

# Efficient simulation of three-dimensional anisotropic cardiac tissue using an adaptive mesh refinement method

Elizabeth M. Cherry

*Department of Physics and Astronomy and Center for Arrhythmia Research, Hofstra University, Hempstead, New York 11549*

Henry S. Greenside

*Department of Physics and Center for Nonlinear and Complex Systems, Duke University, Durham, North Carolina 27708-0305*

Craig S. Henriquez

*Department of Biomedical Engineering, Duke University, Durham, North Carolina 27708-0281*

(Received 4 February 2003; accepted 9 May 2003; published 7 August 2003)

A recently developed space–time adaptive mesh refinement algorithm (AMRA) for simulating isotropic one- and two-dimensional excitable media is generalized to simulate three-dimensional anisotropic media. The accuracy and efficiency of the algorithm is investigated for anisotropic and inhomogeneous 2D and 3D domains using the Luo–Rudy 1 (LR1) and FitzHugh–Nagumo models. For a propagating wave in a 3D slab of tissue with LR1 membrane kinetics and rotational anisotropy comparable to that found in the human heart, factors of 50 and 30 are found, respectively, for the speedup and for the savings in memory compared to an algorithm using a uniform space–time mesh at the finest resolution of the AMRA method. For anisotropic 2D and 3D media, we find no reduction in accuracy compared to a uniform space–time mesh. These results suggest that the AMRA will be able to simulate the 3D electrical dynamics of canine ventricles quantitatively for 1 s using 32 1-GHz Alpha processors in approximately 9 h. © 2003 American Institute of Physics. [DOI: 10.1063/1.1594685]

Cardiac arrhythmias present a serious health problem, resulting in about 250 000 adult deaths annually in the United States alone. Ventricular fibrillation, an often lethal arrhythmia, develops when normally coordinated waves of electrical activation break apart into multiple smaller activations, initiating only localized contractions that cannot effectively pump blood. Understanding the mechanisms responsible for the initiation and evolution of cardiac arrhythmias is a challenging medical problem. Because computer simulations can provide clean and comprehensive data acquisition, reproducible results, and systematic searches of parameter space, they are an important tool for studying arrhythmias. However, simulations of cardiac tissue are computationally intensive because of the differences in temporal and spatial scales of the dynamics. One method already shown to increase the efficiency of simulations for isotropic 2D domains uses the different scales of cardiac dynamics to vary the spatial and temporal resolution of the solution grid automatically according to the local dynamics, resulting in less demanding computations. This paper analyzes the performance of an extension of the earlier adaptive mesh refinement algorithm (AMRA) to anisotropic and inhomogeneous domains in 2D and 3D. Using the AMRA, computational speedups and memory reductions of factors of 50 and 30, respectively, are obtained for 3D propagating wave solutions using the Luo–Rudy 1 model of membrane kinetics. With extensions to support more realistic geometries and parallelization using 32 1-GHz Al-

pha processors, we anticipate that the AMRA can be used to simulate more than 2 s of complex ventricular dynamics in realistic three-dimensional domains such as canine ventricles in under 1 day.

## I. INTRODUCTION

Sudden cardiac death, mostly from ventricular fibrillation, is responsible for 250 000 deaths in the U.S. each year.<sup>1</sup> Despite decades of research, the mechanisms responsible for inducing and sustaining fibrillation are not well understood. Experiments are difficult to conduct and data cannot be obtained easily. In the best case, electrical potential data can be gathered directly from arrays of points on the epicardial and endocardial surfaces and possibly from a few points in the midmyocardium adjacent to the edge of a tissue slab. Optical mapping using voltage-sensitive dyes can provide data at higher spatial resolutions than electrodes, but the voltages must be inferred from optical signals that reflect the dynamics of cells through a certain depth and still usually are obtained only from the epicardial surface, with dual epicardial–endocardial data being gathered only in a small number of experiments in extracted tissue.<sup>2</sup> With data limited primarily to two dimensions, experiments so far have not elucidated the mechanisms responsible for the three-dimensional phenomenon<sup>3,4</sup> of ventricular fibrillation. The roles of properties such as the heart's size,<sup>5</sup> geometry,<sup>6–8</sup> electrical state, anisotropic fiber structure,<sup>9</sup> and inhomogeneities<sup>10,11</sup> have yet to be established.

Because of the experimental difficulties involved in studying the three-dimensional dynamics of the heart and the mathematical complexities of even the simplest cardiac models, simulations play a crucial role both in testing hypotheses and in identifying and understanding experimentally observed dynamics. However, quantitatively accurate simulations of an entire three-dimensional human heart are not yet feasible. The essential difficulty is that human heart muscle is a strongly excitable medium whose electrical dynamics involve narrow, rapidly varying fronts. The width of a front is about two orders of magnitude smaller than the long axis of a human-size right ventricle. Similarly, the fastest time scale in heart dynamics, associated with the rapid depolarization of the cell membrane, is about four orders of magnitude smaller than the time required to observe the establishment of a fibrillating state. As described elsewhere,<sup>12</sup> a simulation of a whole heart for several seconds using a quantitatively accurate model and adequate spatial and temporal resolution may require about  $10^{10}$  variables associated with the spatial mesh to be evolved over about  $10^6$  time steps. A uniform mesh calculation of such magnitude exceeds currently existing computational resources and has not yet been carried out.

In this paper, we discuss new features of an adaptive mesh refinement algorithm (AMRA) introduced in a recent previous paper.<sup>12</sup> There, we showed that the AMRA reduced by a factor of 5 the computation time and memory requirements of a two-dimensional calculation in a many-spiral state using the Luo–Rudy 1 (LR1) model.<sup>13</sup> Here, we examine additional applications not included in the previous study. Specifically, we provide examples in two and three spatial dimensions and demonstrate that the AMRA is sufficiently general to include inhomogeneities and anisotropy easily and to treat them efficiently and accurately.

Previous efforts to improve the efficiency of cardiac simulations have followed three main approaches. The first is to use reduced mathematical models that reproduce some of the behavior observed in more complex models but with only a few coupled fields. One widely studied example is the two-variable FitzHugh–Nagumo model,<sup>14</sup> which describes behavior of a general excitable medium and which can be modified to approximate some types of cardiac dynamics.<sup>8,15</sup> A second example is a three-variable model developed by Fenton and Karma<sup>9</sup> and designed to reproduce the restitution curves of more complex cardiac models. While these and other simplified models can reproduce many known features of cardiac electrical propagation, there may be some differences, such as electrotonic effects.<sup>16,17</sup> In addition, modeling drug effects in reduced models with fewer ionic currents is not as straightforward. For these reasons, efficient algorithms for more quantitatively-based models are still desirable.

A second approach is to retain more complex models but to solve them more efficiently by improved numerical algorithms, such as time integration schemes that are fully or semi-implicit.<sup>9,18–21</sup> Implicit methods avoid a bound on the time step determined by the spatial resolution that explicit schemes incur, and for this reason they permit larger time steps at higher spatial resolutions than explicit methods. Even using a semi-implicit method based on operator

splitting,<sup>22</sup> which treats only linear terms involving high-order spatial derivatives implicitly while continuing to treat other terms explicitly, can result in stability of time steps one to two orders of magnitude larger than those required for stability of a comparable explicit scheme. However, using larger time steps may lead to a decrease in solution accuracy<sup>23,24</sup> and consequently to substantial errors in computed physical quantities. For instance, calculating conduction velocity in the LR1 model at the forward Euler (FE) stability limit gives less than 5% error compared to the velocity obtained by extrapolating linearly to a zero-size time step, while the error increases to 20% and to 50% when compared with a semi-implicit method using time steps 10 and 50 times larger than the FE stability limit, respectively. In addition, the cost of integrating a system of equations for one time step often is substantially higher for an implicit integration scheme compared to explicit schemes, so that implicit schemes also may be computationally inefficient if the overall number of time steps required is not reduced sufficiently.

The spatiotemporal structure of wave dynamics in excitable media suggests a third approach: adjusting the resolution of the calculation locally, dynamically, and automatically. Cardiac electrical dynamics are sparse in that the sharp fronts occupy only a small fraction of the tissue volume. By varying the spatiotemporal resolution to concentrate computational effort primarily along the areas with large spatial and temporal gradients, it is possible to reduce greatly the computational effort and memory required. Thus the algorithm will have complexity proportional to the area or volume of the wave fronts, rather than to the area (or volume in the 3D case) of the domain. For highly excitable media like cardiac tissue, this area or volume fraction generally is small, but it can be much larger for fibrillating states with many spiral or scroll waves.

Until recently, adaptivity for excitable media was implemented by varying either the spatial or temporal resolution, but not both, locally and dynamically. Previous studies<sup>19,20,25–27</sup> have used time-adaptive strategies to vary the size of the time steps used in various regions of the domain depending on the local dynamics, but for a spatially uniform mesh. This technique allowed those areas in the domain without any propagating fronts to use larger time steps, while the spatial resolution remained constant over the entire domain. Moore<sup>21</sup> used a different adaptive approach, in which the spatial resolution was varied, but the time steps did not vary locally with the spatial resolution. However, this method kept the global time step small as long as a front was present in the domain and generally could not yield computational savings as significant as the time-adaptive strategy. Both of these adaptive approaches have the advantages that they are largely model- and method-independent. We have recently shown that combining spatial and temporal adaptivity can yield further savings.<sup>12</sup> Our space–time adaptive algorithm was based on a method developed by Berger and co-workers<sup>28–30</sup> and used successfully for two- and three-dimensional simulations of hyperbolic systems of differential equations, such as the Euler equations of fluid dynamics. In fluid dynamics applications, regions of fine space–time reso-

lution automatically follow traveling shocks, while the rest of the domain uses a coarser spatial mesh and larger time steps. In cardiac applications, higher resolution areas align with propagating wavefronts of electrical activation.

The rest of this paper is organized as follows. Section II contains an overview of the computational models we use for our results, including both the quantitatively-based LR1 model for cardiac tissue<sup>13</sup> and the FitzHugh–Nagumo model for general excitable media.<sup>14</sup> Section III provides a brief review of the AMRA. In Sec. IV, we show how our algorithm correctly and accurately encloses sharp fronts for the Luo–Rudy 1 model with areas of fine spatial and temporal resolution, even in the presence of anisotropy and inhomogeneities. We also present results for 2D and 3D FitzHugh–Nagumo simulations with spiral and scroll waves and establish that there is no significant reduction in accuracy when using the AMRA compared to an algorithm that uses a spatially uniform mesh and constant time step at the finest space–time resolution of the AMRA. Furthermore, we analyze the AMRA’s efficiency and show that for a propagating wave in 3D using the LR1 model, the AMRA provides substantial savings of a factor of 50 in runtime and a factor of 30 in memory. Finally, in Sec. V, we use our results to estimate that a quantitatively accurate AMRA simulation of fibrillation in realistic cardiac anatomies for several seconds with an effective 125  $\mu\text{m}$  resolution should already be practical with existing computers.

## II. MODELS

We used two different models to investigate the accuracy and efficiency of the AMRA. Most of our results were obtained for the physiologically-based LR1 model,<sup>13</sup> which can be written in the form

$$C_m \partial_t V(t, \mathbf{x}) = \frac{1}{\beta} (\nabla \cdot D(\mathbf{x}) \nabla V) - I_{\text{ion}}(V, \mathbf{m}, [\text{Ca}^{2+}]_i) - I_{\text{stim}}(t, \mathbf{x}), \quad (1)$$

$$\frac{d\mathbf{m}}{dt} = \mathbf{f}(V, \mathbf{m}), \quad (2)$$

$$\frac{d[\text{Ca}^{2+}]_i}{dt} = \mathbf{g}(V, \mathbf{m}, [\text{Ca}^{2+}]_i), \quad (3)$$

where  $V(t, \mathbf{x})$  is the membrane potential at time  $t$  and at position  $\mathbf{x}=(x, y)$  in 2D or  $\mathbf{x}=(x, y, z)$  in 3D,  $C_m$  is the membrane capacitance per unit area,  $\beta$  is the surface-to-volume ratio of a heart cell,  $D(\mathbf{x})$  is the conductivity tensor,  $I_{\text{ion}}$  is the total ionic current flowing across the membrane, and  $I_{\text{stim}}$  is a specified current injected to initiate a propagating wave. The six voltage-sensitive membrane variables  $m_i(t, \mathbf{x})$  represent gating variables which, along with the intracellular calcium concentration  $[\text{Ca}^{2+}]_i$ , determine the flow of ions across the membrane. The tissue anisotropy introduced by cardiac muscle fibers, which conduct current quickly along the fiber axis and slowly in other directions, is included through the conductivity tensor  $D$ . If all fiber axes are aligned parallel to one of the coordinate axes,  $D$  becomes a diagonal matrix whose entries are the conductivities along

and across the fibers. In our 3D simulations, we follow Ref. 9 and choose  $D$  to allow the fiber orientation to rotate smoothly with tissue depth from the epicardium to the endocardium, with the fiber axis always located in a plane parallel to the epicardium.

For all calculations reported below, the Neumann boundary conditions  $(\hat{n} \cdot \nabla)V=0$  were used, where  $\hat{n}$  is the unit vector normal to a given boundary point. Except where noted below, the same membrane parameter values as those of Ref. 13 were used except for the maximum calcium conductance  $\bar{G}_{\text{Ca}}$  in the  $I_{\text{ion}}$  term, whose value was changed from 0.09 to 0.045 mS/ $\mu\text{F}$ . The medium in 2D was anisotropic with the conductivities along and across fibers,  $D_{\text{parallel}}$  and  $D_{\text{perp}}$ , respectively, set to 2 mS/cm and 0.2 mS/cm, giving conduction velocities of approximately 60 cm/s and 20 cm/s, respectively, and a 10:1 anisotropy ratio within the range of values observed for cardiac tissue. The surface-to-volume ratio  $\beta$  was set to 2000  $\text{cm}^{-1}$ . In 3D, the medium included rotational anisotropy, with the conductivity along fibers,  $D_{\text{parallel}}$ , set to 2 mS/cm, and the transverse and transmural conductivities,  $D_{\text{perp}}$  and  $D_{\text{trans}}$ , respectively, both set to 0.2 mS/cm. The fiber orientation changed by 120° from the epicardium to the endocardium.

For some simulations, the FitzHugh–Nagumo (FHN) model<sup>31</sup> was used in the following form:

$$\partial_t u(t, \mathbf{x}) = \frac{1}{\epsilon} (u - u^3 - v) + D(\mathbf{x}) \nabla^2 u, \quad (4)$$

$$\partial_t v(t, \mathbf{x}) = \epsilon(u + \beta - \gamma v).$$

Here  $u$  is the excitatory variable and  $v$  is the inhibitory variable at time  $t$  and position  $\mathbf{x}$ . As in the LR1 model,  $D(\mathbf{x})$  is the conductivity tensor, with conductivities along and across fibers set to 1 and 0.2, respectively, a lower anisotropy ratio than we use for the LR1 but also within the range observed in cardiac tissue. The parameter  $\epsilon$  represents the ratio of recovery rate to excitation rate, the parameter  $\beta$  acts as a threshold, and the parameter  $\gamma$  is the slope of the  $v$ -nullcline,<sup>31</sup> which determines whether the system has one stable equilibrium or one unstable and two stable equilibria. As with the LR1 model, Neumann boundary conditions were used. While this FHN model is not based on experimentally measured quantities, it can exhibit phenomena similar to what is found in more quantitatively-based models like LR1, such as rotating spiral waves in 2D and scroll waves in 3D.

We used the dimensionless FHN model to analyze trajectories of spiral wave tips and scroll wave filaments because the dynamics of spiral tips in this model have been well documented.<sup>31</sup> A similar understanding of spiral tip dynamics for the LR1 model has not yet been realized due to the model’s much larger parameter space and often unstable spiral wave behavior. Model parameters in Eq. (4) are set as follows:  $\epsilon=0.1$ ,  $\beta=0.6$ , and  $\gamma=0.5$ . In 2D and 3D, the conductivities along and across fibers are set to 1.0 and 0.2, respectively. In addition to aiding our accuracy analysis, use of the FHN model also demonstrates that our AMRA is not restricted to only one model.

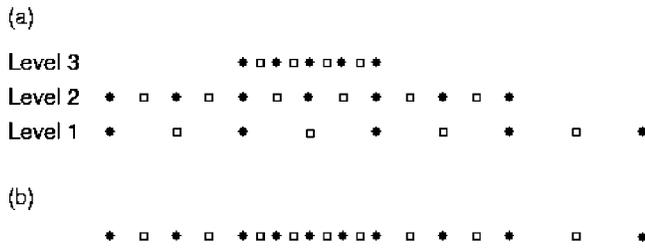


FIG. 1. One-dimensional illustration of the AMRA data structure. Grid points are shown as black circles and data points are shown as open squares. (a) Locations of grid points and data points on three levels of spatial resolution. Level 1 uses the coarsest resolution and level 3 uses the finest. Note that data points from different levels never coincide. (b) Effective grid created by the union of the finest data available throughout the domain shown in (a). Both grid points and data points are distributed nonuniformly.

### III. SOME DETAILS OF THE ALGORITHM

Our algorithm is a straightforward extension of an AMRA that has been used by other researchers to integrate *hyperbolic* systems of conservation laws such as the Euler equations of fluid dynamics.<sup>28–30</sup> Since the algorithm was developed in 1982, it has been successfully used for many applications, including combustion,<sup>32</sup> porous media,<sup>33</sup> compressible<sup>34</sup> and incompressible magnetohydrodynamic flows,<sup>35</sup> incompressible Navier–Stokes fluid flow,<sup>36</sup> biofluid dynamics,<sup>37</sup> cosmology,<sup>38,39</sup> and excitable media.<sup>12</sup> Since key mathematical and algorithmic details are available elsewhere,<sup>12,24,28–30</sup> only some essential ingredients and our modifications to the original formulation are described here. In this section, we first give an overview of the adaptive mesh refinement strategy. Then, the time integration method is discussed. Finally, we describe briefly the use of a lookup table that further improves performance for the LR1 model.

#### A. Overview of the AMRA

The algorithm is founded on the use of Cartesian or logically box-shaped grids. Cartesian grids have the advantage of being substantially simpler to program than unstructured grids. Traditionally, unstructured grids have been able to accommodate boundaries with complex shapes more easily than Cartesian grids, although recent research has expanded the use of irregular boundaries within Cartesian grid structures.<sup>40</sup> Using Cartesian grids allows the use of integration methods for rectangular grids whose convergence properties are well understood.<sup>29</sup> The corresponding code can be parallelized more easily and efficiently by avoiding the costs associated with indirect memory referencing on unstructured grids.<sup>41</sup>

The AMRA approximates a given continuous field such as the cardiac membrane potential  $V(t, \mathbf{x})$  on a set of nested locally-uniform patches of  $d$ -dimensional Cartesian meshes in a  $d$ -dimensional Cartesian box. Each grid patch is defined as a separate data structure, independent of other patches. Field values are represented as cell-centered quantities, as shown in Fig. 1(a) for a one-dimensional example. Grid points align from one level to the next, but because the grid spacing varies and the fields are cell-centered, the data points on different levels never align. The locations of grid points (filled circles) and data points (open squares) are shown in

Fig. 1(a) for three levels of spatial resolution. When combined, these three grid levels give an effectively nonuniform grid, as shown in Fig. 1(b). On each patch, spatial derivatives in the dynamical equations at a given point are approximated by second-order-accurate cell-centered finite differences using neighboring points.

Associated with each level  $l$  are both a spatial resolution  $\Delta x_l$  and a time step  $\Delta t_l$ . While multiple grid patches may exist at a given level of spatial resolution, the same time step is used for all of them to facilitate data synchronization. The spatial resolution  $\Delta x_l$  is an integer multiple  $r_s$  (often two) of the resolution  $\Delta x_{l+1}$  used for the next finer level, where level 1 has the coarsest resolution. Similarly, the time step  $\Delta t_l$  is an integer multiple  $r_t$  of the time step  $\Delta t_{l+1}$ . For parabolic systems of equations such as models of cardiac electrical activity, the stability restrictions for explicit schemes are of the form  $\Delta t < c \Delta x^2$ , and so  $r_t$  generally should equal  $r_s^2$ . The ratio of  $\Delta t_l$  to  $\Delta x_l^2$  for all levels is fixed, which allows the same explicit difference scheme to be stable on all grid patches and also allows larger time steps to be used in parts of the domain using coarser spatial resolution. In some cases it may be desirable to eliminate an intermediate level by setting the spatial resolution of one level equal to the resolution of the next finest level multiplied by  $r_s^2$  instead of  $r_s$ , with the time step similarly adjusted to keep  $\Delta t_l / \Delta x_l^2$  constant on all levels of resolution.

The AMRA assumes that some explicit finite difference scheme (specified by the user) has been chosen to approximate both space and time derivatives. Each grid patch is defined separately and maintains its own solution vector, so that grid patches can be integrated independently of other patches, except for the determination of boundary data (discussed below). Integration of grids of different spatial resolutions proceeds from coarse to fine levels to ensure that internal boundary data for fine grids always can be interpolated from data already computed on coarser grids. Because the temporal resolution varies with the spatial resolution, more time steps are needed for finer patches, which leads to an asynchronous updating of data. Steps on different levels are interleaved, so that before advancing a level  $k$  grid patch to time  $t + \Delta t_k$ , all coarser grid patches (level  $l$  grid patches with  $l < k$ ) have been integrated to time  $t$ . Data on all grid levels are synchronized only after one full time step on the coarsest grid level is completed. The same integration scheme is applied at every level. Because the integration method is contained in only one subroutine that is called whenever needed, the AMRA allows the flexibility of implementing different numerical methods and models as easily as a uniform mesh code.

Communication among grids occurs at two points in the AMRA integration algorithm. When a level  $l$  cell is overlaid by a finer grid level, the cell's value at time  $t$  is replaced by the average of the fine grid values at level  $l+1$  that comprise the coarse cell once they are computed (four cells in 2D, or eight cells in 3D). This process ensures that the most accurate data available are used on all grid levels. The other procedure that requires communication among grid patches is the determination of internal (nonphysical) boundary values for fine grid patches. Nonphysical boundary

values are needed to provide the data necessary to use the finite difference stencil for all grid patches that have at least one edge not touching the physical boundary. These boundary data are either provided directly from neighboring grid patches at the same level (if available) or interpolated from neighboring grid patches at the next coarsest level. Integrating from coarse to fine levels ensures that internal boundary data for fine grids always can be interpolated from data already computed on coarser grids. Once the boundary data have been obtained, their values are stored with the finer grids to permit each grid patch to be integrated independently of the other patches and to allow for greater parallelization potential for future applications.

The power of the AMRA arises from its ability to refine or to coarsen the spatial representations of fields automatically and efficiently by varying the number of grid points locally. Grid patches with higher resolution in space and time are created when an estimate of the local truncation error on a coarser mesh exceeds a specified tolerance and are deleted when no longer required. The errors are estimated on each level every  $k_r$  time steps for some integer  $k_r$ . Errors are estimated more often on finer grids because their time steps are smaller. However, errors are never estimated for grids at the finest level available, since the error in that region cannot be reduced by using a finer grid. We chose to regrid every two time steps, as we found that this value balanced the competing desires to minimize the computational expense of regridding and to keep grid patches closely aligned to wave fronts in order to avoid unnecessary refinement.

Error estimates of a field such as  $V(t,x,y)$  are calculated using Richardson extrapolation.<sup>42</sup> On a given level of resolution, a time step is taken starting from the current time and the solution obtained is coarsened by a factor of 2. This coarsened solution then is compared with the solution obtained from a grid twice as coarse using a coarsened version of the previous solution value at the given level of resolution and a time step twice as large. When the difference in solutions exceeds a specified absolute tolerance for some region, a fine grid patch is added. The algorithm includes flexibility to use refinement criteria other than Richardson extrapolation, such as refining only where spatial gradients exceed some specified tolerance or where some physiological process known to invoke a fast time scale (such as the opening of the sodium channel that excites the cell) occurs.

Once a subset of cells on a given level has been identified as meeting the refinement criterion, the selected cells are formed into box-shaped grid patches at the next finest level. A buffer zone consisting of cells whose error estimates do not require them to be refined, but which are near a refined region, is added to prevent regions of high error from propagating outside the fine grids before regridding next takes place. The size of the buffer zone depends on the regridding frequency  $k_r$ .

Finally, we emphasize that this AMRA scheme can be parallelized. Because grid patches are integrated independently, patches can be updated on separate processors with only limited communication needs to provide boundary conditions for fine grids. Berger showed for a 2D fluid calculation that running the code on four processors resulted in a

speedup of 3.4 compared to a serial code,<sup>43</sup> with an efficiency of about 85%, while Allard and Trangenstein<sup>44</sup> found for an algorithm based on that of Berger and collaborators applied to a linear elasticity problem that a 32-processor distributed memory machine (the Cray T3E) yielded a factor of 28 speedup. We expect that our AMRA code can achieve a similar speedup for cardiac applications.

## B. Time integration method

The AMRA can use a variety of explicit time integration methods.<sup>29</sup> Consideration of the dynamics exhibited by the LR1 model suggests that explicit methods may be acceptable for strongly excitable media like cardiac tissue because small time steps are required to resolve the temporal gradients. The fastest time scale associated with the LR1 model is the opening of the sodium channels, which requires approximately 0.3 ms to increase the fraction of open sodium channels from 10% to 90%. A reasonable temporal resolution would require about ten data points to resolve this change and therefore would need a time step of about 0.03 ms. Some researchers<sup>19</sup> recommend ensuring that the change in potential  $\Delta V$  from one time step to the next is no greater than 1 mV, which implies that over a change from  $-80$  mV to 20 mV, at least 100 time steps should be taken. Since the 100 mV change takes place in roughly 1 ms,  $\Delta t$  can be no larger than 0.01 ms to keep  $\Delta V < 1$  mV during the upstroke. Time steps on the order of 0.01–0.03 ms are stable for the forward Euler (FE) scheme at the spatial resolutions needed to use the LR1 model without numerical artifacts causing propagation to cease (100–200  $\mu\text{m}$ ), suggesting that methods that allow larger time steps are not necessary during the upstroke.

Using FE to solve Eqs. (1)–(3) showed that for values of  $\Delta x$  above 25  $\mu\text{m}$ , the size of the maximum stable time step no longer varied with  $\Delta x$  but remained constant. The source of this constant restriction on the time step size was not the diffusion portion of Eq. (1), but rather came from the coupled ordinary differential equations. As noted in Ref. 45, Eq. (2) can be solved analytically at each time step if  $V$  is treated as a constant potential. We solved Eq. (2) using a semi-implicit backward Euler (BE) method, which gave results indistinguishable from those obtained using the analytical scheme. Equation (3) cannot be solved analytically for a fixed potential and instead was solved by using the old values of the potential  $V$  and the gating variables  $\mathbf{m}$  to calculate values used to update  $[\text{Ca}^{2+}]_i$ , then using BE to integrate in time:  $[\text{Ca}^{2+}]_i^{n+1} = [\text{Ca}^{2+}]_i^n + \Delta t g(V^n, \mathbf{m}^n, [\text{Ca}^{2+}]_i^{n+1})$ , where  $g(V, \mathbf{m}, [\text{Ca}^{2+}]_i)$  is given in Ref. 13. When Eqs. (2) and (3) were integrated using a semi-implicit BE method with the current value of the potential  $V$ , and when the diffusion term was solved using the explicit FE method, the maximum stable time step was 1–2 orders of magnitude larger than what a FE scheme by itself would allow for spatial resolutions of 100–500  $\mu\text{m}$  and resumed the expected dependence on spatial resolution. When incorporated into the AMRA, the integration schemes described yielded greater computational savings at coarser resolutions than the purely explicit FE method while retaining desired accuracy at fine levels.

Finally, we note that numerical stability issues applied to the AMRA as expected for parabolic partial differential equations. For each level of resolution  $l$ , the stability criterion for an explicit method  $\Delta t_l < c \Delta x_l^2$  must be met for a level-independent constant  $c$ . When compared to a uniform mesh, stability constraints were neither appreciably tightened nor appreciably relaxed. We conclude that the AMRA imposed no new stability constraints for explicit methods.

### C. Use of a lookup table

An additional reduction in computational effort by a factor of four to five was obtained by using a lookup table<sup>46</sup> to avoid repeated evaluations of exponentials when integrating the gating variables. Specifically, quantities that depended only on the potential  $V$  and that involved at least one exponentiation were calculated and stored over a range of potentials, from  $-120$  mV to  $70$  mV. When these values were needed, linear interpolation from the two nearest table entries was used, in  $0.05$  mV increments. In simulating a full action potential at spatial resolution  $\Delta x = 125 \mu\text{m}$  with  $\Delta t = 0.002$  ms, use of the lookup table produced a relative error of  $<1\%$  compared to the solution found with no lookup table. Along with a comparison of the action potential shapes, the conduction velocities and action potential durations (APDs) found with the table resolution set to  $0.1$  mV,  $0.05$  mV, and  $0.01$  mV were compared with the same quantities obtained without using the lookup table. Since the velocities and APDs agreed in all cases to two decimal places, we concluded that the table resolution was not a significant source of inaccuracy.

## IV. RESULTS

This section summarizes results obtained by applying the AMRA to the LR1 and the FHN models. Because details showing the accuracy and efficiency of the AMRA in 1D and 2D homogeneous isotropic domains were presented in Ref. 12, we do not reproduce them here. Instead, we focus on domains with anisotropy or inhomogeneities. We demonstrate how the AMRA can resolve 2D anisotropic domains with waves initiated by a point stimulus and analyze what spatial resolution is needed to ensure comparable wave speeds and shapes when fibers are aligned with the computational grid or at nonzero angles relative to it. Next, we show how the AMRA effectively responds to the presence of an inhomogeneous region in the tissue. Then we analyze the accuracy of the AMRA in 2D and 3D anisotropic domains by tracking spiral tips and filaments for the FHN model, whose dynamics are known in various parameter regimes. Finally, we discuss the AMRA's efficiency for a single propagating pulse in 3D LR1 domains with rotational anisotropy and extrapolate the performance to larger domains with more complex dynamics.

### A. Anisotropy in a 2D medium

The human heart is a strongly anisotropic medium, with conductivities in the longitudinal and transverse directions differing by roughly a factor of 4–10. These different conductivities lead to differences in both the wave speed and the

front width depending on the local direction of propagation. In strongly anisotropic domains, achieving an adequate spatial resolution is important to prevent distortion of the wavefronts during propagation. Such distortions can alter the ratio of longitudinal to transverse conduction velocity and change the shapes of the trajectories of individual points in the  $V - \partial V / \partial t$  phase space.

Figure 2 shows some of the effects that can result when  $\Delta x$  and  $\Delta t$  are not fully resolved. In this example, a  $2 \text{ cm} \times 2 \text{ cm}$  domain was stimulated in a small square region in the center of the tissue to initiate propagating waves. Parameters for the LR1 model are as given in Sec. II and the anisotropy ratio is 10:1. Panels (a) and (b) illustrate the shape of the propagating wave after 15 ms using different resolutions. Both use  $125 \mu\text{m}$  and  $0.02$  ms as the coarsest spatial and temporal resolutions, but panel (b) uses the AMRA to add two additional levels of mesh as needed, down to finest resolutions of  $31.25 \mu\text{m}$  and  $0.00125$  ms (factors of 4 and 16 smaller, respectively). As seen in the figure, not only are the conduction velocities along and across fibers faster at higher resolutions, leading to a larger elliptical front, but the ratio of longitudinal to transverse velocity changes as well, altering the overall shape of the front. When the coarser resolutions are used, the front is longer relative to its width than when the finer resolution is used. Figure 2(c) illustrates the AMRA's grid hierarchy corresponding to panel (b) [panel (a) uses a uniform mesh], with intermediate-resolution grid patches shown in yellow and overlaid by high-resolution patches shown in green. It should be noted that the process by which grids are formed does not preserve symmetries, so the grid structure can develop asymmetries even when the underlying dynamics is symmetric.

Figures 2(d) and 2(e) show the time rate of change in the potential  $\partial V / \partial t$  as a function of the potential  $V$  for two locations in space, one to the left of the stimulus site and affected mostly by longitudinal propagation (circles), and the other below the stimulation site and affected mostly by transverse propagation (squares). At adequate resolution, both plots should be qualitatively similar. At the coarser resolution, shown in panel (d), the trace showing transverse propagation has a pronounced hump during the early part of the upstroke and an overshoot at the end of the upstroke compared to the longitudinal trace. At finer resolution, these effects disappear and better agreement is obtained, as shown in Fig. 2(e).

One way to reduce the numerical artifacts associated with coarser resolutions on Cartesian grids in the presence of a strong anisotropy would be to set the ratio  $\Delta x / \Delta y$  not to 1, but instead to the ratio of longitudinal to transverse conduction velocity. This could be done using the AMRA as easily as with a uniform grid. However, this strategy would be useful only for 2D or 3D domains with fibers oriented along the  $x$ - or  $y$ -axes.

When fibers in strongly anisotropic media are oriented at a nonzero angle relative to the  $x$ -axis, discretization errors distort the propagating waves on Cartesian grids unless sufficiently high spatial resolution is used. This effect is similar to what has been shown for fibers aligned to the computational grid, except that the distortions depend on the fiber

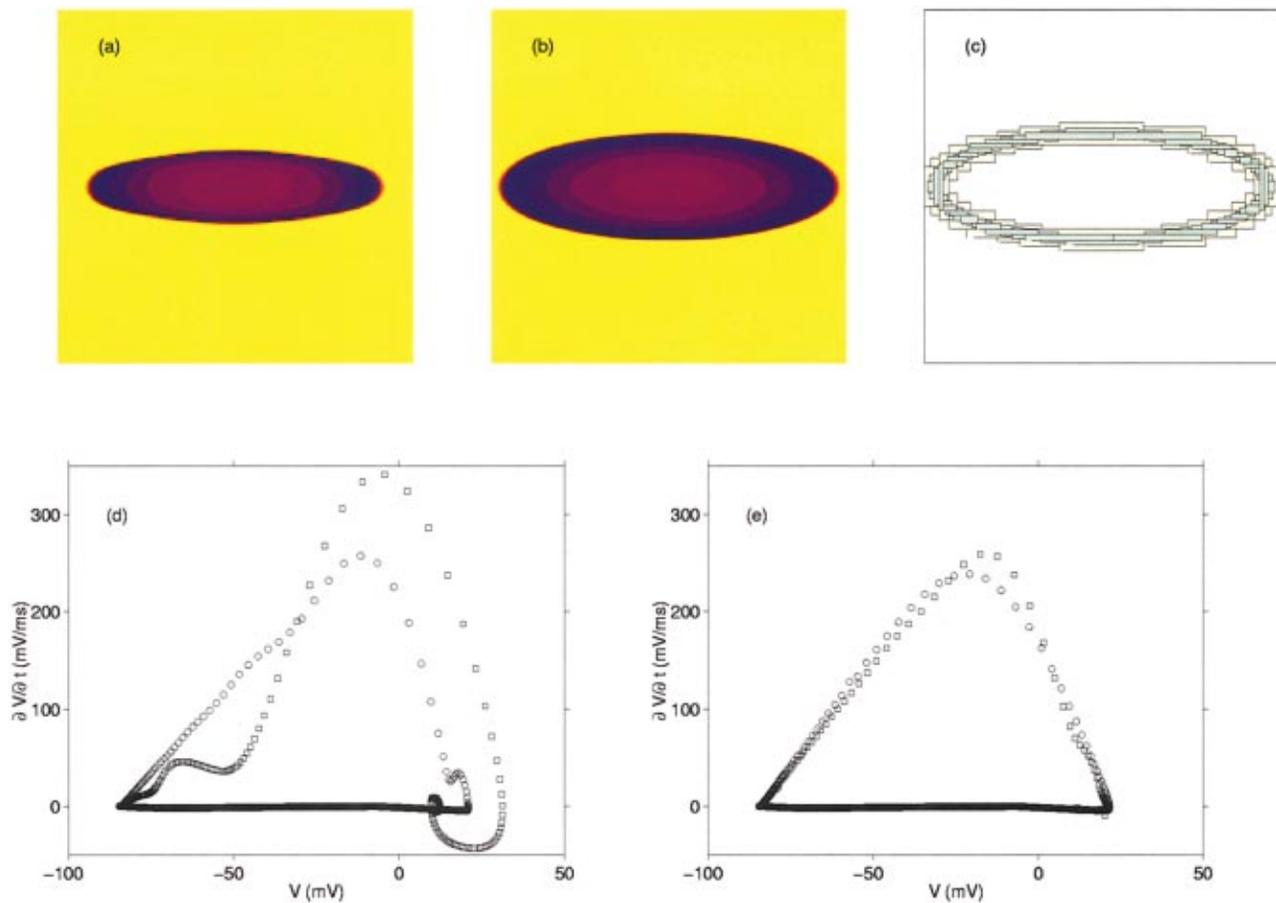


FIG. 2. (Color) Anisotropic propagation using the LR1 model in a  $2\text{ cm} \times 2\text{ cm}$  domain, with conductivities  $D_{\text{parallel}}=2\text{ mS/cm}$  and  $D_{\text{perp}}=0.2\text{ mS/cm}$ . (a) Voltage profile after 15 ms using a uniform mesh with  $\Delta x=125\ \mu\text{m}$  and  $\Delta t=0.02\text{ ms}$ . Values for the voltage  $V$  are color coded with dark blue for  $V \geq -5\text{ mV}$ , red for  $-5 \leq V \leq -65\text{ mV}$ , and yellow for  $V \leq -65\text{ mV}$ . (b) Voltage profile after 15 ms using three levels of mesh with the same coarse  $\Delta x$  and  $\Delta t$  as in (a) and with fine  $\Delta x=31.25\ \mu\text{m}$  and  $\Delta t=0.00125\text{ ms}$ . Note that with a finer  $\Delta x$ , the profile is more rounded and the propagation velocity is slightly faster than can be seen in (a). (c) Instantaneous grid structures corresponding to the voltage profile in (b). The coarsest resolution is shown in white, the intermediate resolution in yellow, and the finest resolution in green. (d) and (e) Plots of the potential time derivative  $\partial V/\partial t$  vs potential  $V$  corresponding to fixed sites in the domains shown in (a) and (b), respectively, over one action potential. The circles show voltage traces gathered from the point (0.3, 1.0), to the left of the stimulus site and thus affected mainly by longitudinal propagation, and the squares show traces from (1.0, 0.8), below the stimulus site and affected mainly by transverse propagation.

angle. Both the shape of the front and the conduction velocities along and across fibers are affected. Using the AMRA, it is straightforward to add additional levels of resolution in order to determine the resolution needed to achieve a desired small variation in the conduction velocities along and across fibers for various fiber angles.

We examined the effect of spatial resolution and fiber angle on the longitudinal and transverse conduction velocities for the LR1 model using parameters as described above in Sec. II with the same anisotropy ratio of 10:1. Figure 3 shows isochrones spaced every 5 ms for 20 ms using angles of  $0^\circ$  and  $45^\circ$  for finest spatial resolutions of  $125\ \mu\text{m}$ ,  $62.5\ \mu\text{m}$ , and  $31.25\ \mu\text{m}$ . To allow easier comparison between the  $0^\circ$  and  $45^\circ$  results, the isochrones obtained using fiber angles of  $45^\circ$  were rotated  $-45^\circ$  in the figure. The isochrones illustrate the decreasing angle dependence of the shape and speed of the front as the resolution decreases. To obtain less than 10% variation in both the longitudinal and the transverse conduction velocities at fiber angles of  $0^\circ$  and  $45^\circ$ , a spatial resolution of roughly  $25\ \mu\text{m}$  must be used, or  $50\ \mu\text{m}$  for  $<25\%$  variation. While reducing the angle depen-

dence may not be necessary for all calculations, there may be some cases where it can affect results. For instance, in 3D calculations with rotational anisotropy, where the fiber angle rotates slowly with depth, it may be important to reduce the angle dependence by using fine spatial resolution when analyzing interactions among several fronts. The angle dependence decreases for smaller anisotropy ratios. In addition, models with upstrokes that are not as sharp as the LR1 upstroke do not require as high resolution to avoid the angle dependence. For example, the Fenton–Karma model<sup>9</sup> with modified LR1 kinetics can achieve  $<10\%$  variation with angle in both longitudinal and transverse conduction velocities at a spatial resolution of roughly  $55\ \mu\text{m}$ , or  $<25\%$  at  $100\ \mu\text{m}$ , about twice as coarse as needed for comparable agreement using the LR1 model.

### B. Inhomogeneities in a 2D medium

In Ref. 12, only homogeneous domains were considered. However, cardiac tissue contains inhomogeneities inherent to its structure, such as varying fiber directions and cell-to-cell

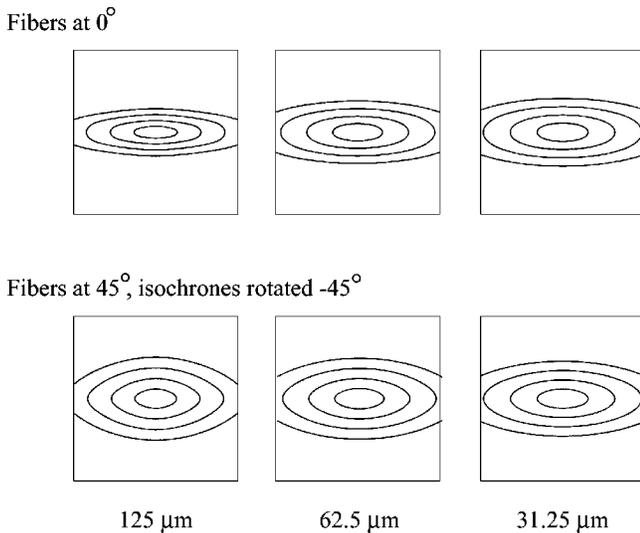


FIG. 3. Isochrones from a propagating wave in an anisotropic medium using the LR1 model with fibers at  $0^\circ$  (top row) and  $45^\circ$  relative to the  $x$ -axis (bottom row). The isochrones from the  $45^\circ$  cases have been rotated  $-45^\circ$  and spatially truncated at the boundaries to facilitate comparison with the corresponding  $0^\circ$  cases. The larger the value of  $\Delta x$ , the slower the propagation velocity and the more different the shapes become for the two fiber angles. Conductivities are 2 mS/cm and 0.2 mS/cm in the longitudinal and transverse directions, respectively. The domain is  $2\text{ cm} \times 2\text{ cm}$ . Isochrones are shown at 5 ms intervals for 20 ms.

variations in electrophysiological properties. Diseased tissue may contain additional inhomogeneities due to conditions like ischemia or infarction. It is important for the AMRA to treat inhomogeneous regions appropriately. In the presence of anisotropy, regions of high error cause refinement only along the wavefront and sometimes along the wave back, depending on the model used and the accuracy required. When the domain no longer is homogeneous, the refinement may last longer in the inhomogeneous tissue if a front lingers there longer due to prolonged APD or to slower conductivity. However, inhomogeneities may cause conduction block, and in that case refinement should be triggered even in the absence of a wavefront or back due to the spatial gradients in voltage.

We tested how the AMRA refined a domain with an inhomogeneity in membrane parameters that led to propagation block. The domain was  $3\text{ cm} \times 3\text{ cm}$  with a  $1\text{ cm} \times 0.1\text{ cm}$  inhomogeneity located in the center. Inside and outside the inhomogeneous region, the calcium conductance  $\bar{g}_{\text{Ca}}$  was set to  $0.075\text{ mS}/\mu\text{F}$  and  $0.045\text{ mS}/\mu\text{F}$ , respectively, causing prolongation of the APD in the inhomogeneous region. The conductivities in the  $x$ - and  $y$ -directions were set to 2 mS/cm and 0.5 mS/cm, giving an anisotropy ratio of 4:1. The medium was paced every 120 ms from a  $0.05\text{ cm}$  square site in front of the inhomogeneous region. Stimuli were unable to propagate through the inhomogeneous tissue when it was still refractory and thus an obstacle to conduction could be created.

Figure 4 shows a series of representative snapshots of the membrane potential and the corresponding grid structure for the inhomogeneous domain. At time  $t=966\text{ ms}$ , a new stimulus had been introduced just 6 ms earlier, while the

center of the medium was still recovering from a blocked stimulus initiated at time  $t=840\text{ ms}$ . Because of the large spatial gradient, much of the inhomogeneous region was still refined although no wave front or back was present in that area. At time 986 ms, the new stimulus was propagating around but not through the refractory area, which caused the front to split into the two parts shown. In the corresponding grid structure, the refined regions from the refractory area and from the propagating front had merged into a heart-shaped area. Since the refractory region was small and the medium highly excitable, the front reformed on the other side of the inhomogeneity, as illustrated at  $t=998\text{ ms}$ . At time  $t=1030\text{ ms}$ , the front had completely passed the inhomogeneous region and had left behind an area that was not excited by the most recent stimulus, because the refractory region had not recovered enough to be re-excited. At this time and later at  $t=1056\text{ ms}$ , an area in the middle of the domain remained refined due to the spatial gradients there.

Another stimulus was applied at  $t=1080\text{ ms}$ , and because the domain was quiescent near the stimulus site, propagation began, as shown at time  $t=1094\text{ ms}$ . However, the wave was blocked because the rest of the medium was refractory. While the exterior portion of the domain recovered, as shown at time  $t=1146\text{ ms}$ , the central region remained excited from the blocked stimulus, and the corresponding grid structure reflects the spatial gradients that were still present in the center of the domain even after the wave back had passed through nearly all the rest of the medium. At time  $t=1197\text{ ms}$ , just 3 ms before the next stimulus was applied, the medium remained refractory in the middle of the domain, with refinement persisting in the areas with spatial gradients, similar to the plot shown at  $t=966\text{ ms}$ .

To summarize, the AMRA does not treat inhomogeneous domains in an intrinsically different manner than it treats the homogeneous domains. Wavefronts and backs still trigger refinement. However, in an inhomogeneous region, propagation may be blocked while most of the medium recovers from a wave that propagated successfully. This type of dynamics gives rise to sharp spatial gradients that trigger refinement in areas that may be distant from the wavefronts and backs.

### C. Accuracy

Having shown that the AMRA is capable of handling anisotropy and inhomogeneities, we next analyze its accuracy. In Ref. 12, we established that the action potential shape, APD, and conduction velocity found using the AMRA agreed with comparable uniform mesh values to within 1%, except at the peak of the action potential, where the relative error was roughly 6%. Here, we use measures of accuracy applicable to 2D and 3D domains to establish that no inaccuracies are incurred when using the AMRA in higher spatial dimensions. To quantify the accuracy of the AMRA code in 2D and 3D media, spiral tip and filament trajectories were analyzed. We used the FHN model rather than the LR1 model, since spirals generally are unstable in LR1 and the dependence of their trajectories on the LR1 model param-

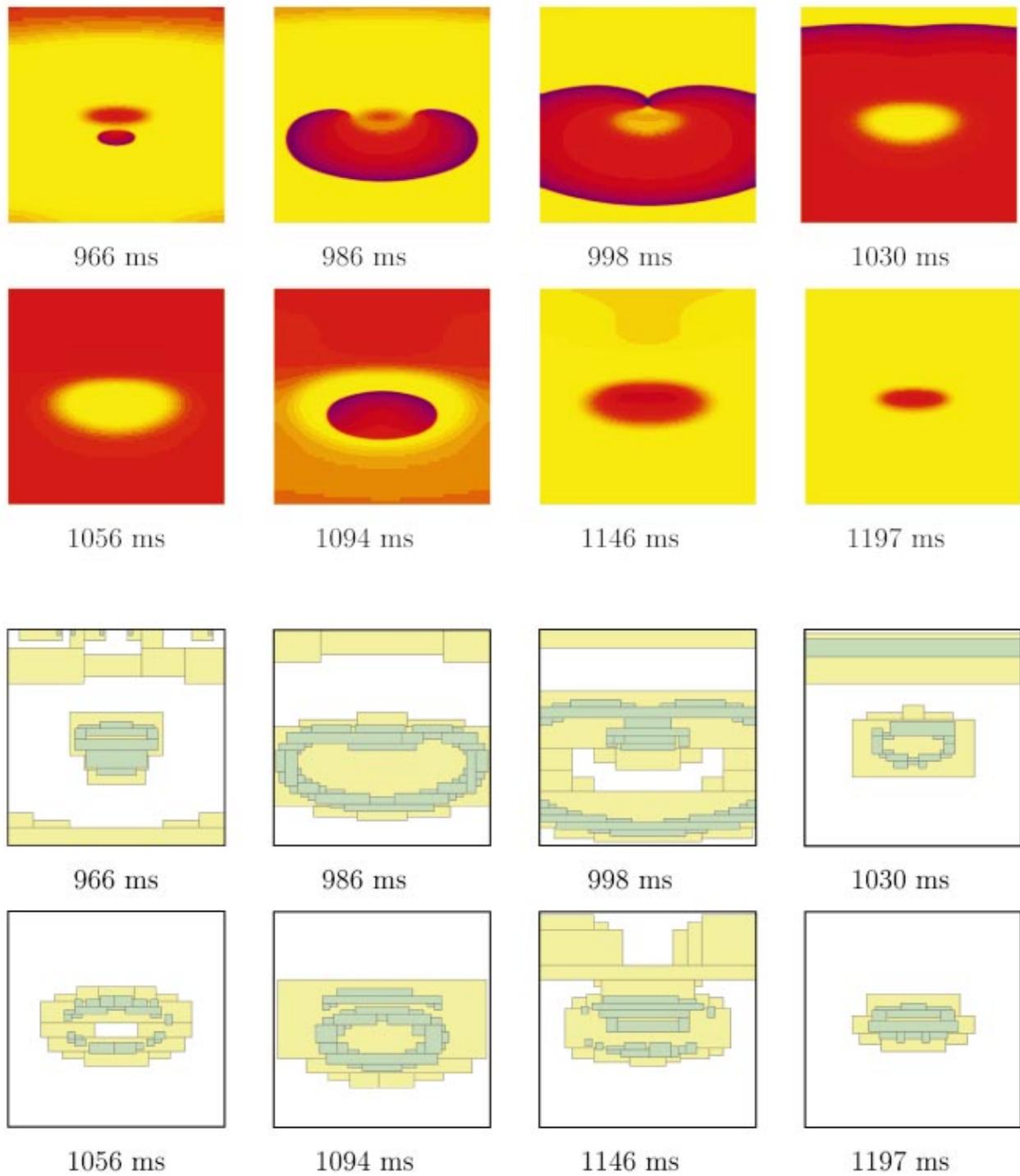


FIG. 4. (Color) Front profiles in a medium with an inhomogeneity in  $\bar{g}_{Ca}$ . The calculation used three levels of resolution, from  $\Delta x=500 \mu\text{m}$  and  $\Delta t=0.32 \text{ ms}$  to  $\Delta x=125 \mu\text{m}$  and  $\Delta t=0.02 \text{ ms}$ . Field value colors are as described in Fig. 2(a) for the voltage profiles and in Fig. 2(c) for the grid structures. The first stimulus shown does not affect a small region of refractoriness in the center of the domain, which causes the next stimulus to be blocked in most of the domain. Additional details are given in the text.

eters has not been studied as thoroughly.<sup>31</sup> Model parameters were set as discussed in Sec. II, producing a spiral tip which in an isotropic domain traced out a three-petal flower pattern that rotated slowly over time. The petals were stretched somewhat in the presence of anisotropy of strength comparable to what is seen in cardiac tissue (we used a ratio of 5:1 compared to the isotropic pattern).

In 2D, the tip trajectory traced by the AMRA code with three levels, from  $\Delta x=0.25$  to  $\Delta x=0.0625$  in space and from  $\Delta t=0.01$  to  $\Delta t=0.000625$  in time, was compared with that traced by a uniform mesh code using the finest spatial and temporal resolution,  $\Delta x=0.0625$ ,  $\Delta t=0.000625$  in a square domain with a side length of 30 spatial units. The location of each tip was identified by finding the intersection

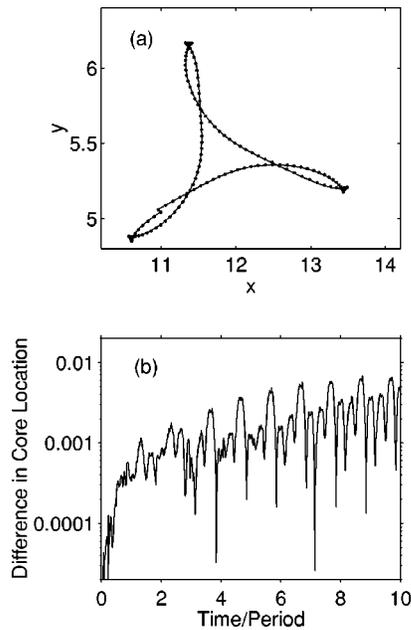


FIG. 5. (a) Very good agreement is obtained for the spiral tip trajectories produced for the FHN model using the AMRA (filled circles) and a uniform mesh at the finest space–time resolution of the AMRA (line). This is the final flower produced after about ten rotations. Parameter values are given in the text. The  $x$ -axis has been scaled differently than the  $y$ -axis to clarify the shape of the flower. (b) Difference in the spiral tip trajectory produced for the FHN model by the AMRA compared to the trajectory produced by a uniform mesh at the finest space–time resolution of the AMRA. About ten rotations have occurred during the time plotted. The error grows quickly as the spiral initially forms and then grows more slowly over the remainder of the time, but after ten rotations the maximum difference is less than 0.01 spatial units, which is less than the fine spatial resolution  $\Delta x$ . Parameter values are given in the text.

of the  $u=0.5$  and  $\partial_t u=0$  contours every 0.4 time units to identify points with zero normal velocity.<sup>9</sup> Specifically, the coarse grid was swept to find squares where the contours intersected, and the location of the spiral tip was calculated using linear interpolation. By comparing the tip locations found with images of the solution, it was determined that no false tips were identified.

Figure 5(a) illustrates the trajectories produced by both the uniform mesh (filled circles) and the AMRA (line) codes after about ten rotations. The two trajectories are nearly identical. Figure 5(b) shows the corresponding difference of the AMRA tip position compared to the uniform mesh tip position over the entire time integrated. After the first few rotations, the difference appears to grow linearly over time, which is consistent with the error being caused by a slight phase difference. Over ten rotations, the average difference in tip location is approximately 0.002 spatial units, with the maximum value less than 0.01 spatial units, or roughly 0.03% of the length of one side of the square domain and still much less than the fine spatial resolution  $\Delta x$ . These data indicate that the AMRA is not a significant source of error in the tracing of spiral tips in our 2D anisotropic FHN medium.

In 3D, we combined rotational anisotropy with the parameters already used in 2D. This produced a stack of spiral waves whose tips formed a filament stretching between the  $z=0$  and  $z=L_z$  planes. As the spirals meandered within the

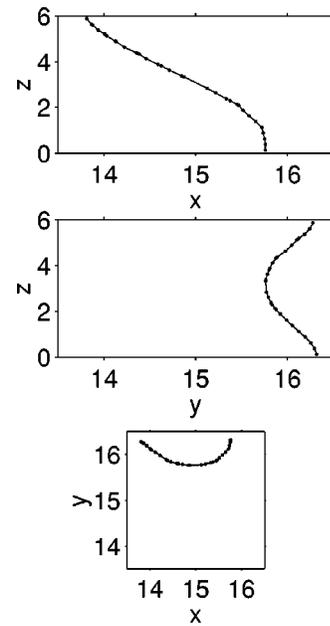


FIG. 6. Comparison of a filament in the FHN model using the AMRA (lines) and a uniform grid at the finest space–time resolution of the AMRA (circles) after 140 time units, or about 12 rotations of the scroll wave. The filaments are compared by depicting their projections in the  $x$ - $z$ ,  $y$ - $z$ , and  $x$ - $y$  planes. The difference in all three projections never exceeds the fine spatial resolution. Parameter values are given in the text.

planes, the filaments bent and twisted. The system was 30 spatial units in the  $x$ - and  $y$ -directions and 6 spatial units thick, which was not enough to give rise to transmural filament breakup. The uniform grid used resolutions of  $\Delta x=0.25$  and  $\Delta t=0.01$ , while the adaptive calculation used these same values at the third and highest level of resolution and values of  $\Delta x=1$  and  $\Delta x=0.16$  at the coarsest level. To avoid large jumps in the fiber orientation from one  $x$ - $y$  plane to the next, we used a rotation rate of  $3^\circ$  per fine  $\Delta x$ , for a total of  $72^\circ$  from the top to the bottom surface. Filaments were found by the same method used for identifying spiral tips in 2D, by looking at the intersection of the  $u=0.5$  and  $\partial_t u=0$  contours, in this case at every fine time step. The same code was used as for the 2D tip identification, with sweeps over the  $x$ - $y$ ,  $x$ - $z$ , and  $y$ - $z$  planes.

Figure 6 illustrates the filament projections in the  $x$ - $z$ ,  $y$ - $z$ , and  $x$ - $y$  planes after 140 time units, or approximately 12 rotations of the scroll wave. Data from the uniform mesh and AMRA calculations are shown as circles and lines, respectively. In all planes, the filaments are nearly identical, and the difference in the filament projections in all cases is less than the fine spatial resolution  $\Delta x$ . We conclude that the use of the AMRA does not contribute a significant error to the tracking of filaments in this parameter regime for the FHN model.

#### D. Efficiency in a 3D medium

Having demonstrated the accuracy and utility of the AMRA in 2D and 3D, we next quantify its efficiency in 3D using the LR1 model. To test the efficiency of the AMRA for a 3D tissue slab at the same spatial and temporal resolutions required to achieve  $<25\%$  variation in longitudinal and

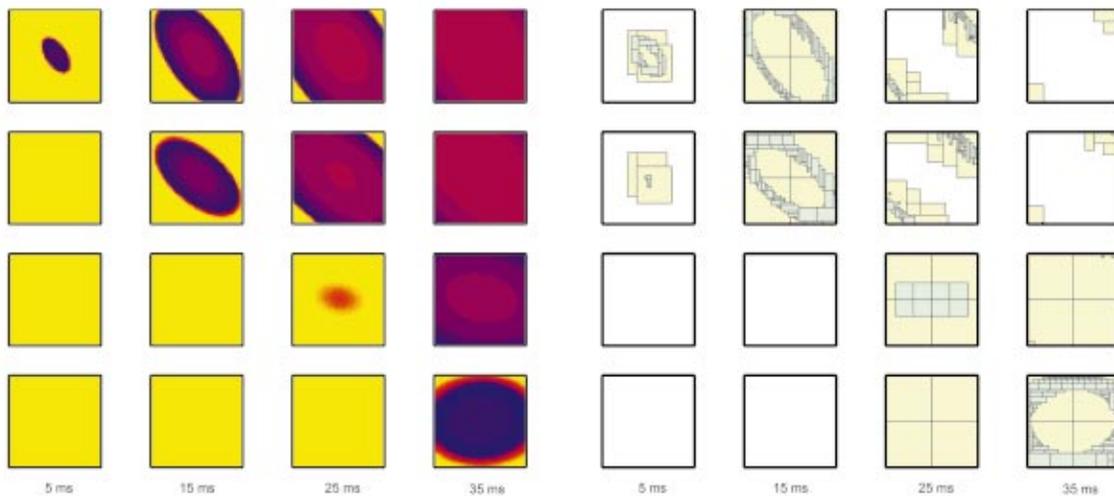


FIG. 7. (Color) Instantaneous voltage and grid structures during propagation in the LR1 model with rotational anisotropy. To facilitate visualization of the 3D data, four 2D slices spaced evenly from the epicardium to the endocardium and parallel to both surfaces are shown, with the slices stacked and the epicardium on top. The activation propagates from the epicardium down through the tissue and is affected by the rotational anisotropy, as can be seen from the different orientations of the propagating fronts in the slices shown. The color scale is the same as in Fig. 2 for both the voltage and the grid structures. The anisotropy ratio is 10:1. Additional parameter values are given in the text.

transverse conduction velocities for the 2D case already described (using three levels of refinement:  $\Delta x = 400 \mu\text{m}$ ,  $\Delta t = 0.2 \text{ ms}$ ;  $\Delta x = 100 \mu\text{m}$ ,  $\Delta t = 0.0125 \text{ ms}$ ; and  $\Delta x = 50 \mu\text{m}$ ,  $\Delta t = 0.003125 \text{ ms}$ ), a single front was initiated in a cube centered in the epicardium of a domain measuring  $1.6 \text{ cm} \times 1.6 \text{ cm} \times 0.64 \text{ cm}$ . Figure 7 shows four 2D slices of the 3D tissue from epicardium to endocardium to illustrate the dynamics within the thickness, along with the corresponding grid hierarchies used on each slice. The front spread throughout the domain and the simulation lasted until the entire medium was again quiescent, 260 ms in all. The AMRA used the finest level of resolution only when the front was present in the domain, which allowed coarser spatial and temporal resolution to be used during the much longer refractory period. This local and dynamic adaptivity led to a substantial computational savings: while 84% of the total number of function evaluations were performed for the finest resolution, on average the finest resolution covered  $< 2\%$  of the physical domain. Overall, the AMRA required a factor of 50 less computational time and a factor of 30 less memory on average than a comparable calculation on a uniform space–time mesh using the finest AMRA resolution. The simulation was performed in about 12.8 h on a 750-MHz Alpha 264 processor with 2 GB of memory.

Although the AMRA substantially reduces the resources required, a simulation of more complex dynamics using the LR1 model at this spatial and temporal resolution is still beyond our group's current computational resources. However, we can extrapolate the algorithm's performance to domains with more complex dynamics. In a 2D domain with a single propagating pulse, we found a speedup of about a factor of 20 and an average savings in memory of a factor of 10. With multiple spiral waves, we found a speedup of a factor of 5 with a comparable reduction in memory.<sup>12</sup> In moving to a domain with more complex dynamics in 2D, then, the speedup was reduced by a factor of 4 and the

memory savings by a factor of 2. If the same scaling holds for regimes with more complex dynamics in 3D, we can expect a speedup of about a factor of 12 and a reduction in required memory of about a factor of 15 for the LR1 model.

## V. DISCUSSION AND CONCLUSIONS

In this paper, we have discussed new features, new applications, and new capabilities of a space–time adaptive method recently introduced for simulations of cardiac tissue and other excitable media.<sup>12</sup> We have shown the AMRA's ability to simulate inhomogeneous and strongly anisotropic domains using the LR1 model. Using the AMRA has no significant effect on solution accuracy, as quantified by comparing spiral wave tips and filament locations in the FHN model found using the AMRA with those found using a uniform space–time mesh at the finest resolution of the AMRA calculations. For a single propagating pulse in a 3D slab of cardiac tissue using the LR1 model and incorporating rotational anisotropy with high spatial resolution to minimize angle dependence, we have demonstrated that our AMRA reduces the computational time and memory requirements necessary for a comparable calculation on a uniform space–time mesh by factors of 50 and 30, respectively. We emphasize that the AMRA does not require the use of simpler models or special integration schemes. Moreover, the algorithm can be parallelized straightforwardly,<sup>43,44</sup> which will facilitate calculations on larger domains for longer times, particularly for 3D calculations.

Using the AMRA can provide a variety of advances in addition to reductions in time and memory requirements. For instance, it can allow more complex ionic models to be used in place of simpler ones, either to perform a series of simulations or to compare results across models, without altering the runtime significantly. Similarly, it can allow the use of higher spatial and temporal resolution to reduce the angle

dependence of propagation patterns in domains with rotational anisotropy or to verify that a calculation is sufficiently resolved so that an increase in resolution does not affect the results.

Furthermore, using the AMRA can allow simulation of larger domains, including published anatomical models, such as rabbit ventricles<sup>7</sup> and canine ventricles.<sup>6</sup> Based on the known efficiencies of the AMRA in 2D and 3D domains with a single propagating pulse and in 2D with more complex dynamics, we have estimated that using the AMRA in a 3D domain with complex dynamics will result in a savings in memory of a factor of 15 on average and a speedup of a factor of 12 compared to a uniform space–time grid. Assuming that the effective spatial and temporal resolutions for the models are 50  $\mu\text{m}$  and 0.003125 ms, respectively, and anticipating a modest factor of 2 speedup from using more modern processors (such as the 1-GHz Alpha chip), we can estimate that simulating 1 s of complex dynamics in the rabbit ventricles will require approximately 25 days on a single processor, and parallelization would be necessary to reduce this to a more tractable problem. If we gain a speedup of a factor of 28 using 32 processors, as Allard and Trangenstein<sup>44</sup> achieved using a similar algorithm for a different class of problems, the computation time required to simulate 1 s of fibrillation in the rabbit ventricles at this resolution would be reduced to less than a day. Simulating 1 s of activity on the much larger and thicker canine ventricles would require substantially more time, an estimated factor of 17 longer, and likely would require the use of a coarser resolution to obtain results in a reasonable amount of time. For instance, if 125  $\mu\text{m}$  and 0.02 ms were used, 1 s using the canine ventricular anatomy would require just over 9 h using 32 processors that provide a speedup of 28.

Our estimate of the speedup anticipated for fibrillating 3D domains is based on the known efficiency of a plane wave in 3D and the known reduction in efficiency of the AMRA for 2D domains containing multiple waves compared to a single plane wave. However, it is possible that waves may pack more densely in 3D than in 2D, in which case our estimate will be too optimistic. Nevertheless, we believe that in most cases beneficial speedup will be obtained. We also note that while implementing other models into the AMRA code is straightforward, more complex models may pose new challenges. For instance, the Luo–Rudy dynamic (LRd) model<sup>47,48</sup> is known to be sensitive to calcium and requires smaller time steps than the LR1 model. In addition, the LRd and other models can exhibit calcium overload and the subsequent release of intracellular calcium during any phase of the action potential, and these releases may occur more often during fibrillation. To ensure that adequate resolution is used for these events, it may be necessary to incorporate calcium variables into the error estimate that indicates where refinement is required. With finer resolution needed more often to resolve calcium processes and possibly more complex error estimation, the savings from using the AMRA may be more modest for models more complex than LR1. However, it is important to note that the usefulness of the AMRA depends on the question being studied. Investigations of the onset of fibrillation, which would involve fewer waves, likely would

benefit more from use of the AMRA than studies of fully developed fibrillation.

Along with parallelization, there are several other ways to enhance the capabilities of codes using the AMRA. One particularly important extension to develop and implement is the capability to represent curved physical boundaries within the Cartesian meshes of the AMRA. Recently, Calhoun and LeVeque<sup>40</sup> introduced a technique to represent irregular boundaries for a 2D adaptive method (which should generalize to 3D) based on Berger's original algorithm. Their method uses a capacity function to identify what fraction of the area of a boundary cell is contained in the computational domain and from this information varies the treatment of the boundary cell. Another technique that has been used for uniform grids is the phase field method, which originally was developed to follow advancing fronts in solidification where boundaries are continually changing,<sup>49</sup> and which has been extended to model stationary curved boundaries such as those found on the surfaces of the heart.<sup>50,51</sup> In this formulation, an additional matrix describing the structure is set up to describe a given geometry, with values of 1 representing points in the domain and values of 0 representing points outside the domain. A phase field is generated by assigning intermediate values along the edge of the domain to smooth the transition from 1's to 0's. Neumann boundary conditions are imposed correctly provided that algorithmic parameters governing the width of the phase field and the values for intermediate points are set to appropriate values for the problem. We anticipate that the phase field method will work within the AMRA framework by generating phase field matrices for all spatial resolutions before starting the calculation and using the correct matrices on the corresponding grid levels.

The algorithm also could be made more efficient by incorporating known information about the dynamics into the regridding algorithm. Because regridding does not occur every step, fine patches currently are slightly enlarged in all directions to allow propagating fronts to remain refined until regridding next occurs. However, the fronts represented on these fine patches can propagate only into neighboring quiescent regions. Using this knowledge, a front-prediction scheme could be developed that would enlarge fine patches only in the directions in which propagation can occur, thereby limiting the portion of the domain resolved to finer levels. In addition, the use of refinement criteria specific to cardiac models and computationally less expensive than Richardson extrapolation, such as refining in portions of the grid where the sodium channel is opening, also could be beneficial.

With parallelization, curved boundaries, and front-prediction added, AMRA-based codes will provide extremely powerful tools for understanding and eventually treating cardiac arrhythmias.

## ACKNOWLEDGMENTS

We thank M. Berger and F. Fenton for useful discussions and M. Berger for making available to us her 2D and 3D AMRA codes that we modified to obtain the results reported here.

- <sup>1</sup>American Heart Association, *Heart Disease and Stroke Statistics—2003 Update* (American Heart Association, Dallas, 2002).
- <sup>2</sup>A.V. Zaitsev, O. Berenfeld, S.F. Mironov, J. Jalife, and A.M. Pertsov, *Circ. Res.* **86**, 408–417 (2000).
- <sup>3</sup>W. Garrey, *Am. J. Physiol.* **33**, 397–414 (1914).
- <sup>4</sup>M.J. Janse, F. Wilms-Schopman, and R. Coronel, *J. Cardiovasc. Electrophysiol.* **6**, 512–521 (1995).
- <sup>5</sup>A. Winfree, *Science* **266**, 1003–1006 (1994).
- <sup>6</sup>P. Nielsen, I.L. Grice, B. Smaill, and P. Hunter, *Am. J. Physiol.* **260**, H1365–H1378 (1991).
- <sup>7</sup>F.J. Vetter and A.D. McCulloch, *Prog. Biophys. Mol. Biol.* **69**, 157–183 (1998).
- <sup>8</sup>A.V. Panfilov, *Phys. Rev. E* **59**, R6251–R6254 (1999).
- <sup>9</sup>F. Fenton and A. Karma, *Chaos* **8**, 20–47 (1998).
- <sup>10</sup>C. Antzelevitch, W. Shimizu, G.-X. Yan, S. Sicouri, J. Weissenburger, V.V. Nesterenko, A. Burashnikov, J. DiDiego, J. Saffitz, and G.P. Thomas, *J. Cardiovasc. Electrophysiol.* **10**, 1124–1152 (1999).
- <sup>11</sup>R. Wolk, S.M. Cobbe, M.N. Hicks, and K.A. Kane, *Pharmacol. Ther.* **84**, 207–231 (1999).
- <sup>12</sup>E.M. Cherry, H.S. Greenside, and C.S. Henriquez, *Phys. Rev. Lett.* **84**, 1343–1346 (2000).
- <sup>13</sup>C.-H. Luo and Y. Rudy, *Circ. Res.* **68**, 1501–1526 (1991).
- <sup>14</sup>R.A. FitzHugh, *Biophys. J.* **1**, 445–466 (1961).
- <sup>15</sup>O. Berenfeld and A.M. Pertsov, *J. Theor. Biol.* **199**, 383–394 (1999).
- <sup>16</sup>F. Fenton, *Comput. Cardiol.* **27**, 251–254 (2000).
- <sup>17</sup>F.H. Fenton, E.M. Cherry, H.M. Hastings, and S.J. Evans, *Chaos* **12**, 852–892 (2002).
- <sup>18</sup>D. Barkley, *Physica D* **49**, 61–70 (1991).
- <sup>19</sup>W. Quan, S.J. Evans, and H.M. Hastings, *IEEE Trans. Biomed. Eng.* **45**, 372–385 (1998).
- <sup>20</sup>Z. Qu and A. Garfinkel, *IEEE Trans. Biomed. Eng.* **46**, 1166–1168 (1999).
- <sup>21</sup>P.K. Moore, *SIAM J. Sci. Comput.* **21**, 1567–1586 (2000).
- <sup>22</sup>G. Strang, *SIAM J. Numer. Anal.* **5**, 506–517 (1968).
- <sup>23</sup>A.E. Pollard, N. Hooke, and C.S. Henriquez, *Crit. Rev. Biomed. Eng.* **20**, 171–210 (1992).
- <sup>24</sup>E.M. Cherry, Ph.D. thesis, Duke University, 2000.
- <sup>25</sup>E.J. Vigmond and L.J. Leon, *Ann. Biomed. Eng.* **27**, 160–170 (1999).
- <sup>26</sup>G. Rousseau and R. Kapral, *Chaos* **10**, 812–825 (2000).
- <sup>27</sup>D. Allexandre and N.F. Otani, *Comput. Cardiol.* **29**, 9–12 (2002).
- <sup>28</sup>M.J. Berger and J. Oliger, *J. Comput. Phys.* **53**, 484–512 (1984).
- <sup>29</sup>M.J. Berger and P. Colella, *J. Comput. Phys.* **82**, 64–84 (1989).
- <sup>30</sup>J. Bell, M. Berger, J. Saltzman, and M. Welcome, *SIAM J. Sci. Comput.* **15**, 127–138 (1994).
- <sup>31</sup>A.T. Winfree, *Chaos* **1**, 303–334 (1991).
- <sup>32</sup>R. Trompert and J. Verwer, *Appl. Numer. Math.* **8**, 65–90 (1991).
- <sup>33</sup>R. Hornung and J. Trangenstein, *J. Comput. Phys.* **136**, 522–545 (1997).
- <sup>34</sup>U. Ziegler, *Comput. Phys. Commun.* **116**, 65–77 (1999).
- <sup>35</sup>H. Friedel, R. Grauer, and C. Marliani, *J. Comput. Phys.* **134**, 190–198 (1997).
- <sup>36</sup>A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, and M.L. Welcome, *J. Comput. Phys.* **142**, 1–46 (1998).
- <sup>37</sup>A.M. Roma, C.S. Peskin, and M.J. Berger, *J. Comput. Phys.* **153**, 509–534 (1999).
- <sup>38</sup>G.L. Bryan, *Comput. Sci. Eng.* **1**, 46–53 (1999).
- <sup>39</sup>R.I. Klein, *J. Comput. Appl. Math.* **109**, 123–152 (1999).
- <sup>40</sup>D. Calhoun and R.J. LeVeque, *J. Comput. Phys.* **157**, 143–180 (2000).
- <sup>41</sup>P. MacNeice, K.M. Olson, C. Mobarry, R. de Fainchtein, and C. Packer, *Comput. Phys. Commun.* **126**, 330–3554 (2000).
- <sup>42</sup>D. Kincaid and W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*, 2nd ed. (Brooks/Cole, Pacific Grove, 1996).
- <sup>43</sup>M.J. Berger, in *Adaptive Methods for Partial Differential Equations*, edited by J.E. Flaherty, P.J. Paslow, M.S. Shephard, and J.D. Vasilakis (SIAM, Philadelphia, 1989), pp. 150–159.
- <sup>44</sup>W.K. Allard and J.A. Trangenstein, preprint available online at <http://www2.math.duke.edu/publications/1174> (unpublished).
- <sup>45</sup>S. Rush and H. Larsen, *IEEE Trans. Biomed. Eng.* **25**, 389–392 (1978).
- <sup>46</sup>B. Victorri, A. Vinet, F.A. Roberge, and J.-P. Drouhard, *Comput. Biomed. Res.* **18**, 10–23 (1985).
- <sup>47</sup>C.-H. Luo and Y. Rudy, *Circ. Res.* **74**, 1071–1096 (1994).
- <sup>48</sup>G.M. Faber and Y. Rudy, *Biophys. J.* **78**, 2392–2404 (2000).
- <sup>49</sup>A. Karma and W.-J. Rappel, *Phys. Rev. E* **53**, R3017–R3020 (1996).
- <sup>50</sup>F. Fenton, E.M. Cherry, I. Banville, R.A. Gray, H.M. Hastings, and S.J. Evans, *Pacing Clin. Electrophysiol.* **24**, 538 (2002).
- <sup>51</sup>E.M. Cherry, F. Fenton, H.M. Hastings, F. Xie, A. Garfinkel, J.N. Weiss, and S.J. Evans, *Pacing Clin. Electrophysiol.* **24**, 538 (2002).