A Multi-Scale Heart Simulation on Massively Parallel Computers

Akira Hosoi
FUJITSU LTD.
Email: hosoi.akira@jp.fujitsu.com

Takumi Washio
and Jun-ichi Okada
Graduate School of Frontier Sciences,
The University of Tokyo
Email: {washio,okada}@sml.k.u-tokyo.ac.jp

Yoshimasa Kadooka
FUJITSU LTD.
Email: y-kadooka@jp.fujitsu.com

Kengo Nakajima
Supercomputing Division,
Information Technology Center, The University of Tokyo
Email: nakajima@cc.u-tokyo.ac.jp

Toshiaki Hisada
Graduate School of Frontier Sciences,
The University of Tokyo
Email: hisada@mech.t.u-tokyo.ac.jp

***Abstract*—To understand the macroscopic function of the human heart based on sub-cellular microscopic events, multi-scale analysis is indispensable. Our heart simulator uses the so-called homogenization method, where both the human heart and the myocardial cells are modeled and solved simultaneously by the finite element method. Because the contraction and deformation of each finite element in the heart model are governed by their respective cell model, the NDOF of all the cells becomes prohibitively large. Furthermore, the phenomena are highly nonlinear and transient. This challenging problem has been tackled by our group for many years, and a novel algorithm to accelerate the computation was implemented in the code. We have recently tested its performance using the T2K Open Supercomputer (Tokyo). A pulsation of the heart with a total NDOF of 160 million was successfully simulated using 6144 CPU cores in ten hours. Scalability and other computational performances were measured and are discussed.**

I. Introduction

In the post-genome era, the integration of molecular and cellular findings from studies on the functions of organs and individuals is considered an important field in medical science and physiology. Computational modeling plays a central role in this field, which is referred to as Physiome[1][2]. However, despite advancements in computational science, it remains difficult to accomplish this task. In addition to embracing multiple disciplines, including electricity, physical chemistry, solid mechanics, and fluid dynamics, computational modeling must also integrate events over a wide range of scales.

Our group, which includes clinical practitioners, has been tackling this problem for several years, with the focus on the human heart. Our research has now reached the point where it can be useful in clinical applications[3][4]. Our simulations have already been utilized in the design of an implantable cardioverter defibrillator (ICD) and have resulted in a breakthrough in non-invasive defibrillation. Clinical investigation of the new ICD is scheduled for 2012. The morphology of our heart model was reconstructed from human multi-detector computed tomography data and discretized using the finite element method. The model consists of a structure part and a fluid part that can be solved simultaneously using the strong coupling method developed by one of the coauthors[5], thus successfully solving the so-called fluid structure interaction problem of the heart[6]. However, a more important aspect of the simulation involves modeling the underlying mechanism driving the myocardium, i.e., the origin of the pulsation of the heart, which includes electrophysiological regulation and cross-bridge kinetics in the cardiac cells. To integrate such microscopic phenomena with the macroscopic function of the organ in a seamless manner, the cardiac cells are also modeled using the finite element method, based on the cell physiology for every finite element in the heart model. The mathematical linkage is realized using the so-called homogenization method. All the cell models and the heart model[4][7] are then solved simultaneously, because the instantaneous states of the macroscopic model, such as the strains and strain rates over the heart wall, also regulate each cell response. Nevertheless, it is apparent that the NDOF of all the cell models becomes prohibitively large and far exceeds that of the heart model. Moreover, the phenomena are highly nonlinear and transient.

In the following sections, we show the essence of the physical governing equations and the simultaneous equations resulting from the homogenization method (Section II), and how the parallel algorithm can be created by taking advantage of the features of the simultaneous equations (Section III). Thereafter, we outline the implementation of a program for a massively parallel computer, the T2K Open Supercomputer (Tokyo)(T2K/Tokyo)[8] (Section IV), and discuss the simulation results for further massive parallelization (Section V).

II. Governing Equations and Homogenization Method for Multi-scale Simulation

A. Variational Formulation of the Whole System

The heart consists of muscle (solid) and blood (fluid), the interactions of which cause the pulsation. In our simulator both fields are modeled by the finite element method, and the so-called strong coupling technique is applied so that a stable

SC10 November 2010, New Orleans, Louisiana, USA 978-1- 4244-7558-2/10/$26.00

Fluid Structure Interaction (FSI) analysis can be carried out throughout the pulsation. The essence of our formulation is as follows.

The large deformation of the muscle necessitates an exact approach based on Continuum Mechanics, i.e., a total Lagrangian formulation must be employed. On the other hand, the fluid field is governed by Navier-Stokes equations, which need to be described from moving and deforming coordinates, since the fluid domain largely changes with time. The Arbitrary Lagrangian-Eulerian (ALE) formulation is thus indispensable. The whole system is described by the following variational formulation in weak form:

$$\begin{aligned}\int_{\Omega_f} \boldsymbol{\delta v} \cdot \rho^f \boldsymbol{a}_\chi d\chi + 2\mu \int_{\Omega_f} \boldsymbol{\delta D} : \boldsymbol{D} d\chi \\ - \int_{\Omega_f} p \nabla_\chi \cdot \boldsymbol{\delta v} d\chi - \int_{\Omega_f} \delta p \nabla_\chi \cdot \boldsymbol{v} d\chi \\ - \int_{\Gamma_f} \boldsymbol{\delta v} \cdot \boldsymbol{\tau}_f ds + \int_{\bar{\omega}} \boldsymbol{\delta \bar{u}} \cdot \rho^s \ddot{\bar{\boldsymbol{u}}} d\bar{X} \\ + \int_{\bar{\Omega}} \frac{1}{|\Omega_{\bar{X}}|} \int_{\Omega_{\bar{X}}} \boldsymbol{\delta Z} : \boldsymbol{\Pi} dX d\bar{X} = 0. \qquad (1)\end{aligned}$$

The first two lines express the equation for the fluid part. χ represents the ALE coordinate system on the fluid domain denoted by $\Omega_f = \Omega_f(t)$ at time t. The acceleration of fluid includes an artificial convective term and is expressed as:

$$\boldsymbol{a}(\chi, t) = \frac{\partial \boldsymbol{v}(\chi, t)}{\partial t} + \boldsymbol{c} \cdot \nabla_\chi \boldsymbol{v} \equiv \boldsymbol{a}_\chi \text{ on } \Omega_f(t)$$

where $\boldsymbol{c} = \boldsymbol{v} - \hat{\boldsymbol{v}}$ is the relative velocity observed from the ALE coordinate system with velocity $\hat{\boldsymbol{v}}$. We assume that the fluid is incompressible and Newtonian, with the Cauchy stress tensor given by the deformation velocity tensor $\boldsymbol{D}$ and pressure p as:

$$\boldsymbol{T}_f = -p\boldsymbol{I} + 2\mu\boldsymbol{D}, \ \boldsymbol{D} = \frac{1}{2}\left\{\nabla_\chi \boldsymbol{v} + \nabla_\chi \boldsymbol{v}^T\right\} \text{ on } \Omega_f$$

Γ_f corresponds to the inlets and outlets of the ventricles, with the traction force $\boldsymbol{\tau}_f$ determined through interaction with the circulatory systems of the body, modeled as electrical analog circuits[6]. Note that strong coupling is realized on the fluid-structure interfaces, i.e., in the FEM framework the common velocity nodes are shared on the interface under the non-slip condition of viscous fluids, thereby preserving the mechanical equilibrium and the geometrical compatibility between the two fields [5].

The last two lines represent the equations for the structural part, where X and $\bar{X}$ are the Lagrange coordinates on the macroscopic structural domain $\bar{\Omega}$ and the unit microscopic structural domain $\Omega_{\bar{X}}$ located at $\bar{X}$, respectively. $\bar{u}$ is the displacement of the macroscopic structure, $\ddot{\bar{\boldsymbol{u}}}$ its acceleration, and ρ^s the density thereof. The last term originates from the basic assumption of the homogenization method, which reflects the multi-scale nature of biological tissue, i.e., the muscle consists of and is driven by cardiac cells. This is explained below.

B. Homogenization Method for Multi-scale Simulation

The last term of Eq. (1), rewritten below, represents the virtual work done by the muscle, and the instantaneous equilibrium of the whole system is established through the work done by the fluid, external forces, and its own inertial forces.

$$\delta W_S = \int_{\bar{\Omega}} \frac{1}{|\Omega_{\bar{X}}|} \int_{\Omega_{\bar{X}}} \boldsymbol{\delta Z} : \boldsymbol{\Pi} dX d\bar{X}. \qquad (2)$$

This equation states that the virtual work at a macroscopic point $\bar{X}$ in the muscle is given by the spatial average of the virtual work of the microscopic unit over its domain $\Omega_{\bar{X}}$ located at $\bar{X}$. Here, $\boldsymbol{\Pi} = \boldsymbol{\Pi}(\boldsymbol{Z}, \dot{\boldsymbol{Z}})$ is the first Piola-Kirchhoff stress tensor as a function of the displacement gradient tensor $\boldsymbol{Z} = \partial \boldsymbol{u}/\partial X$ on $\Omega_{\bar{X}}$ and its time derivative.

In the homogenization method[9][10], the microscopic displacement vector is defined as

$$\boldsymbol{u}(X, \bar{X}) = \boldsymbol{w}(X, \bar{X}) + \bar{\boldsymbol{Z}} X,$$

where $\bar{\boldsymbol{Z}} = \partial \bar{\boldsymbol{u}}/\partial \bar{X}$ is the macroscopic displacement gradient tensor at $\bar{X}$ and $\boldsymbol{w}$ is the deviation of the displacement on $\Omega_{\bar{X}}$ due to microscopic structural effects. Therefore, the displacement gradient tensor is given as

$$\boldsymbol{Z} = \frac{\partial \boldsymbol{u}}{\partial X} = \widetilde{\boldsymbol{Z}} + \bar{\boldsymbol{Z}}$$

with $\widetilde{\boldsymbol{Z}} = \partial \boldsymbol{w}/\partial X$ being the microscopic displacement gradient tensor. By assuming that the spatial average of $\boldsymbol{Z}$ on $\Omega_{\bar{X}}$ returns to $\bar{\boldsymbol{Z}}$, the periodicity of $\boldsymbol{w}$ is deduced using the divergence theorem. In the homogenization method it is thus assumed that infinite microscopic units are periodically connected to constitute a material point in the macrostructure. In the case of heart muscle (myocardium), the orientation of the cardiac cells plays an important role in the contraction of the chamber and varies spatially. However, as shown in Section IV, cardiac cells are arrayed in a pseudo-periodic manner if we focus on a very circumscribed domain. This enables us to apply the homogenization method to a multi-scale simulation of the heart.

C. Finite Element Discretization for a Multi-scale Term

In our heart simulator, the equilibrium equation is discretized by the finite element method with tetrahedral elements for the macroscopic domain, i.e., left and right ventricles, and hexahedral elements for the microscopic unit domain, i.e., myocardial cells. The response of a macroscopic finite element is determined by a infinite number of microscopic units which are periodically connected.

Here, we focus on the discretization of Eq. (2) and its derivative for the infinitesimal increment of the displacement vectors. Although the stress tensor $\boldsymbol{\Pi}$ is a function of the velocity gradient tensor $\dot{\boldsymbol{Z}}$ as well as $\boldsymbol{Z}$, due to the viscosity of the myocardium and the stretch rate dependence of the contraction force in cross-bridge kinematics, we show a simplified formulation by assuming that $\boldsymbol{\Pi} = \boldsymbol{\Pi}(\boldsymbol{Z})$ apart from the actual simulator. In addition, for the sake of simplicity, we omit the volume factor $\Omega_{\bar{X}}$ and the notation for the integral domains in the following equations.

In the finite element analysis, the nodal displacement vectors are interpolated on each element and integrations are carried out element by element. We represent the linear relations between the nodal displacement vectors $\{w\}$, $\{\bar{u}\}$ and the spatial derivatives of their interpolations as

$$[\widetilde{Z}] = [\widetilde{B}]\{w\},\ [\bar{Z}] = [\bar{B}]\{\bar{u}\},$$

where the 3 by 3 displacement gradient tensors $\widetilde{\boldsymbol{Z}}$ and $\bar{\boldsymbol{Z}}$ are represented by 9 dimensional vectors. With these B-matrices, the virtual work δW_S is given as follows.

$$\begin{aligned} \delta W_S = & \int\int \{\delta w\}^T [\widetilde{B}]^T [\Pi] dX d\bar{X} \\ & + \int \{\delta \bar{u}\}^T [\bar{B}]^T \int [\Pi] dX d\bar{X} \\ & = \{\delta w\}^T \{f\} + \{\delta \bar{u}\}^T \{\bar{f}\} \end{aligned}$$

Here, the stress tensor is represented by the notation $[\Pi]$ as a 9 dimensional vector, and the equivalent nodal forces are given by

$$\begin{aligned} \{f\} &= \int\int [\widetilde{B}]^T [\Pi] dX d\bar{X}, \\ \{\bar{f}\} &= \int\int [\bar{B}]^T [\Pi] dX d\bar{X}. \end{aligned}$$

Let $[A]$ be the 9 by 9 matrix given by the derivative of $[\Pi]$ with respect to $[Z]$. Then, the following relation holds for the infinitesimal increment.

$$[\Delta\Pi] = [A][\Delta Z].$$

Thus, we obtain

$$\begin{aligned} \{\Delta f\} \quad &= \int\int [\widetilde{B}]^T [A] [\widetilde{B}] \{\Delta w\} dX d\bar{X} \\ &\quad + \int\int [\widetilde{B}]^T [A] dX [\bar{B}] \{\Delta \bar{u}\} d\bar{X}, \\ \{\Delta \bar{f}\} \quad &= \int [\bar{B}]^T \int [A] [\widetilde{B}] \{\Delta w\} dX d\bar{X} \\ &\quad + \int [\bar{B}]^T \int [A] dX [\bar{B}] \{\Delta \bar{u}\} d\bar{X}. \end{aligned}$$

Let $\{\bar{e}_k\}_{k=1,\cdots,n}$ be the tetrahedral elements in $\bar{\Omega}$ and Δw_k and Δf_k be the components of Δw and Δf, respectively, in the microscopic unit model at $\bar{e}_k$. Then, the above increments can be represented as follows.

$$\begin{aligned} \{\Delta f_k\} \quad &= |\bar{e}_k| \int [\widetilde{B}_k]^T [A_k] [\widetilde{B}_k] \{\Delta w_k\} dX \\ &\quad + |\bar{e}_k| \int [\widetilde{B}_k]^T [A_k] dX [\bar{B}_k] \{\Delta \bar{u}\}, \qquad (3) \\ \{\Delta \bar{f}\} \quad &= \textstyle\sum_k |\bar{e}_k| [\bar{B}_k]^T \int [A_k] [\widetilde{B}_k] \{\Delta w_k\} dX \\ &\quad + \textstyle\sum_k |\bar{e}_k| [\bar{B}_k]^T \int [A_k] dX [\bar{B}_k] \{\Delta \bar{u}\}. \qquad (4) \end{aligned}$$

Here, $|\bar{e}_k|$ denotes the volume of the tetrahedral element $\bar{e}_k$, and the integral over $\bar{e}_k$ is approximated with the value at the central integral point $\bar{X}_k$ in $\bar{e}_k$. Furthermore, $[\widetilde{B}_k]$ is constant over the microscopic unit domain $\Omega_{\bar{X}_k}$ and thus, can be placed outside the integral over $\Omega_{\bar{X}_k}$.

III. Strategy for Massively Parallel Computation

The relations in Eqs. (3) and (4) between the infinitesimal increments of the nodal displacements and the equivalent nodal forces determine the nonzero structure of the coefficient matrix in the linear system that appears in the Newton-Raphson process to solve the nonlinear equation originating from Eq. (1). The linear system to be solved can be written in block matrix form as follows.

$$\begin{bmatrix} A_{w_1} & & & C_1 \\ & \ddots & & \vdots \\ & & A_{w_n} & C_n \\ C_1^T & \cdots & C_n^T & \bar{A} \end{bmatrix} \begin{bmatrix} \Delta w_1 \\ \vdots \\ \Delta w_n \\ \Delta \bar{u} \end{bmatrix} = \begin{bmatrix} r_{w_1} \\ \vdots \\ r_{w_n} \\ \bar{r} \end{bmatrix}, \qquad (5)$$

where the right-hand side is composed of the residual vectors for the nonlinear equilibrium equation,

$$\begin{aligned} \{r_{w_k}\} &= -|\bar{e}_k| \int [\widetilde{B}_k]^T [\Pi] dX, \\ \{\bar{r}\} &= \sum_k \{\bar{r}_k\}, \\ \text{where} \quad & \{\bar{r}_k\} = -|\bar{e}_k| [\bar{B}_k]^T \int [\Pi] dX, \end{aligned}$$

and the block matrices are given by

$$\begin{aligned} A_{w_k} &= |\bar{e}_k| \int [\widetilde{B}_k]^T [A_k] [\widetilde{B}_k] dX, \\ C_k &= G_k \bar{B}_k, \\ \text{where} \quad & G_k = \int [\widetilde{B}_k]^T [A_k] dX, \\ \text{and} \quad & \bar{B}_k = |\bar{e}_k| [\bar{B}_k], \\ \bar{A} &= \sum_k \bar{A}_k, \\ \text{where} \quad & \bar{A}_k = |\bar{e}_k| [\bar{B}_k]^T \int [A_k] dX [\bar{B}_k]. \end{aligned}$$

The nonzero structure of the coefficient matrix is depicted in Fig. 1. Note that matrices A_{w_k} and $\bar{A}$ have the random sparse nonzero structure seen in ordinary finite element analysis, and C_k is also sparse in the column direction, with nonzero columns only for those nodes contained in $\bar{e}_k$.

A. Algorithm for Basic Solution

The point of the present simulation is now reduced to how we solve Eq. (5) in an affordable time. The problem is nonlinear and transient, so this huge linear system must be solved at least thousands of times. From this aspect, we have studied some strategies including Mode Superposition Method [4]. This paper focuses on the block LU decomposition of Eq. (5), and by assessing the solution process, an efficient solution algorithm is newly proposed.

To solve the linear system in Eq. (5) by block LU decomposition, we need to compute the Schur complement in the last block corresponding to the macro displacement vectors. For this purpose, it is convenient to use characteristic modes χ_k for each microscopic unit [10]. The characteristic modes are

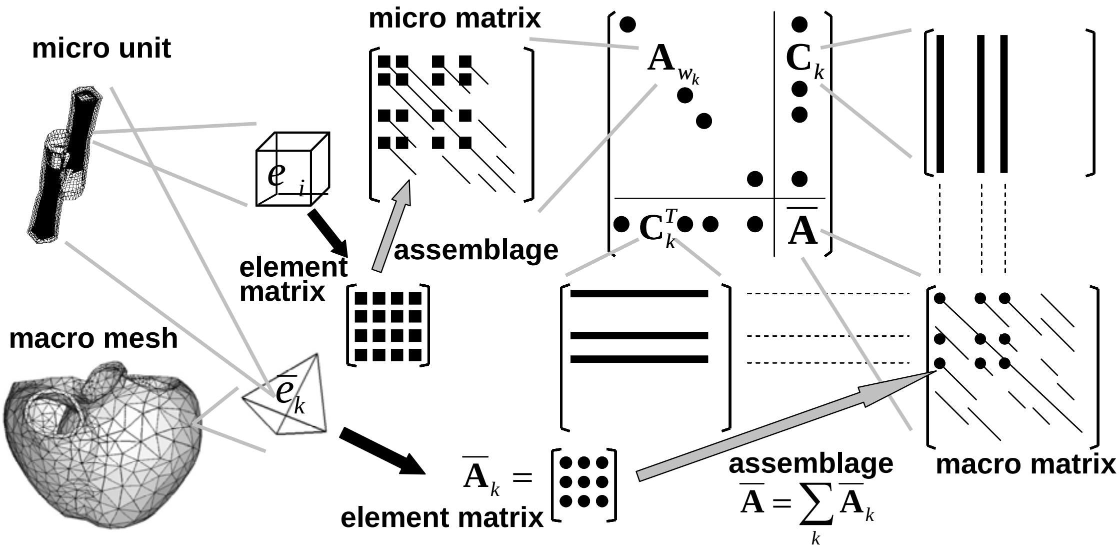

Fig. 1. Nonzero structure of the coefficient matrix in multi-scale analysis.

composed of 9 displacement vectors satisfying the following linear equation.

$$\begin{aligned} A_{w_k}\chi_k &= G_k, \qquad (6)\\ \chi_k &= [\chi_{k,11}, \chi_{k,21}, \cdots, \chi_{k,23}, \chi_{k,33}],\\ G_k &= [G_{k,11}, G_{k,21}, \cdots, G_{k,23}, G_{k,33}]. \end{aligned}$$

Once all the characteristic modes have been given, the Schur complement is computed as

$$\bar{S} = \sum_k \bar{S}_k$$

with

$$\begin{aligned} \bar{S}_k &= \bar{A}_k - \bar{B}_k^T G_k^T A_{w_k}^{-1} G_k \bar{B}_k\\ &= \bar{A}_k - \bar{B}_k^T \chi_k^T A_{w_k} \chi_k \bar{B}_k. \qquad (7) \end{aligned}$$

With the Schur complement, we can obtain the solution vector of Eq. (5) by forward and backward substitutions as follows.

$$\text{Step1} \quad \text{Solve } A_{w_k}\{\Delta w_k\} = \{r_{w_k}\}, 1 \le k \le n \qquad (8)$$

$$\text{Step2} \quad \{g_k\} = \{\bar{r}_k\} - \bar{B}_k^T \chi_k^T \{r_{w_k}\}, 1 \le k \le n \qquad (9)$$

$$\text{Step3} \quad \text{Solve } \bar{S}\{\Delta\bar{u}\} = \textstyle\sum_k \{g_k\} \qquad (10)$$

$$\text{Step4} \quad \{\Delta w_k\} = \{\Delta w_k\} - \chi_k \bar{B}_k \{\Delta\bar{u}\}, 1 \le k \le n \qquad (11)$$

Vector $\{g_k\}$ in Step 2 is given by the following equality of the solution vector $\{\Delta w_k\}$ in Step 1:

$$\bar{B}_k^T G_k^T \{\Delta w_k\} = \bar{B}_k^T G_k^T A_{w_k}^{-1}\{r_{w_k}\} = \bar{B}_k^T \chi_k^T \{r_{w_k}\}.$$

When performing the Newton-Raphson iterations, Step 4 can be omitted, since Step 1 in the next Newton-Raphson iteration has almost the equivalent effect. In the Newton-Raphson iterations, an accurate solution for the stiffness matrix is not always indispensable. Thus, for example, we may skip the update of the characteristic modes in Eq. (6) for ten time steps or more without degradation of the convergence. However, to attain the best accuracy of the Schur complement $\bar{S}_k$ in Eq. (7) with the available approximated characteristic modes, computation of the Schur complement $\bar{S}_k$ by Eq. (7) is performed in every Newton-Raphson step using the updated matrices $\bar{A}_k$ and A_{w_k}.

In our analysis, an implicit time integration scheme (based on the Newmark-β method) is adopted for the total system, and thus the coefficient matrices and the residual vectors given by the fluid elements are superposed on the macroscopic coefficient matrix and the right-hand side vector. Thus, updates of the fluid variables are computed simultaneously with the macroscopic structural variables in Step 3.

In addition to the aforementioned strategies, the LU factorization-based preconditioner for GMRES Micro-solver was reused as addressed in Section IV. By virtue of the new solution algorithm thus streamlined, a large multi-scale heart simulation including the interaction with blood flow was finally realized as demonstrated in Section V.

B. Basic Parallelization Strategy

It turns out that the computation of the solutions for the characteristic modes χ_k in Eq. (6), the Schur complement $\bar{S}_k$, and the solutions for Δw_k in Step 1 can be executed in parallel for $k = 1, \cdots, n$. Since these operations have the highest computational cost, the nonlinear homogenization computation is fairly well suited to execution on a massively parallel computer as exemplified later. On the other hand, in the macroscopic solution phase in Step 3, the NDOF is usually far less than the total NDOF. Thus, in the present implementation of our simulator, this solution phase is parallelized with an appropriate number of processes (np_{ma}), which is typically far less than the total number of processes in a massively parallel computer system. Thus, the basic parallel strategy adopted is as follows.

1) Tasks for n microscopic units are distributed to the available processes.

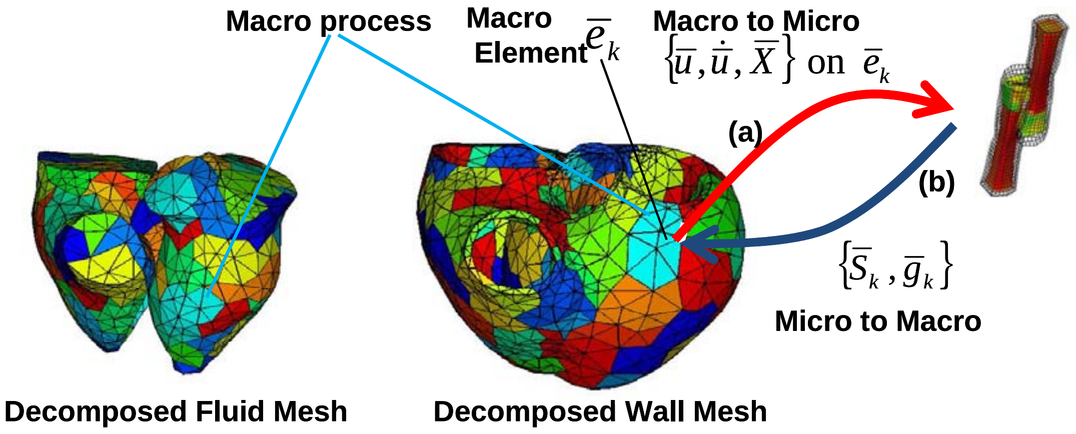
Macro process
Macro Element $\bar{e}_k$
Macro to Micro
$\{\bar{u}, \dot{\bar{u}}, \bar{X}\}$ on $\bar{e}_k$
(a)
(b)
$\{\bar{S}_k, \bar{g}_k\}$
Micro to Macro
Decomposed Fluid Mesh
Decomposed Wall Mesh

Fig. 2. Communication data between the macroscopic subdomain process and the process for each microscopic unit.

2) Among the available processes, np_{ma} macro processes are chosen to perform Step3 in parallel.

In Fig. 2 we illustrate the communication required when the solution phase switches from the macro to the micro and vice versa. In this figure, each color on the macroscopic mesh of the heart wall represents a subdomain assigned to one process. To begin the microscopic computation, the macroscopic displacement $\{\bar{u}\}$, its time derivative $\{\dot{\bar{u}}\}$, and the coordinates $\{\bar{X}\}$ on the macroscopic element $\bar{e}_k$ must be sent from the macro process that owns $\bar{e}_k$ to the process that is responsible for the computation of the k-th microscopic unit. The number of floating point data for each vector is $3 \times 4 = 12$ per element. Thus, the number of communication data is quite small compared with the computational load in the microscopic solution phase. On the other hand, before starting Step 3, vector $\{g_k\}$ and matrix $\bar{S}_k$ must be sent to the macro process that owns $\bar{e}_k$ in its subdomain. The size of each of these in terms of floating point data is $4 \times 3 = 12$ and $12^2 = 144$, respectively, which is also fairly small compared with the computational load. Consequently, the communication overhead should be small.

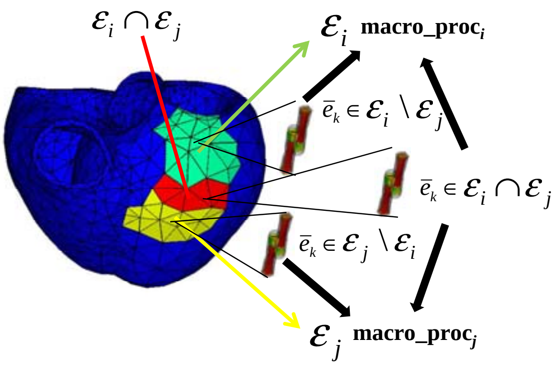
$\mathcal{E}_i \cap \mathcal{E}_j$
$\mathcal{E}_i$ macro_proc$_i$
$\bar{e}_k \in \mathcal{E}_i \setminus \mathcal{E}_j$
$\bar{e}_k \in \mathcal{E}_i \cap \mathcal{E}_j$
$\bar{e}_k \in \mathcal{E}_j \setminus \mathcal{E}_i$
$\mathcal{E}_j$ macro_proc$_j$

Fig. 3. Overlapping of elements and the micro to macro communication pattern.

To determine the distribution of computational loads in Step 3, each of the fluid and structural nodes are decomposed into np_{ma} exclusive subsets and a pair of fluid and structural subsets is assigned to each macro process as depicted in Fig. 2. We define this initial exclusive decomposition of the nodes by

$$\mathcal{N} = \cup_{i=1}^{np_{\mathrm{ma}}} \widehat{\mathcal{N}}_i.$$

We adopted such an initial decomposition for load balancing in the assembly of the matrix for the fluid part and in the ALE mesh control. Next, we define the subset of elements $\mathcal{E}_i$ by gathering elements that include at least one node in $\widehat{\mathcal{N}}_i$. Finally, we define subset $\mathcal{N}_i$ of the nodes, since it is composed of nodes on elements in $\mathcal{E}_i$. Then, we assign $\{\mathcal{E}_i, \mathcal{N}_i\}$ to the i-th macro process $\mathrm{macro\_proc}_i$. We also determine an exclusive decomposition of elements $\mathcal{E}$ by

$$\mathcal{E} = \cup_{i=1}^{np_{\mathrm{ma}}} \widehat{\mathcal{E}}_i \tag{12}$$

such that

$$\widehat{\mathcal{E}}_i \cap \widehat{\mathcal{E}}_j = \phi, \ i \neq j,$$
$$\widehat{\mathcal{E}}_i \subset \mathcal{E}_i, \ i = 1, \cdots, np_{\mathrm{ma}}.$$

The initial exclusive decomposition of nodes determines the responsibilities of the calculations, such as the dot product or the matrix vector product. The overlapping decomposition of elements by $\{\mathcal{E}_i\}$ is useful for matrix assembly, while the exclusive decomposition of elements by $\{\widehat{\mathcal{E}}_i\}$ is useful for the total volume integration, for example, of ventricle cavities. In our implementation, we also exploit the overlap of nodes

$$\mathcal{N} = \cup_{i=1}^{np_{\mathrm{ma}}} \mathcal{N}_i$$

to realize a robust parallel preconditioner for the GMRES method applied in the macroscopic solution phase (Step 3). Here, we apply the overlapping Schwartz preconditioner P defined as:

$$\bar{P}^{-1} = \sum_{i=1}^{np_{\mathrm{ma}}} R_{\widehat{\mathcal{N}}_i}{}^T \bar{P}_i^{-1} R_{\mathcal{N}_i},$$

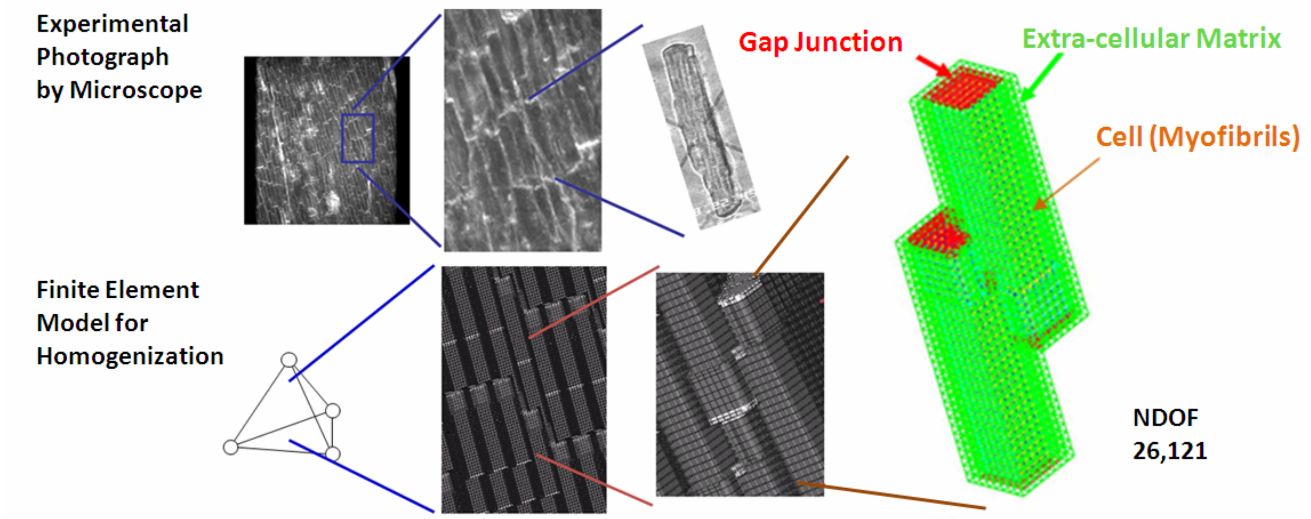
Experimental Photograph by Microscope
Finite Element Model for Homogenization
Gap Junction
Extra-cellular Matrix
Cell (Myofibrils)
NDOF 26,121

Fig. 4. Finite Element Model for Homogenization and Close-up photograph of myocardium by microscope

where $R_{\widehat{\mathcal{N}}_i}$ and $R_{\mathcal{N}_i}$ are restrictions from the total vector space to the subspaces on $\widehat{\mathcal{N}}_i$ and $\mathcal{N}_i$, respectively. $\bar{P}_i$ is constructed so as to approximate the restriction of the coefficient matrix $\bar{S}$ on $\mathcal{N}_i$:

$$\bar{S}_{\mathcal{N}_i} = R_{\mathcal{N}_i}{}^T \bar{S} R_{\mathcal{N}_i}.$$

For this purpose, we employ the ILU preconditioner proposed in [11], where a special fill control strategy is applied to deal with the saddle point problem caused by the incompressibility constraints. Note that the overlap of nodes at the subdomain boundaries also contributes to stabilizing this ILU factorization.

Since the macroscopic elements overlap in the decomposition

$$\mathcal{E} = \cup_{i=1}^{np_{\mathrm{ma}}} \mathcal{E}_i, \tag{13}$$

the communication pattern from the micro processes to the macro processes may be one-to-many for the process that owns the micro unit contained in the overlapping regions as depicted in Fig. 3. For each macro element $\bar{e}_k, k = 1, \cdots, n$, let us define a subset $\mathcal{S}_k^{\mathrm{macro}}$ of the macro processes as:

$$\mathcal{S}_k^{\mathrm{macro}} = \{\mathrm{macro\_proc}_i, \bar{e}_k \in \mathcal{E}_i\}.$$

The number of processes in $\mathcal{S}_k^{\mathrm{macro}}$ depends on the overlapping condition of $\bar{e}_k$ in Eq. (13). In our implementation, the process that owns the micro unit at $\bar{e}_k$ sends the data to all the macro processes in $\mathcal{S}_k^{\mathrm{macro}}$. On the other hand, communication from the macro processes to the micro processes is performed by referring to the exclusive decomposition in Eq. (12).

IV. Implementation on T2K/Tokyo

A. Target Problem Size

As shown in the right panel of Fig. 4, the microscopic unit consists of two cells, and an infinite number of these units are connected periodically to constitute a finite element of the macroscopic myocardium in the context of the homogenization method. The upper-left panel shows a close-up photograph of the myocardium through a microscope for comparison.

Let n_{κ} be the NDOF on the k-th microscopic unit and $\bar{n}$ be the NDOF on the macroscopic domain. Then, the total NDOF in Eq. (5) is given by:

$$n_{total} = \sum_k n_{\kappa} + \bar{n}.$$

In the present simulation, n_{κ} is equal to 26,121 for every microscopic unit, the number of microscopic units is 6144, and $\bar{n}$ is 46,165. Thus, n_{total} is about 1.6×10^8.

B. MPI process mapping

We used 6144 CPU cores and created 6144 MPI processes to solve this target program. Because the majority of the computation time is expected to be spent simulating microscopic units, we created a few MPI processes capable of handling both the macroscopic subdomain and a microscopic unit. More concretely, as the macroscopic domain is decomposed into 64 subdomains at this time, processes ranked 0 to 63 handle a single macroscopic subdomain as well as one microscopic unit, while those ranked 64 to 6143 handle only one microscopic unit.

Since 16 continuous MPI processes, ranked $16n$ to $16n + 15 (n \geq 0)$, are assigned to a single node in the case of the T2K/Tokyo, 64 macro processes are assigned to only 4 nodes. As the network performance between any two nodes used in this experiment is equally 5GB/s $\times$ 2, the way in which the macro processes are distributed among the nodes does not have much effect on the communication time between the macroscopic domain and the microscopic units.

C. The algorithm of multi-scale heart simulation

An outline of the algorithm for our multi-scale heart simulator is given in Fig. 5. At each time step, lines (a) to (j)

```
for $istep = 0, \cdots$ $(time\ step)$
  for $iter = 1, \cdots$ $(Newton -\ Raphson\ step)$
    comm1 : Macro → Micro                                                   $(a)$
    for $k = 1, \cdots$ $(Computation\ for\ each\ Micro\ Unit)$
      assemble $A_{w_k}, \{\overline{r}_k\}, \{r_{w_k}\}, G_k, \overline{A}_k$                    $(b)$
      Micro_Solver$(N_t)$                                                   $(c)$
      compute $\overline{S}_k$                                                   $(d)$
      compute $\{\overline{g}_k\} = \{\overline{r}_k\} - \overline{B}_k \chi_k^T \{r_{w_k}\}$          $(e)$
    endfor
    comm2 : Micro → Macro                                                   $(f)$
    compute $\overline{S} = \sum_{k=1}^{n} \overline{S}_k,\ \{\overline{g}\} = \sum_{k=1}^{n} \{\overline{g}_k\}$                $(g)$
    Solve $\overline{S}\{\Delta\overline{u}\} = \{\overline{g}\}$                                   $(h)$
    if $(\{\overline{r}\}\ and\ \{r_w\}\ are\ small\ enough)$
      goto $next\ istep$                                                    $(i)$
    update $\overline{u}\ and\ w$                                                $(j)$
  endfor
endfor
```

Fig. 5. Algorithm of the Multi-Scale Heart Simulator (Outline)

```
procedure Micro_Solver$(N_t)$
  if $(mod(istep, N_t) == 0\ \&\&\ iter == 1)$
    $LU\ factorization\ A_{w_k} = L_{w_k} U_{w_k}$                          $(c1)$
  solve $A_{w_k}\{\Delta w_k\} = \{r_{w_k}\}$ $by\ preconditioned$
    $GMRES\ using\ L_{w_k} U_{w_k} as\ the\ preconditioner$                  $(c2)$

  if $(mod(istep, N_t) == 0\ \&\&\ iter == 1)$
    solve $L_{w_k} U_{w_k} \chi_k = G_k$ $by\ preconditioned$
      $GMRES\ using\ L_{w_k} U_{w_k} as\ the\ preconditioner$                $(c3)$
end procedure
```

Fig. 6. Micro_Solver procedure.

are repeatedly executed for each Newton-Raphson iteration. In particular, the following should be noted.

- In Line (a), $\{\overline{u}\}$, $\{\dot{\overline{u}}\}$ and $\{\overline{X}\}$ on each macro element $\overline{e}_k$ are sent to the corresponding microscopic unit (Fig. 2(a)).
- Line (e) and (h) are corresponding to Eqs. (9)(10).
- In Line (f), $\{\overline{S}_k\}$ and $\{\overline{g}_k\}$ of each microscopic unit k are sent to the core in which macro element $\overline{e}_k$ is handled (Fig. 2(b)).

An outline of the Micro_Solver procedure (Fig. 5(c)) is given in Fig. 6. Lines $(c1)$ and $(c2)$ correspond to Eq. (8), while line $(c3)$ corresponds to Eq. (6). However, lines $(c1)$ and $(c3)$ are executed only in the first Newton-Raphson iteration for every N_t time step. The effect of this idea will be discussed in the next section.

The time spent executing lines (b) to (e) corresponds to the computation time of the microscopic unit. In this simulation, to measure the time taken to handle all microscopic units at each Newton-Raphson iteration, we insert barrier synchronization before line (a) and after line (f). This may result in a delay, caused by load imbalance among the microscopic units. Furthermore, the time spent executing lines (a) and (f) corresponds to the communication time. Here we refer to the sum of these three times as *Micro+L+C Time*. We also refer to the time taken to execute lines (g) and (h) as *Macro Time* and the total time for the execution of lines (a) to (j) as *Entire Time*.

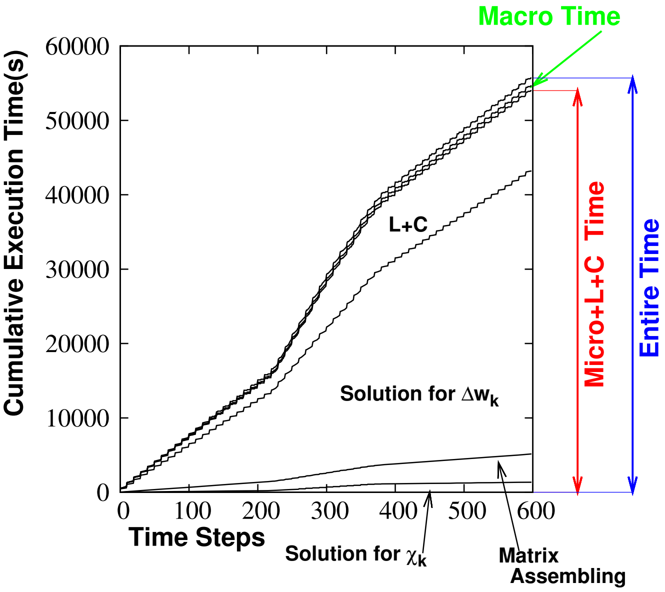

Fig. 7. Cumulative Entire, Micro+L+C and Macro Time, and Classification of Micro+L+C in case of rank 0 MPI process

V. Simulation Results

A. Computation Time

Shown in Fig.7, are the cumulative *Entire*, *Micro+L+C* and *Macro Times* using 6144 microscopic units and 6144 CPU cores. It took about 16 hours (600 time steps) to simulate 1.5 heart beats, meaning about 10 hours (400 time steps) to simulate 1 heart beat. *Micro+L+C Time* consistently consumes more than 96% of *Entire Time*, whereas *Macro Time* consumes only about 1% of *Entire Time*.

Figure 7 also shows that *Micro+L+C Time* consumes more time in the systolic phase, corresponding to the time steps from about 220 to 380, compared with the other time steps, which comprise the diastolic phase. The reasons for this are given below. In the systolic phase, it takes more time to solve Eq. (8) (Fig. 6($c1$)+($c2$)) and the behavior of the microscopic units differs greatly from one to the other. Therefore, there may be a greater delay due to the load imbalance among them.

In Fig. 7, we also show the classification of the cumulative *Micro+L+C Time* for process 0. The following three computation times account for the greater part thereof: Solution for $\{\Delta w_k\}$ (Eq. (8) and Fig. 6($c1$) + ($c2$)), Solution for χ_k (Eq. (6) and Fig. 6($c3$)), and Matrix Assembling (Fig. 5(b)).

The time for executing lines (d) and (e) is so negligible that it could not be depicted in Fig. 7.

As described later, the time taken to execute lines (a) and (f) is also negligible. Therefore, we assume that the majority of the *L+C Time* is caused by load imbalances among the microscopic units. In the case of the other processes, it is possible that the delay caused by the load imbalance may become rather large

In Fig. 8, we show the three computation times per time step. The upper and lower graphs in Fig. 8 show, respectively,

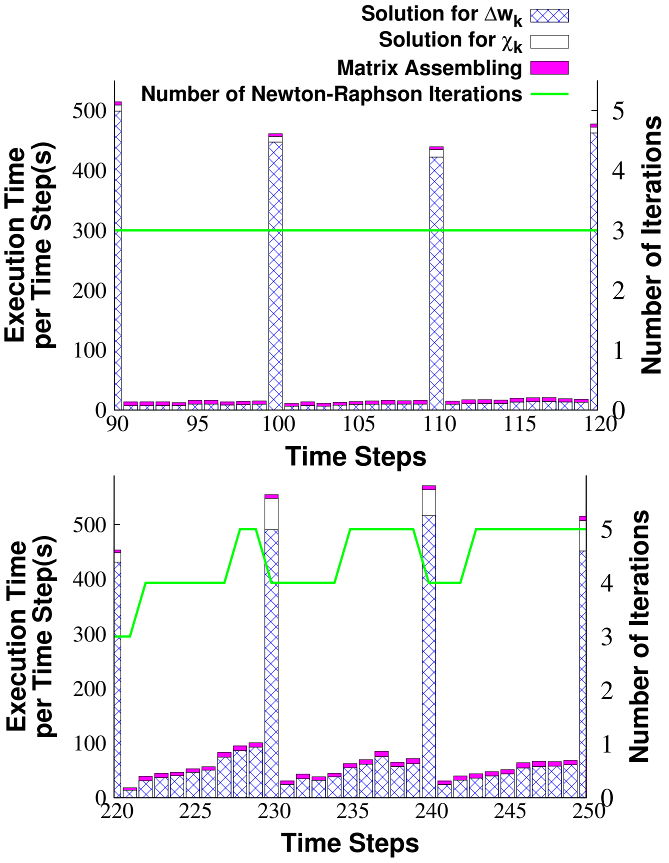

Fig. 8. Execution Time per Time Step of Solution for Δw_k,χ_k and Matrix Assembling, and Number of Newton-Raphson Iterations in the Diastolic Phase (Upper) and Systolic Phase (Lower)

a part of the diastolic and systolic phases. In this simulation, LU factorization was only carried out in the first Newton-Raphson iteration of every N_t=10 time steps. In both phases, $T_{\Delta w_k}$ increases greatly only at those time steps with LU factorization, while LU factorization occupies a great deal of $T_{\Delta w_k}$. In [4], LU factorization was done in every Newton-Raphson iteration for every time step. Thus, the present method can greatly reduce the computation time.

At the time steps with LU factorization, $T_{\Delta w_k}$ in the systolic phase is almost the same as that in the diastolic phase. Nevertheless, at the time steps without LU factorization, $T_{\Delta w_k}$ becomes greater in the systolic phase. The first reason for this is that nonlinearity strengthens and the number of Newton-Raphson iterations increases, while the second is that A_{w_k} becomes asymmetric and thus, it is more difficult to solve Eq. (8). In the current implementation, A_{w_k} is treated as symmetric in both phases to reduce memory usage. Therefore, the LU (LL^T) factor in the systolic phase is not as good a preconditioner for GMRES as that in the diastolic phase, and the time for GMRES iterations increases.

In our simulation, the update of the characteristic modes is also only done in the first Newton-Raphson iteration of every N_t=10 time steps, at which time, 9 systems of equations with coefficient matrices A_{w_k}, are solved and A_{w_k} becomes asymmetric. Therefore, T_{χ_k} also increases in the systolic phase.

Consideration of the frequency of LU factorization: In the diastolic phase, $T_{\Delta w_k}$ is always much smaller at the time steps without LU factorization than at time steps with LU factorization. Consequently, the cumulative $T_{\Delta w_k}$ becomes smaller if LU factorization is done in the first Newton-Raphson iteration of every 20 time steps or more. Even in the systolic phase, it may become smaller. It remains a future work to decide the frequency of LU factorization, optimally or adaptively.

TABLE I
Micro+L+C/Entire RATIO, AND *Macro/Entire* RATIO IN USING 1536, 2048, 3072 AND 6144 CORES

Number of CPU cores	1536	2048	3072	6144
Micro+L+C/Entire	0.99	0.99	0.98	0.96
Macro/Entire	0.0027	0.0037	0.0063	0.013

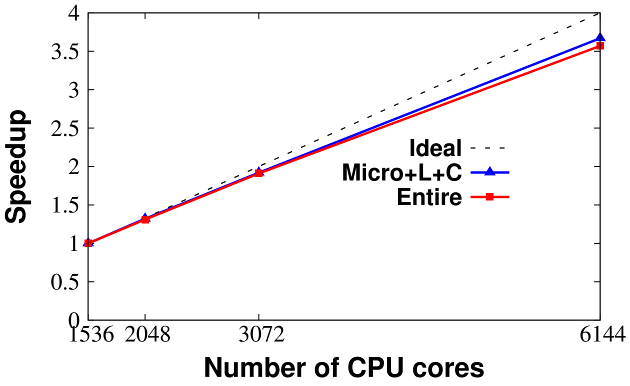

Fig. 9. Ideal Speedup, and Relative Performances based on *Micro+L+C* and *Entire Time* at the 31st Time Step

B. Scalability

Performance of strong scaling was evaluated using 1536, 2048, 3072, and 6144 cores of the T2K/Tokyo. Table I gives the ratios of *Micro+L+C Time* and *Macro Time* to *Entire Time*. In every case, *Micro+L+C Time* consumes more than 96% of *Entire Time*, and also in every case, *Macro Time* consumes at most about 1% of *Entire Time*.

Figure 9 provides the results of strong scaling. Both speedups based on *Micro+L+C Time* and *Entire Time* show good scalability. The performance of *Entire Time* using 6144 cores is 89.3% of the ideal, while that of *Micro+L+C Time* is 91.8%.

Compared with the ideal speedup, performance deteriorates slightly using 3072 cores, and even more using 6144 cores. We assume the reasons for this are as follows. *Micro+L+C Time* contains a delay caused by load imbalance which is unable to be negligible. And if each core handles less microscopic units, the delay caused by load imbalance may increase. The synchronization overhead may increase with more nodes.

The scalability of *Entire* slightly lowers that of *Micro+L+C*. This is because the macro domain is always executed with 64 cores. Thus, *Macro Time* does not essentially decrease, and it shares a relatively larger portion of *Entire Time* as the number of cores increases.

C. Communication Cost

Compared with the NDOF of each microscopic unit, the volume of communication is very small. Concretely, the volume of communication for each of $\{\bar{u}\}, \{\dot{\bar{u}}\}, \{\bar{X}\}$ and $\{\bar{S}_k\}$ in Fig. 2 is about 100 bytes, and for $\{\bar{g}_k\}$ is about 2 KB. On a previous occasion, we measured the communication times for comm1 and comm2 in Fig. 5. After 149 time steps using 3072 cores, the cumulative times for comm1 and comm2 were 107.4 and 21.4 sec, respectively. On the other hand, *Micro+L+C Time* was 19,731 sec. Thus, communication times between the macroscopic domain and the microscopic units are very short. In fact, both communication times are less than 0.6% of *Micro+L+C Time*.

D. Performance of Multi-core Cluster

It is well-known that performance deteriorates as more cores on each socket/node are used, due to memory/cache contention. The cumulative times for two executions with 6144 cores are compared. 12 cores are assigned to each of 512 nodes in the first case, while 16 cores are assigned to each of 384 nodes in the second case. In this comparison, almost a 17% performance improvement is observed for a 25% (4/16) decrease in the number of cores compared with the full use of the cores. In the case of the T2K/Tokyo, each CPU core has its own L1 and L2 cache, but the L3 cache is shared between four cores on a socket. We may, therefore, for example, need to improve the L1 and L2 cache hit ratio and reduce the L3 cache contention in the matrix and vector operations.

E. Simulated Responses of the Heart and Cardiac Cells

The left panel of Fig. 10 shows the computed response of the heart, i.e., the changes in ventricle pressures and volumes in a cardiac cycle. The red and blue lines indicate values for the left and right ventricles, respectively. Due to the deformation of the ventricle wall originating from the contractions of 6144 cell models, the maximum blood pressure reached 15 kPa in the left ventricle, ejecting about half the volume of blood (65 cc) with a slight time delay. Including the response of the right ventricle, this heart simulation accurately reproduced clinical observations. The right panel depicts the blood flow patterns at various representative time phases, with the arrows indicating the velocity vectors. At the end of the filling phase (0.0 sec), a vortex is clearly observed in the left ventricle. Five snapshots of the heart and a cell model chosen from the endocardium are displayed in Fig. 11. The period depicted in this figure corresponds to the systolic phase, with red colors highlighting the distribution of the contraction force in the cell model calculated from cross-bridge kinematics. It is of clinical importance to analyze how these micro and macro events affect each other.

VI. Conclusions

In this paper, we first outlined the governing equations of our multi-scale heart simulator based on the finite element method, although we omitted an introduction of the underlying subcellular electrochemical-mechanical models due to the page limitation. Taking advantage of the mathematical feature of the homogenization method that bridges the macroscopic ventricle models and the microscopic cardiac cells, an efficient parallelization strategy was designed. The algorithm developed for the multi-scale simulation was implemented on a massively parallel computer, the T2K/Tokyo, and the performance was measured. The results are summarized below.

1) We achieved a stable computation of heart pulsation with 6144 cell models and a total NDOF of 160 million. For a simulation of one cardiac cycle, i.e., one second, it took about 10 hours using 6144 CPU cores.
2) The computed responses of the heart, such as blood pressure and volume change, were found to be physiologically reasonable.
3) A good scalability up to 6144 CPU cores was confirmed by virtue of the homogenization method. Using 6144 cores, the performance based on *Entire Time* was 89.3% of the ideal, and that of *Micro+L+C Time* was 91.8%.
4) According to our analysis, *Micro+L+C Time* dominated *Entire Time*, compared with *Macro Time*. In a more detailed analysis of *Micro+L+C Time*, we found that the communication times were negligible, but that the delay caused by load imbalance could not be negligible. More work needs to be done to decrease load imbalance in the future.

VII. Future Work

In this study, we used a very coarse cell model, which is suitable for the computation resources of the T2K/Tokyo. When, in the near future, we have access to peta-scale computers with more than 80,000 nodes (600,000 cores), we intend carrying out the following simulation.

In the discretization, there will be more than 600,000 finite elements in the macroscopic domain, while the number of microscopic units will also be greater than 600,000. The NDOF of each microscopic unit will be between 100,000 and 200,000. Therefore, the total NDOF of the whole heart will be approximately between 6×10^{10} and 1.2×10^{11}. We believe we can contribute to medical practice by carrying out simulations with this NDOF.

Using more than 80,000 nodes, the load imbalance among microscopic units may become even more significant.

We assume that the macroscopic domain will be executed in parallel using less than 1000 macro processes. Because the communication costs between any two nodes may not always be equal on peta-scale computers, communication time may change according to how the macro processes are distributed among the 80,000 nodes.

Due to the fact that overhead may become excessive when using about 10^6 MPI processes, we intend rewriting our simulator using a hybrid programming model, MPI with OpenMP.

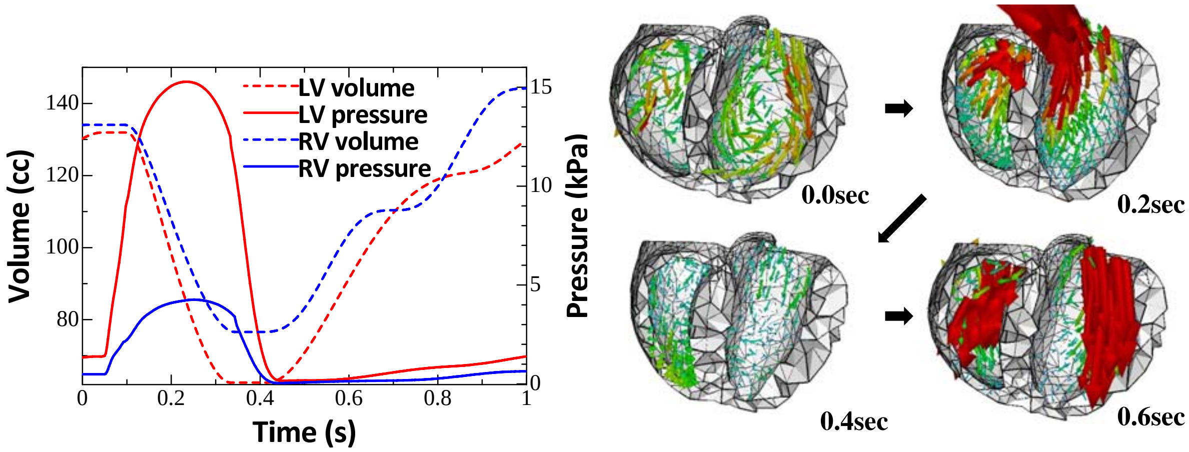

Fig. 10. Blood pressures and cavity volumes of the left and right ventricles in the cardiac cycle and blood flow patterns at representative time phases.

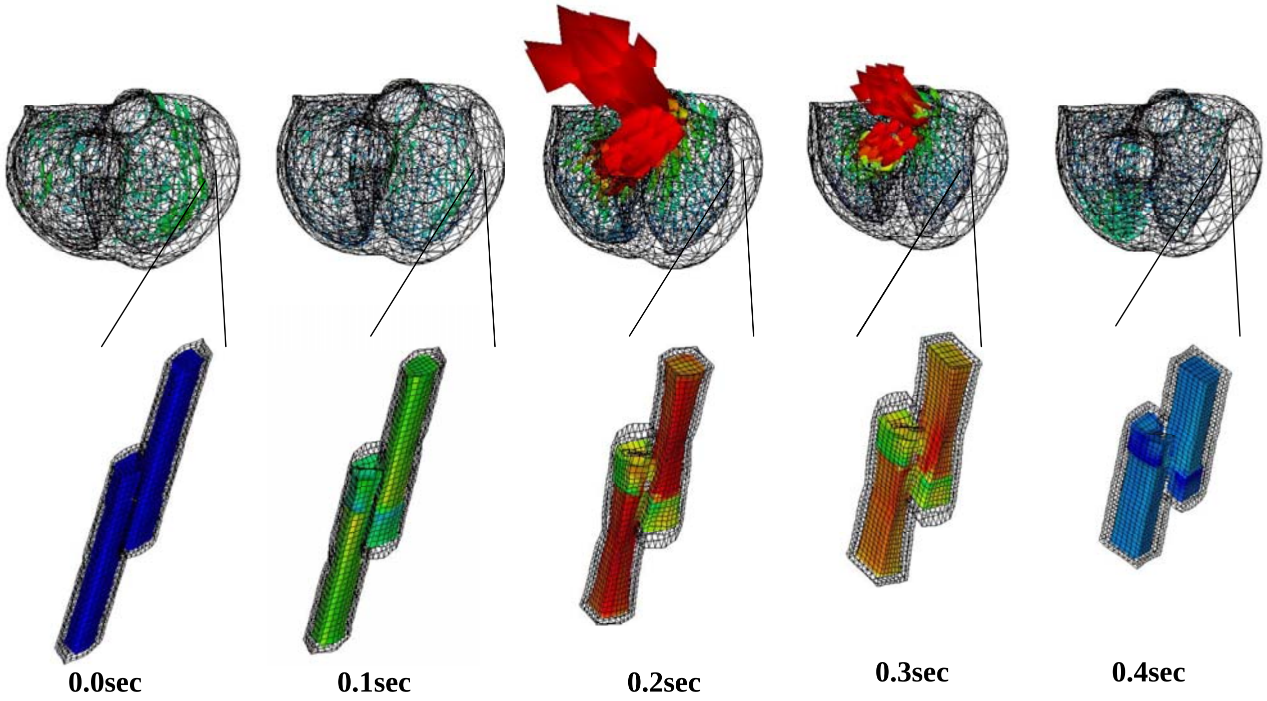

Fig. 11. Contraction of left and right ventricles, outflow of blood, and cardiac cells.

Acknowledgment

This work was supported by the Japan Science and Technology Agency under grant *Collaborative Development of Innovative Seeds-Practicability Verification Stage*, and Japan Society for the Promotion of Science under *Funding Program for World-Leading Innovative R&D on Science and Technology*. We also acknowledge the facilities and assistance of the Supercomputing Division, Information Technology Center, The University of Tokyo.

References

[1] P. J. Hunter and T. K. Borg, "Integration from proteins to organs: the physiome project," *Nat. Rev. Mol. Cell Biol*, vol. 4, pp. 237–243, 2003.

[2] D. Noble, "Modeling the heart: from genes to cells to the whole organ," *Science*, vol. 295, pp. 1678–1682, 2002.

[3] T. Washio, J. Okada, and T. Hisada, "A parallel multilevel technique for solving the bidomain equation on a human heart with purkinje fibers and a torso model," *SIAM J Sci Comput*, vol. 30(6), pp. 2855–2881, 2008.

[4] J. Okada, T. Washio, and T. Hisada, "Study of efficient homogenization algorithms for nonlinear problems approximation of a homogenizedtangent stiffness to reduce computational cost," *Computational Mechanics*, 2009, doi:10.1007/s00466 - 009 - 0432 - 1.

[5] Q. Zhang and T. Hisada, "Analysis of fluid-structure interaction prob-

lems with structural buckling and large domain changes by ale finite element method," *Comput. Methods Appl. Mech. Engrg*, vol. 190, pp. 6341–6357, 2001.

[6] H. Watanabe, S. Sugiura, H. Kafuku, and T. Hisada, "Multiphysics simulation of left ventricular filling dynamics using fluid-structure interaction finite element method," *Biophys*, vol. J87, pp. 2074–2085, 2004.

[7] J. Okada, S. Sugiura, S. Nishimura, and T. Hisada, "3D simulation of calcium waves and contraction in cardiomyocytes using the finite element method," *Am J Physiol Cell Physiol*, pp. C510–C522, 2005.

[8] I. T. C. The University of Tokyo, http://www.cc.u-tokyo.ac.jp/.

[9] C. Miehe, J. Schröder, and J. Schotte, "Computational homogenization analysis in finite plasticity: Simulation of texture development in polycrystalline materials," *Comput Methods Appl. Mech. Engrg.*, vol. 171, pp. 387–418, 1999.

[10] G. Allaire, "Homogenization and two-scale convergence," *SIAM J. Math. Anal.*, vol. 23, pp. 1482–1518, 1992.

[11] T. Washio, T. Hisada, H. Watanabe, and T. E. Tezduyar, "A robust and efficient iterative linear solver for strongly coupled fluid-structure interaction problems," *Comput. Methods Appl. Mech. Engrg.*, vol. 194, pp. 4027–4047, 2005.