ Very large-scale numerical computation
- ■ Very large-scale data processing
- □ Very large capacity network technology
- □ Very large-scale information systems

(3) Roles of Project Members

Kengo Nakajima (U Tokyo): Project administration, numerical algorithms and parallel programming.

Xing Cai (Simula/Norway): Numerical algorithms, code parallelization and optimization, as well as project coordination together with Prof. Nakajima.

Glenn Terje Lines (Simula/Norway): Cardiac electrophysiology, mathematical modeling and running simulations.

Akihiro Ida (U Tokyo): Numerical algorithms and parallel programming.

Toshihiro Hanawa (U Tokyo): Code parallelization, profiling and optimization.

Masatoshi Kawai (U Tokyo): Numerical algorithms and parallel programming.

Tetsuya Hoshino (U Tokyo): Code parallelization, profiling and optimization.

Chad Jarvis (Simula/Norway): Code parallelization, profiling and optimization, as well as running simulations.

Johannes Langguth (Simula/Norway): Code parallelization, profiling and optimization.

Jonas van den Brink (Simula/Norway): Preparation of geometries and physiological parameters for subcellular simulations.

2. Purpose and Significance of the Research

Synchronous and stable calcium handling is vital for the normal heart contraction. During a heartbeat, calcium is released from over 10,000 calcium release units (CRUs) inside each heart cell, with structural details at the nanometer scale. State-of-the-art simulations of subcellular calcium dynamics, however, have insufficient physiological realism due to two related reasons. First, the 3D geometries of the calcium release units are not properly resolved. Even with realistic data produced by advanced biomedical imaging technique, the lack of high-performance parallel code prevents simulations with the desirable spatial resolution. Second, due to the lack of both performant simulators and access to top-notch supercomputers, most computational cardiologists only study a tiny piece of the cell covering one or a few calcium release units. Such small-scale simulations cannot reveal the true interactions between calcium release units, giving another reason for insufficient physiological realism. All these factors seriously hamper the research work on understanding, for example, the impact of disease-driven structural changes on the excitation-contraction coupling within the heart.

The software starting point of this project was an experimental subcellular simulator previously developed by the Norwegian partner. Despite its potential capability of incorporating sufficient physiological realism, the existing simulator could not fully utilize modern processors due to e.g. inefficient data structures, performance-unfriendly loop nests and suboptimal parallelization. Therefore, a restructuring and optimization of the experimental simulator was urgent.

With help of an efficient simulator (to be developed in this project) and very large-scale simulations thus enabled, this project will consolidate a multi-scale mathematical model that gives a physiologically accurate description of healthy and pathological calcium releases, thereby advancing the scientific understanding of subcellular calcium dynamics. The work related to optimizing the subcellular simulator for Oakforest-PACS will also produce new knowledge about efficiently coding and parallelizing multiple inter-tangled stencil computations for the Knights Landing architecture. Additionally, important experience will also be obtained on using high-speed file cache systems with respect to in-situ huge-scale data analysis.

3. Significance as a JHPCN Joint Research Project

The significance of this JHPCN joint research project has two aspects. First, UTokyo has world-leading expertise in implementing and optimizing advanced numerical code for real-world applications to run on cutting-edge supercomputers. Such hands-on experience in supercomputing is lacking for the Norwegian partner. Second, the Oakforest-PACS system is of a suitable size for achieving the ambitious goal of this project, whereas access to world-leading supercomputers has traditionally been very scarce for the Norwegian partner. The high-speed file cache systems available at UTokyo also provide new possibilities of in-situ huge-scale data analysis.

4. Outline of the Research Achievements up to FY2017

N/A

5. Details of FY2018 Research Achievements

To properly describe the research achievements, it is necessary to first briefly describe the physiology, mathematical model, and numerical algorithm involved.

The tiny calcium release units (CRUs) are of irregular shapes, as well as their internal compartments and channels. The actual geometries can be captured by advanced medical imaging technique. The sarcoplasmic reticulum (SR) within each CRU, which is its internal calcium storage, consists of multiple compartments, such as NSR and JSR. Moreover, the dyadic cleft sits between JSR and T-tubule, whereas the exterior of each CRU is cytosol. The mathematical model thus divides the entire solution domain into, e.g., five physiological domains: NSR, JSR, cleft, T-tubule & cytosol. The primary calcium concentration and the additional calcium buffer concentrations are considered as different species living in all the physiological domains, expressed as functions of space coordinates and time. Most of the species are diffusive, thus modeled by 3D reaction-diffusion equations, where the reaction terms describe the interaction between the different species. On the surface of the SR there are calcium sensitive channels, called ryanodine receptors (RyRs). The opening and closing of each channel is considered as stochastic, for which a higher calcium concentration will give rise to a higher probability of channel opening. While open, a small influx of calcium from the outside will trigger a much larger calcium release from the SR. This important process is called calcium induced calcium release.

The numerical algorithm assumes a 3D box-shaped solution domain covered by a uniform 3D mesh of voxels. The irregular geometries of the CRUs (produced by high-resolution imaging) are imbedded into the mesh, where each voxel belongs uniquely to one of the physiological domains. Explicit time stepping, combined with operator splitting, is adopted. This also means that the diffusion computation is separated from the reaction computation per time step. We remark that a diffusive species may have different diffusion constants in the different physiological domains. The diffusibility also varies between the different species. Numerical solutions are sought at the center of each voxel. The diffusion terms are discretized by a finite volume method, which results in a 7-point computational stencil.

The research activities carried out in FY 2018 can be roughly divided into five major tasks as follows. (The first, second and fourth tasks were done during the first half of FY2018, whereas the third and fifth tasks during the second half.)

(1) OpenMP parallelization

As our software starting point, the existing simulator was primitively implemented, in the sense that only MPI parallelization was programmed. Moreover, the incurred MPI communication (needed for updating each diffusive species per time step) was not overlapped with computation. In particular, the lack of shared memory-based parallelization (via threads) is a clear obstacle to fully utilizing the Oakforest-PACS system, because an MPI-all-the-way approach will very likely fail to scale when using many Knights Landing (KNL) nodes each with 68 physical cores. Our first effort of code improvement was thus focused on incorporating OpenMP parallelization into the various computational tasks. The required effort

is much more than simply inserting #omp pragma parallel for (often with the collapse clause) in front of the loop nests. We have painstakingly avoided pitfalls such as race conditions and false sharing. This is often done together with appropriate code restructuring. The efficiency of the OpenMP-enabled new version is demonstrated in Table 1, where we have used one KNL node on the Oakforest-PACS system. The entire 3D solution domain is of a physical size of $(2\mu m)^3$, imbedding one CRU of realistic geometry and RyR details. A 168x168x168 global computational mesh is used, with each voxel having a spatial resolution of $(12nm)^3$. For this small test problem, 10000 time steps are used to simulate 1ms of calcium dynamics. The new version uses 256 OpenMP threads, whereas the original version uses 256 MPI processes (with an 8x8x4 partitioning of the global domain). We can see from Table 1 that the largest benefit of the OpenMP version is avoiding the overhead of explicit inter-process communication. The time usage of the OpenMP version on the diffusion and reaction computations is comparable with the MPI counterpart, where the latter has the benefit of better data locality. The reason for the considerably slower speed of the "Add increment" computation will require a thorough study during the second year of the project.

Table 1. Time consumption (in seconds) of the new (OpenMP) and original (MPI) versions of the simulator.

	OpenMP	Average per MPI process	Max per MPI process	Min per MPI process
Communication	N/A	25.69	27.78	3.56
Diffusion comp.	31.17	27.18	29.57	25.58
Reaction comp.	16.86	14.50	15.46	14.34
Add increment	12.21	3.94	4.68	3.45
Whole simulation	64.05	79.70	79.76	79.60

(2) Reducing memory footprint

For each diffusive species function $u_s(x,y,z,t)$, its own spatial interaction can be extracted from the original reaction-diffusion equation as the following 3D pure diffusion equation:

$$\frac{\partial u_s}{\partial t} = \nabla \cdot (\alpha_s(x,y,z)\nabla u_s)$$

where $\alpha_s(x,y,z)$ is a generalized variable coefficient incorporating the different diffusion coefficients valid in different physiological domains, as well as the possible flux between a pair of connecting physiological domains. For each diffusive species, the original simulator adopted three separate 3D arrays, each of dimension $n_x x n_y x n_z$, to store the discrete α values at the face midpoints between all pairs of neighboring computational voxels in the x,y,z directions, respectively. More specifically, at each interior voxel, the following code (termed "coefficient-array" implementation) calculates the increment per time step:

```
du[xi][yi][zi] =
(alpha_x[xi][yi][zi]*(u[xi-1][yi][zi]-u[xi][yi][zi])
+ alpha_x[xi+1][yi][zi]*(u[xi+1][yi][zi]-u[xi][yi][zi])
+ alpha_y[xi][yi][zi]*(u[xi][yi-1][zi]-u[xi][yi][zi])
+ alpha_y[xi][yi+1][zi]*(u[xi][yi+1][zi]-u[xi][yi][zi])
+ alpha_z[xi][yi][zi]*(u[xi][yi][zi]-u[xi][yi][zi-1])
+ alpha_z[xi][yi][zi+1]*(u[xi][yi][zi+1]-
u[xi][yi][zi]))/h/h*dt
```

Actually, these alpha arrays are strictly speaking not needed. This is because the diffusion coefficient for a species is constant within one physiological domain. In the case of five physiological domain types, the effective diffusion coefficient on any face midpoint between two neighboring voxels can be easily found by a 5x5 lookup table (which is also symmetric). Consequently, we re-programmed the diffusion computation thus eliminated all the alpha arrays from the simulator. The new kernel for the diffusion computation (a "lookup-table" approach) looks like

```
int di = domain_ids[xi][yi][zi];
int di_xm = domain_ids[xi-1][yi][zi];
int di_xp = domain_ids[xi+1][yi][zi];
int di_ym = domain_ids[xi][yi-1][zi];
int di_yp = domain_ids[xi][yi+1][zi];
int di_zm = domain_ids[xi][yi][zi-1];
int di_zp = domain_ids[x][yi][zi+1];
du[xi][yi][zi] =
(lookup[di][di_xm]*(u[xi-1][yi][zi]-u[xi][yi][zi])
+ lookup[di][di_xp]*(u[xi+1][yi][zi]-u[xi][yi][zi])
+l ookup[di][di_ym]*(u[xi][yi-1][zi]-u[xi][yi][zi])
+ lookup[di][di_yp]*(u[xi][yi+1][zi]-u[xi][yi][zi])
+ lookup[di][di_zm]*(u[xi][yi][zi-1]-u[xi][yi][zi])
+ lookup[di][di_zp]*(u[xi][yi][zi+1]-u[xi][yi][zi]))/h/h*dt
```

An immediate benefit is that the memory footprint of the simulator is considerably reduced. Take for instance the mesh of size 168x168x168. Four diffusive species would have required 4x3x168x168x168x8=455MB (double precision) for the 12 alpha arrays alone.

(3) Explicit code vectorization with AVX-512

The "lookup table" approach, as described above, requires care in coding. While considerably shrinking the memory footprint and the associated volume of memory traffic, the "lookup table" approach induces more instructions to be carried out on the processor level. To offset this potential performance disadvantage, we explicitly vectorized the diffusion and reaction computations using AVX-512 intrinsics. (This is because experiments show that compiler-enabled auto vectorization is not sufficiently effective.) In addition, we also restructured the code to enable overlapping MPI communication with computation.

We note that manual vectorization by using AVX-512 intrinsics also benefits the "coefficient array" implementation approach, for which we will show below a code segment. (The details of manual vectorization for the "lookup table" implementation are too extensive to be included in this report.)

```
__m512d uc, duc, um, up, uu, ud, ub, uf;
__m512d coefm, coefp, coefu, coefd, coefb, coeff;
__m512d tm, tp, tu, td, tb, tf, s1, s2, s3, s4, s5, s6;

for (xi=1; xi<nx-1; xi++) {
  for (yi=1; yi<ny-1; yi++) {
    for (zi=1; zi<nz-1; zi+=8) {
      pos= xi*x_offset + yi*y_offset + zi;
      uc = _mm512_loadu_pd(&u[pos]);
      duc= _mm512_loadu_pd(&du[pos]);

      um = _mm512_loadu_pd(&u[pos-1]);
      up = _mm512_loadu_pd(&u[pos+1]);
      coefm = _mm512_loadu_pd(&alpha_z[pos-1]);
      coefp = _mm512_loadu_pd(&alpha_z[pos]);
      uu = _mm512_loadu_pd(&u[pos-y_offset]);
      ud = _mm512_loadu_pd(&u[pos+y_offset]);
      coefu = _mm512_loadu_pd(&alpha_y[pos-y_offset]);
      coefd = _mm512_loadu_pd(&alpha_y[pos]);
      ub = _mm512_loadu_pd(&u[pos-x_offset]);
      uf = _mm512_loadu_pd(&u[pos+x_offset]);
      coefb = _mm512_loadu_pd(&alpha_x[pos-x_offset]);
      coeff = _mm512_loadu_pd(&alpha_x[pos]);

      tm = _mm512_mul_pd(coefm, _mm512_sub_pd(um, uc));
      tp = _mm512_mul_pd(coefp, _mm512_sub_pd(up, uc));
      tu = _mm512_mul_pd(coefu, _mm512_sub_pd(uu, uc));
      td = _mm512_mul_pd(coefd, _mm512_sub_pd(ud, uc));
      tb = _mm512_mul_pd(coefb, _mm512_sub_pd(ub, uc));
      tf = _mm512_mul_pd(coeff, _mm512_sub_pd(uf, uc));

      s1 = _mm512_add_pd(tm, tp); s2 = _mm512_add_pd(tu, td);
      s3 = _mm512_add_pd(tb, tf); s4 = _mm512_add_pd(s1, s2);
      s5 = _mm512_add_pd(s3, s4); s6 = _mm512_add_pd(s5, duc);
      _mm512_storeu_pd(&du[pos], s6);
    }
  }
}
```

In Table 2, we report some time measurements of the diffusion and reaction computations, denoted respectively by T_D and T_R. These are associated with different programming and vectorization approaches. Particularly, CA denotes the "coefficient array" programming

approach, and LUT denotes the "lookup table" approach, whereas auto denotes compiler auto vectorization and man denotes manual AVX-512 vectorization. (For the "lookup table" approach, there are two versions of manual vectorization, with version "man2" involving additional code optimizations than in version "man1".) We can observe that, with manual AVX-512 vectorization, the "lookup table" approach (in particular LUT_man2) clearly wins over the other versions.

Table 2. Time measurements of diffusion and reaction computations using auto and manual vectorizations

Global computational mesh: $672 \times 672 \times 168$, time steps: 1000										
Version	CA_auto		LUT_auto		CA_man		LUT_man1		LUT_man2	
MPI procs	T_R	T_D	T_R	T_D	T_R	T_D	T_R	T_D	T_R	T_D
$4 \times 2 \times 2 = 16$	73.7	127.7	73.7	184.7	24.8	102.5	24.6	84.6	24.5	60.1
$4 \times 4 \times 2 = 32$	36.9	65.4	36.9	92.0	12.4	50.7	12.4	42.7	12.4	30.5
$4 \times 4 \times 4 = 64$	18.0	48.8	18.0	59.7	6.7	36.1	6.7	27.7	6.7	21.5
$8 \times 4 \times 4 = 128$	11.9	45.9	12.0	51.1	4.6	37.2	4.5	24.0	4.7	20.7
$8 \times 8 \times 4 = 256$	10.7	41.2	10.6	43.2	4.8	36.9	4.5	24.3	4.4	21.7

(4) Quantitative study of the impact of SR load

Many factors can affect the calcium-induced calcium release process, including geometrical configuration, RyR sensitivity and initial SR concentration. A quantitative understanding of the impact of each factor requires a series of simulations. The actual number of simulations needed is considerably large (an ensemble per data point) due to the stochastic property of channel closing and opening. As an example, we investigate the impact of varying the SR concentration (also called SR load). We use the same single-CRU configuration as before, with the 168x168x168 uniform mesh. Figure 1 shows the SR in blue and green while the individual RyRs are shown in pink. The cleft space is right under the RyRs, where the outer membrane has been removed here to reveal the location of the RyRs.

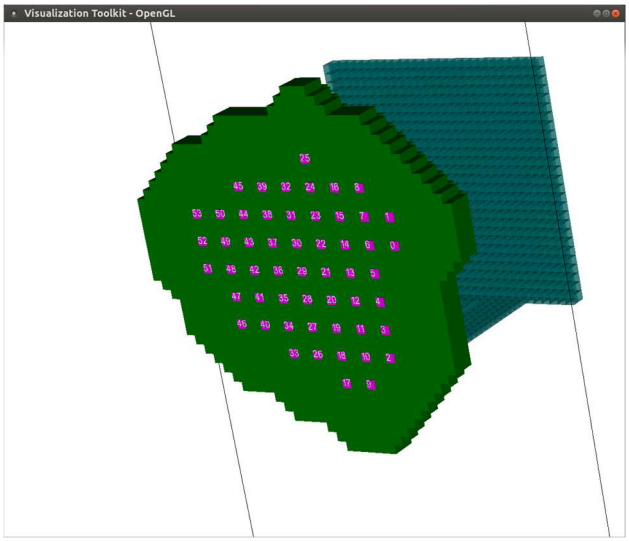

Figure 1. Detailed geometry of SR and RyR distribution

It is known that cells behave differently under different SR loads. A higher concentration generally leads to a larger fidelity and longer sparks. This behavior is often modeled phenomenologically by assuming that the open probability of the RyR is a function of the SR load. However, there is no data to support the notion of a calcium sensor on the SR side. In our mathematical model, we only include calcium sensitivity on the cytosolic side. Theoretically, this could still explain the dependency on the SR load, albeit through an indirect route: A larger SR load would upon release yield a higher concentration also in the cleft space, and also it should take longer time to drain the SR. Combined, this could contribute to both a higher fidelity and longer "calcium sparks".

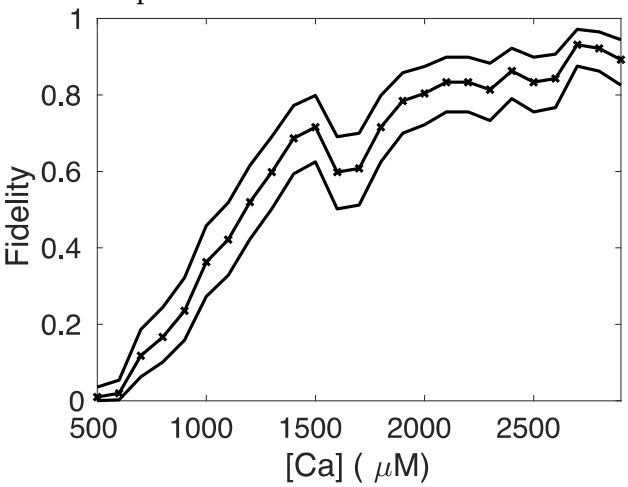

Figure 2. Study of the impact of SR load using 100 runs

To test whether the model has this emergent property, we have performed a series of simulations where the SR load is varied from

500μM to 2900μM. Since the results are stochastic we have repeated each configuration several times. In Figure 2, we see the result of 100 runs. The centerline shows the expected fidelity, while the outer lines indicate the 95% confidence intervals. Clearly there is a strong dependency on the SR load, and the fidelity seems to increase in a monotonic fashion and saturate at high calcium levels. The dip around 1650μM seems to break this trend, but the confidence intervals indicate that the true function still could be monotonic. We have then increased the number of runs to 1000 in order to narrow the confidence interval. The results are shown in Figure 3. Clearly the tip was caused by the relatively low sampling number, and with 1000 runs we get a much more accurate result.

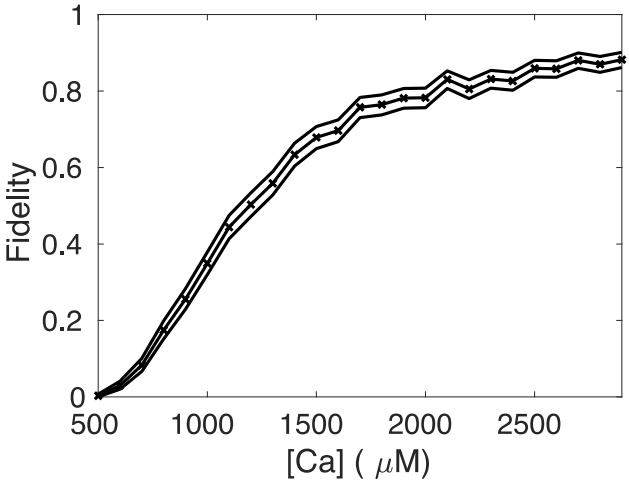

Figure 3. Study of the SR load impact using 1000 runs

It should be noted that these small scale single-CRU simulations are necessary for validating the mathematical model, as a preparation step for very large-scale simulations that involve many CRUs.

(5) Simulations involving many CRUs

Another topic of great research interest is the impact due to disease-driven changes of the microanatomical structures inside the cardiac cells. For example, chronic heart failure can lead to a disintegration of the calcium release units (CRUs), in terms of a change in the number, distribution and calcium release channels. For this research topic, it is very important to adopt simulations that involve many CRUs with realistic geometries and microanatomical details.

In Figure 4, we show three snapshots, which are 2D slices of the 3D computational domain. The snapshots are from a simulation that involves 93 CRUs. (Such many-CRU simulations were not done before by the Norwegian partner of this project, due to inefficient code and no sustained access to supercomputers.) The entire 3D domain is of dimension 10x10x2 μm. The grey outlines in the plots denote the realistic boundaries of the CRUs, whereas the different colors denote the different concentration levels of the primary calcium species. Specifically, the snapshots show a calcium wave that can occur under pathological conditions. Time goes from left to right. Some of the RyRs are activated by the initial condition (left plot), then in the middle plot this activation has spread, and in the right plot the activation covers most of the space.

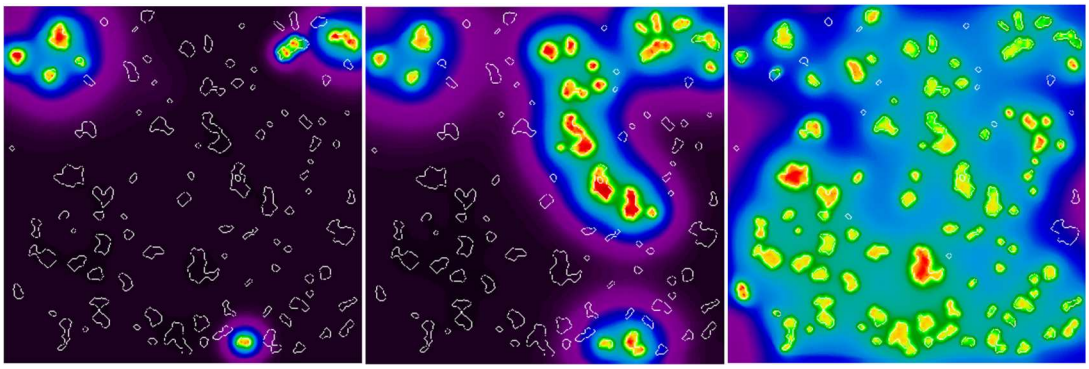

Figure 4. Snapshots of a simulation that involves 93 CRUs

6. Progress of FY2018 and Future Prospect

The work carried out in FY2018 was in good agreement with the original project plan. For the second year of the project, while continuing with small and medium scale simulations, our main focus will be on ensuring good performance of the parallel simulator running on many nodes of the Oakforest-PACS system. We plan to adopt detailed performance profiling to assist further code analysis and optimization. For the scale-out purpose (using many KNL nodes), we will also try to improve the MPI-specific parts in the code. One specific topic is to investigate about a suitable combination of MPI processes and OpenMP threads. Another topic is "in situ" data analysis (simultaneously with an ongoing simulation) with help of high-speed file cache systems available at UTokyo.

The physiological target of research for the upcoming very large-scale simulations will be to simulate a much larger portion of one cardiac cell (even an entire cell), where all the CRUs have realistic geometries provided by the latest technique of super-resolution microscopy. These simulations will allow us to explore multiscale phenomena that emerge from the interplay between different CRUs.

7. List of Publications and Presentations

(1) Journal Papers

None so far, but work has started on drafting one journal submission to report the scientific findings related to multi-calcium-release-unit simulations and new validation results of the mathematical model.

(2) Conference Papers

Chad Jarvis, Glenn Terje Lines, Johannes Langguth, Kengo Nakajima, Xing Cai. *Combining algorithmic rethinking and AVX-512 intrinsics for efficient simulation of subcellular calcium signaling.* Proceedings of ICCS 2019 Conference, 2019.

(3) Oral Presentations

Xing Cai. *Heterogeneous Computing: Programming, Performance and Applications*. Invited keynote talk at CoSaS 2018 Symposium, September 5-7, 2018, Erlangen, Germany.

Johannes Langguth, Hermenegild Arevalo, Chad Jarvis, Xing Cai. *Towards detailed organ-scale simulations in cardiac electrophysiology*. Poster presentation at CoSaS 2018 Symposium, September 5-7, 2018, Erlangen, Germany.

Xing Cai. *Use of modern processor architectures for computing the electrical activity in the heart*. Invited webinar for the Applied Mathematics Special Interest Group at Schlumberger, March 13, 2019.

Xing Cai. *Heterogeneous computing for cardiac electrophysiology*. Invited keynote talk at EFFECT Workshop, April 25-26, 2019, University of Tromsø, Norway.

(4) Others

N/A