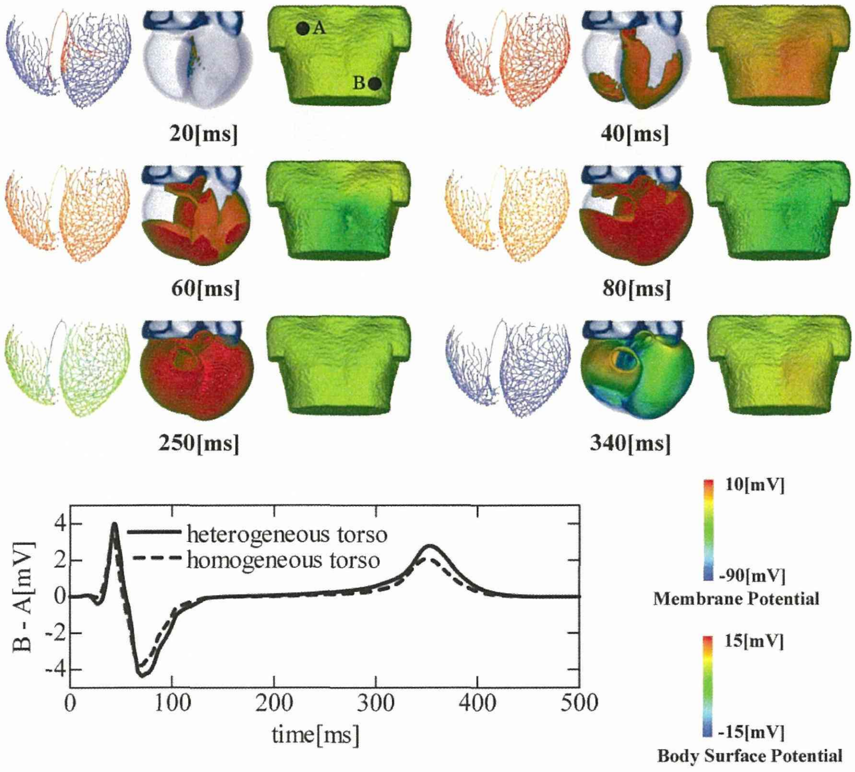

Fig. 1.5 *Membrane potentials on the Purkinje fiber network and the heart muscle, and the surface potential on the torso, at typical times. At the bottom, the second-lead ECG (difference of the potential values between points A and B) is presented for cases with the heterogeneous (rigid line) and homogeneous (broken line) torso models.*

torso domain outside the heart, Γ_C its boundaries, $\boldsymbol{\sigma}_{\mathrm{c}}$ the electric conductivity tensor, and ϕ_{c} the potential on Ω_C. We assume that there are some regions inside Ω_C where electrodes with given potentials ϕ_{p} are imbedded in the case of an ICD simulation. We represent these electrode regions by Ω_P. In this situation, the following equations are further imposed on Ω_C in addition to (1.5)–(1.8):

$$-\nabla \cdot \boldsymbol{\sigma}_{\mathrm{c}} \nabla \phi_{\mathrm{c}} = 0 \text{ on } \Omega_C \backslash \Omega_P, \tag{1.11}$$

$$\vec{n}_H \cdot \boldsymbol{\sigma}_{\mathrm{c}} \nabla \phi_{\mathrm{c}} = \vec{n}_H \cdot \boldsymbol{\sigma}_{\mathrm{e}} \nabla \phi_{\mathrm{e}} = J_H \text{ and } \phi_{\mathrm{c}} = \phi_{\mathrm{e}} \text{ on } \Gamma_H, \tag{1.12}$$

$$\vec{n}_C \cdot \boldsymbol{\sigma}_{\mathrm{c}} \nabla \phi_{\mathrm{c}} = 0 \text{ on } \Gamma_C, \tag{1.13}$$

$$\phi_{\mathrm{c}} = \phi_{\mathrm{p}} \text{ on } \Omega_P. \tag{1.14}$$

As shown in Figure 1.7, the torso is composed of several kinds of organs that have different electric conductivities. Table 1.2 lists the conductivities of the organs present in our simulation. As depicted by the broken line in Figure 1.5, we observed substantially different computational results for the body surface potential when we assumed a totally homogeneous conductivity (3 mS/cm) on the torso. This indicates the importance of modeling the torso conductance accurately.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

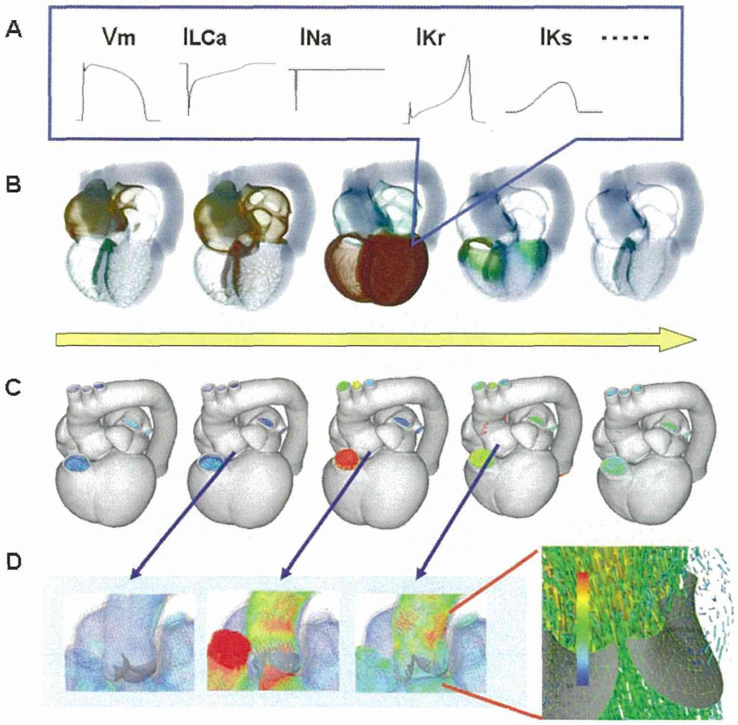

Fig. 1.6 *Computational results generated by the UT heart simulator. Time courses of the transmembrane potential and the ion channel currents at a specific point in the heart muscle* (A)*, the distribution of the transmembrane potential* (B)*, structural deformation of the heart muscle and blood flow* (C)*, and blood flow and valve leaflets at the aorta* (D)*. The yellow arrow indicates the time axis for one heartbeat.*

In this paper, we present a technique for a parallel solution that handles the bidomain problem with the torso model in a robust and efficient way. In particular, we focus on a technique to overcome the difficulty of very high spatial resolution by using a sophisticated multigrid algorithm and parallel computation techniques. The multigrid algorithm is constructed for a conservative finite element discretization of the given potential problem on a composite mesh consisting of a fine local voxel mesh around the heart and a coarse global voxel mesh covering the torso. We also briefly present strategies to resolve the difficulty of time resolution. On one hand, more sophisticated adaptive refinement techniques have been proposed in earlier studies [4, 16, 27], where the refinement is varied over time depending on the ongoing electric activity. However, on the other hand, this study is significantly more complex since the method is shown to be applicable to a problem of practical interest, whereas the earlier study merely showed the feasibility of the method on a simple two-dimensional grid. Some previous works [19, 29] also adopted a multigrid method for the bidomain problem. However, neither the composite mesh problem nor the Purkinje network is dealt with in these studies. The differences between this research and the earlier studies mentioned above are summarized below.

1. A stability analysis of a typical explicit time integration scheme of the bidomain equation [27, 29] is given.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

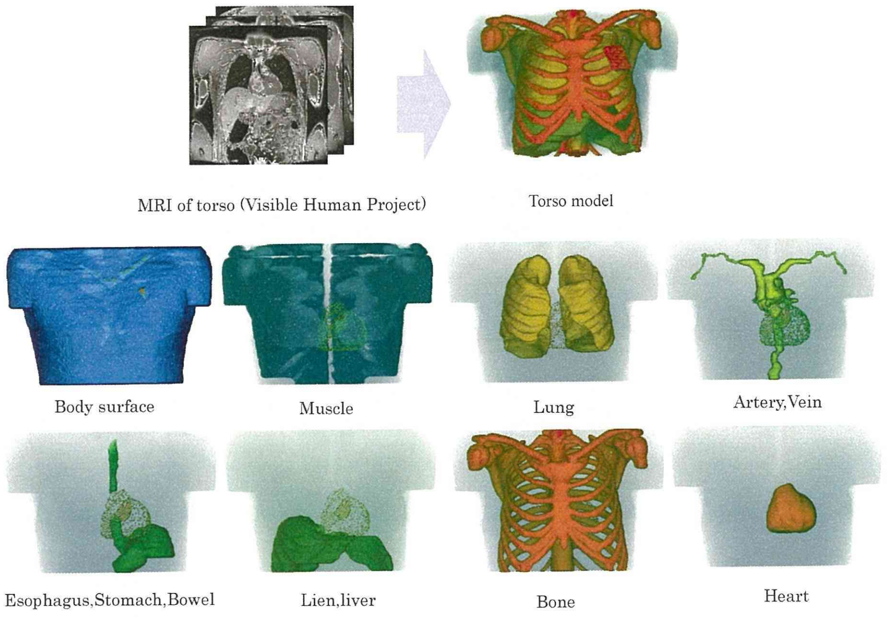

Fig. 1.7 *Organs in the torso.*

Table 1.2 *Conductivities [mS/cm] of the organs outside the heart muscle.*

Blood	7.8	Capsula cordis	15.0
Muscle	2.56	Lung	0.83
Artery, vein	6.8	Esophagus	2.0
Stomach, bowel	2.0	Lien, liver	1.67
Bone	0.06	Others	0.5

2. A conservative finite element discretization of the potential problem on the composite mesh is proposed by applying the Lagrange multiplier technique for the constraints at the interface of the local and global meshes.
3. A multigrid solution method for the discretization on the composite mesh is derived naturally by the Lagrange multiplier at the interface.
4. The conservation property of the electric currents passing through the electrodes is investigated. This is important when considering the interaction with an external electrical device, such as an ICD.
5. An efficient way of dealing with the Purkinje fiber network in the multigrid method is proposed.

In our simulator, the Purkinje fibers are modeled as one-dimensional multiway branching lines connected to the voxel nodes at these end-points, as has been reported in the literature [25]. Bidomain equations similar to (1.5), (1.6), and (1.9) are then discretized by the one-dimensional finite elements constructed on the network. In this situation, the voxel mesh can be coarsened in a standard way by exploiting its regular structure, whereas the Purkinje network is irregular. Thus, combining the Purkinje network with the multigrid algorithm is not straightforward. We also present a practical parallel implementation method on a distributed memory machine. Good performance of the implemented algorithm is demonstrated through numerical experiments

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

with a realistic human heart model that includes the torso. Note that unstructured grids are preferred for defibrillation studies with an ICD, since smooth boundaries are required to avoid artificial currents at the strong shock. Efficiency of the algebraic multigrid method in such cases is confirmed in the literature [17] for rabbit ventricles. In our case, we adopted the voxel mesh approach due to the simplicity of modeling and data handling, ease of parallelization, and the faster computation speed obtained by avoiding indirect memory addressing.

2. Finite Element Discretization. In this section, we introduce the temporal and spatial discretization of the bidomain equation. The weak forms of (1.5)–(1.8) on Ω_H are given by

$$\int_{\Omega_H} \nabla w_{\mathrm{i}} \cdot \boldsymbol{\sigma}_{\mathrm{i}} \nabla \phi_{\mathrm{i}} d\Omega = -\int_{\Omega_H} w_{\mathrm{i}} \beta I_{\mathrm{m}} d\Omega, \tag{2.1}$$

$$\int_{\Omega_H} \nabla w_{\mathrm{e}} \cdot \boldsymbol{\sigma}_{\mathrm{e}} \nabla \phi_{\mathrm{e}} d\Omega = \int_{\Omega_H} w_{\mathrm{e}} \beta I_{\mathrm{m}} d\Omega + \int_{\Gamma_H} w_{\mathrm{e}} J_H d\Gamma. \tag{2.2}$$

Furthermore, the weak form of (1.11)–(1.13) on Ω_C are given by

$$\int_{\Omega_C \backslash \Omega_P} \nabla w_{\mathrm{c}} \cdot \boldsymbol{\sigma}_{\mathrm{c}} \nabla \phi_{\mathrm{c}} d\Omega = -\int_{\Gamma_H} w_{\mathrm{c}} J_H d\Gamma, \ w_{\mathrm{c}} = 0 \text{ on } \Gamma_P. \tag{2.3}$$

Here, $w_{\mathrm{i}}, w_{\mathrm{e}}$, and w_{c} are arbitrary test functions. With the boundary conditions on Γ_H given in (1.12), we can superpose the two extracellular equations on Ω_H and Ω_C, and replace (2.2) and (2.3) by

$$\int_{\Omega} \nabla w_{\mathrm{e}} \cdot \boldsymbol{\sigma}_{\mathrm{e}} \nabla \phi_{\mathrm{e}} d\Omega = \int_{\Omega_H} w_{\mathrm{e}} \beta I_{\mathrm{m}} d\Omega, \tag{2.4}$$

$$\phi_{\mathrm{e}} = \phi_{\mathrm{p}}, \ w_{\mathrm{e}} = 0 \text{ on } \Gamma_P. \tag{2.5}$$

Here, $\Omega = \Omega_H \cup \Omega_C \backslash \Omega_P$ and Γ_P is the boundary of Ω_P. The extracellular potentials and the conductivity tensors are combined on the whole domain Ω as

$$\phi_{\mathrm{e}} = \begin{cases} \phi_{\mathrm{e}} & \text{on} \quad \Omega_H, \\ \phi_{\mathrm{c}} & \text{on} \quad \Omega_C, \end{cases} \qquad \boldsymbol{\sigma}_{\mathrm{e}} = \begin{cases} \boldsymbol{\sigma}_{\mathrm{e}} & \text{on} \quad \Omega_H, \\ \boldsymbol{\sigma}_{\mathrm{c}} & \text{on} \quad \Omega_C. \end{cases} \tag{2.6}$$

The matrix representation for a finite element discretization of the intracellular equation on Ω_H in (2.1) and the extracellular equation on Ω in (2.4) are given by

$$\boldsymbol{K}_{\mathrm{i}} \boldsymbol{\phi}_{\mathrm{i}} = -\beta \boldsymbol{I}_{\mathrm{m}}, \tag{2.7}$$

$$\boldsymbol{K}_{\mathrm{e}} \boldsymbol{\phi}_{\mathrm{e}} = \boldsymbol{R}_H{}^T \beta \boldsymbol{I}_{\mathrm{m}} - \boldsymbol{K}_{\mathrm{p}} \boldsymbol{\phi}_{\mathrm{p}}. \tag{2.8}$$

Here, we assume that the nodes on the electrodes are deleted in the vector representation $\boldsymbol{\phi}_{\mathrm{e}}$, and that $\boldsymbol{R}_H$ represents a restriction operator from the whole domain Ω to the subdomain Ω_H on the heart muscle. Since $\boldsymbol{R}_H$ is a simple injection on Ω_H in our case, we will omit $\boldsymbol{R}_H$ and its transpose $\boldsymbol{R}_H^T$ in the equations hereafter. For example, the matrices $\boldsymbol{R}_H^T \boldsymbol{K}_{\mathrm{i}}$ and $\boldsymbol{R}_H^T \boldsymbol{K}_{\mathrm{i}} \boldsymbol{R}_H$ will simply be represented by $\boldsymbol{K}_{\mathrm{i}}$.

2.1. Explicit Time Integration Scheme. In order to establish a stable time integration scheme, (2.7) and (2.8) are rewritten as

$$\boldsymbol{K}_{\mathrm{i}} \boldsymbol{V}_{\mathrm{m}} + \beta \boldsymbol{I}_{\mathrm{m}} + \boldsymbol{K}_{\mathrm{i}} \boldsymbol{\phi}_{\mathrm{e}} = \boldsymbol{0}, \tag{2.9}$$

$$\boldsymbol{K}_{\mathrm{i}} \boldsymbol{V}_{\mathrm{m}} + (\boldsymbol{K}_{\mathrm{i}} + \boldsymbol{K}_{\mathrm{e}}) \boldsymbol{\phi}_{\mathrm{e}} = -\boldsymbol{K}_{\mathrm{p}} \boldsymbol{\phi}_{\mathrm{p}}. \tag{2.10}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Equation (2.10) is obtained by adding (2.7) and (2.8) and the relation between the potentials, $\boldsymbol{V}_{\mathrm{m}} = \boldsymbol{\phi}_{\mathrm{i}} - \boldsymbol{\phi}_{\mathrm{e}}$ on Ω_H. Note that the intracellular potential disappears in this representation, and the second equation (2.10) does not contain the transmembrane current I_{m}. The system of (2.9) and (2.10) can be integrated stably along the temporal axis in either an explicit or a semi-implicit manner. In our case, we adopt the following explicit method to reduce computational costs [27]. The matrix representation of the explicit scheme is given by

$$\begin{bmatrix} \frac{\beta C_{\mathrm{m}}}{\Delta t}\boldsymbol{M}_H & \boldsymbol{0} \\ \boldsymbol{K}_{\mathrm{i}} & \boldsymbol{K}_{\mathrm{i}} + \boldsymbol{K}_{\mathrm{e}} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_{\mathrm{m}}^{t+\Delta t} \\ \boldsymbol{\phi}_{\mathrm{e}}^{t+\Delta t} \end{bmatrix} = \begin{bmatrix} \frac{\beta C_{\mathrm{m}}}{\Delta t}\boldsymbol{M}_H - \boldsymbol{K}_{\mathrm{i}} & -\boldsymbol{K}_{\mathrm{i}} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_{\mathrm{m}}^{t} \\ \boldsymbol{\phi}_{\mathrm{e}}^{t} \end{bmatrix}$$

$$(2.11) \qquad - \begin{bmatrix} \beta \boldsymbol{M}_H & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_{\mathrm{p}} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_{\mathrm{ion}}(\boldsymbol{V}_{\mathrm{m}}^{t}, \boldsymbol{S}^{t}) \\ \boldsymbol{\phi}_{\mathrm{p}}^{t+\Delta t} \end{bmatrix},$$

where $\boldsymbol{M}_H$ is the lumped matrix on Ω_H and $\boldsymbol{\phi}_{\mathrm{p}}^{t+\Delta t}$ is a prescribed potential on the electrodes. In the semi-implicit scheme as adopted in the literature [19], the entire PDE system (except for the ODEs to compute $\boldsymbol{S}$) is solved simultaneously. This scheme requires almost quadruple the computational costs for the matrix-vector multiplication and the relaxation phase. On the other hand, the time step size Δt is usually determined in accordance with the ODE system to compute the state vector $\boldsymbol{S}$, as described later. Thus, we have adopted the explicit scheme in our simulator. If we ignore the time dependence of I_{ion} in (2.11), the stability of the explicit scheme depends on the spectral radius of the matrix:

$$(2.12) \qquad \mathcal{M} = \begin{bmatrix} \boldsymbol{D}_{\mathrm{m}} & \boldsymbol{0} \\ \boldsymbol{K}_{\mathrm{i}} & \boldsymbol{K}_{\mathrm{i}} + \boldsymbol{K}_{\mathrm{e}} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{D}_{\mathrm{m}} - \boldsymbol{K}_{\mathrm{i}} & -\boldsymbol{K}_{\mathrm{i}} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix},$$

with

$$(2.13) \qquad \boldsymbol{D}_{\mathrm{m}} = \frac{\beta C_{\mathrm{m}}}{\Delta t}\boldsymbol{M}_H.$$

The spectral radius of $\mathcal{M}$ is bounded above as described by the following theorem.

THEOREM 2.1. *Assume the spectral radius of $\boldsymbol{D}_{\mathrm{m}}^{-1}\boldsymbol{K}_{\mathrm{i}}$ is less than one,*

$$(2.14) \qquad \rho(\boldsymbol{D}_{\mathrm{m}}^{-1}\boldsymbol{K}_{\mathrm{i}}) < 1.$$

Then the spectral radius of $\mathcal{M}$ in (2.12) *is bounded as*

$$(2.15) \qquad \rho(\mathcal{M}) \leq 1 - \lambda_{\min}(\boldsymbol{D}_{\mathrm{m}}^{-1}\boldsymbol{K}_{\mathrm{i}})(1 - \lambda_{\max}((\boldsymbol{K}_{\mathrm{i}} + \boldsymbol{K}_{\mathrm{e}})^{-1}\boldsymbol{K}_{\mathrm{i}})).$$

Here, $\lambda_{\min}$ and $\lambda_{\max}$ represent, respectively, the minimum and maximum eigenvalues of the matrices in brackets.

A proof of the theorem is given in the literature [30]. From the theorem, we see that the scheme is stable if condition (2.14) is satisfied and the time dependence of I_{ion} is ignored. In the actual simulation, however, a fairly small time step, far smaller than the above limitation, is required to compute the ion currents $I_{\mathrm{ion}}(V_{\mathrm{m}}, \boldsymbol{S})$ in (1.9) (see [26]). Thus, we can adopt an "inner-outer" time integration strategy where the intracellular (2.9) is integrated with a small time step in the inner iterations while fixing the extracellular potential ϕ_{e}, and the extracellular potential ϕ_{e} is updated only in the outer iteration with a large time step. This strategy has been adopted extensively by many studies [27]. The algorithm is presented in Figure 2.1. Here, the superscripts denote the time indices for the variables, where T is the time index for

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

```
*The inner-outer time integration scheme.*
Given $\boldsymbol{V}_{\mathrm{m}}^{t_0}$
Solve $(\boldsymbol{K}_{\mathrm{i}} + \boldsymbol{K}_{\mathrm{e}})\boldsymbol{\phi}_{\mathrm{e}}^{t_0} = -\boldsymbol{K}_{\mathrm{p}}\boldsymbol{\phi}_{\mathrm{p}}^{t_0} - \boldsymbol{K}_{\mathrm{i}}\boldsymbol{V}_{\mathrm{m}}^{t_0}$
for $k = 1, \dots, k_{\mathrm{end}}$
    $T = t_0 + (k-1)\Delta T$
    for $i = 1, \dots, \Delta T/\Delta t$
        $t = (i-1)\Delta t + T$
        $\boldsymbol{V}_{\mathrm{m}}^{t+\Delta t} = \boldsymbol{V}_{\mathrm{m}}^{t} - \frac{\Delta t}{\beta C_{\mathrm{m}}}\boldsymbol{M}_H^{-1}\boldsymbol{K}_{\mathrm{i}}\boldsymbol{V}_{\mathrm{m}}^{t} - \boldsymbol{K}_{\mathrm{i}}\boldsymbol{\phi}_{\mathrm{e}}^{T} - \frac{\Delta t}{\beta C_{\mathrm{m}}}\boldsymbol{I}_{\mathrm{ion}}(\boldsymbol{V}_{\mathrm{m}}^{t}, \boldsymbol{S}^{t})$
    next $i$
    Solve $(\boldsymbol{K}_{\mathrm{i}} + \boldsymbol{K}_{\mathrm{e}})\boldsymbol{\phi}_{\mathrm{e}}^{T+\Delta T} = -\boldsymbol{K}_{\mathrm{p}}\boldsymbol{\phi}_{\mathrm{p}}^{T+\Delta T} - \boldsymbol{K}_{\mathrm{i}}\boldsymbol{V}_{\mathrm{m}}^{T+\Delta T}$
next $k$
```

Fig. 2.1 *The inner-outer time integration scheme.*

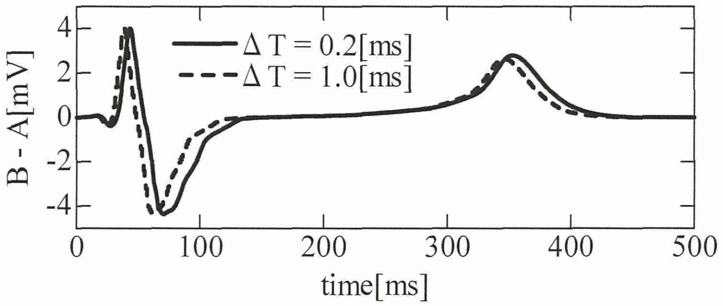

Fig. 2.2 *The second-lead ECG for two different time steps.*

solving the extracellular potential problem and t is the time index for integrating the membrane potential. At each inner iteration, the ODE to compute the state vector $\boldsymbol{S}$ is solved by the explicit Euler scheme. The time step δt (typically $10{\sim}100\,\mu$s) is flexibly varied in time and space depending on the ongoing electric activity as described in the literature [27]. At each outer iteration, a solution to the potential problem on Ω is required. The time steps ΔT and Δt are adjusted flexibly depending on the simulation conditions and the desired accuracy of the solution. For example, in Figure 2.2, the two curves represent the second-lead ECG results as explained in section 1 for time steps $\Delta T = 0.2\,$ms and $\Delta T = 1\,$ms, respectively. Though both curves show a similar tendency and magnitude of fluctuation, substantial phase lag is observed.

Even when adopting a large time step for ΔT in the above inner-outer time integration scheme, the speedup of the solution to the extracellular potential problem is the key issue for the overall performance. Henceforth, we will focus on an efficient discretization of and solution method for the potential problem.

2.2. Spatial Discretization of a Composite Voxel Mesh. Hereafter, $\boldsymbol{\sigma}$ denotes $\boldsymbol{\sigma}_{\mathrm{e}} + \boldsymbol{\sigma}_{\mathrm{i}}$ and ϕ denotes ϕ_e for notational convenience. The weak form corresponding to (2.10) is given by

$$\int_{\Omega} \nabla w \cdot \boldsymbol{\sigma} \nabla \phi d\Omega = -\int_{\Omega_H} \nabla w \cdot \boldsymbol{\sigma}_{\mathrm{i}} \nabla V_{\mathrm{m}} d\Omega,$$
$$w = 0 \text{ and } \phi = \phi_{\mathrm{p}} \text{ on } \Gamma_P. \tag{2.16}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

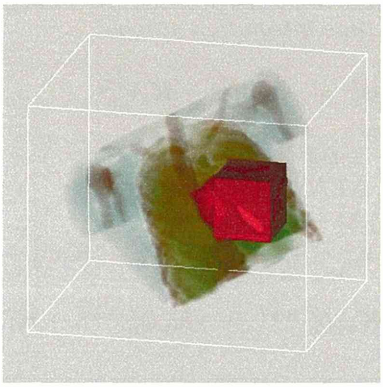

Fig. 2.3 *The fine local voxel mesh domain around the heart and the global coarse voxel mesh domain covering the torso.*

To discretize the above equation, we adopt a uniform three-dimensional voxel mesh because this allows the input data to be prepared easily, and it facilitates the implementation of the multigrid solver and the parallelization of the code. For the intracellular equations (1.5) and (1.9) on the heart, a fine spatial resolution (0.15 mm $\sim$ 0.4 mm) is required to attain sufficient accuracy. On the other hand, such a fine resolution is unnecessary for the extracellular equation (2.4) outside the heart. The above fact naturally leads to the use of two meshes with different spatial resolutions. One is a finer mesh on the local rectangular parallelepiped domain around the heart. The other is a coarser mesh on the global rectangular parallelepiped domain covering the whole torso (see Figure 2.3). Henceforth, we refer to the former mesh as the local mesh and to the latter as the global mesh. We assume that the nodes in the global mesh on the local rectangular parallelepiped domain are given at the same positions as those in the local mesh, and the interval ratio of the global to local mesh is a power of two. Note that it is not necessarily exactly two, as in the usual local refinement. For example, four or eight can also be handled in our framework.

In order to discretize (2.16) on the composite mesh, we apply the Lagrange multiplier method for the constraints at the interface of the local and global meshes. Therefore, we start with a variational formulation of the problem. The energy functional for the formulation is given by

$$\mathcal{E}(\phi) = \int_{\Omega} \frac{1}{2} \nabla\phi \cdot \boldsymbol{\sigma} \nabla\phi d\Omega + \int_{\Omega_H} \nabla\phi \cdot \boldsymbol{\sigma}_{\mathrm{i}} \nabla V_{\mathrm{m}} d\Omega,$$
$$\phi = \phi_{\mathrm{p}} \text{ on } \Gamma_P. \tag{2.17}$$

The extremum of the energy functional in (2.17) satisfies (2.16). Let Ω^L and Ω^G be the local and global meshes, respectively. In particular, Ω^L is a local refinement of Ω^G. In the following, Ω^L and Ω^G are identified either as the domains covered by the meshes or as the sets of nodes in them. Let E^L and E^G be the sets of finite elements contained in Ω^L and Ω^G, respectively. Let Ω^G_L and $\overline{\Omega^G_L}$ be the subsets of Ω^G on Ω^L and outside of Ω^L, respectively. Let E^G_L and $\overline{E^G_L}$ be the subsets of E^G which lie in Ω^G_L and $\overline{\Omega^G_L}$, respectively (see Figure 2.4). We represent an interpolated function ϕ^L

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

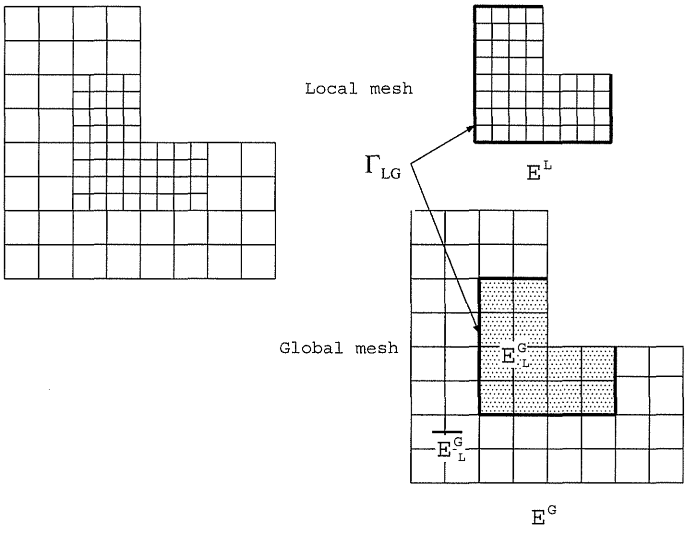

Fig. 2.4 *A composite mesh (left) and decomposition of the elements into the subsets for each level (right).*

for the nodal values $\boldsymbol{\phi}^L$ on the local mesh Ω^L by

$$\phi^L = \boldsymbol{N}^L \cdot \boldsymbol{\phi}^L = \sum_{i\in\Omega^L} N_i^L \phi_i^L. \tag{2.18}$$

Here, $\{N_i^L\}_{i\in\Omega^L}$ are the shape functions on Ω^L. We use similar notation, $\boldsymbol{N}^G, N_i^G, \boldsymbol{\phi}^G$, etc., for the global mesh Ω^G.

Under the above definitions, we define an energy functional for a given nodal function $\boldsymbol{\phi} = \{\boldsymbol{\phi}^L, \boldsymbol{\phi}^G\}$ on the composite mesh by

$$\begin{aligned}\mathcal{E}(\boldsymbol{\phi}) &= \int_{\Omega_L} \frac{1}{2}\nabla\phi^L \cdot \boldsymbol{\sigma}\nabla\phi^L d\Omega + \int_{\overline{\Omega}^G_L} \frac{1}{2}\nabla\phi^G \cdot \boldsymbol{\sigma}\nabla\phi^G d\Omega + \int_{\Omega_H} \nabla\phi^L \cdot \boldsymbol{\sigma}_{\mathrm{i}}\nabla V^L_{\mathrm{m}} d\Omega \\ &= \sum_{e^L\in E^L} \int_{e^L} \frac{1}{2}\nabla\boldsymbol{N}^L\boldsymbol{\phi}^L \cdot \boldsymbol{\sigma}\nabla\boldsymbol{N}^L\boldsymbol{\phi}^L d\Omega + \sum_{e^G\in\overline{E}^G_L} \int_{e^G} \frac{1}{2}\nabla\boldsymbol{N}^G\boldsymbol{\phi}^G \cdot \boldsymbol{\sigma}\nabla\boldsymbol{N}^G\boldsymbol{\phi}^G d\Omega \\ &\quad + \sum_{e^L\in E^L_H} \int_{e^L} \nabla\boldsymbol{N}^L\boldsymbol{\phi}^L \cdot \boldsymbol{\sigma}_{\mathrm{i}}\nabla\boldsymbol{N}^L\boldsymbol{V}^L_{\mathrm{m}} d\Omega.\end{aligned} \tag{2.19}$$

Here, E^L_H are elements in E^L that lie in Ω_H. Note that we assume that Ω_H is composed of a subset of E^L. Now, we impose the following constraint conditions on the variational problem:

$$\boldsymbol{\phi}^L = \boldsymbol{I}^L_G\boldsymbol{\phi}^G \text{ on } \Gamma_{LG}. \tag{2.20}$$

Here, $\boldsymbol{I}^L_G$ is an interpolation operator from Ω^G to Ω^L and Γ_{LG} is the set of nodes situated at the interface boundaries of Ω^L with Ω^G. Note that Γ_{LG} is not necessarily identical to the boundary of Ω^L. As shown by the thick lines in the diagram on the right of Figure 2.4, the intersections of $\partial\Omega$ and $\partial\Omega_L$ are not contained in Γ_{LG}. The weights of the interpolation are determined by the shape functions $\boldsymbol{N}^G$ on Ω^G.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Let $\dot{\Omega}_L^G$ be the set of internal nodes in Ω_L^G where the nodes on Γ_{LG} are excluded. The components of $\boldsymbol{\phi}^G$ on $\dot{\Omega}_L^G$ do not affect the functional in (2.19). However, these components are set identical to $\boldsymbol{\phi}^L$ by an injection of the solution as described in section 3.1.

The following equation is obtained by applying the Lagrange multiplier method to the variational problem (2.19):

$$\begin{aligned}&\int_{\Omega^L} \nabla w^L \cdot \boldsymbol{\sigma} \nabla \phi^L d\Omega + \int_{\overline{\Omega_L^G}} \nabla w^G \cdot \boldsymbol{\sigma} \nabla \phi^G d\Omega + \int_{\Omega_H^L} \nabla w^L \cdot \boldsymbol{\sigma}_{\mathrm{i}} \nabla V_{\mathrm{m}}^L d\Omega \\ &\qquad + \left(\boldsymbol{w}^L - \boldsymbol{I}_G^L \boldsymbol{w}^G\right)_{\Gamma_{LG}} \cdot \boldsymbol{\lambda} + \boldsymbol{w}_\lambda \cdot \left(\boldsymbol{\phi}^L - \boldsymbol{I}_G^L \boldsymbol{\phi}^G\right)_{\Gamma_{LG}} = 0. \end{aligned} \tag{2.21}$$

Here, w^L and w^G are arbitrary test functions equal to zero on Γ_P. The brackets $(\ )_{\Gamma_{LG}}$ denote the restriction of a vector to the nodes on Γ_{LG}, $\boldsymbol{\lambda}$ is the Lagrange multiplier defined at the nodes on Γ_{LG}, and $\boldsymbol{w}_\lambda$ is a test vector associated with the Lagrange multiplier. Equation (2.21) can be rewritten in matrix form as follows:

$$\begin{aligned}&\boldsymbol{w}^L \cdot \left(\boldsymbol{K}^L \boldsymbol{\phi}^L + \boldsymbol{K}_{\mathrm{i}}^L \boldsymbol{V}_{\mathrm{m}}^L\right) + \boldsymbol{w}^G \cdot \overline{\boldsymbol{K}_L^G} \boldsymbol{\phi}^G + \left(\boldsymbol{w}^L - \boldsymbol{I}_G^L \boldsymbol{w}^G\right)_{\Gamma_{LG}} \cdot \boldsymbol{\lambda} \\ &\qquad + \boldsymbol{w}_\lambda \cdot \left(\boldsymbol{\phi}^L - \boldsymbol{I}_G^L \boldsymbol{\phi}^G\right)_{\Gamma_{LG}} = 0. \end{aligned} \tag{2.22}$$

Here, the matrices $\boldsymbol{K}^L$, $\boldsymbol{K}_{\mathrm{i}}^L$, and $\overline{\boldsymbol{K}_L^G}$ are obtained by superposing element matrices as

$$\boldsymbol{K}^L = \sum_{e^L \in E^L} \boldsymbol{K}^L(e^L), \tag{2.23}$$

$$\boldsymbol{K}_{\mathrm{i}}^L = \sum_{e^L \in E^L} \boldsymbol{K}_{\mathrm{i}}^L(e^L), \tag{2.24}$$

$$\overline{\boldsymbol{K}_L^G} = \sum_{e^G \in \overline{E_L^G}} \boldsymbol{K}^G(e^G), \tag{2.25}$$

where the element matrices are given by

$$\boldsymbol{K}^L(e^L)_{ij} = \int_{e^L} \nabla N_i^L \cdot \boldsymbol{\sigma} \nabla N_j^L d\Omega, \tag{2.26}$$

$$\boldsymbol{K}_{\mathrm{i}}^L(e^L)_{ij} = \int_{e^L} \nabla N_i^L \cdot \boldsymbol{\sigma}_{\mathrm{i}} \nabla N_j^L d\Omega, \tag{2.27}$$

$$\boldsymbol{K}^G(e^G)_{ij} = \int_{e^G} \nabla N_i^G \cdot \boldsymbol{\sigma} \nabla N_j^G d\Omega \tag{2.28}$$

for nodes i and j of elements e^L and e^G. From (2.22), we finally obtain

$$\boldsymbol{K}^L \boldsymbol{\phi}^L + \boldsymbol{K}_{\mathrm{i}}^L \boldsymbol{V}_{\mathrm{m}}^L + \boldsymbol{\lambda} = \boldsymbol{0} \text{ on } \Omega^L \backslash \Gamma_P, \tag{2.29}$$

$$\overline{\boldsymbol{K}_L^G} \boldsymbol{\phi}^G - {\boldsymbol{I}_G^L}^T \boldsymbol{\lambda} = \boldsymbol{0} \text{ on } \overline{\Omega_L^G} \backslash \Gamma_P, \tag{2.30}$$

$$\boldsymbol{\phi}^L = \boldsymbol{\phi}_{\mathrm{p}} \text{ and } \boldsymbol{\phi}^G = \boldsymbol{\phi}_{\mathrm{p}} \text{ on } \Gamma_P,$$

with the constraint condition in (2.20). In section 3.1, we derive a local-global multigrid solution algorithm for (2.29) and (2.30). From (2.29), the nodal values of the Lagrange multiplier at Γ_{LG} can be interpreted as the electric currents from the local

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

mesh. In other words, the electric currents passing through the element surface from the local mesh elements are integrated on the local nodes at the interface. In (2.30), these nodal values of the currents from the local mesh are distributed by ${\boldsymbol{I}_G^L}^T$ to the global mesh nodes at $\overline{\Gamma_{LG}}$, and they are balanced with the currents from the global mesh elements in $\overline{E_L^G}$. In this way, the current balance is ensured at the interface. In the next section, we will see that this results in the conservation of the electric currents passing through the electrodes.

2.3. Conservation Properties of the Electric Currents Passing through the Electrodes. For simplicity, at first we assume that the boundaries Γ_P of the electrodes do not intersect with the interface boundary Γ_{LG}. Later, this condition is relaxed to some extent. The above assumption implies that the nodes on Γ_P are exclusively divided into local and global sections:

$$\Gamma_P = \Gamma_P^L \cup \Gamma_P^G. \tag{2.31}$$

We define the nodal residual components at nodes $i \in \Gamma_P^L$ and $j \in \Gamma_P^G$ by

$$r_i^L = -\int_{\Omega^L} \nabla N_i^L \cdot (\boldsymbol{\sigma}\nabla\phi^L + \boldsymbol{\sigma}_{\mathrm{i}}\nabla V_{\mathrm{m}}^L) d\Omega = -(\boldsymbol{K}^L\boldsymbol{\phi}^L + \boldsymbol{K}_{\mathrm{i}}^L\boldsymbol{V}_{\mathrm{m}}^L)_i,\ i \in \Gamma_P^L, \tag{2.32}$$

$$r_j^G = -\int_{\overline{\Omega_L^G}} \nabla N_j^G \cdot \boldsymbol{\sigma}\nabla\phi^G d\Omega = -(\overline{\boldsymbol{K}_L^G}\boldsymbol{\phi}^G)_j,\ j \in \Gamma_P^G. \tag{2.33}$$

The conservation law through the entire domain Ω in the finite element context is stated below.

THEOREM 2.2. *Assume that $\{\phi^L, \phi^G\}$ is the finite element solution to* (2.21). *Then*

$$\sum_{i\in\Gamma_P^L} r_i^L + \sum_{j\in\Gamma_P^G} r_j^G = 0. \tag{2.34}$$

Proof. For the local and global meshes, let us define the following test functions:

$$w^L = 1 - \sum_{i\in\Gamma_P^L} N_i^L, \tag{2.35}$$

$$w^G = 1 - \sum_{j\in\Gamma_P^G} N_j^G. \tag{2.36}$$

From the assumption on Γ_P, $\boldsymbol{w}^L \equiv 1$ on Γ_{LG} and $\boldsymbol{w}^G \equiv 1$ on Γ_{LG}. Thus, from the natural requirement on the interpolation $\boldsymbol{I}_G^L$, we see that

$$\boldsymbol{I}_G^L\boldsymbol{w}^G \equiv \boldsymbol{w}^L \equiv 1 \text{ on } \Gamma_{LG}. \tag{2.37}$$

By substituting (2.35), (2.36), and (2.37) into (2.21), we obtain

$$\int_{\Omega^L} \nabla\left(1 - \sum_{i\in\Gamma_P^L} N_i^L\right) \cdot \boldsymbol{\sigma}\nabla\phi^L d\Omega + \int_{\overline{\Omega_L^G}} \nabla\left(1 - \sum_{j\in\Gamma_P^G} N_j^G\right) \cdot \boldsymbol{\sigma}\nabla\phi^G d\Omega$$
$$+ \int_{\Omega_H^L} \nabla\left(1 - \sum_{i\in\Gamma_P^L} N_i^L\right) \cdot \boldsymbol{\sigma}_{\mathrm{i}}\nabla V_{\mathrm{m}}^L d\Omega = 0.$$

By expanding the above equation, we obtain (2.34). □

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

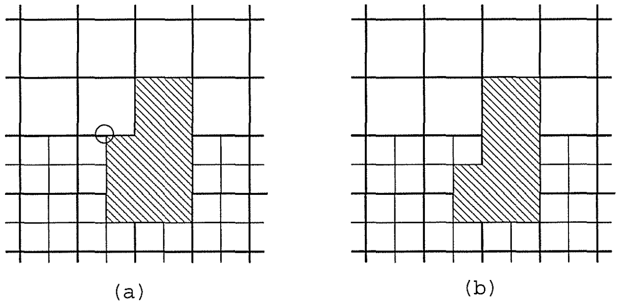

Fig. 2.5 *An undesired example for conservation* (a) *and the correction making this example conservative* (b). *In* (a), *a hanging node (marked with a circle) is located at the corner of the electrode.*

Physically, the residual at the electrode boundaries Γ_P can be interpreted as the current entering the torso through the surface of the electrodes. Here we assume that the weak solution ϕ to (2.16) is also the strong solution around Γ_p under a certain regularity of $\boldsymbol{\sigma}_{\mathrm{e}}$. That is, if the strong form equation

$$-\nabla \cdot \boldsymbol{\sigma}_{\mathrm{e}} \nabla \phi = 0 \text{ around } \Gamma_P \tag{2.38}$$

holds, we obtain from the Gauss divergence theorem

$$-\int_{\Omega^L} \nabla N_i^L \cdot \boldsymbol{\sigma}_{\mathrm{e}} \nabla \phi d\Omega = -\int_{\Gamma_P^L} N_i^L \vec{n} \cdot \boldsymbol{\sigma}_{\mathrm{e}} \nabla \phi d\Gamma, \ i \in \Gamma_P^L, \tag{2.39}$$

$$-\int_{\Omega^G} \nabla N_j^G \cdot \boldsymbol{\sigma}_{\mathrm{e}} \nabla \phi d\Omega = -\int_{\Gamma_P^G} N_j^G \vec{n} \cdot \boldsymbol{\sigma}_{\mathrm{e}} \nabla \phi d\Gamma, \ j \in \Gamma_P^G. \tag{2.40}$$

Here, $\vec{n}$ is the normal outward unit vector at Γ_P. Note that we assume that the electrode boundaries Γ_P are not attached to the heart muscle Ω_H. Thus, V_{m} and $\boldsymbol{\sigma}_{\mathrm{i}}$ do not appear in the equations. If we replace the analytical solutions on the left-hand sides of (2.39) and (2.40) by their discrete finite element approximations, they are, in fact, the residual components. Thus, the residual components $r_i^L, i \in \Gamma_P^L$, and $r_j^G, j \in \Gamma_P^G$, are the approximations of the fluxes on the right-hand sides of (2.39) and (2.40), respectively. If the interaction with the electric circuit connected to the electrodes is taken into account, the summations of the residual components at the electrode surfaces can be identified with the currents from the electrodes.

So far we have assumed that the electrode surface Γ_P does not intersect the interface boundary Γ_{LG}. However, this restriction is not essential for conservation, which is ensured as long as the boundaries Γ_{LG} follow the edges of the coarser elements at any interface. Figure 2.5 depicts an undesired example for conservation (a) and the correction making this example conservative (b). Note that a hanging node (marked with a circle) is located at the corner of the electrode in the undesired example. In general, we modify the definition of the test function w^L in (2.35) to match w^G in (2.36) at the interface:

$$w^L = 1 - \sum_{i \in \Gamma_P \setminus \Gamma_{LG}} N_i^L - \sum_{j \in \Gamma_P \cap \Gamma_{LG}} \boldsymbol{N}^L \left(\boldsymbol{I}_G^L \boldsymbol{\delta}_j^G\right)_{\Gamma_{LG}}. \tag{2.41}$$

Here, $\boldsymbol{\delta}_j^G$ is the vector on $\overline{\Omega_L^G}$ set to 1 at j and zero at the other nodes. From the above definition, we see that $\boldsymbol{I}_G^L \boldsymbol{w}^G \equiv \boldsymbol{w}^L$ holds at Γ_{LG}. By substituting this test

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

function into (2.21), we once again obtain (2.34) with the following definition of the residual at the interface:

$$r_j = -(\overline{\boldsymbol{K}_L^G}\boldsymbol{\phi}^G)_j - \left(\boldsymbol{I}_G^{L^T}(\boldsymbol{K}^L\boldsymbol{\phi}^L + \boldsymbol{K}_{\mathrm{i}}^L\boldsymbol{V}_{\mathrm{m}}^L)\right)_j, \; j \in \Gamma_{LG}\cap\Gamma_P. \tag{2.42}$$

Thus, in computing the residual component at the interface, the contributions from the residual components on the neighboring fine mesh hanging nodes must be taken into account.

3. The Local-Global Multigrid Algorithm on a Composite Mesh.

3.1. Derivation of the Algorithm.

Let us define the matrices $\boldsymbol{K}_L^G$ and $\boldsymbol{K}^G$ on the global mesh by

$$\boldsymbol{K}_L^G = \sum_{e^G \in E_L^G} \boldsymbol{K}^G(e^G), \tag{3.1}$$

$$\boldsymbol{K}^G = \sum_{e^G \in E^G} \boldsymbol{K}^G(e^G). \tag{3.2}$$

From the above definitions and (2.25), $\boldsymbol{K}^G$ is obtained by superposing $\boldsymbol{K}_L^G$ and $\overline{\boldsymbol{K}_L^G}$. Thus,

$$\boldsymbol{K}^G = \boldsymbol{K}_L^G + \overline{\boldsymbol{K}_L^G}. \tag{3.3}$$

Based on an idea proposed by Brandt [2], we add $\boldsymbol{K}_L^G\boldsymbol{\phi}^G$ to both sides of (2.30). Then we obtain the following equations equivalent to (2.29) and (2.30):

$$\boldsymbol{K}^L\boldsymbol{\phi}^L = -\boldsymbol{K}_{\mathrm{i}}^L\boldsymbol{V}_{\mathrm{m}}^L \text{ on } \Omega^L\backslash(\Gamma_{LG}\cup\Gamma_P), \tag{3.4}$$

$$\boldsymbol{\lambda} = -\boldsymbol{K}^L\boldsymbol{\phi}^L - \boldsymbol{K}_{\mathrm{i}}^L\boldsymbol{V}_{\mathrm{m}}^L \text{ on } \Gamma_{LG}, \tag{3.5}$$

$$\boldsymbol{K}^G\boldsymbol{\phi}^G = \boldsymbol{I}_G^{L^T}\boldsymbol{\lambda} + \boldsymbol{K}_L^G\boldsymbol{\phi}^G \text{ on } \Omega^G\backslash\Gamma_P. \tag{3.6}$$

If we define the residual vector $\boldsymbol{r}^L$ on the local mesh from (3.4), not only on $\Omega^L\backslash(\Gamma_{LG}\cup\Gamma_P)$ but also on Γ_{LG}, by

$$\boldsymbol{r}^L = -\boldsymbol{K}_{\mathrm{i}}^L\boldsymbol{V}_{\mathrm{m}}^L - \boldsymbol{K}^L\boldsymbol{\phi}^L \text{ on } \Omega^L\backslash\Gamma_P, \tag{3.7}$$

we see that it is identical to the Lagrange multiplier on Γ_{LG} from (3.5),

$$\boldsymbol{\lambda} = \boldsymbol{r}^L \text{ on } \Gamma_{LG}. \tag{3.8}$$

The above consideration naturally leads to the local-global solution process, shown in Figure 3.1, where steps (1) and (2) are iterated. Here, the injection $\hat{\boldsymbol{I}}_L^G$ is performed by injecting the local mesh nodal values into the global mesh nodes on Ω_L^G. Note that the correction with the interpolation is also performed on the interface Γ_{LG} in the global mesh correction phase. Thus, together with the assumption on the initial guess, the constraint condition in (2.20) is always satisfied. As for the relaxation on the local and global meshes, a multigrid V-cycle can be applied. In particular, for the local mesh relaxation, one V-cycle is sufficient to smooth the error with respect to the global mesh resolution, where the coarsest mesh of the V-cycle on the local mesh has the same spatial resolution as the global mesh in our implementation, as depicted in Figure 3.2.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

```
The local-global multigrid algorithm.
(0) Start
    - $\boldsymbol{g}^L := -\boldsymbol{K}_{\mathrm{i}}^L \boldsymbol{V}_{\mathrm{m}}^L$
    - Prepare an initial guess $\{\boldsymbol{\phi}^L, \boldsymbol{\phi}^G\}$ such that $\boldsymbol{\phi}^L = \boldsymbol{I}_G^L \boldsymbol{\phi}^G$ on $\Gamma_{LG}$
(1) Global mesh correction
    - Compute the residual on the local mesh $\boldsymbol{r}^L := \mathbf{g}^L - \boldsymbol{K}^L \boldsymbol{\phi}^L$ on $\Omega^L \backslash \Gamma_P$
    - Inject the local mesh solution $\boldsymbol{\phi}^G := \hat{\boldsymbol{I}}_L^G \boldsymbol{\phi}^L$ on $\Omega_{\mathrm{i}}^G$
    - Store $\boldsymbol{\phi}^G$ in another vector $\hat{\boldsymbol{\phi}}^G := \boldsymbol{\phi}^G$
    - Compute the right-hand side $\mathbf{g}^G := \boldsymbol{K}_L^G \hat{\boldsymbol{\phi}}^G + \boldsymbol{I}_G^{L^T} \boldsymbol{r}^L$
    - Compute an approximation for $\boldsymbol{K}^G \boldsymbol{\phi}^G = \mathbf{g}^G$
      with the fixed boundary values on $\Gamma_P$
    - Correction with the interpolation $\boldsymbol{\phi}^L := \boldsymbol{\phi}^L + \boldsymbol{I}_G^L(\boldsymbol{\phi}^G - \hat{\boldsymbol{\phi}}^G)$ on $\Omega^L$
(2) Smoothing on the local mesh
    - Relax the interior components of $\boldsymbol{\phi}^L$ for $\boldsymbol{K}^L \boldsymbol{\phi}^L = \mathbf{g}^L$
      with the fixed boundary values on $\Gamma_{LG} \cup \Gamma_P$
```

Fig. 3.1 *The local-global multigrid algorithm.*

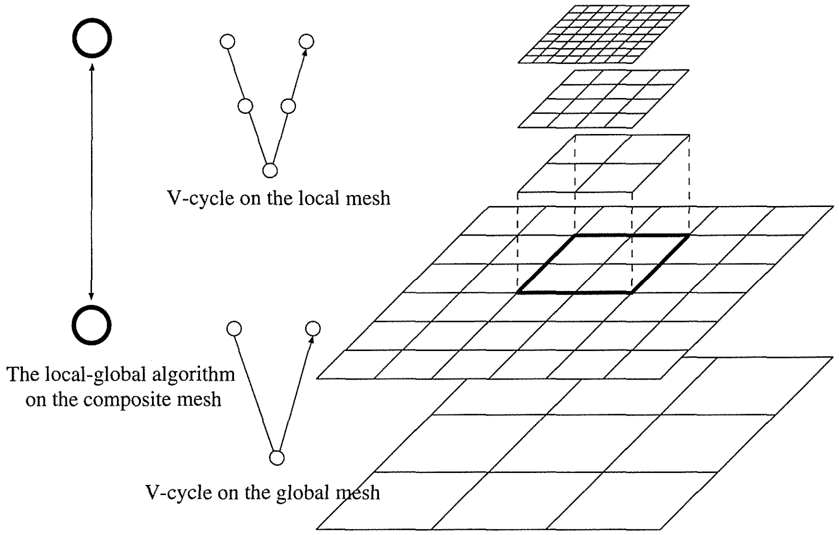
V-cycle on the local mesh
The local-global algorithm
on the composite mesh
V-cycle on the global mesh

Fig. 3.2 *A sketch of the local-global multigrid algorithm. In each mesh, the multigrid V-cycle is applied as a smoother. In particular, the coarsest grid of the local mesh V-cycle has the same resolution as the finest global mesh.*

The local-global multigrid algorithm obtained above is similar to the multilevel adaptive technique (MLAT) proposed by Brandt [2]. However, the MLAT was described for finite difference or finite volume discretizations [22] and derived from the full approximation scheme (FAS) [2, 22], originally proposed to solve a nonlinear problem with a multigrid. An interesting point here is that the MLAT is naturally derived by extending the Lagrange multiplier in (3.6) at the local-global interface Γ_{LG} to the inside of the fine finite element mesh, where it can be interpreted as the residual. Also note that in the standard implementation of MLAT [22], the residual at the fine

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

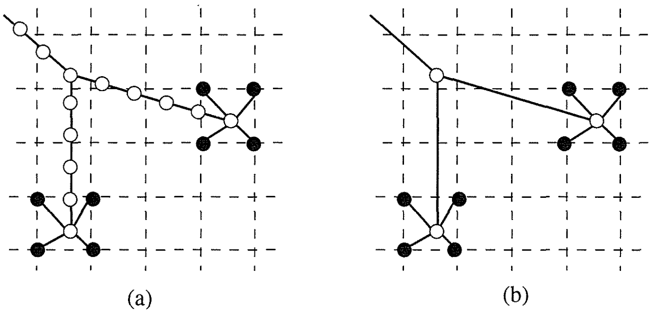

Fig. 3.3 *Original Purkinje network* (a) *and the network after the elimination* (b). *Black nodes are shared nodes with the voxel mesh.*

grid boundary is not transferred to the coarse grid, whereas in the above algorithm the residual components at the local mesh interface boundaries certainly affect the right-hand side of the coarse mesh equation. As we have seen in section 2.3, this is an essential point to ensure the conservation property at the local-global interface. Other techniques to ensure conservation have been introduced, for example, in [1, 14, 31] for finite volume discretizations. On the other hand, in common adaptive finite element approaches, a special refinement strategy is adopted at the fine-coarse interface so that hanging nodes are not present. In these approaches, conservation is automatically ensured. However, a method that allows hanging nodes provides easier mesh generation, in particular, for hexahedral elements.

3.2. Treatment of the Purkinje Fiber Network. In this section, we describe the special treatment of the Purkinje fibers in the local-global multigrid algorithm. As mentioned in section 1, the Purkinje network is modeled by one-dimensional elements, as is commonly done in the cardiovascular literature. In our simulator, only end-point nodes of the Purkinje network are connected to the voxel mesh nodes, as shown in Figure 3.3(a). Although a fairly fine spatial resolution is required for the Purkinje one-dimensional elements, we can eliminate most of the unknowns before solving the potential problem. This situation is illustrated in Figure 3.3. Nodes on the Purkinje network with only two edges connected can be eliminated without increasing the number of edges. Therefore, we do not apply any coarsening to the matrix on the reduced Purkinje nodes when constructing the matrix at the coarse level. In this case, although we have to invert the matrix completely on the reduced Purkinje nodes at the smoothing steps at each level of the multigrid cycle, this does not result in a crucial overhead due to the smaller size of the matrix on the reduced Purkinje nodes.

Here, we show how to construct the matrix on the global mesh for the reduced Purkinje network. In the following, the subscript letters "s" and "r" represent shared nodes and reduced Purkinje nodes, respectively. We assume that the shared nodes are included in the local voxel mesh nodes. Under this notation, a vector $\tilde{\phi}^L$ on the local mesh (involving the reduced Purkinje nodes) is represented as

$$\tilde{\phi}^L = \begin{bmatrix} \phi^L \\ \phi^L_{\mathrm{r}} \end{bmatrix}. \tag{3.9}$$

According to the above block representation, the coefficient matrix on the local mesh

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

is then represented by

$$\tilde{\boldsymbol{K}}^L = \begin{bmatrix} \boldsymbol{K}^L + \boldsymbol{D}_{\mathrm{s}} & \boldsymbol{K}_{\mathrm{sr}} \\ \boldsymbol{K}_{\mathrm{rs}} & \boldsymbol{K}_{\mathrm{rr}} \end{bmatrix}. \tag{3.10}$$

Here, the matrices $\boldsymbol{D}_{\mathrm{s}}$, $\boldsymbol{K}_{\mathrm{sr}}$, $\boldsymbol{K}_{\mathrm{rs}}$, and $\boldsymbol{K}_{\mathrm{rr}}$ originate from the reduced stiffness matrices on the reduced Purkinje network. In order to construct an appropriate local-global multigrid algorithm, the reduced Purkinje network should also be connected to the global voxel mesh. In our approach, the matrix on the reduced Purkinje network is not coarsened, as mentioned above. Thus, the coefficient matrix on the global mesh is given by

$$\tilde{\boldsymbol{K}}^G = \begin{bmatrix} \boldsymbol{K}^G + {\boldsymbol{I}_G^L}^T \boldsymbol{D}_{\mathrm{s}} \boldsymbol{I}_G^L & {\boldsymbol{I}_G^L}^T \boldsymbol{K}_{\mathrm{sr}} \\ \boldsymbol{K}_{\mathrm{rs}} \boldsymbol{I}_G^L & \boldsymbol{K}_{\mathrm{rr}} \end{bmatrix}. \tag{3.11}$$

The interpolation $\tilde{\boldsymbol{I}}_G^L$ from the global to the local mesh involving the reduced Purkinje nodes is defined by

$$\tilde{\boldsymbol{I}}_G^L = \begin{bmatrix} \boldsymbol{I}_G^L & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix}. \tag{3.12}$$

Here, $\boldsymbol{I}$ denotes the identity mapping on the reduced Purkinje nodes. Under these matrices and mappings, the local-global multigrid algorithm in section 3.1 can also be performed with the Purkinje fiber network.

3.3. Relaxation of the Local-Global Multigrid Algorithm. In this section, we describe some details of the multigrid V-cycle that is applied "as the relaxation" in the local-global multigrid algorithm. As for the interpolation from the coarse to the fine mesh, we define two different operators, $\boldsymbol{I}_c^f$ and $\acute{\boldsymbol{I}}_c^f$, as follows. For $\boldsymbol{I}_c^f$, standard weights are chosen, whereby weights for the voxel mesh nodes that are outside Ω are set to zero. Note that the sum of the interpolation weights of $\boldsymbol{I}_c^f$ is not equal to one if one of the neighboring coarse nodes is outside Ω. In the definition of $\acute{\boldsymbol{I}}_c^f$, the weights are adjusted so that their sum is equal to one, except for the fine nodes, all of whose neighboring coarse nodes are outside Ω. In our implementation, we apply $\boldsymbol{I}_c^f$ as the interpolation operator and its transpose as the restriction operator in the multigrid V-cycles, whereas in the determination of the coefficient matrix $\boldsymbol{K}^c$ on the coarse mesh from the coefficient matrix $\boldsymbol{K}^f$ on the fine mesh, we apply $\acute{\boldsymbol{I}}_c^f$ as follows:

$$\boldsymbol{K}^c = (\acute{\boldsymbol{I}}_c^f)^T \boldsymbol{K}^f \acute{\boldsymbol{I}}_c^f. \tag{3.13}$$

In our experience, the above-mentioned strategy (using different interpolations in the multigrid cycle and in the determination of the coarse mesh matrix) results in the best convergence. For example, using $\acute{\boldsymbol{I}}_c^f$ for both stages leads to convergence stagnation when the Purkinje fiber network is connected, while using $\boldsymbol{I}_c^f$ for both stages results in slower convergence. Further study of this will be part of our future research.

The other key issue for robust convergence in the given potential problem is the choice of smoother in the multigrid. In this application, the electric conductivity has an anisotropy in the heart muscle along the fabric construction and also jumps in coefficients at the interfaces between different organs. Furthermore, the torso boundaries given on the finest level on the global mesh do not necessarily fit with the coarser voxel elements. These problems trigger convergence difficulties for the standard multigrid method. Therefore, we adopt an incomplete Cholesky (IC) smoother since it is

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

more powerful than a Gauss–Seidel smoother for jumping coefficient and anisotropic problems (see, for example, [22] or [33]). Thus, the coarse mesh correction may be somewhat less accurate. In the case where the Purkinje fiber network is connected, we apply the IC smoother on the voxel part and a sparse direct solution method on the reduced Purkinje network part with the two-block representation as in (3.10). In general, the linear equation to be solved at an arbitrary level can be represented as follows (see (3.11)):

$$\begin{bmatrix} \boldsymbol{K}+\boldsymbol{I}^{\mathrm{s}T}\boldsymbol{D}_{\mathrm{s}}\boldsymbol{I}^{\mathrm{s}} & \boldsymbol{I}^{\mathrm{s}T}\boldsymbol{K}_{\mathrm{sr}} \\ \boldsymbol{K}_{\mathrm{rs}}\boldsymbol{I}^{\mathrm{s}} & \boldsymbol{K}_{\mathrm{rr}} \end{bmatrix}\begin{bmatrix} \boldsymbol{\phi} \\ \boldsymbol{\phi}_{\mathrm{r}} \end{bmatrix}=\begin{bmatrix} \boldsymbol{g} \\ \boldsymbol{g}_{\mathrm{r}} \end{bmatrix}. \tag{3.14}$$

Here, $\boldsymbol{K}$ is the coefficient matrix on the grid where the smoother is applied, and $\boldsymbol{I}^{\mathrm{s}}$ denotes the interpolation operator of the shared nodes on the finest local mesh Ω^L from the grid where the smoother is applied. Under the above notation, one smoothing step is described as follows:

$$\boldsymbol{r}^{(k)}=\boldsymbol{g}-(\boldsymbol{K}+\boldsymbol{I}^{\mathrm{s}T}\boldsymbol{D}_{\mathrm{s}}\boldsymbol{I}^{\mathrm{s}})\boldsymbol{\phi}^{(k)}-\boldsymbol{I}^{\mathrm{s}T}\boldsymbol{K}_{\mathrm{sr}}\boldsymbol{\phi}_{\mathrm{r}}{}^{(k)}, \tag{3.15}$$

$$\text{Solve } \boldsymbol{M}\Delta\boldsymbol{\phi}^{(k+1)}=\boldsymbol{r}^{(k)}, \tag{3.16}$$

$$\boldsymbol{\phi}^{(k+1)}=\boldsymbol{\phi}^{(k)}+\Delta\boldsymbol{\phi}^{(k+1)}, \tag{3.17}$$

$$\boldsymbol{r}_{\mathrm{r}}{}^{(k+1/2)}=\boldsymbol{g}_{\mathrm{r}}-\boldsymbol{K}_{\mathrm{rr}}\boldsymbol{\phi}_{\mathrm{r}}^{(k)}-\boldsymbol{K}_{\mathrm{rs}}\boldsymbol{I}^{\mathrm{s}}\boldsymbol{\phi}^{(k+1)}, \tag{3.18}$$

$$\text{Solve } \boldsymbol{K}_{\mathrm{rr}}\Delta\boldsymbol{\phi}_{\mathrm{r}}^{(k+1)}=\boldsymbol{r}_{\mathrm{r}}^{(k+1/2)}, \tag{3.19}$$

$$\boldsymbol{\phi}_{\mathrm{r}}^{(k+1)}=\boldsymbol{\phi}_{\mathrm{r}}^{(k)}+\Delta\boldsymbol{\phi}_{\mathrm{r}}^{(k+1)}, \tag{3.20}$$

where $\boldsymbol{M}$ denotes the IC factorization of the matrix $\boldsymbol{K}+\boldsymbol{I}^{\mathrm{s}T}\boldsymbol{D}_{\mathrm{s}}\boldsymbol{I}^{\mathrm{s}}$.

Another important technique to improve robustness is the Krylov subspace acceleration technique. One iteration of the multilevel solution is applied as a preconditioner for the Krylov subspace method. In the case of a composite mesh application, the implementation of a matrix-vector product may require considerable effort. However, as shown in the literature [3, 32], one can obtain a Krylov subspace acceleration by recombining iterants when their residuals are available. Details of our acceleration algorithm for this application are given in the literature [30].

4. Numerical Experiments with a Realistic Torso Model and Purkinje Fiber Network.

In this section, we evaluate the performance of the local-global multigrid algorithm for a realistic model on which the real-life simulations described in section 1 are performed. In the model, the voxel mesh data of the organs in Figure 1.7 are prepared based on the Visible Human dataset [28]. The mesh sizes and intervals of the local and global voxel meshes are described in Table 4.1. The ratio of the local and global mesh intervals is equal to 4. An illustration of the Purkinje fiber network adopted in the simulation is depicted in Figure 4.1. The geometry of the network is based on an anatomical observation in [20]. Its conductivity is set to 100 mS/cm, which is much larger than the values in Table 1.1. The conductivity has been adjusted in order to reproduce the experimental observation of excitation propagation given in the literature [8]. The Purkinje-ventricular delay [15] is not taken into account in the current simulation, and the radius of the cross-section of the network is set to 0.05 cm except near the interconnection points with the heart muscle, where 0.01 cm or 0.03 cm is used. These radius values have also been adjusted in order to reproduce proper excitation conduction from the Purkinje fiber to the ventricular muscle. The total number of Purkinje nodes is 24509, of which 9882 nodes are shared with the local voxel mesh. The number of nonzero components in the matrix on the Purkinje

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Table 4.1 *Local and global mesh sizes and intervals.*

Local mesh		Global mesh	
Size (x, y, z)	Interval length	Size (x, y, z)	Interval length
(288, 288, 368)	0.4 mm	(256, 256, 288)	1.6 mm

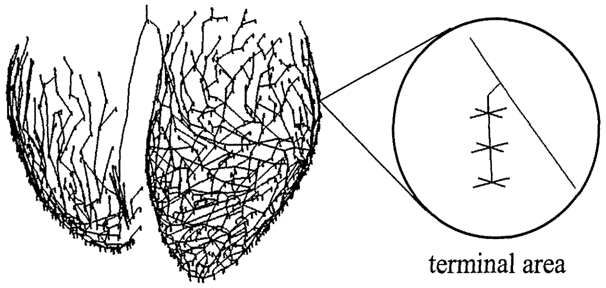

Fig. 4.1 *The Purkinje fiber network. At each terminal, the fiber is attached with* 12 *shared nodes.*

Table 4.2 *Elapsed time (in seconds) of one iteration of the local-global multigrid algorithm for the different divisions of the meshes.*

Division	L to G	G to L	Krylov	Smoother		Total
				Voxel	Purkinje	
$4 \times 4 \times 2$	0.09 (3.5%)	0.13 (4.9%)	0.12 (4.4%)	0.79 (30.1%)	0.30 (11.5%)	2.62
$4 \times 4 \times 3$	0.07 (3.4%)	0.09 (4.7%)	0.08 (4.4%)	0.56 (29.7%)	0.27 (14.4%)	1.89
$4 \times 4 \times 4$	0.05 (3.4%)	0.07 (4.7%)	0.06 (4.0%)	0.42 (27.8%)	0.22 (14.4%)	1.51
$4 \times 4 \times 5$	0.04 (3.3%)	0.06 (4.5%)	0.05 (3.9%)	0.33 (27.3%)	0.20 (16.1%)	1.22

network is 73709. However, after the reduction to the reduced Purkinje nodes, only 3937 nodes are left (besides the shared nodes). Finally, matrix $\boldsymbol{K}_{\mathrm{rr}}$ in (3.10) consists of 11989 nonzero components. The sparse LU factorization of matrix $\boldsymbol{K}_{\mathrm{rr}}$ has only 16481 nonzeros with a fill-in reduced ordering. Thus, the solution to $\boldsymbol{K}_{\mathrm{rr}}$ in (3.19) in the smoother is unlikely to produce a crucial overhead.

As for the relaxation on the local and global meshes, the multigrid V-cycle, with one pre- and one postsmoothing iteration, is applied. The V-cycle on the local mesh consists of three levels as the coarsest local mesh then has the same spatial resolution as the finest global mesh. One smoothing iteration is performed on the coarsest local mesh. The V-cycle on the global mesh consists of six levels. On the coarsest global mesh, twenty smoothing iterations are performed.

First, we examine the parallel performance of the local-global multigrid algorithm. Parallelization strategies used in our implementation are described in Appendix A. The timing results are measured on a PC-cluster composed of Pentium 4 processors (3.2GHz) connected via Myrinet. Shown in Table 4.2 are the elapsed times for the main processes in one iteration of the local-global multigrid algorithm, where the performance was examined for up to 80 processors. In the table, "L to G" and "G to L" denote the elapsed times for the local-to-global and global-to-local data transformations in (A.1), respectively. "Krylov" denotes the elapsed time for the Krylov acceleration. "Smoother" denotes the elapsed time for the smoothing iterations at

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

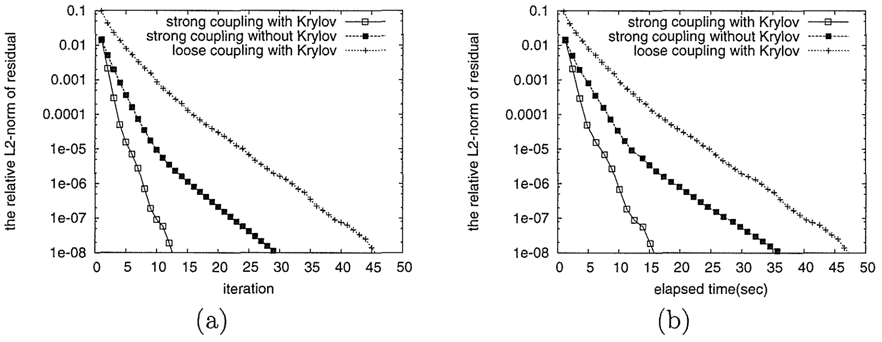

Fig. 4.2 *Convergence history with respect to* (a) *iterations and* (b) *elapsed time.*

all levels, where "Voxel" corresponds to the relaxation on the voxel meshes in (3.16) and (3.17), and "Purkinje" corresponds to the processes on the shared and reduced Purkinje nodes in (3.15), (3.18), (3.19), and (3.20). The numbers in parentheses are the ratios to the total time.

We observe a very satisfactory scaling. The elapsed times for the data transformations between the local and global meshes are relatively small in all cases. However, the elapsed time for the smoothing on the reduced Purkinje nodes is more pronounced as the number of processors increases. In particular, in (3.15) and (3.18), one-to-many communications are necessary to transfer the data to the shared nodes. It seems that this influences the parallel performance negatively.

Further, we analyze the convergence of the local-global multigrid algorithm. The convergence histories with respect to the number of iterations and the elapsed times are presented in Figure 4.2 for three different approaches. The notations "strong coupling with Krylov" and "strong coupling without Krylov" denote the local-global multigrid algorithm, respectively, with and without the Krylov acceleration technique, where the reduced Purkinje nodes are taken into account in the smoothing at every level, as described in section 3.3. On the other hand, the notation "loose coupling with Krylov" denotes the solution algorithm with the Krylov acceleration technique but solving the reduced Purkinje nodes and the voxel nodes in a decoupled way. This means that the direct solution on the reduced Purkinje nodes (including the shared nodes) and the local-global multigrid algorithm on the voxel mesh are performed alternately. In this case, the coupling with the reduced Purkinje nodes is dealt with only at the finest level on the local mesh.

In the Krylov acceleration technique, up to five iterants are recombined for the acceleration, and the acceleration process is restarted every five iterations. The results in Figure 4.2 show the effectiveness of the Krylov acceleration technique and the importance of the smoothing on the reduced Purkinje nodes at every level. The convergence speed of the loose coupling approach is obviously much slower than that of the proposed local-global multigrid algorithm. Even though there is considerable overhead in dealing with the reduced Purkinje nodes at every level, the proposed algorithm is still significantly faster than the loose coupling approach with respect to the elapsed time. In our real-life simulations, we commonly adopt 10^{-5} as the convergence tolerance for the relative L2-norm of the residual. Therefore, one solution takes approximately 8 seconds. If we solve the potential problem every 1 or 0.2 ms, about 8000 or 40000 seconds, respectively, are needed for the solutions to a 1 second

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

simulation with 80 processors. This is approximately 40 or 70%, respectively, of the total elapsed time in the ECG simulation to obtain the results given in Figure 2.2.

5. Conclusions. A parallel solution to the bidomain equation that appears in the excitation propagation analysis of the human heart was constructed. The stability of the explicit scheme was analyzed and an efficient multigrid technique to solve the potential problem with the torso model was introduced. In our approach, the potential problem was discretized on the composite mesh composed of a fine local mesh around the heart and a coarse global mesh covering the torso. A conservative finite element discretization adopting the Lagrange multiplier approach was introduced and a multigrid solution technique for this discretization was naturally derived. Furthermore, a method to combine the Purkinje fiber network with the multigrid solution technique was shown, whereby the matrix on the Purkinje network was reduced before entering the multilevel solver and the reduced matrix on the network was dealt with by a direct solution method at every level. The parallel efficiency and good convergence results were proved through an experiment with a realistic simulation model.

Appendix A. Parallelization Strategy. Here, we introduce the parallelization strategy of the bidomain analysis on a distributed memory parallel computer. Figure A.1 depicts the general situation for a composite mesh.

Note that there is no problem regarding the conservation discussed in section 2.3, even in the case where the local voxel mesh covers more than the torso region, as long as the boundaries of the torso on the local mesh coincide with the global mesh edges. In such a case, the local voxel mesh nodes on the torso boundaries are excluded from Γ_{LG}. Thus, there is no exchange of current there.

As shown on the right-hand side of Figure A.1, the solution process is decomposed into three phases with respect to data distribution between the processors. The first

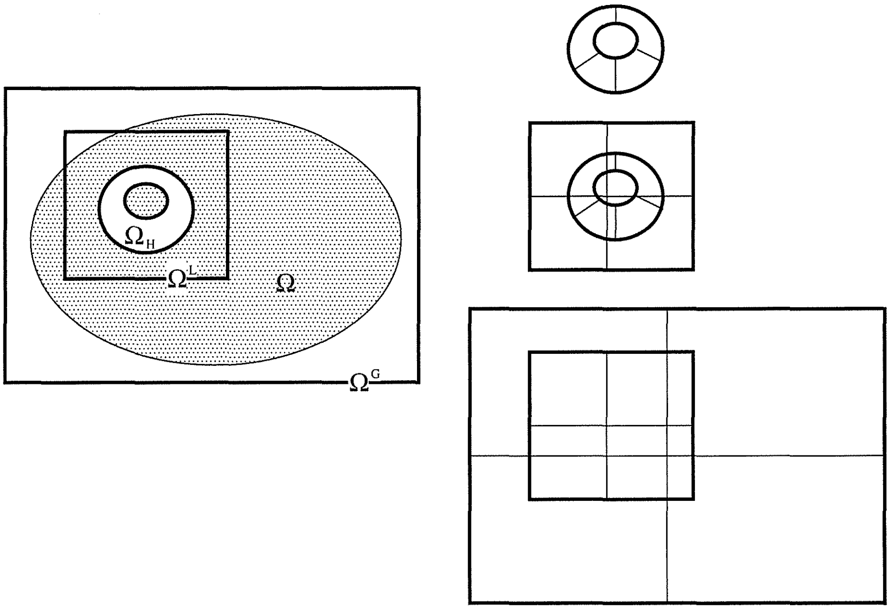

Fig. A.1 *(Left) two-dimensional image of three meshes (Ω_H (on the heart), Ω^L (local rectangular mesh), Ω^G (global rectangular mesh)) and (right) their partitioning (for 4 processes).*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

phase corresponds to the explicit Euler time integration of the inner iteration described in section 2.1. Here, the computation is performed only at the nodes on the heart muscle Ω_H. The second and third phases include the local-global multigrid algorithm described in section 3.1 on the local and global voxel meshes, respectively. For the second and third phases, the nodes are partitioned regularly in each direction. In contrast, the first phase is based on a partitioning of an unstructured graph which consists only of the nodes on the heart muscle Ω_H. For the partitioning, we adopt the graph partitioning tool ParMETIS [10] to obtain the partitioning information. Since these data distributions are not consistent, as can be seen in Figure A.1, redistribution of variables takes place at each phase change. The variables that are redistributed at each phase change are described in the diagram below:

$$\begin{tikzcd}
\Omega_H \arrow[d, shift right, leftarrow, "\boldsymbol{\phi}^L"'] \arrow[d, shift left, "\boldsymbol{K}_{\mathrm{i}}\boldsymbol{V}_{\mathrm{m}}"] \\
\Omega^L \arrow[d, shift right, leftarrow, "\boldsymbol{\phi}^G - \hat{\boldsymbol{\phi}}^G"'] \arrow[d, shift left, "{\boldsymbol{I}_G^L}^T \boldsymbol{r}^L, \hat{\boldsymbol{I}}_G^L \boldsymbol{\phi}^L."] \\
\Omega^G
\end{tikzcd} \tag{A.1}$$

The communication speed of the redistribution between the local and global meshes is crucial to the overall parallel performance because this redistribution is performed every cycle in the local-global multigrid algorithm. The overhead cost for the redistribution in the real-life application has been presented in section 4.

In each phase, parallelism is obtained in a standard way except for the IC smoother in the second and third phases. The IC smoother is modified, as it can easily be parallelized under the regular partitioning; that is, the local IC smoother is performed in each processor with one layer overlap at the subdomain boundaries.

In the case where the Purkinje fiber network is connected, the vectors $\boldsymbol{\phi}_{\mathrm{r}}$, $\boldsymbol{g}_{\mathrm{r}}$ and the matrices $\boldsymbol{K}_{\mathrm{sr}}$, $\boldsymbol{K}_{\mathrm{rs}}$, $\boldsymbol{K}_{\mathrm{rr}}$ in (3.14) are stored in one of the processors (say, Proc0), while the matrix for the interpolation on the shared nodes $\boldsymbol{I}^{\mathrm{s}}$ is distributed along the regular partitioning of the voxel mesh. Thus, the vector data $\boldsymbol{K}_{\mathrm{sr}}\boldsymbol{\phi}_{\mathrm{r}}^{(k)}$ on Proc0 is scattered in (3.15), and the distributed vector data $\boldsymbol{I}^{\mathrm{s}}\boldsymbol{\phi}^{(k+1)}$ is gathered to Proc0 in (3.18). This process may generate considerable overhead when the number of shared nodes is large.

Acknowledgments. The authors would like to thank S. Sugiura (University of Tokyo, M.D.), K. Sunagawa (Kyushu University Faculty of Medicine, M.D.), and M. Sugimachi and M. Inagaki (National Cardiovascular Center Research Institute, M.D.) for their support in constructing the heart muscle and torso models and for their useful suggestions in improving the accuracy of the simulation. They would like to thank H. Koga for his great contribution in constructing the Purkinje fiber network model. Finally, they would also like to thank C. W. Oosterlee, Ph.D., for reading the manuscript and for the useful comments.

REFERENCES

[1] D. Bai and A. Brandt, *Local mesh refinement multilevel techniques*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 109–134.

[2] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.

[3] A. Brandt and V. Mikulinsky, *On recombining iterants in multigrid algorithms and problems with small islands*, SIAM J. Sci. Comput., 16 (1995), pp. 20–28.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.