Available online at www.sciencedirect.com
ScienceDirect

ELSEVIER

Procedia Computer Science 1 (2012) 935–944

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Science, ICCS 2010

Anisotropic mesh adaptivity for cardiac electrophysiology

J. Southern[a,*], G.J. Gorman[b], M.D. Piggott[b], P.E. Farrell[b], M.O. Bernabeu[c], J. Pitt-Francis[c]

[a]Fujitsu Laboratories of Europe, Hayes Park Central, Hayes End Road, Uxbridge, Middlesex, UB4 8FE, UK
[b]Applied Modelling and Computation Group, Department of Earth Science and Engineering, South Kensington Campus, Imperial College London, SW7 2AZ, UK
[c]Oxford University Computing Laboratory, Wolfson Building, University of Oxford, Parks Road, Oxford, OX1 3QD, UK

Abstract

The simulation of cardiac electrophysiology requires small time steps and a fine mesh in order to resolve very sharp, but highly localized, wavefronts. The use of very high resolution meshes containing large numbers of nodes results in a high computational cost, both in terms of CPU hours and memory footprint. In this paper an anisotropic mesh adaptivity technique is implemented in the Chaste physiological simulation library in order to reduce the mesh resolution away from the depolarization front. Adapting the mesh results in a reduction in the number of degrees of freedom of the system to be solved by an order of magnitude during propagation and 2–3 orders of magnitude in the subsequent plateau phase. As a result, a computational speedup by a factor of between 5 and 12 has been obtained with no loss of accuracy, both in a slab-like geometry and for a realistic heart mesh with a spatial resolution of 0.125 mm.

Keywords: Cardiac electrophysiology, simulation, anisotropic mesh adaptivity, bidomain.

1. Introduction

The normal sinus rhythm of the heart is characterized by waves with a sharp depolarization front that travel across the whole myocardium. This rapid variation in transmembrane potential means cardiac modellers require a very fine mesh resolution (node spacing of 0.2 mm or less [1]) in their simulations in order to ensure that the results are sufficiently accurate. At this resolution, a mesh containing approximately 30 million nodes would be required to represent a typical human heart (with a volume of 250 cm^3). A recently published rabbit ventricular mesh with a mean spatial resolution of 0.125 mm consists of 4.3 million nodes [2]. Running cardiac simulations on meshes with this number of nodes is computationally expensive as it requires the repeated solution of linear systems with nearly 10 million degrees of freedom (there are two independent variables at each node for the bidomain equations): simulating 400 ms of cardiac activity on this rabbit mesh using the Chaste software package [3] takes 15 CPU days. Further, the memory requirements of the simulations (both while they are running and in storing the data generated) become increasingly large. The use of adaptive mesh methods provides a solution to both of these problems. By maintaining the extremely fine resolution only where it is needed (i.e. in the vicinity of the wavefront) and coarsening elsewhere the number of

[*]Corresponding author
Email address: James.Southern@uk.fujitsu.com (J. Southern)

doi:10.1016/j.procs.2010.04.103

degrees of freedom could be greatly reduced, resulting in faster calculations, a lower memory overhead while running the simulations and smaller output files to be processed following the simulation in order to extract the important information. The key advances in this work are: the use of *anisotropic* adaptive mesh methods for *unstructured finite element meshes* to efficiently capture solution curvature within complex, realistic *in three dimensional* geometries; and to concentrate the resolution in the direction of rapid variations in the solution fields and *reducing the frequency at which the mesh is adapted* to limit the overhead incurred by remeshing.

2. Numerical Modelling of Cardiac Electrophysiology

In this work an adaptive method is used to increase the computational efficiency of solving the bidomain equations of cardiac electrophysiology [4], which can be written as:

$$\chi\left(C_{\mathrm{m}}\frac{\partial V_{\mathrm{m}}}{\partial t} + I_{\mathrm{ion}}(\mathbf{u}, V_{\mathrm{m}})\right) - \nabla\cdot(\sigma_{\mathrm{i}}\nabla(V_{\mathrm{m}} + \phi_{\mathrm{e}})) = 0, \tag{1}$$

$$\nabla\cdot((\sigma_{\mathrm{i}} + \sigma_{\mathrm{e}})\nabla\phi_{\mathrm{e}} + \sigma_{\mathrm{i}}\nabla V_{\mathrm{m}}) = -I_{\mathrm{e}}, \tag{2}$$

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}, V_{\mathrm{m}}), \tag{3}$$

plus appropriate boundary conditions, where $V_{\mathrm{m}}(\mathbf{x}, t)$ is the transmembrane potential, $\phi_{\mathrm{e}}(\mathbf{x}, t)$ is the extracellular potential, $\mathbf{u}(t)$ is a vector of dependent variables, I_{ion} is the total ionic current across the cell membrane per unit area, χ is the cell surface to volume ratio, C_{m} is the membrane capacitance per unit area, σ_{i} and σ_{e} are the intracellular and extracellular conductivity tensors respectively, I_{e} is an extracellular stimulus current (e.g. to model an electric shock during defibrillation), $\mathbf{f}$ is a vector-valued function that describes the cellular level behaviour, $\mathbf{x}$ is position and t is time. Here, the Luo-Rudy I model [5], a system of seven state variables (plus the transmembrane potential), is used for the function $\mathbf{f}$.

Physically, this describes cardiac cells that are electrically connected both via gap junctions between neighbouring cells and through the extracellular matrix that surrounds them. Cell membranes contain ion channels and pumps that selectively allow ions to cross. The activity of these channels and pumps sets up potential gradients both across gap junctions and in extracellular space, generating action potentials (APs) that propagate across the entire heart, both in the extra- and intracellular spaces.

There are many different finite difference, finite volume and finite element methods that can be used for solving the bidomain equations [6–12]. Given the complex geometry of the domain and the potential need to develop accurate error estimators in order to implement mesh adaptivity, the finite element method is the strategy adopted here.

Chaste [3] is a software development project that was set up in order to produce high-quality software for solving a variety of related problems in the field of computational biology, where "high-quality" means that the software is to be extensible, robust, fast, accurate, maintainable and is to use state-of-the-art numerical techniques. In order to ensure these high-quality goals are met, the library is written using modern agile software development techniques [13].

Chaste has been designed with the aim of being a generic multi-purpose library supporting computational simulations for a wide range of biological problems, but much of the software development to date has focused on the mature area of cardiac electrophysiology. The mathematical modelling of the cardiac bidomain equations (and other biological models) relies heavily on (i) accurate representation of a complex, realistic geometry provided through finite-element meshes; (ii) finite-element assembly of PDEs from a given domain into a sparse system of linear equations; (iii) the iterative solution of large sparse systems via the PETSc scientific library [14] and (iv) the numerical solution of ODE systems. These four common features of the model become four component libraries in the Chaste code: `mesh`, `pde`, `linalg` and `ode` respectively. They are supported by two lower-level libraries, `global` (which is responsible for initialization, book-keeping and exceptions) and `io` (which handles input and output). Classes for solving the bidomain equations are built upon these generic components. See [3] for details.

3. Brief Survey of the Use of Mesh Adaptivity in Cardiac Electrophysiology Applications

Several previous studies have implemented adaptive mesh methods for the solution of problems in cardiac electrophysiology, with varying degrees of success. However, many have been restricted to two-dimensional or structured grids (or both) and will most likely not be suitable for use with the three-dimensional unstructured finite element meshes used to represent realistic geometries. Further, only one existing method exploits anisotropic mesh adaptivity in order to ensure a higher resolution in the direction of greatest change in transmembrane potential. To date, almost all researchers have adapted the mesh at each time step. This appears to be an unnecessary expense as the solution does not change very much from one time step to the next. As a result of all of these factors, no previous study has been able to demonstrate a reduction in computational time by using adaptive mesh methods on a realistic three-dimensional heart geometry.

A spatio-temporal finite difference adaptivity method is presented in [15]. This uses nested patches of uniform Cartesian grids, guided by error estimators obtained from Richardson extrapolation. Results are presented which show a factor five reduction in computational effort and memory use without loss of accuracy. The authors comment that it is not necessary to adapt the mesh every time step, although they do adapt every second time step in the work presented. In [16] it is observed that the anisotropic nature of conductivities in the human heart (potential variation by a factor of 4–10 between the longitudinal and transverse directions) leads to differences in wave speed and front width depending on the direction of propagation. Hence the desired grid resolution is direction-dependent. The authors comment that one can only achieve this with both Cartesian grids (whether fixed or adaptive) if the anisotropy is aligned with coordinate axes. When fibres are not aligned this can lead to distorted waves when using Cartesian grids unless the resolution is very high. Two- and three-dimensional examples in simple geometries show a factor 50 increase in computational efficiency. The inability of a Cartesian grid method to handle the curved boundaries typical of realistic heart geometries is mentioned as a further potential drawback.

Mortar elements are used in [17] to form composite grid equations that define different finite element meshes in different parts of the domain. Adaptive mesh refinement (AMR) methods on quadrilateral grids are used in two dimensions and the approach has the ability to use local time steps for each grid. The error estimator is based on a hierarchical approach used to control both spatial and temporal adaptivity. Examples are given of wave propagation for 460 ms on a two-dimensional problem with simple geometry. Using a 320-by-320 uniform grid requires 4863 s time of simulation, whereas AMR with five levels of refinement had a timing of 2812 s. However, the mesh refinement is isotropic, has not been extended to three dimensions and suffers from the drawbacks of using structured grids mentioned above.

In [18] the multi-scale nature of the problem is noted due to the presence of very steep wave fronts. Simple two-dimensional geometries are considered and the gradient of solution variables is used to flag elements close to the front. A coarse mesh is used throughout the domain, with a finer mesh used in the flagged region (both are structured grids). The fine mesh is obtained by subdividing coarse elements into a higher number; nine are used here. The author comments on the additional costs associated with recalculating discretization matrices on the fine mesh and that speedup therefore depends on the proportion of the domain using the fine mesh. Analysis of two-dimensional results indicate that factor 250 efficiency gains are possible with adaptive methods and it was postulated that speedups are likely to be greater in three dimensions. However, no anisotropic or three-dimensional implementation has yet been tested—and the method's dependence on a structured mesh would likely make it unsuitable for use with realistic heart geometries.

A series of recent papers [19–21] describe the application of the KARDOS library (for the solution of nonlinear systems of parabolic PDEs) as an adaptive finite element solver for electrocardiology. In [19] simple geometries are considered with an adaptive approach based upon hierarchical refinement of the mesh according to multi-level *a posteriori* error estimators. The level of refinement or the total number of grid points is generally limited. This work is extended upon in [20], with examples in realistic three-dimensional geometries. Up to five levels of hierarchical (isotropic) refinement are used and this leads to the use of 2.1 million vertices. The authors comment that a computation simulating 800 ms on an eight dual-core AMD machine takes six weeks of CPU time (representing a degradation in performance compared with a non-adaptive scheme). The authors also use their previous experience with KARDOS to discuss the likely effects of the multi-scale nature of the problem and to compare the complexities associated with solving the bidomain and monodomain eqauations with various choices of cell model and how this might impact on the effectiveness of an adaptive scheme. Most recently in [21] an interesting discussion is presented on the mixed results of using adaptive methods in cardiac simulation to date. For example, a factor 150 reduction in the number of vertices compared with fixed grids with the same finest resolution has been seen, but this did not translate into reduced CPU requirements. Some reasons for this are given and include: time step limitations by the front speed and its width; the computation of error estimates being a significant part of the overall computational

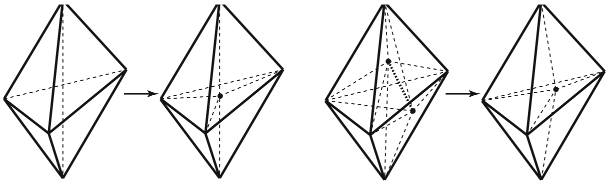

Figure 1: Example of the edge split (left) and edge collapse (right) operations in 3D. The edge is split at its midpoint and the newly created node is then connected to the vertices of the elements containing the edge, thus creating additional elements. For edge collapse, the elements surrounding the edge are deleted and a node placed at the midpoint of the removed edge.

work; mesh modifications leading to the need to regularly re-compute discretization matrices, and this could happen multiple times per time step; and finally the cost of the mesh modifications themselves and the associated non-local data structures also take their toll. Note, however, that generally adapts have been performed at each time step. Since the location of the wave front moves relatively slowly compared with the time step used this is likely to be much more frequent than necessary.

The possible enhancements (compared with previous studies) in representing steep wave fronts that could result from using anistotropic mesh refinement are reported for the first time in [22]. Examples of cardiac applications in simple two-dimensional geometries are presented. The author comments that using this scheme the mesh must be adapted every time step and that this uses 8–10 iterations of their mesh optimisation algorithm. It is also commented that mesh-to-mesh interpolation can lead to numerical diffusion; this will be exacerbated by adapting the mesh every time step. However, recently optimally accurate interpolation methods that help to minimize this issue have been described in [23, 24]. On an analytical two-dimensional problem a factor 64 improvement in element count for the same resulting accuracy is observed. On a two-dimensional spiral wave problem there was a factor 18 improvement. In [25] the extension of the previous work to both three-dimensional problems and realistic geometries (canine heart) is presented. For simplicity the monodomain equations are used. In contrast to the two-dimensional applications which used hierarchical error estimators, here a Hessian based error indicator was used as the former approach is not available in three dimensions. Adaptivity is again applied every time step. Several metrics are used to gauge solution quality including Richardson extrapolation of the depolarization time at the centre of the box geometry. The adaptive results are shown to behave favourably compared to uniformly high resolution meshes. On a cubic geometry a factor 43 reduction in degree of freedom count corresponds to a computational speedup of 6.4. Part of this discrepancy is due to the computation of the error measure and the generation of a new mesh. However, no reduction in computational time is reported on the heart mesh used. And, given that on this mesh the number of degrees of freedom is only reduced by a factor of 2.3 it seems likely that the adaptive simulation will in fact have been slower than the fixed mesh simulation.

4. Anisotropic Mesh Adaptivity

In any numerical model the quality of the underlying computational mesh, in addition to the discretization method, is crucial. The approach taken in this work is to apply the mesh optimization algorithm of [26] to modify the mesh so that it is appropriate for the representation of the dynamics throughout the simulation. This class of mesh adaptivity method can be seen as a generalisation of the classical *h*-method as it makes use of local changes to the mesh connectivity rather than hierarchical refinement and coarsening.

The local operations employed in three dimensions are:

- Edge split: a node is inserted at the centre of an edge and surrounding elements created (Figure 1).
- Edge collapse: all elements belonging to the edge are deleted and the two nodes of the edge collapsed to its centre (Figure 1), in cases where one of the nodes is used to define the domain geometry the nodes are collapsed to that point, if both nodes define some geometrical structure then this operation is not permitted.
- Edge-face swap: if two tetrahedra share a common face, and provided their combined interior is convex, the face is deleted and a new edge introduced between the two nodes not shared thus producing three tetrahedra with different alignment (Figure 2). The inverse operation where an edge is replaced by a face is also allowed.

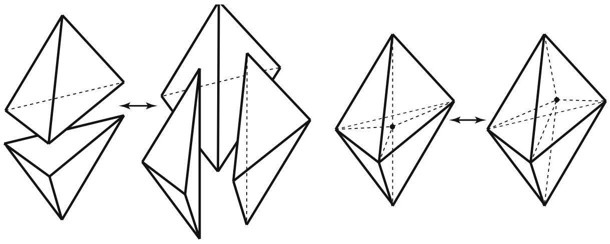

Figure 2: An example of the face to edge and edge to face swapping operation (left) and node movement (right) in 3D.

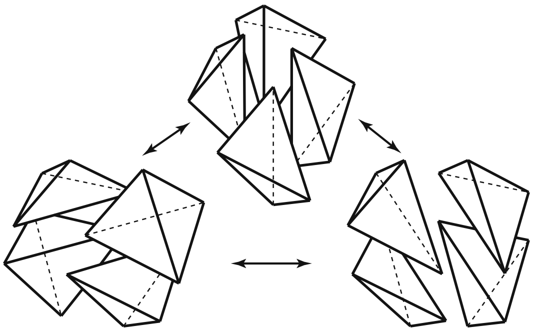

Figure 3: An example of the topological operation of edge to edge swapping in 3D.

- Edge-edge swap: S elements are assumed to lie around an edge which may be replaced by a different edge resulting in $2S - 4$ elements with different alignment (Figure 3).
- Limited node movement: the local topology of the mesh is preserved but mesh quality is improved by visiting each node and moving it to the centroid of all surrounding nodes, i.e. ensuring that when measured in metric space the lengths of all edges attached to this node are approximately equal (Figure 2).

Approaches based on these operations are often referred to as mesh optimisation methods since they involve the definition of some objective function, usually a norm over the entire mesh of a local mesh quality estimate. Optimisation of the mesh progresses by improving the worst elements through a series of elemental modifications to the mesh in an attempt to minimise the functional and hence improve the overall quality of the mesh. Dynamic mesh optimisation is achieved through the construction of metric with which the objective functional is evaluated and which is itself dependent on the computed solution fields [26–30]. This allows local anisotropic features in a solution to be made isotropic in a warped domain through a coordinate transformation. Mesh optimization is a subclass of adaptive remeshing methods. Adaptive remeshing [31] is characterized by the absence of any necessary relationship between the previous and adapted meshes; this is in contrast to hierarchical refinement methods, where the adapted mesh is intimately related to and constrained by the previous mesh. Mesh optimisation proceeds by identifying elements of the mesh which do not satisfy the desired sizing and orientation criteria, by means of an element quality functional. The worst elements are then improved by means of a series of local operations until the mesh satisfies the input specification. See [26] for a detailed discussion of the mesh quality functional and local optimisation operations used in this work.

The desired output mesh is specified by means of a symmetric positive-definite metric tensor field (*i.e.* the discrete form of a non-Euclidean space), which describes the ideal element sizing and orientation at every point in space. This metric is usually derived from consideration of interpolation error. In one dimension, the interpolation error over a linear element is bounded by the second derivative of the function being interpolated. This result carries over to higher dimensions, with the second derivative replaced by the Hessian of the function being interpolated, the tensor field of second partial derivatives [32]. The Hessian describes the curvature of the solution at each point in every direction; using the Hessian for the error metric to guide anisotropic mesh adaptivity places resolution where the solution field curves. For elliptic problems, Céa's lemma shows that the interpolation error bounds the discretisation error in the

Table 1: Quantifying the error in the solutions on different meshes. The second column records the L^2-norm of the error in the spatial variation of the solutions compared with a "gold standard" solution on a mesh containing 3.8 million nodes at a representative time step when the wavefront is approximately midway across the cube (2 ms). The third column records the L^2-norm of the error in the APs at the centre of the cube during the first 5 ms of simulation compared with the same gold standard solution.

Mesh	Spatial error	Temporal error
1,611 nodes	0.5935	1.1931
11,048 nodes	0.2541	0.4847
81,374 nodes	0.0618	0.1143
Adaptive	0.0688	0.1225

energy norm; in practice, minimising the interpolation error in this manner has been found to be highly effective at controlling the discretisation error, even for non-elliptic problems.

For a mesh of tetrahedra in 3D, assume a sufficiently smooth function $u \equiv u(\boldsymbol{x})$ is approximated by its piecewise-linear Lagrange interpolant $\Pi_h u$ on element $\triangle$. The interpolation error on element $\triangle$ satisfies

$$\epsilon_u \equiv ||u - \Pi_h u||_{\infty,\triangle} \leq c \cdot \max_{\boldsymbol{x}\in\triangle} \max_{\boldsymbol{v}\subset\triangle} \boldsymbol{v}^T |H| \boldsymbol{v} \leq c \cdot \max_{\boldsymbol{e}\in E_\triangle} \boldsymbol{e}^T |\bar{H}| \boldsymbol{e}, \tag{4}$$

where c is a constant independent of the mesh, $\boldsymbol{v}$ is a vector in $\triangle$, $E_\triangle$ is the set of edges of $\triangle$, $||\cdot||_{\infty,\triangle}$ is the infinity norm over $\triangle$, and $\bar{H}$ is an element-valued Hessian defined such that (4) holds [32]. $|H|$ denotes the positive definite metric formed by taking the absolute value of the eigenvalues of H and reflects the fact that it is the magnitude of the curvature that is of interest, rather than the sign of the curvature.

In this work, the metric was designed to resolve efficiently the transmembrane potential field, V_m, as it propagates through the heart. In general, any number of metric tensor fields can be combined through a superposition procedure so that a single metric tensor field respects the edge length and requirements for controlling the error in each field, see [26, 33, 34].

The adaptive algorithm employed below applies interpolation by the basis functions to transfer data between the previous and adapted meshes. Other approaches such as Galerkin projection [23, 24] are possible: Galerkin projection is conservative, optimally accurate, and can be made bounded, but these features are not of first-order importance in this study.

The adaptive library used was provided by the Applied Modelling and Computational Group, Imperial College London. The serial adaptive code is written in Fortran 77, and is wrapped with a C++ interface for use with third party applications such as Chaste. The adaptive library is also parallelised; however work interfacing parallel adaptivity with parallel Chaste was not completed in time for inclusion here and so only serial results are presented.

5. Results

In evaluating the performance of the adaptive method, two benchmark cases were considered. A simple cubic geometry was used to determine the maximum potential speedup that might be available where no allowance needs to be made for complex geometries, and a realistic heart geometry was used to quantify the impact of requiring additional mesh nodes to be retained on the boundary in order to encapsulate the curved surface. In both cases the non-adaptive simulations were run using the standard open-source release of Chaste [3]. Matrix assembly and solves were performed using the same Chaste data structures and routines as for the non-adaptive case. The benchmarks were run sequentially on a single node of a Linux cluster at Fujitsu Laboratories of Europe, consisting of a dual-core Xeon 5160 processor (3.0 GHz).

5.1. Slab Benchmark

The results of simulating of 400 ms of bidomain activity with a Luo-Rudy I cell model on a simple 2 mm cube of cardiac tissue are shown in Figure 4. The face $x = 0$ is stimulated at $t = 0$ and again at $t = 375$ ms in order to generate APs. Time steps of 0.005 and 0.01 ms were used for the ODE and PDE solves respectively. Fixed mesh simulations were run on meshes containing 81,374 nodes (average mesh spacing $\approx$ 0.06 mm), 11,048 nodes (average

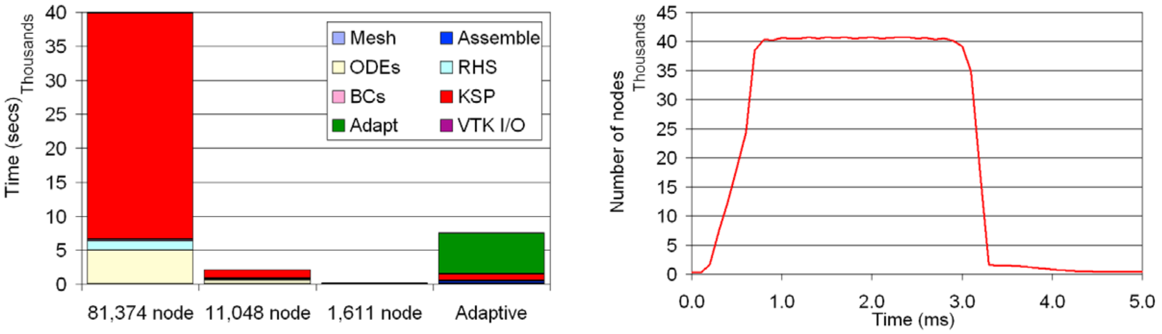

Figure 4: The computational load of simulating 400 ms of cardiac activity on a slab. On the left is shown the total computational time required using three fixed meshes and an adaptive mesh. On the right is the number of nodes in the adaptive mesh during the first 5 ms of the simulation.

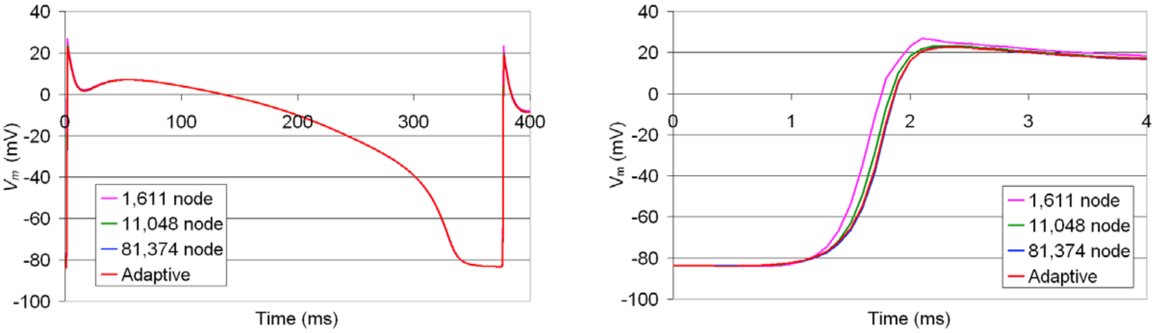

Figure 5: The action potentials at the point (0.1, 0.1, 0.1) for each mesh. The action potentials for the full duration of the simulation are shown on the left. On the right the differences in the four solutions are shown by zooming in on the AP upstroke (0–4 ms).

mesh spacing ≈ 0.12 mm) and 1,611 nodes (average mesh spacing ≈ 0.23 mm). For the adaptive simulations the mesh was adapted every 0.1 ms, based on the values of V_m. During each adapt, the maximum and minimum permitted edge lengths were 0.4 and 0.025 mm and a maximum of 100,000 nodes was imposed.

Using adaptivity reduces the computational time by a factor of 5.4 compared with the highest resolution fixed mesh. To determine the accuracy of the adaptive method a "gold standard" solution was also computed on a very fine mesh containing 608,687 nodes (resolution ≈ 0.03 mm). The L^2-norms of the difference in V_m between this solution and each of the others was computed at various time steps via the construction of an intermediate supermesh, as described in [35]. Further, the APs predicted at the mid-point of the cube for each simulation were computed (see Figure 5) and the L^2-norms of the errors in the first 5 ms (since by this time the upstroke is complete everywhere in the domain and the differences in solutions are much reduced) of these with respect to the "gold standard" solution AP were also computed (see Table 1). It can be seen that both the adaptive solution and that on the 81,734 node mesh are of approximately equivalent accuracy and that both are much more accurate than the solutions on the two coarser meshes. So, the speedup obtained by using an adaptive method does not come at the expense of accuracy. Further, it can be seen from Figure 5 that the adaptive simulation is able to resolve the second AP upstroke (at 375 ms), so coarsening the mesh during the plateau phase does not result in errors if subsequent stimuli are applied.

5.2. Heart Benchmark

Figure 6 shows the results of simulating of 400 ms of bidomain activity with a Luo-Rudy I cell model on a realistic heart geometry [2]. A volume stimulus of magnitude -5×10^3 mV is applied to the apex of the heart (nodes with $z < 0.04248$ are stimulated) for 0.5 ms starting at $t = 0$. Time steps of 0.005 and 0.01 ms were used for the ODE and PDE solves respectively. The non-adaptive simulations were run on meshes containing 4,310,704 nodes (equivalent

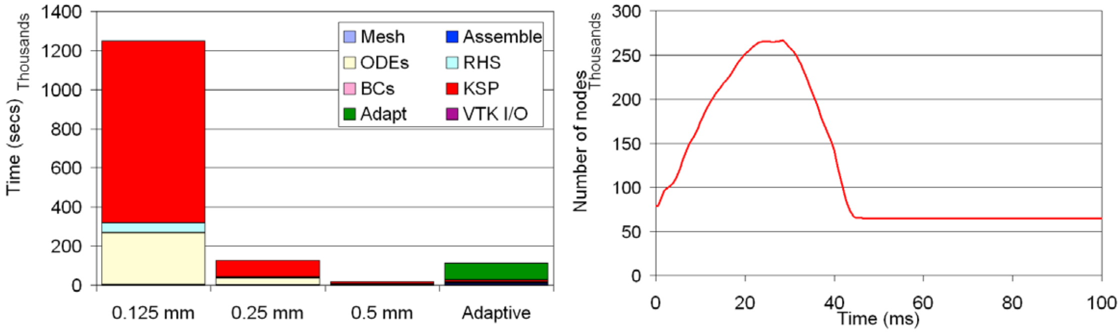

Figure 6: The computational load of simulating 400 ms of cardiac activity on the Oxford heart geometry. On the left is shown the total computational time required using three fixed meshes and an adaptive mesh. On the right is the number of nodes in the adaptive mesh during the first 100 ms of the simulation.

Table 2: Quantifying the error in the solutions on different meshes. The second column records the L^2-norm of the error in the spatial variation of the solutions compared with the solution on the adaptive mesh at a representative time step when the action potential wavefront is approximately in the centre of the heart (12.5 ms). The third column records the L^2-norm of the error in the APs at the point $(1.0, 1.0, 0.55)$ during the first 20 ms of simulation compared with the AP at the same point in the adaptive mesh.

Mesh spacing	Spatial error	Temporal error
0.500 mm	22.7498	0.3970
0.250 mm	22.4878	1.4415
0.125 mm	2.1022	0.1783

to an average mesh spacing of approximately 0.125 mm), 572,066 nodes (average mesh spacing ≈ 0.25 mm) and 82,619 nodes (average mesh spacing ≈ 0.5 mm). These three meshes are representations of the same geometry, but at different resolutions—with the two coarser meshes obtained by downsampling the finest. For the adaptive simulations the mesh was adapted every 50 PDE time steps (i.e. every 0.5 ms), based on the values of V_{m}. When adapting, the maximum and minimum permitted edge lengths were 5 and 0.1 mm respectively and a maximum of 600,000 nodes were permitted. In this case, the minimum permitted edge length is two-thirds of the average mesh spacing in the highest resolution fixed mesh. The mesh is adapted every 0.5 ms compared with every 0.1 ms for the slab benchmark. This is justified since the minimum permitted edge length is four times as large for the heart as in the slab and the tissue conductivities are the same in both benchmarks. So, the wavefront travels approximately as far relative to the length of the edges in its neighbourhood in 0.5 ms in the heart as it does in 0.1 ms in the slab.

A speedup by a factor of 11.2 is obtained by using adaptivity compared with using the finest fixed mesh (and the adaptive method is slightly faster even than using the medium resolution fixed mesh). The relative accuracies of the simulations computed on each of the meshes were estimated as for the slab by computing the L^2-norms of the differences between the solutions at a typical timestep and in the APs generated at the point $(1.0, 1.0, 0.55)$ on each mesh. Since no higher resolution heart mesh is available the errors for each of the fixed mesh simulations were compared with the adaptive solution. The results are shown in Table 2. It can be seen that the adaptive solution is very much closer to that on the fine mesh than it is to those on either coarse mesh. This indicates that there is no significant loss of accuracy incurred by adapting the mesh. In the worst case scenario, the adaptive solution is an order of magnitude more accurate than that on the medium resolution fixed mesh (compared with that on the fine mesh) — and the solution is computed faster on the adaptive mesh than the medium one.

6. Discussion

In both benchmarks a significant speedup has been obtained by utilizing adaptive methods, with no loss of accuracy. Perhaps surprisingly, the speedup factor is greater for the heart (11.2) than for the slab (5.4), despite the requirement to retain otherwise superfluous nodes to represent the curved surface geometry in regions away from the wavefront. This increased speedup is due to the much larger size of the heart (approximately 23 × 21 × 29 mm) compared with the slab (0.2 mm in each dimension), the wavefront occupies a much smaller proportion of the volume of the heart than in the slab, allowing coarsening compared with the fixed mesh in a much larger region. For the heart, the maximum number of nodes in the adaptive mesh is 266,499, or 6% of the number in the highest resolution fixed mesh (Figure 6), whereas the equivalent proportion is 50% for the slab (Figure 4). As expected, this is offset by the more complex geometry of the heart, which requires more mesh nodes to be retained on the surface of the heart even during the plateau phase. Thus, the minimum number of nodes in the adaptive heart mesh is 65,011, which is equivalent to 1.51% of the number in the highest resolution fixed mesh, compared with just 0.45% for the slab. Note, however, that the balance between these two factors is critically dependent on the initial geometrical representation of the surface of the heart. In the results presented here, the coarse representation of the surface was used and the mesh refined. Starting with the finest representation and trying to coarsen would require more boundary nodes and may greatly reduce the overall speedup.

The increased speed of the adaptive simulations results primarily from reducing the computational load of solving the linear system resulting from the discretization of the PDE part of the bidomain equations. These linear solves dominate the non-adaptive simulations (75% of computational load for the finest fixed heart mesh and 81% for the finest fixed slab mesh), but account for only 12% of the load in both adaptive cases (a speedup factor of 36 for the slab and 70 for the heart). The greatly reduced load is a direct consequence of the reduction in the number of mesh nodes. The ODE solve and RHS vector assembly parts of the load (20% and 4% respectively of the total for the finest fixed heart mesh: the two largest contributions after the linear solve) are also greatly reduced (speedup factors of 57 and 53 respectively for the heart and 89 and 105 for the slab) in the adaptive simulations as they also depend on the number of mesh nodes. These values suggest that if no additional expenses were incurred in adapting the mesh then an overall speedup of more than 50 could be possible on the heart mesh and close to 40 for the slab.

However, speedups of this magnitude are not seen in this work since the computational cost of repeatedly adapting the mesh dominates the adaptive simulations, accounting for 79% of the simulation time for the slab and 74% for the heart. Further, it is also necessary to reassemble the system matrix at each time step in the adaptive simulations, leading to an increase in the expense of this part of the code from well below 1% in the fixed mesh cases to 6% when adaptivity is used. This leads to two conclusions: reducing the cost of adapting the mesh is likely to lead to greater reductions in the computational cost of solving the bidomain equations than improving solution methods for the linear system or ODEs; and a more systematic way of determining when it is necessary to adapt should be investigated, rather than simply adapting every *n* steps (to stop, for example, unnecessary adapts during the plateau phase incurring additional costs). This is likely to be particularly significant when extending the adaptive method to simulations of re-entrant arrhythmia (when there is no plateau phase, so it is likely that more mesh nodes will be required throughout the simulation) in. While both Chaste and the adaptivity library have been parallelised and been shown to scale well to at least the order of 1000 processes; work on parallelizing the interface between Chaste and the adaptivity library was not completed for the results presented here. This work is expected to be published in the near future. Finally, research to determine the most effective way of representing the surface geometry away from a wavefront will also be carried out in an attempt to try to reduce the number of nodes used to encapsulate curved surfaces without reducing the accuracy of the resulting simulations.

Acknowledgements

The work of J.S. and M.O.B., part of the preDiCT project, was supported by a grant from the European Commission DG-INFSO – grant number 224381. G.J.G. gratefully acknowledges the support of the Leverhulme Trust Early Career Fellowship. M.D.P. gratefully acknowledges the support of the Grantham Insitute at Imperial College. P.E.F. would like to thank AWE for their funding of his research through the Institute of Shock Physics at Imperial College. P. E. Farrell would like to thank AWE for their funding of his research through the Institute of Shock Physics.

References

[1] P. J. Hunter, T. K. Borg, Integration from proteins to organs: the Physiome Project, Nat. Rev. 4 (2003) 237–243.

[2] G. Plank, R. A. B. Burton, P. Hales, M. Bishop, T. Mansoori, M. O. Bernabeu, A. Garny, A. J. Prassl, C. Bollensdorff, F. Mason, F. Mahmood, B. Rodriguez, V. Grau, J. E. Schneider, D. Gavaghan, P. Kohl, Generation of histo-anatomically representative models of the individual heart: tools and applications, Phil. Trans. Roy. Soc. A 367 (2009) 2257–2292.

[3] J. Pitt-Francis, P. Pathmanathan, M. O. Bernabeu, R. Bordas, J. Cooper, A. G. Fletcher, G. R. Mirams, P. Murray, J. M. Osborne, A. Walter, S. J. Chapman, A. Garny, I. M. M. van Leeuwen, P. K. Maini, B. Rodriguez, S. L. Waters, J. P. Whiteley, H. M. Byrne, D. J. Gavaghan, Chaste: a test-driven approach to software development for biological modelling, Comp. Phys. Comm. 180 (2009) 2452–2471.

[4] J. P. Keener, J. Sneyd, Mathematical Physiology, Springer, New York, 1998, ch. 11.

[5] C. Luo, Y. Rudy, A model of the ventricular cardiac action potential. Depolarization, repolarization, and their interaction, Circ. Res. 68 (1991) 1501–1526.

[6] D. M. Harrild, C. S. Henriquez, A finite volume model of cardiac propagation, Ann. Biomed. Eng. 25 (1997) 315–334.

[7] J. P. Keener, K. Bogar, A numerical method for the solution of the bidomain equations in cardiac tissue, Chaos 8 (1998) 234–241.

[8] E. J. Vigmond, F. Aguel, N. A. Trayanova, Computational techniques for solving the bidomain equations in three dimensions, IEEE Trans. Biomed. Eng. 49 (2002) 1260–1269.

[9] E. Vigmond, M. Hughes, G. Plank, L. Leon, Computational tools for modeling electrical activation in cardiac tissues, J. Electrocardiol. 36 (Suppl.) (2003) 69–74.

[10] M. L. Trew, I. LeGrice, B. H. Smaill, A. Pullan, A finite volume method for modeling discontinuous electrical activation in cardiac tissue, Ann. Biomed. Eng. 33 (2005) 590–602.

[11] M. L. Trew, B. H. Smaill, D. P. Bullivant, P. J. Hunter, A. J. Pullan, A generalized finite difference method for modeling cardiac electrical activation on arbitrary, irregular computational meshes, Math. Biosci. 198 (2005) 169–189.

[12] E. J. Vigmond, R. Weber dos Santos, A. J. Prassl, M. Deo, G. Plank, Solvers for the cardiac bidomain equations, Prog. Biophys. Mol. Biol. 96 (2007) 3–18.

[13] J. Pitt-Francis, M. O. Bernabeu, J. Cooper, A. Garny, L. Momtahan, J. Osborne, P. Pathmanathan, B. Rodriguez, J. P. Whiteley, D. J. Gavaghan, Chaste: using agile programming techniques to develop computational biology software, Phil. Trans. Roy. Soc. A 366 (2008) 3111–3136.

[14] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, B. F. Smith, H. Zhang, PETSc users manual, Tech. Rep. ANL-95/11 – Revision 2.3.2, Argonne National Laboratory (2006).

[15] E. M. Cherry, H. S. Greenside, C. S. Henriquez, A space-time adaptive method for simulating complex cardiac dynamics, Phys. Rev. Lett. 84 (2000) 1343–1346.

[16] E. M. Cherry, H. S. Greenside, C. S. Henriquez, Efficient simulation of three-dimensional anisotropic cardiac tissue using an adaptive mesh refinement method, Chaos 13 (2003) 853.

[17] J. A. Trangenstein, C. Kim, Operator splitting and adaptive mesh refinement for the Luo–Rudy I model, J. Comp. Phys. 196 (2004) 645–679.

[18] J. P. Whiteley, Physiology driven adaptivity for the numerical solution of the bidomain equations, Ann. Biomed. Eng. 35 (2007) 1510–1520.

[19] P. Colli Franzone, P. Deuflhard, B. Erdmann, J. Lang, L. F. Pavarino, Adaptivity in space and time for reaction-diffusion systems in electrocardiology, SIAM J. Sci. Comput. 28 (2006) 942–962.

[20] P. Deuflhard, B. Erdmann, R. Roitzsch, G. T. Lines, Adaptive finite element simulation of ventricular fibrillation dynamics, Comput. Vis. Sci. 12 (2009) 201–205.

[21] M. Weiser, B. Erdmann, P. Deuflhard, On efficiency and accuracy in cardioelectric simulation, Tech. rep., Konrad-Zuse-Zentrum fur Informationstechnik Berlin, Berlin, Germany, ZIB-Report 08-41 (2008).

[22] Y. Belhamadia, A time-dependent adaptive remeshing for electrical waves of the heart, IEEE Trans. Biomed. Eng. 55 (2) (2008) 443–452.

[23] P. E. Farrell, M. D. Piggott, C. C. Pain, G. J. Gorman, Conservative interpolation between unstructured meshes via supermesh construction, Comput. Methods Appl. Mech. Eng. 198 (2009) 2632–2642.

[24] P. E. Farrell, J. R. Maddison, Interpolation between discontinuous volume meshes by local Galerkin projection, Comput. Methods Appl. Mech. Eng., in review (2009).

[25] Y. Belhamadia, A. Fortin, Y. Bourgault, Towards accurate numerical methods for monodomain models using a realistic heart geometry, Math. Biosci. 220 (2009) 89–101. doi:10.1016/j.mbs.2009.05.003.

[26] C. Pain, A. Umpleby, C. de Oliveira, A. Goddard, Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations, Comput. Methods Appl. Mech. Eng. 190 (2001) 3771–3796.

[27] R. B. Simpson, Anisotropic mesh transformations and optimal error control, Applied Numerical Mathematics 14 (1994) 183–198.

[28] G. C. Buscaglia, E. A. Dari, Anisotropic mesh optimization and its applications in adaptivity, Int. J. Num. Meth. Eng. 40 (1997) 4119–4136.

[29] M. D. Piggott, C. C. Pain, G. J. Gorman, P. W. Power, A. J. H. Goddard, *h*, *r*, and *hr* adaptivity with applications in numerical ocean modelling, Ocean Modelling 10 (2005) 95–113.

[30] P. W. Power, C. C. Pain, M. D. Piggott, G. J. Gorman, D. P. Marshall, A. J. H. Goddard, I. M. Navon, Adjoint goal-based error norms for adaptive mesh ocean modelling, Ocean Modelling 15 (2006) 3–38.

[31] J. Peraire, M. Vahdati, K. Morgan, O. Zienkiewicz, Adaptive remeshing for compressible flow computations, J. Comp. Phys. 72 (1987) 449–466.

[32] P. J. Frey, F. Alauzet, Anisotropic mesh adaptation for CFD computations, Comput. Methods Appl. Mech Eng. 194 (2005) 5068–5082.

[33] M. J. Castro-Diaz, F. Hecht, B. Mohammadi, O. Pironneau, Anisotropic unstructured mesh adaptation for flow simulations, Int. J. Num. Meth. Fluids 25 (1997) 475–491.

[34] P. L. George, H. Borouchaki, Delaunay Triangulation and Meshing: Application to Finite Elements, Hermes, 1998.

[35] P. E. Farrell, Galerkin projection of discrete fields via supermesh construction, Ph.D. thesis, Imperial College London (2009).