Parallel Solution of the Bidomain Equations with High Resolutions

Xing Cai,\textsuperscript{a,b} Glenn Terje Lines\textsuperscript{a} and Aslak Tveito\textsuperscript{a,b}

\textsuperscript{a}Simula Research Laboratory, P.O. Box 134, N-1325 Lysaker, Norway.
\textsuperscript{b}Department of Informatics, University of Oslo, P.O. Box 1080, Blindern, N-0316 Oslo, Norway

This paper is concerned with the parallel solution of the Bidomain equations and an associated forward problem, which can be used to simulate the electrical activity in the heart and torso. For achieving high-resolution simulations on parallel computers, a scalable and parallelizable numerical strategy must be used. We present therefore an advanced parallel preconditioner for a 2 × 2 block linear system that arises from finite element discretizations. The scalability of the preconditioner is studied by numerical experiments that involve up to 81 million unknowns in the block linear system.

1. INTRODUCTION

Normal electrical activity in the heart is very important for the heart’s pumping function. In order to study the relationship between disorders in the heart and the measurable ECG signals on the body surface, researchers have developed advanced mathematical models involving ordinary differential equations (ODEs) and partial differential equations (PDEs). Numerical strategies have thus been devised to enable computer simulations, see e.g. [1–3,7].

Due to factors such as the complexity of the equations and the irregular shapes of the domains, high resolutions in both time and space are needed to achieve sufficiently accurate computational results. One of the desired goals is to run full-scale 3D simulations with 0.1ms temporal resolution and 0.2mm spatial resolution in the heart domain. This gives rise to computations on a 3D grid with about 5 × 10^7 grid points, involving several thousand time steps. Clearly, efficient numerical strategies are required for doing such large-scale simulations. An important property of a desirable numerical strategy is good scalability, meaning that the computational effort per degree of freedom should remain roughly constant, independent of the growth of the problem size. Scalability is also essential for a parallel solution of the mathematical model, i.e., the overall computational demand also roughly remains constant independent of the number of processors used.

Following the work of [8] on designing an order-optimal sequential block preconditioner for Krylov iterative methods, we present in this paper its parallel counterpart that preserves the same efficiency. The scalability of this parallel preconditioner is studied through numerical experiments, which indicate that the desired spatial resolution of 0.2mm is achievable even on a moderately large parallel computing system.

The remainder of the paper is organized as follows. First, Section 2 gives a brief description of the mathematical model involving the Bidomain equations and an elliptic PDE in the torso. Then, Section 3 explains a numerical strategy for the mathematical model, which needs to solve a 2 × 2 block linear system per time step. Thereafter, Section 4 presents the design of a parallel block preconditioner for achieving scalable performance when solving such a 2 × 2 block linear system, whereas Section 5 is devoted to issues about parallelization. Finally, Section 6 shows the measurements of some numerical experiments before Section 7 gives a few concluding remarks.
2. THE MATHEMATICAL MODEL

A well-established mathematical model for describing the electrical activity in the heart is the Bidomain equations, see [9]. In this paper, the domain of the heart is denoted by $H$, and we consider the following form of the Bidomain equations involving the pair of primary unknowns: the transmembrane potential $v$ and the extracellular potential $u_e$.

\[
\chi C_m \frac{\partial v}{\partial t} + \chi I_{ion} = \nabla \cdot (M_i \nabla v) + \nabla \cdot (M_i \nabla u_e) \quad \text{in } H, \tag{1}
\]

\[
0 = \nabla \cdot (M_i \nabla v) + \nabla \cdot ((M_i + M_e) \nabla u_e) \quad \text{in } H. \tag{2}
\]

In (1), the nonlinear function $I_{ion}(v, s)$ represents the ionic current where $s$ denotes a vector of state variables describing the state of the cell membrane. There exist many different models of $I_{ion}$, almost all of them rely on solving an ODE system to find the $s$ vector, see e.g. [4,10]. That is, an ODE system of the form $d s / d t = F(v, s)$ models the electrical behavior of the cardiac cells. Moreover, $\chi$ is a surface-to-volume scaling factor, $C_m$ denotes the capacitance of the membrane, and $M_i$ denotes the intracellular conductivity tensor. Similarly, $M_e$ denotes the extracellular conductivity tensor in (2).

![Figure 1. A schematic 2D slice of the entire solution domain $\Omega = H \cup T$.](image)

In order to compute the electrical potential on the body surface, we supplement the Bidomain equations (1)-(2) with a forward problem, which is the following elliptic PDE describing the propagation of the electrical signal in the torso $T$ exterior to the heart:

\[
-\nabla \cdot (M_o \nabla u_o) = 0 \quad \text{in } T, \tag{3}
\]

where $u_o$ is the electrical potential in the torso and $M_o$ denotes the associated conductivity tensor. Therefore, our mathematical model consists of three PDEs (1)-(3), for which the entire solution domain $\Omega = H \cup T$ is depicted in Figure 1. As for boundary conditions, we have

\[
M_i \frac{\partial}{\partial n} (v + u_e) = 0, \quad u_e = u_o, \quad M_e \frac{\partial u_e}{\partial n} - M_o \frac{\partial u_o}{\partial n} = 0 \quad \text{on } \partial H, \quad M_o \frac{\partial u_o}{\partial n} = 0 \quad \text{on } \partial T.
\]

In the temporal direction, the mathematical model is to be solved for a time period with known initial values for $v$, $s$, $u_e$, and $u_o$.

3. NUMERICAL STRATEGY

The basic idea behind discretizing (1) is to split this nonlinear PDE into two parts:

\[
\frac{\partial v}{\partial t} = -\frac{1}{C_m} I_{ion}(v, s) \quad \text{and} \quad \chi C_m \frac{\partial v}{\partial t} = \nabla \cdot (M_i \nabla v) + \nabla \cdot (M_i \nabla u_e), \tag{4}
\]
where the first part is to be solved together with an ODE system modeling the state variables $s$. In the temporal direction, the simulation time period for the entire mathematical model is divided into discrete time levels:

$$0 = t_0 < t_1 < t_2 \cdots$$

At each time level $t_l$, $l \geq 1$, the solutions from the previous time level, $v^{l-1}, u^{l-1}_v, u^{l-1}_o, s^{l-1}$, are used as the starting values. Using a $\theta$-rule, where $0 \leq \theta \leq 1$, we can construct a flexible numerical strategy whose work per time step consists of solving an ODE system twice, separated by the solution of a system of PDEs. We refer to [8] for a detailed explanation of the strategy, and remark that $\theta = 1/2$ gives rise to a second-order accurate temporal discretization. In summary, the computational work for the time step $t_{l-1} \rightarrow t_l$ consists of the following sub-steps:

- First, a chosen ODE system

$$\frac{dv}{dt} = -\frac{1}{C_m}I_{ion}(v, s)$$
$$\frac{ds}{dt} = f(v, s)$$

is solved for $t \in (t_{l-1}, t_l + \theta(t_l - t_{l-1}))$, using the initial values $v^{l-1}$ and $s^{l-1}$. The results are an intermediate transmembrane potential solution $\hat{v}^{l-1}$ and an updated $\hat{s}^{l-1}$ vector.

- Second, the three PDEs (1)-(3) are solved simultaneously, where (1) only uses its second part in the splitting given in (4). The temporal discretization is of the following form:

$$\chi C_m \frac{\hat{v}^l - \hat{v}^{l-1}}{\Delta t} = (1 - \theta) \left( \nabla \cdot (M_i \nabla \hat{v}^{l-1}) + \nabla \cdot (M_e \nabla u^{l-1}_e) \right)$$
$$+ \theta \left( \nabla \cdot (M_i \nabla \hat{v}^l) + \nabla \cdot (M_e \nabla u^{l}_e) \right),$$

$$0 = (1 - \theta) \left( \nabla \cdot (M_o \nabla \hat{u}^{l-1}) + \nabla \cdot ((M_i + M_e) \nabla u^{l-1}_e) \right)$$
$$+ \theta \left( \nabla \cdot (M_o \nabla \hat{u}^l) + \nabla \cdot ((M_i + M_e) \nabla u^{l}_e) \right),$$

$$0 = (1 - \theta) \nabla \cdot (M_o \nabla u^{l-1}_o) + \theta \nabla \cdot (M_o \nabla u^{l}_o).$$

We remark that $\hat{v}^l, u^{l}_v, u^{l}_o$ are the unknowns to be found. The spatial discretization, using, e.g., finite elements, will give rise to a $2 \times 2$ block linear system:

$$\begin{bmatrix}
\chi C_m I + \theta \Delta t A_v & \theta \Delta t A_v \\
\theta \Delta t A_v^T & \theta \Delta t A_u
\end{bmatrix}
\begin{bmatrix}
\hat{v}^l \\
\hat{u}^l
\end{bmatrix} = \begin{bmatrix}
b \\
0
\end{bmatrix}.$$  

In the above block linear system, $I$ and $A_v$ represent the mass matrix and the stiffness matrix associated with $M_i$ inside $H$, respectively. Moreover, we have combined the unknown values of $u^{l}_v$ and $u^{l}_o$ into one vector $u^l$. The sparse matrix $A_v$ arises from discretizing (2) and (3) together. That is, we solve a combined elliptic problem $-\nabla \cdot (M \nabla u) = 0$ in $\Omega$, using $M = M_i + M_e$ in $H$ and $M = M_o$ in $T$, see [8] for more details. The $A_v$ matrix is the same as $A_u$, except that $\hat{A}_v$ is padded with some additional zero columns to allow the multiplication $A_u u$, whereas the $\hat{A}_v^T$ matrix is the transpose of $\hat{A}_v$.

By extending the proof that is given in [5] to also include the torso domain, we can deduce that the $2 \times 2$ block matrix in (9) is symmetric and positive semi-definite. We remark that this property arises from the symmetry and positive definiteness of the conductivity tensors $M_i, M_e$ and $M_o$. Experiments have shown that the conjugate gradient (CG) method is an appropriate choice for (9). A suitable preconditioner is thus important for achieving rapid convergence. We present a scalable parallel preconditioner in Section 4.

- Third, the chosen ODE system (5) is solved again for $t \in (t_{l-1} + \theta(t_l - t_{l-1}), t_l]$, using the initial values $\hat{v}^l$ and $\hat{s}^{l-1}$. The computational results are stored in $v^l$ and $s^l$.

Since the above numerical strategy splits the solution of (1) into different parts, we denote it by the “semi-implicit” approach.
4. A PARALLEL PRECONDITIONER

We know from [8] that the following diagonal block system

\[
\begin{bmatrix}
\chi C_m I + \theta \Delta t A_v & 0 \\
0 & \theta \Delta t A_u
\end{bmatrix}
\] (10)

can work as an efficient preconditioner for (9). Here, we remark that the notation for the matrices $I$, $A_v$ and $A_u$ is the same as for (9). The rapid convergence of this preconditioner is due to the fact that (10) is spectrally equivalent to the $2 \times 2$ block matrix in (9), see [8]. During each preconditioning operation, the inverse of the diagonal block matrix (10) can be found by inverting $\chi C_m I + \theta \Delta t A_v$ and $\theta \Delta t A_u$ separately. It can be shown that the number of needed CG iterations, using the above preconditioner, is constant independent of the number of unknowns in (9).

To make the above preconditioner parallel while maintaining its scalability, we suggest a layered design. First, one or several additive Schwarz iterations (see [6]) work as a parallel solver for both $\chi C_m I + \theta \Delta t A_v$ and $\theta \Delta t A_u$. Let us demonstrate this for the latter system. One additive Schwarz iteration for $A_u$ can be expressed as

\[
A_u^{-1} \approx \sum_{i=0}^{P} A_{u,i}^{-1},
\] (11)

where it is assumed that the body domain $\Omega$ is partitioned into $P$ overlapping subdomains $\Omega_i$, $1 \leq i \leq P$. Thus, $A_{u,i}$ denotes a subdomain matrix that arises from a discretization restricted to $\Omega_i$. Note that $A_{u,0}$ is associated with a discretization on a very coarse global grid. The use of $A_{u,0}^{-1}$, also called coarse grid correction, is to ensure convergence independent of the number of subdomains $P$, see e.g. [6].

Second, as an approximate subdomain solver, we use multigrid V-cycles. This is because the complexity of such cycles is linear with respect to the number of subdomain unknowns. The construction of the required hierarchy of subdomain grids is described in Section 5.

In summary, the combination of additive Schwarz iterations on the “global layer” and multigrid V-cycles on the “subdomain layer” constitutes our parallel preconditioner. Scalability with respect to the number of unknowns arises from the spectral equivalence between (9) and (10), together with using multigrid as subdomain solvers. Convergence independent of the number of subdomains is due to Schwarz iterations with coarse grid correction.

5. PARALLEL COMPUTING

Let $P$ denote the number of processors, we adopt the approach of explicit domain partitioning that divides the entire computational work among the processors. Recall that the body domain $\Omega$ consists of the heart $H$ and the torso $T$, so we partition both $H$ and $T$ into $P$ pieces, see Figure 2. Processor $i$ is responsible for the composite subdomain $\Omega_i = H_i \cup T_i$. In addition, we also introduce a certain amount of overlap between the subdomains to enable the additive Schwarz iterations that are used in the parallel preconditioner.

During parallel computation, the work on processor $i$ consists of local operations that are restricted to $H_i$ and $T_i$. That is, local finite element discretizations are carried out independently on the processors. The ODE system is also solved independently by the processors on their local heart subdomain points. No communication between processors is needed for these two tasks. However, during the parallel iterations for solving the global $2 \times 2$ block linear system, subdomain local operations need to be interleaved with inter-processor communication.

Recall also that we want to run multigrid V-cycles as subdomain solvers in the additive Schwarz iterations for approximating $(\chi C_m I + \theta \Delta t A_v)^{-1}$ and $A_u^{-1}$. This requires two associated hierarchies of subdomain grids. One approach to achieving this is that we start with a global $H$ grid and a global $T$ grid, both of medium resolution. Then, we partition the two global grids into $P$ overlapping parts respectively. Afterwards, the subdomain $H$ and $T$ grids are refined several times on each subdomain, giving rise to a hierarchy of subdomain $H$ grids and a hierarchy of subdomain $T$ grids. Note that a hierarchy of subdomain $\Omega$ grids is just the union of the associated hierarchies of subdomain $H$ and $T$ grids.
6. NUMERICAL EXPERIMENTS

In Table 1, we show the number of CG iterations needed for achieving convergence when solving the $2 \times 2$ block linear system (9) with $\theta = 1/2$. We have used the parallel preconditioner that is described in Section 4. The number of local grids in the subdomain grid hierarchy is listed in the first column, whereas the summed number of $v$ and $u$ unknowns is shown in the second column. We can observe convergence scalability by noting that the number of CG iterations remains almost constant, independent of the number of unknowns or $P$. We also remark that the largest system size in Table 1 corresponds to over 39 million grid points in $H$, quite close to a spatial resolution of 0.2mm.

Table 1
The number of CG iterations needed to solve the block linear system (9), using the parallel block preconditioner. Convergence is claimed when the $L_2$-norm of the global residual is reduced by a factor of $10^4$.

<table>
<thead>
<tr>
<th>grid levels</th>
<th># unknowns $(v + u)$</th>
<th>$P = 2$</th>
<th>$P = 4$</th>
<th>$P = 8$</th>
<th>$P = 16$</th>
<th>$P = 32$</th>
<th>$P = 64$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>302,166</td>
<td>9</td>
<td>9</td>
<td>10</td>
<td>12</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1,552,283</td>
<td>9</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10,705,353</td>
<td></td>
<td></td>
<td>13</td>
<td>13</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>81,151,611</td>
<td></td>
<td></td>
<td>14</td>
<td>15</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

Table 2 is meant to reveal the parallel efficiency of the preconditioned parallel CG iterations. The wall-clock time measurements are obtained on an SGI Origin 3800 system. As a comparison, we also list the corresponding wall-clock time measurements of a so-called “decoupled” strategy, which first solves (1), and then a combined system for (2)-(3). In terms of solving linear systems, the “decoupled” strategy first solves a linear system $(\chi C_{1m}I + \theta \Delta t A_{m})v = b_v$, and then a linear system $A_{u}u = b_u$, each system only once per time step. Note that the wall-clock time measurements of the “decoupled” strategy in Table 2 is the sum of solving these two linear systems.

The measurements in Table 2 show that it is possible for the advanced “semi-implicit” strategy to achieve the same level of computational efficiency as the simple but accuracy-wise much inferior “decoupled” strategy. We remark that the scalability of the “semi-implicit” strategy is due to the parallel block preconditioner. We also remark that the measurements in Table 2 are obtained on a heavily-loaded SGI Origin system with many users competing for the same resource, so care should be taken when interpreting the speedup results.
Table 2
Wall-clock time measurements (in seconds) of two different strategies for solving the linear system(s) for one time step.

<table>
<thead>
<tr>
<th>grid levels</th>
<th># unknowns (v + u)</th>
<th>“decoupled” strategy</th>
<th>“semi-implicit” strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1,552,283</td>
<td>92.66 41.46 26.60</td>
<td>32.66 46.08 44.11</td>
</tr>
<tr>
<td>4</td>
<td>10,705,353</td>
<td>248.65 144.65 50.77</td>
<td>399.64 224.15 148.28</td>
</tr>
<tr>
<td>5</td>
<td>81,151,611</td>
<td>4406.61 2538.32 1412.63</td>
<td>4094.97 2915.88 1902.91</td>
</tr>
</tbody>
</table>

7. CONCLUDING REMARKS

We have presented a layered design of a parallel preconditioner for the $2 \times 2$ block system (9). This parallel preconditioner enables scalable performance of the advanced “semi-implicit” numerical strategy for the mathematical model (1)-(3). Even for a $2 \times 2$ block system involving more than 81 million unknowns, solution is obtained within a reasonable amount of time on a moderately large parallel system.

Acknowledgement

We acknowledge the support from the Research Council of Norway through a grant of computing time (Programme for Supercomputing).

REFERENCES