Adaptive Mesh Refinement and Adaptive Time Integration for Electrical Wave Propagation on the Purkinje System

Wenjun Ying*

Department of Mathematics, MOE-LSC and Institute of Natural Sciences, Shanghai Jiao Tong University, Minhang, Shanghai 200240, P. R. China.

Craig S. Henriquez†

Departments of Biomedical Engineering and computer science, Duke University, Durham, NC 27708-0281

Abstract

A both space and time adaptive algorithm is presented for simulating electrical wave propagation in the Purkinje system of the heart. The equations governing the distribution of electric potential over the system are solved in time with the method of lines. At each timestep, by an operator splitting technique, the space-dependent but linear diffusion part and the nonlinear but space-independent reactions part in the partial differential equations are integrated separately with implicit schemes, which have better stability and allow larger timesteps than explicit ones. The linear diffusion equation on each edge of the system is spatially discretized with the continuous piecewise linear finite element method. The adaptive algorithm can automatically recognize when and where the electrical wave starts to leave or enter the computational domain due to external current/voltage stimulation, self-exciting, or local change of membrane properties. Numerical examples demonstrating efficiency and accuracy of the adaptive algorithm are presented.

*wying@sjtu.edu.cn

†ch@duke.edu

I. INTRODUCTION

One of the long-recognized challenges in modeling cardiac dynamics [1–4] is developing efficient and accurate algorithms that can accommodate the widely varying scales in both space and time [5]. The electrical wave fronts typically occupy only a small fraction of the domain, are very sharp (in space) and change very rapidly (in time) while, in the region away from the wave fronts, the electrical potential is spatially broad and changes more slowly. With standard numerical methods on uniform grids, very small mesh parameters and very small timesteps must be used to correctly resolve the fine details of the sharp and rapidly changing wave fronts. These discretization parameters are often chosen heuristically and are fixed throughout the simulation, even if conditions change. Adaptive mesh-refinement (AMR) methods have been proposed as a solution, in which coarse grids and large timesteps are used in the area where the electrical potential is changes slowly and fine grids and small timesteps are applied only in the region where the sharp electrical waves are located and the action potential changes very rapidly. Using this approach, the numbers of grid nodes and timesteps used with the adaptive algorithm are to some extent optimized. The original AMR algorithm was first proposed by Berger and Oliger for hyperbolic equations [6] and shock hydrodynamics [7]. The methods has been applied to cardiac simulations by Cherry, Greenside and Henriquez [8, 9] and Trangenstein [10].

The Berger-Oliger's AMR algorithm is a hierarchical and recursive integration method for time-dependent partial differential equations. It starts time integration on a relatively coarse grid with a large time step. The coarse grid is further locally refined if the computed solution at part of the domain is estimated to have large errors. Better solutions are obtained by continuing time integration on the fine grid with a smaller time step until both coarse and fine grids reach the same time, called synchronization of levels. The fine grid may be further locally refined and is dynamically changing, which leads to both space and time adaptive algorithm.

The standard implementation of Berger-Oliger's AMR algorithm uses block-structured grids and assumes that the underlying grids are logically rectangular and can be mapped onto a single index space. While the method has been shown provide computational and accuracy advantages, it is challenging to apply to domains with complex geometry.

In this paper, we present an AMR algorithm that can be used for unstructured grids.

The method is demonstrated on both idealized and realistic tree-like domains similar that found in the His-Purkinje system. The algorithm is based on that proposed by Trangenstein [6] where operator splitting technique is used to separate the space-independent reactions part from the linear diffusion part during time integration. Both the reactions and the diffusion parts are integrated with an implicit scheme. This allows larger and adaptive timesteps. As the AMR algorithm naturally provides a hierarchy of multilevel grids, the linear systems resulting from space discretization of the linear diffusion on the adaptively refined grids are solved by a standard geometric multigrid solver. The results show that uniform coarse discretization can lead to conduction failure or changes in dynamics in some parts of the branching network when compared to a uniform fine grid. The results also show that AMR scheme with adaptive time integration (AMR-ATI) can yield results as accurate as the uniform fine grid but with a speedup of 15 times.

The remainder of the work is organized as follows. In section II, we describe the partial differential equations, which model electrical wave propagation on the Purkinje system. In section III and section IV, we discuss on time integration and space discretization for the reaction-diffusion equations. In section V, the adaptive mesh refinement and adaptive time integration algorithm is outlined. In section VI, some simulation results with the AMR-ATI algorithm are presented.

II. DIFFERENTIAL EQUATIONS

Suppose that we are given a fiber network of the Purkinje system with N_V vertices and N_E edges. Let d_i be the number of edges connected to vertices V_i, for each $i = 1, 2, \cdots, N_V$. We call d_i as the *degree* of the vertex V_i. A vertex V_i with $d_i = 1$ is called a *leaf-vertex*. Otherwise, it is called a *nonleaf-vertex*. Denote the j^{th} edge E_j in the fiber network by $E_j = [x_0^{(j)}, x_1^{(j)}]$. Denote the edges that have vertex V_i as the common endpoint by $E_{i_1}, E_{i_2}, \cdots, E_{i_{d_i}}$. Denote by $x_{b_i}^{(i_1)}, x_{b_i}^{(i_2)}, \cdots, x_{b_i}^{(i_{d_i})}$ the endpoints of the adjacent edges, which overlap with vertex V_i. The subscript b_i is either 0 or 1.

On each edge E_j of the fiber network, the distribution of the electric/action potential is determined by the conservation of currents, and could be described by a partial differential equation coupled with a set of ordinary differential equations,

$$\frac{a}{2}\frac{\partial J(t,x)}{\partial x} + C_m\frac{\partial u(t,x)}{\partial t} + I_{ion}(u,\mathbf{q}) = I_{stim}(t,x) \quad \text{for } x_0^{(j)} < x < x_1^{(j)}, \tag{1}$$

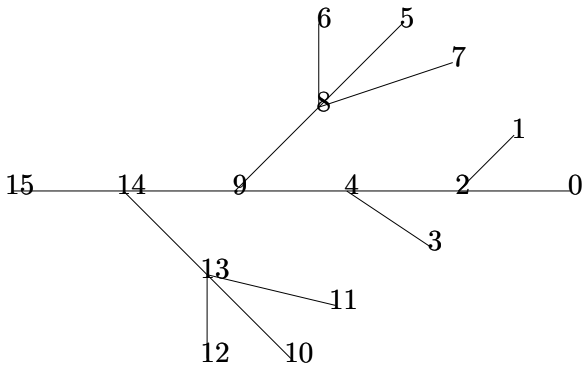


FIG. 1: An idealized branch of the Purkinje system

$$\frac{d\mathbf{q}(t,x)}{dt} = \mathcal{M}(u,\mathbf{q}), \tag{2}$$

with

$$J(t,x) = -\frac{1}{R}\frac{\partial u(t,x)}{\partial x}$$

be the flux. Here, a (units: cm) denotes a typical radius of the fiber; C_m (units: μF/cm^2) be the membrane capacitance constant; R (units: k$\Omega \cdot$ cm) be the electrical resistivity. The vector $\mathbf{q}$ denotes a vector of dimensionless gating variables. The functions $I_{ion}(u,\mathbf{q})$ and $\mathcal{M}(u,\mathbf{q})$ are typically nonlinear, describing the membrane dynamics of the fiber. One specific model is the Hodgkin-Huxley equations [11].

We assume the electric potential $u(t,x)$ is continuous,

$$u(t,x_{b_i}^{(i_1)}) = u(t,x_{b_i}^{(i_2)}) = \cdots = u(t,x_{b_i}^{(i_{d_i})}), \tag{3}$$

and the electric current is conserved,

$$\sum_{r=1}^{d_i} J(t,x_{b_i}^{(i_r)}) = 0, \tag{4}$$

at each vertex V_i at any time $t > 0$.

Combined with some appropriate initial/boundary conditions, the differential equations above can be uniquely solved.

III. TIME INTEGRATION AND OPERATOR SPLITTING

We will adapt the method of lines to temporally integrate the differential equations (1)-(2). Let t^n be the discrete times, at which the equations will be discretized. At each timestep

from t^n to t^{n+1}, we use an operator splitting technique to advance the space-dependent part,

$$\frac{a}{2}\frac{\partial J(t,x)}{\partial x} + C_m\frac{\partial u(t,x)}{\partial t} = 0 \quad \text{for } x_0^{(j)} < x < x_1^{(j)}, \tag{5}$$

separately from the space-independent part,

$$C_m\frac{\partial u(t,x)}{\partial t} + I_{ion}(u,\mathbf{q}) = I_{stim}(t,x), \tag{6a}$$

$$\frac{d\mathbf{q}(t,x)}{dt} = \mathcal{M}(u,\mathbf{q}), \tag{6b}$$

in the differential equations (1)-(2). Note the space-dependent part (5) is simply a linear diffusion equation. The space-independent part (6) is simply a set of ordinary differential equations (ODEs).

With respect to time integration, both the linear diffusion and the nonlinear reaction parts can be in principle integrated with any standard ODE solver. In this work, we adapt implicit time integration schemes to integrate both the linear diffusion equation and the nonlinear ODEs or so-called reaction equations. An implicit scheme allows relatively larger timesteps than those imposed by the stability restriction associated with an explicit scheme.

In the next sections, we will focus on the discretization of the linear diffusion and the solution of the discrete system. For simplicity, we assume the linear diffusion equation (5) is discretized in time with the backward Euler method,

$$\frac{a}{2}\frac{\partial J(t^{n+1},x)}{\partial x} + C_m\frac{u(t^{n+1},x)-u(t^n,x)}{\Delta t} = 0 \quad \text{for } x_0^{(j)} < x < x_1^{(j)}, \tag{7}$$

with

$$J(t^{n+1},x) = -\frac{1}{R}\frac{\partial u(t^{n+1},x)}{\partial x}.$$

Here, $\Delta t = t^{n+1} - t^n$.

IV. SPACE DISCRETIZATION WITH THE FINITE ELEMENT METHOD

For conciseness, we will omit the time-dependency of the electric potential $u(t^{n+1},x)$ and the electric flux $J(t^{n+1},x)$. Let

$$u(x) \equiv u(t^{n+1},x), \qquad J(x) \equiv J(t^{n+1},x) \quad \text{and} \quad b(x) \equiv ku(t^n,x).$$

with $\kappa = 2C_m/(a\Delta t)$. Note $b(x)$ is known in the timestep from t^n to t^{n+1}. The semi-discrete equation (7) can be rewritten as

$$J'(x) + \kappa u(x) = b(x) \quad \text{for } x_0^{(j)} < x < x_1^{(j)}. \tag{8}$$

We will further discretize equation (7) with the continuous piecewise linear finite element method.

Suppose the j^{th} edge E_j in the fiber network is partitioned into a possibly non-uniform grid, denoted by $\mathcal{G}_j$. The grid $\mathcal{G}_j$ has $(m_j + 1)$ nodes, denoted by $\{x_i^{(j)}\}_{i=0}^{m_j}$.

Let $u_i^{(j)}$ be the potential variable associated with the grid node $x_i^{(j)}$. Let $\mathbf{u}^{(j)} = (u_0^{(j)}, u_1^{(j)}, \cdots, u_{m_j}^{(j)})^T$. With the continuous piecewise linear finite element method, from the semi-discrete equation (8), we can get a set of linear equations,

$$\mathbf{A}^{(j)}\mathbf{u}^{(j)} + \mathbf{J}^{(j)} = \mathbf{b}^{(j)}, \tag{9}$$

with $\mathbf{A}^{(j)} = (a_{r,s}^{(j)})_{(m_j+1)\times(m_j+1)}$ be the conductance matrix, $\mathbf{b}^{(j)} = (b_r^j)_{m_j+1}$ be the current vector, and $\mathbf{J}^{(j)}$ be the flux vector. Typically, the conductance matrix $\mathbf{A}^{(j)}$ is tridiagonal and symmetric nonnegative definite. In each row, at most the three entries right on the diagonal ($a_{r,r-1}^{(j)}$, $a_{r,r}^{(j)}$ and $a_{r,r+1}^{(j)}$) are non-zero. The flux vector $\mathbf{J}^{(j)}$ has the form

$$\mathbf{J}^{(j)} = (-J(x_0^{(j)}), 0, \cdots, 0, J(x_1^{(j)}))^T,$$

where most entries are zeros except the first and the last ones.

As the fluxes $J(x_0^{(j)})$ and $J(x_1^{(j)})$ through the endpoints of each edge E_j are unknown, the tri-diagonal system (9) involves two more unknowns than equations. So, for the global system to be uniquely solvable, we need totally $2N_E$ equations/conditions.

Fortunately, at each nonleaf-vertex V_i, which has degree $d_i > 1$, we can get $(d_i - 1)$ equations by the continuity (3) of potentials, and one more equation by the conservation (4) of electric currents. Suppose at each leaf-vertex V_i with $d_i = 1$, exactly one boundary condition is specified. Totally, we have $2N_E = \sum_{i=1}^{N_V} d_i$ additional equations. The number of equations in the final system is thus the same as that of unknowns. Normally, the system is well-determined.

V. ADAPTIVE MESH REFINEMENT

The adaptive mesh refinement (AMR) algorithm applied for this study follows Berger-Oliger's approach in time-stepping [6–10]. It uses a multilevel approach to recursively integrate the reaction-diffusion system. It first integrates the system with a large timestep on a coarse level grid. Next a locally refined fine grid is generated based on available coarse

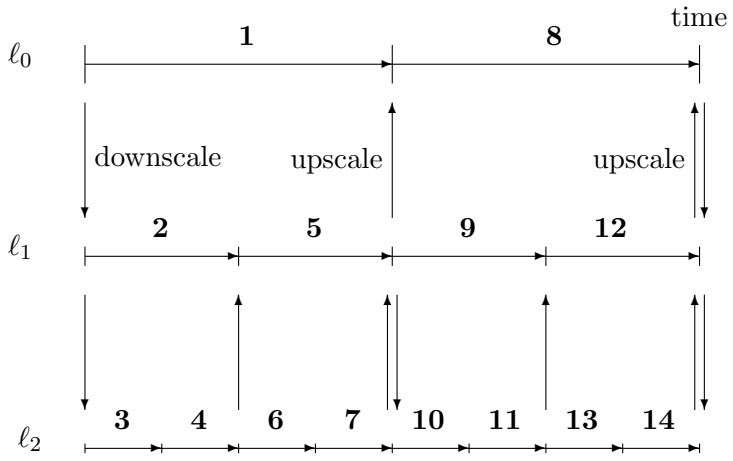


FIG. 2: The grid levels are recursively integrated. Coarse levels are integrated before fine levels. Each level has its own timestep of different size: a coarser level has a larger timestep size and a finer level has a smaller timestep. The algorithm first advances level ℓ_0 by Δt_0 (indicated by "1"), next advances level ℓ_1 by $\Delta t_1 = \Delta t_0/2$ (indicated by "2"), and then advance level ℓ_2 by two steps with $\Delta t_2 = \Delta t_1/2$ (indicated by "3" and "4"). Upon the synchronization of levels ℓ_1 and ℓ_2, data on the fine level ℓ_2 are upscaled to the coarser level ℓ_1. The recursive integration continues with steps "5", "6" and "7" etc..

data. Then the algorithm integrates the system on the fine level grid with a small timestep. Error estimation is achieved by Richardson extrapolation. By the recursive nature of the algorithm, fine grids are dynamically and locally created based on the data on coarse grids. On fine grids, smaller timesteps are used for time integration. The AMR algorithm synchronizes adjacent coarse and fine levels from time to time. At the moment of synchronization, typical routines such as mesh regridding and data up-/downscaling are performed.

The design of the AMR algorithm used for this work was based on a few assumptions proposed by Trangenstein [10]. First we assume the number of elements in the base grid, which describes the computational domain and is provided by the user, is relatively small. This will make linear systems on the grid be solved very quickly in the middle of a multilevel (composite grid) iteration, and also improve the performance of the adaptive algorithm. The base grid is fixed during the process of adaptive mesh refinement. Other grids on fine levels are in general dynamically and recursively generated by local or uniform refinement of those on coarse levels.

Second, we assume that the region covered by the grid on a fine level is contained in the

FIG. 3: Adaptively refined grids from a simulation, which use five refinement levels

interior of that on the coarser level unless both coarse and fine grids coincides with physical boundary of the computational domain. This assumption can prevent recursive searching for element neighbors. This assumption is called *proper level nesting*.

Third, we assume that if an element of a coarse grid is refined in any part of its physical space, it must be refined everywhere. As a result, the boundary of the main grid on a fine level aligns with the boundary of a subgrid of that on the previous coarser level. By a subgrid of a grid, we mean the union of a subset of its elements. This assumption is called *alignment of level grids*.

Fourth, after the data on a coarse level grid are advanced to some time, we assume that the data on its finer level are advanced by as several time steps as required by stability and accuracy to reach exactly the same time as the coarse level. This assumption implies that a coarse level is integrated before its finer levels and the time step on a coarse level is an integer multiple of that on the next finer level. It also implies that the time stepping algorithm must be applied recursively within each time step on all but the finest level. This assumption is called *synchronization of advancement*.

Fifth, by the time a fine level and its coarser level are synchronized, we assume that data on the fine level is more accurate than that on the coarse level. So, the data on a fine level must be upscaled to its coarser level before the coarse level is further advanced by another

coarse time step. This assumption is called *fine data preference*

As stated, all grids except the coarsest one in the AMR algorithm are changing dynamically. It is necessary for a coarse level to regrid its finer level from time to time. But, it will be extremely costly to tag elements and regrid levels every coarse time step. So, we would rather make regridding infrequently. At a coarse level, the number of time steps between times of regridding its finer level is called *regrid interval*. In the AMR algorithm, the regrid interval may be chosen to be an integer divisor of the refinement ratio [12], which could be any even integer number in the implementation. This assumption is called *infrequent regridding*.

There is one more assumption on the adaptive algorithm, called *finite termination*. This means that the user shall specify a maximum number of refinement levels. Once the refinement reaches the maximum level, no further mesh refinement is performed.

As determined by the nature of the Purkinje system, the grids resulting from space discretization of the computational domain are essentially unstructured. This restricts direct application of Berger-Oliger's original AMR algorithm, which requires the underlying grid be Cartesian or logically rectangular. The AMR algorithm adapted here for the Purkinje system uses unstructured grids. An unstructured grid is represented by lists of edges and nodes. A grid node may have multiple connected edges and the degree of a node could be greater than two. The unstructured grid can naturally represent the idealized Purkinje system, which primarily is a tree-like structure but has some loops inside.

The AMR algorithm can automatically recognize when (and where) the wave fronts start to leave or enter the computational domain (due to external current/voltage stimulation or self-exciting). Once the algorithm determines that there is no need to use space mesh refinement, it will turn it off and use adaptive time integration (ATI) only. The implementation of timestep size control for the adaptive time integration is standard. In each timestep, the ODEs/reactions are integrated with a full timestep once and independently integrated with a half timestep twice. Then the two solutions are compared and an estimation of relative numerical error is obtained by the standard Richardson extrapolation technique [12]. If the estimated (maximum) relative error, denoted by $\|E\|_{max}$, is greater than a maximum relative tolerance $rtol_{ATI}^{(max)}$, the timestep size was rejected and a new timestep will be estimated by

$$\Delta t^{(new)} = \gamma \cdot \left(rtol_{ATI}^{(max)} / \|E\|_{max}\right)^{1/(p+1)} \cdot \Delta t^{(old)}, \tag{10}$$

and the Richardson extrapolation process is repeated. Otherwise, the solution with two half-timesteps is simply accepted (actually an extrapolated solution could be used for better accuracy). Here, the coefficient $\gamma = 0.8$ is called a safety factor, which ensures the new timestep size be strictly smaller than the old one to avoid repeated rejection of timesteps. The constant p in the exponent of (10) is the accuracy order of the overall scheme.

If the suggested timestep size is less than a threshold timestep, which usually indicates that there is an abrupt/quick change of solution states, a local change of membrane properties or arrival of an external stimulus, the adaptive mesh refinement process is then automatically turned back on again. In the implementation, the threshold timestep is not explicitly specified by the user. It is set as the time step that is used on the base grid.

In addition, during the ATI period, if the estimated relative error $\|E\|_{max}$ is less than a minimum relative tolerance $rtol_{ATI}^{(min)}$, a slightly larger time step,

$$\Delta t^{(new)} = 1.1\, \Delta t^{(old)}, \tag{11}$$

is suggested for integration in the next step. The adaptive time integration does not like timestep size to be significantly increased within two consecutive steps for the implicit solver to have a good initial guess. For the same reason, a maximum timestep size $\Delta t^{(max)}$ is imposed during the adaptive time integration. This usually guarantees that the Newton solver used converges within a few iterations.

Finally, it is worth mentioning that in the AMR algorithm, there is a critical parameter, called AMR tolerance and denoted by tol_{AMR}, which is used by the Richardson extrapolation process for tagging coarse grid elements. The AMR tolerance closely influences efficiency and accuracy of the algorithm. The larger the tolerance is, the more efficient is the algorithm but less accurate is the solution. The smaller the tolerance is, the more accurate is the solution but less efficient is the algorithm. Due to the limitation of space, the detailed explanation of this tolerance and other components of the AMR algorithm, such as tagging and buffering of coarse grid elements, implementation of multigrid iteration on the adaptively refined grids, data up-/downscaling between coarse and fine grids, are all omitted. We refer interested readers to Ying's thesis work for more information [12].

VI. RESULTS

The AMR algorithm proposed in the work was implemented in custom codes written in C++. The simulations presented in this section were all performed in double precision on a dual Xeon 3.6 GHz computer. No data was output when the programs were run for timing studies.

In all of the numerical studies, the electrical resistivity R and the membrane capacitance C_m are fixed to be constants, $0.5\,\mathrm{k}\Omega \cdot \mathrm{cm}$ and $1\,\mu\mathrm{F/cm}^2$, respectively. The membrane dynamics is described by the Beeler-Reuter model [13]. The ODEs resulting from operator splitting are integrated with a second-order singly diagonally implicit Runge-Kutta scheme. The linear diffusion is integrated with the Crank-Nicolson scheme. For solving the partial differential equations (linear diffusion), no-flux (homogeneous Neumann-type) boundary conditions were applied at the leaf-nodes of the fiber network. Together with the second-order operator splitting technique, called Strang splitting, the resulting scheme has second-order global accuracy if the tolerances in each component of the adaptive algorithm are consistently selected [12].

In the adaptive simulations, the mesh refinement ratio is fixed to be two and the critical AMR tolerance tol_{AMR} is set as 0.02. The maximum and minimum relative tolerances, which are used in the ATI period, are chosen to be $rtol_{ATI}^{(max)} = 10^{-2}$ and $rtol_{ATI}^{(min)} = 10^{-4}$, respectively. We restrict the timestep sizes used in the adaptive time integration be not greater than one millisecond, i.e., $\Delta t^{(max)} = 1\,\mathrm{msec}$. The value of p in the exponent of (10) is equal to two as the scheme has second-order accuracy.

Example 1. *Simulations on a two-dimensional branch with thickness/radius-variable and non-uniform branch segments. Voltage stimulation is applied from right.*

The two-dimensional branch shown in Fig. 4 has a dimension of $4\,\mathrm{cm} \times 2\,\mathrm{cm}$, which is bounded by the rectangular domain $[0, 4] \times [1, 3]\,\mathrm{cm}^2$. This two-dimensional branch has variable (piece-wise constant) radius. Branch segments "6" and "7" have the smallest radius, equal to $20\,\mu\mathrm{m}$. Branch segment "1" has the largest radius, equal to $160\,\mu\mathrm{m}$ and eight times that of branch segment "6". The radius of branch segment "2" is six times that of "6". Branch segments "3" and "4" have the same radius, four times that of "6". The radius of branch segment "5" is twice that of branch segment "6".

A voltage stimulation is applied from the right end of branch segment "1" through a

FIG. 4: A typical branch in the heart conduction system is highly non-uniform in the sense that the lengths of branch segments, each of which is bounded by two adjacent leaf or branching vertices, may vary significantly. In this figure, branch segments "1" and "4" are not directly connected. Instead, they are connected through a very short branch segment. Similarly, branch segments "4" and "6" are also connected through a very short segment.

discontinuous initial action potential, which is given by

$$V_m(x,y) = \begin{cases} 25.4\,\text{mvolts} & \text{if } x > 3.5 \\ -84.6\,\text{mvolts} & \text{otherwise} \end{cases}. \tag{12}$$

The base grid used by the simulation with adaptive mesh refinement is a coarse partition of the branch structure (Fig. 4) and has 240 edges with minimum element size equal to 0.0078 cm and maximum element size equal to 0.078 cm. Note this is a highly non-uniform grid as the ratio of the maximum to the minimum element size is much greater than one. Actually, in the simulation, the base grid was generated by three times uniform bisection from a much coarser grid, which only has 30 edges. This makes the multigrid iterations for linear systems more efficient even though its contribution to the speed-up of the overall algorithm is marginal as most ($> 90\%$) of the computer time used by the simulation was spent integrating nonlinear ODEs/reactions. The adaptive simulation effectively uses three refinement levels. To apply the Richardson extrapolation technique, the second level grid was created from uniform bisection of the base grid. In fact, only the grids at the third level were dynamically changing. They are regridded every other time step.

In the adaptive simulation, the timestep sizes vary from $\Delta t = 0.0098$ msecs to $\Delta t = 1.0$ msec. The AMR process was automatically turned off at time $t = 42.85$ msecs as the algorithm determines there is no need to make spatially local refinement. After that, the ATI process is activated. The simulation on the time interval $[0, 400]$ msecs totally used 99.1 secs in computer time.

The two-dimensional branch is also partitioned into a fine, quasi-uniform grid, which has 726 edges with minimum element size equal to 0.0104 cm and maximum element size equal to 0.0127 cm. The simulation with this fine, quasi-uniform grid and time step $\Delta t = 0.00635$ msecs on the same time interval as the adaptive one used about 1406.0 secs in computer time. We call this one as "uniform" simulation.

Another simulation simply working with the coarse grid, which is used as the base grid for the AMR one, is also performed. We call this as "no-refinement" simulation. The timestep is fixed to be $\Delta t = 0.039$ msecs. This one used 76.7 secs in computer time.

In each of the simulations above, the action potential is initiated at the right end of branch segment "1" and propagates across the whole computational domain. We collected action potentials at those seven marked points (see Fig. 4) during the simulations. The action potentials from the adaptive and uniform mesh refinement simulations match very well, and their difference at the peak of the upstroke phase is uniformly bounded by 0.1 mV. The action potential from the "no-refinement" simulation has least accurate results. Potential oscillation is even observed at point "7" (see Fig. 6) in the "no-refinement" simulation.

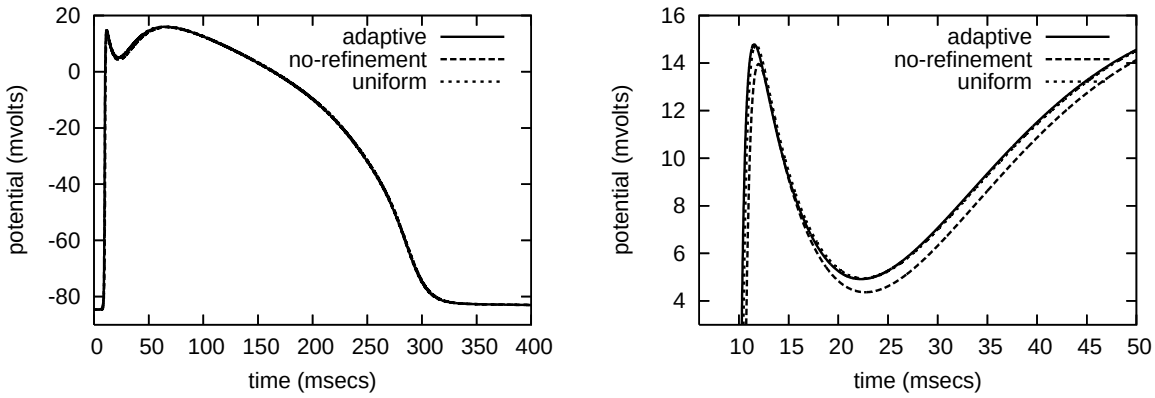
Example 2. *Simulations on a two-dimensional branch with thickness/radius-variable and non-uniform branch segments. Voltage stimulation is applied from top.*

We once again run three simulations respectively with adaptive, uniform mesh refinement and no-refinement. The computational domain and parameters are all the same as those used previously. The only difference now is to stimulate the tissue from top instead of from right.

The voltage stimulation is applied from the upper part of branch "7" through a discontinuous initial action potential, which is given by

$$V_m(x, y) = \begin{cases} 25.4\,\text{mvolts} & \text{if } y > 2.8 \\ -84.6\,\text{mvolts} & \text{otherwise} \end{cases}. \tag{13}$$

The action potential successfully propagates across the whole computational domain in



(a) (b)

FIG. 5: Traces of action potentials during the simulation period $[0, 400]$ msecs at marked point "4" in the two-dimensional branch shown in Fig. 4. The solid curves were from the adaptive simulation. The dotted curves were from the uniform simulation. The dashed curves were from the "no-refinement" simulation. In these simulations, a voltage stimulation is applied from the right of the radius-variable branch structure. The right plot is a close-up of the left plot.

each of the adaptive and uniform refinement simulations while it fails to pass across branching points where the fiber radius changes from small to large in the "no-refinement" simulation (see Fig. 7).

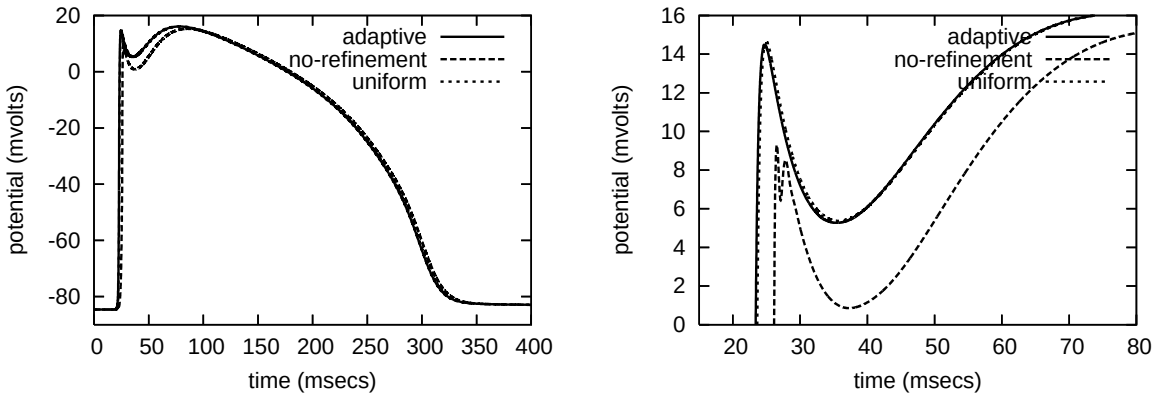
To verify accuracy of the solution from the adaptive simulation, action potentials at the seven marked points (see Fig. 4) are also collected and compared with those from the simulation with the fine and quasi-uniform grid. It is observed that the adaptive and uniform results match very well too, and their difference at the peak of the upstroke phase is bounded by 0.15 mvolts.

The simulation with adaptive mesh refinement used 100.4 msecs in computer time. The timestep sizes used in the adaptive simulation vary from $\Delta t = 0.0098$ msecs to $\Delta t = 1.0$ msec. The adaptive mesh refinement was automatically turned off at time $t = 44.02$ msecs.

The one with the uniform grid and timestep $\Delta t = 0.00635$ msecs used 1405.0 msecs in computer time.

Example 3. *Simulation on a three-dimensional Purkinje system with thickness/radius-variable and non-uniform branch segments.*

The three-dimensional Purkinje system used in the following simulations was originally from 3DSCIENCE.COM. The fiber radius of the system was modified to be piece-wise



(a) (b)

FIG. 6: Traces of action potentials during the simulation period $[0, 400]$ msecs at the point marked as "7" in the two-dimensional branch shown in Fig. 4. The solid curves were from the adaptive simulation. The dotted curves were from the uniform simulation. The dashed curves were from the "no-refinement" simulation. In these simulations, a voltage stimulation is applied from the right of the thickness/radius-variable branch structure. The right plot is a close-up of the left plot.

constant for simplicity. The radius of each branch segment takes one of the five values as the two-dimensional case. The minimum and the maximum radius of the fiber are respectively $20\,\mu$m and $160\,\mu$m. Intermediate values are 40, 80 and 120 cm. See Fig. 8 for an illustration of the topological structure, where the ratios of variable radii are not courteously shown.

The idealized Purkinje system has a dimension $2.04 \times 2.16 \times 2.98\,\text{cm}^3$. It is bounded by the rectangular domain, $[-0.39, 1.65] \times [4.05, 6.21] \times [-1.31, 1.67]\,\text{cm}^3$. The maximum of the shortest paths from the top-most vertex to other leaf-vertices, computed by Dijkstra's algorithm, is about 5.29 cm.

As in the two-dimensional examples, in each of the simulations with the idealized Purkinje system, a voltage stimulation is applied from the top branch through a discontinuous initial action potential, given by

$$V_m(x, y, z) = \begin{cases} 25.4\,\text{mvolts} & \text{if } z > 1.372 \\ -84.6\,\text{mvolts} & \text{otherwise} \end{cases}. \tag{14}$$

The base grid used by the adaptive simulation is a coarse partition of the Purkinje system and has 1412 edges/elements with minimum element size equal to 0.006 cm and maximum element size equal to 0.075 cm. This is also a highly non-uniform grid as the

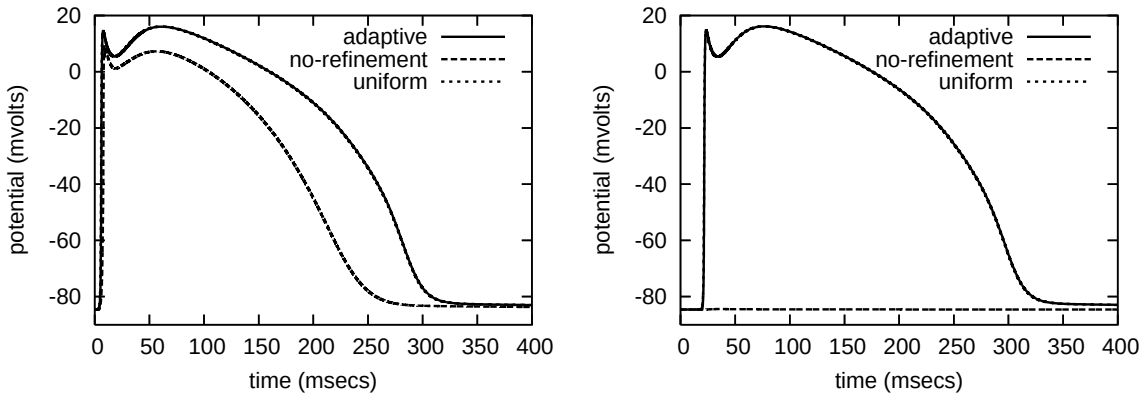


FIG. 7: Traces of action potentials during the simulation period $[0, 400]$ msecs at marked points "7" and "6" in the two-dimensional branch shown in Fig. 4. The solid curves were from the adaptive simulation. The dotted curves were from the uniform simulation. The dashed curves were from the "no-refinement" simulation. In these simulations, a voltage stimulation is applied from the top of the radius-variable branch structure. The left plot corresponds to the marked point "7" and the right plot corresponds to the marked point "6".

ratio of the maximum to the minimum element size is up to 12.5. Once again, in this simulation, the base grid was actually generated from uniform bisection of a coarser grid, which has 706 edges/elements as this will make the multigrid iterations for linear systems more efficient. The adaptive simulation effectively uses three refinement levels. Similar to the two-dimensional case, the second level grid was created from uniform bisection of the base grid and only the grids at the third level were dynamically changing, regridded every other timestep.

The timestep sizes used in the adaptive simulation vary from 0.00935 msecs to 1.0 msec. The adaptive mesh refinement was automatically turned off at time $t = 42.74$ msecs. The simulation on the time interval $[0, 400]$ msecs totally used 626.7 secs in computer time.

The three dimensional Purkinje system is also partitioned into a fine and quasi-uniform grid, which has 6311 edges/elements with minimum element size equal to 0.0095 cm and maximum element size equal to 0.016 cm. The simulation with this fine quasi-uniform grid and time step $\Delta t = 0.0082$ msecs used about 9696.0 secs in computer times.

We also run a simulation with time step $\Delta t = 0.0374$ msecs on the coarse grid that is used as the base grid for the adaptive simulation. This "no-refinement" simulation used 458.0 secs in computer time.

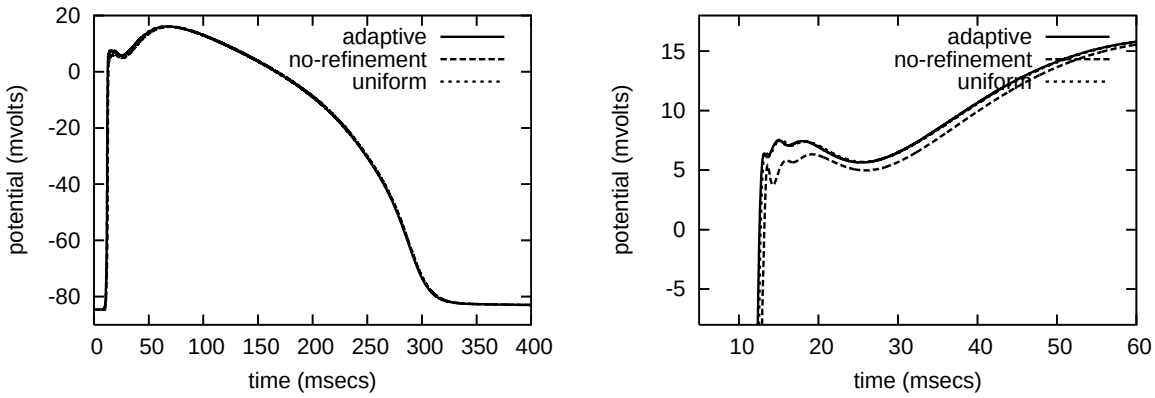
FIG. 8: The idealized Purkinje system of the heart with the thickness/radius variable.

It is observed that, in each of the adaptive, uniform refinement and no-refinement simulations, the action potential successfully propagates and goes through the whole domain. The solution from the adaptive simulation is in very good agreement with that from the uniform simulation. We collected action potentials at sixteen points located in different regions of the domain, and found that the corresponding potential traces almost overlap. Their difference at the peak of the upstroke phases is uniformly bounded by 0.4 mvolts. See Figs. 9-12 for some visualized results, which correspond to action potentials at the four marked points shown in Fig. 8. The adaptive simulation roughly uses the same computer time as the "no-refinement" one but yields much more accurate results, and gains about 15.5 times speed-up over the uniform one.

VII. DISCUSSION

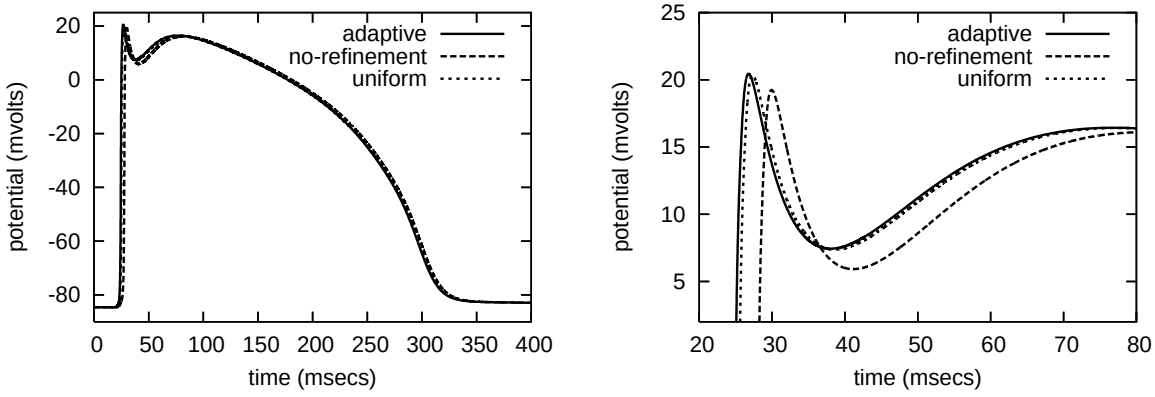
This paper demonstrates the promising application of a both space and time adaptive algorithm for simulating wave propagation on the His-Purkinje system.

In this work, only the the numerical results with membrane dynamics described by Beeler-



(a) (b)

FIG. 9: Traces of action potentials during the simulation period $[0, 400]$ msecs at the point marked as "3" in Fig. 8. The right plot is a close-up of the left one.



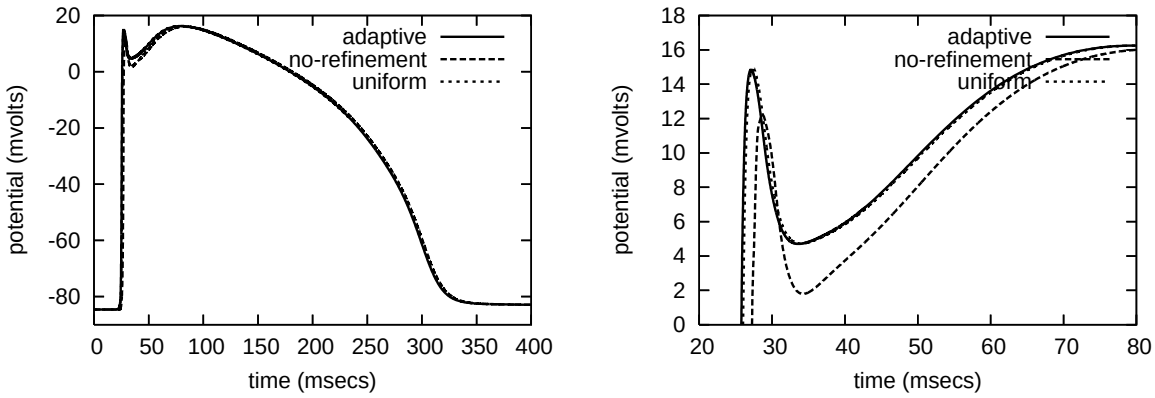
(a) (b)

FIG. 10: Traces of action potentials during the simulation period $[0, 400]$ msecs at the point marked as "20" in Fig. 8. The right plot is a close-up of the left one.

Reuter model are presented. The adaptive algorithm, however, is by no means restricted to the simple model. In fact, in our implementation, the algorithm successfully works with other physically realistic models, such as Luo-Rudy dynamic model and DiFrancescoNoble model. As the advantage of the algorithm due to application of the operator splitting technique, principally any standard cardiac model of reaction-diffusion type for modeling wave propagation can be easily incorporated into the adaptive algorithm.

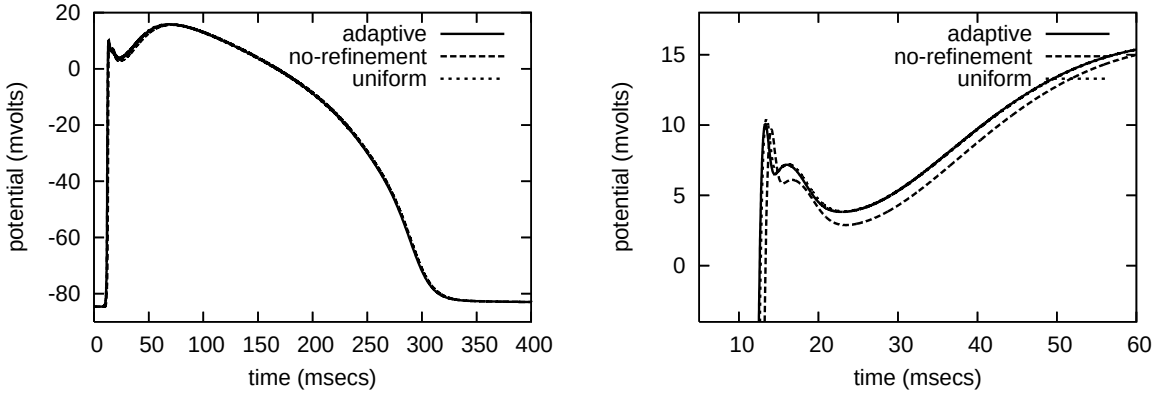
In addition, the adaptive algorithm can also be straightforwardly applied to modeling signal propagation on the neural system of the body or even the brain.

Other than its advantages, admittedly the algorithm has its own limitations. For example,



(a) (b)

FIG. 11: Traces of action potentials during the simulation period $[0, 400]$ msecs at the point marked as "76" in Fig. 8. The right plot is a close-up of the left one.



(a) (b)

FIG. 12: Traces of action potentials during the simulation period $[0, 400]$ msecs at the point marked as "109" in Fig. 8. The right plot is a close-up of the left one.

efficiency of the adaptive algorithm heavily relies on the existence of a relatively coarse base grid, which describes the computational domain. As the speed-up of an adaptive mesh refinement algorithm is roughly bounded by the ratio of the maximum degree of freedoms (DOFs) of the grid with highest resolution to the mean DOFs of all grids that ever been created during the adaptive process, the existence of a coarse base grid makes the ratio as large as possible and hence its speedup over a simulation on the grid with highest resolution. The use of a relatively coarse base grid also makes the multigrid/multilevel solver for linear systems more efficient. It is noticeable that in our simulations the base grids were all selected to have the number of edges/elements as small as possible.

FIG. 13: The membrane dynamics follows the Beeler-Reuter model.

In the AMR algorithm [10, 12] designed for uniform or quasi-uniform grids, the timestep size is typically selected to be proportional to the minimum of element sizes (multiplied by an estimated wave speed) in a grid in order that the numerical waves or discontinuities will never propagate more than one element within a single timestep. As this strategy guarantees

that the fronts of traveling waves are always bounded and tracked by fine grids, the ODEs resulting from operator splitting applied to the reaction-diffusion systems are only integrated on the fine grids while the solution in other regions covered by coarser level grids are simply obtained by time interpolation.

However, a typical branch in the heart conduction system or the whole system its own is highly non-uniform in the sense that the lengths of branch segments, each of which is bounded by two adjacent leaf or branching vertices, may vary significantly (see Fig. 4), which determines that the coarse partition of the domain is also highly non-uniform since the algorithm requires the number of edges/elements as small as possible for the efficiency considerations. On this kind of computational domain, the determination of time steps based on minimum element sizes is found to be less efficient. Here, in the proposed adaptive algorithm, timestep sizes are instead chosen to be proportional to the maximum of the element sizes in a grid. This alternative strategy is experimentally proved to be efficient and accurate enough in simulating electric waves that travel moderately fast even though its rigidity need further investigation, in particular when the fiber radius is much larger than those currently used.

In the simulations presented in this work, error estimations in both the AMR and ATI processes were all based on Richardson extrapolation, which is in general more expensive but more rigorous and reliable than other simpler ones such as the gradient detector. If the gradient detector is used, the second-coarsest grid in the AMR process does not need to be uniformly bisected and can also be changing dynamically, and the ATI period can use less computer time. The potential gain in algorithm efficiency and the possible loss in solution accuracy with the alternative use of a gradient detector will be studied and compared in a future work.

Conflict of Intersects

The authors declare that there is no conflict of interests regarding the publication of this article.

---

[1] E. J. Vigmond and C. Clements, IEEE Trans. Biomed. Engr. **54**, 389 (2007).

[2] E. J. Vigmond, R. W. dos Santos, A. J. Prassl, M. Deo, and G. Plank, Prog. Biophys. Mol. Biol. **96**, 3 (2008).

[3] S. Linge, J. Sundnes, M. Hanslien, G. T. Lines, and T. Tveito, Phil. Trans. R. Soc. A **367**, 1931 (2009).

[4] P. Pathmanathan, M. O. Bernabeu, R. Bordas, J. Cooper, A. Garny, J. M. Pitt-Francis, J. P. Whiteley, and D. J. Gavaghan, Prog. Biophys. Mol. Biol. **102**, 136 (2010).

[5] C. S. Henriquez, IEEE Transactions on Biomedical Engineering **61**, 1457 (2014).

[6] M. J. Berger and J. Oliger, Journal of Computational Physics **53**, 484 (1984).

[7] M. J. Berger and P. Colella, J. Comp. Phys. **82**, 64 (1989).

[8] E. M. Cherry, H. S. Greenside, and C. S. Henriquez, Physical Review Letters **84**, 1343 (2000).

[9] E. M. Cherry, H. S. Greenside, and C. S. Henriquez, Chaos **13**, 853 (2003).

[10] J. A. Trangenstein and C. Kim, J. Computational Physics **196**, 645 (2004).

[11] A. L. Hodgkin and A. F. Huxley, Journal of Physiology **117**, 500 (1952).

[12] W. Ying, Ph.D. thesis, Department of Mathematics, Duke University (2005).

[13] G. W. Beeler and H. Reuter, J. Physiol. **268**, 177 (1977).