



Report Number 09/07

**Solving the Coupled System Improves  
Computational Efficiency of the Bidomain  
Equations**

by

**James A. Southern, Gernot Plank, Edward J.  
Vigmond, Jonathan P. Whiteley**



Oxford Centre for Collaborative Applied Mathematics  
Mathematical Institute  
24 - 29 St Giles'  
Oxford  
OX1 3LB  
England



# Solving the Coupled System Improves Computational Efficiency of the Bidomain Equations

James A. Southern\*, Gernot Plank, Edward J. Vigmond,  
and Jonathan P. Whiteley

## Abstract

The bidomain equations are frequently used to model the propagation of cardiac action potentials across cardiac tissue. At the whole organ level the size of the computational mesh required makes their solution a significant computational challenge. As the accuracy of the numerical solution cannot be compromised, efficiency of the solution technique is important to ensure that the results of the simulation can be obtained in a reasonable time whilst still encapsulating the complexities of the system.

In an attempt to increase efficiency of the solver, the bidomain equations are often decoupled into one parabolic equation that is computationally very cheap to solve and an elliptic equation that is much more expensive to solve. In this study the performance of this uncoupled solution method is compared with an alternative strategy in which the bidomain equations are solved as a coupled system. This seems counter-intuitive as the alternative method requires the solution of a much larger linear system at each time step. However, in tests on two 3-D rabbit ventricle benchmarks it is shown that the coupled method is up to 80% faster than the conventional uncoupled method — and that parallel performance is better for the larger coupled problem.

## Index Terms

*Asterisk indicates corresponding author.*

\*J. A. Southern is with Fujitsu Laboratories of Europe Ltd, Hayes, Middlesex, UB4 8FE, UK (email: James.Southern@uk.fujitsu.com)

G. Plank and J. P. Whiteley are with the Computing Laboratory, University of Oxford, Oxford, OX1 3QD, UK

E. J. Vigmond is with the Department of Electrical and Computer Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada

Bidomain model, cardiac simulation, finite element methods, operator splitting, parallel computing.

## I. INTRODUCTION

The bidomain equations [1]–[3] have been widely used for many years to model the propagation of electrical waves across cardiac tissue. A system of two partial differential equations (PDEs), they model the tissue as a homogenized two-phase material (intra- and extra-cellular space), with the potential difference between the two phases at a given point driving a capacitive current across the cell membrane that separates these spaces. An ionic current also flows across the cell membrane. This ionic current is calculated using an electrophysiological cell model that is usually described by a system of ordinary differential equations (ODEs).

The numerical solution of the bidomain equations is usually calculated using standard finite element (FE), finite volume (FV) or finite difference (FD) methods and there are many different numerical schemes that can be used to do this [4]–[11]. However, as the discretization of a whole heart with an average nodal spacing of  $250\ \mu\text{m}$  generates a mesh with many millions of nodes, the linear systems resulting from FE, FV or FD methods are very large. Whole heart simulation using the bidomain model is therefore a non-trivial scientific computing problem. It is desirable to choose a numerical method that gives the required accuracy as efficiently as possible.

The need for relatively fine meshes, however, is driven by detailed anatomical and physiological considerations and this inevitably leads to a high computational cost. For example, the simulation of a single second of cardiac activity on a rabbit ventricular mesh consisting of 862,515 nodes using the CARP simulator [7] and a modified Beeler-Reuter cardiac cell model [12], [13] requires approximately 40 hours of CPU time on 64 nodes of the HPCx supercomputer [14]. The equivalent simulation on a whole human heart, using the same mesh spacing, would necessitate a grid containing 30 million nodes and the simulation would take around 1400 hours — about seven weeks — assuming (somewhat optimistically) that the increase in run-time would scale linearly with the problem size. This predicted time frame makes the use of whole heart simulations to investigate arrhythmias or the effect of new drugs infeasible using current algorithms and technology.

In this study we show that the bidomain equations can be solved more efficiently as a coupled system than by decoupling the extra-cellular and transmembrane potentials. The coupled method

requires the solution of a much larger linear system than the uncoupled method, but the iterative solvers converge in many fewer iterations resulting in an overall speedup of between 50% and 80% for the same level of accuracy. The parallel scaling of the coupled and uncoupled methods are also considered as future whole heart simulations are very likely to require the use of massively parallel computing facilities in addition to improved algorithms. We also show that using the coupled method allows for the boundary conditions on the system to be applied more rigorously than is the case for the uncoupled system (since one boundary condition depends on both the transmembrane and extra-cellular potentials and cannot be decoupled).

## II. METHODS AND MODELS

### A. The Bidomain Model

The bidomain equations are given in [3] as:

$$\chi \left( C_m \frac{\partial V_m}{\partial t} + I_{\text{ion}}(\mathbf{u}, V_m) \right) - \nabla \cdot (\sigma_i \nabla (V_m + \phi_e)) = 0, \quad (1)$$

$$\nabla \cdot ((\sigma_i + \sigma_e) \nabla \phi_e + \sigma_i \nabla V_m) = 0, \quad (2)$$

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}, V_m), \quad (3)$$

where  $V_m(\mathbf{x}, t)$  is the transmembrane potential,  $\phi_e(\mathbf{x}, t)$  is the extra-cellular potential,  $\mathbf{u}$  is a vector of dependent variables,  $I_{\text{ion}}$  is the total ionic current across the cell membrane,  $\chi$  is the cell surface to volume ratio,  $C_m$  is the membrane capacitance,  $\sigma_i$  and  $\sigma_e$  are the intra-cellular and extra-cellular conductivity tensors respectively,  $\mathbf{f}$  is a vector-valued function,  $\mathbf{x}$  is position and  $t$  is time. The function  $\mathbf{f}$ , the components of  $\mathbf{u}$ , and  $I_{\text{ion}}$  are determined by an ODE model of a cardiac cell.

In order for the bidomain equations to be well-posed it is also necessary to impose approximate initial and boundary conditions. For the simulations described in this paper the simulated heart is surrounded by a conducting bath of fluid with electrodes used to stimulate parts of the surface of the bath. The bath is modelled as an extension of the extra-cellular space, so  $\phi_e$  and the extra-cellular current  $i_e = -\sigma_e \nabla \phi_e$  should be continuous across the interface between heart and bath. The FE method implemented below will enforce these conditions automatically. Since there is no intra-cellular space in the bath  $V_m$  is not defined in this region and there is no flow of

intra-cellular current across the interface, i.e.

$$\mathbf{n} \cdot (\sigma_i \nabla (V_m + \phi_e)) = 0 \quad \text{on the surface of the heart,} \quad (4)$$

where  $\mathbf{n}$  is the unit outward-pointing normal (i.e. from the heart into the bath) to the surface. Finally, it is necessary to specify a boundary condition on  $\phi_e$  on the outer surface of the bath. Away from the electrodes it is assumed that no current crosses the boundary of the bath:

$$\mathbf{n} \cdot (\sigma_e \nabla \phi_e) = 0 \quad \text{away from the electrodes.} \quad (5)$$

One of the electrodes is grounded, so the correct boundary condition on this electrode is

$$\phi_e = 0 \quad \text{on the ground electrode.} \quad (6)$$

Current is injected through the second electrode, so that:

$$\nabla \cdot (\sigma_e \nabla \phi_e) = -I_e(t) \quad \text{on the stimulation electrode,} \quad (7)$$

where  $I_e(t)$  is the external stimulus current applied (which, in general, is zero except for short periods of stimulation).

## B. Solution Methods

Two families of numerical algorithms for solving the bidomain equations using FE methods are considered in this work. One is very typical of the way the bidomain equations are solved in most simulation software: by decoupling the PDEs, Eqns. (1) and (2), for  $\phi_e$  and  $V_m$  and solving separately. The motivation behind this strategy is to increase efficiency by reducing the size of the linear systems to be solved. Note also that, historically, reducing the size of the linear system size could be equally important, or more important, than increasing efficiency: when Cray T3D's only had 16 or 32 MB of memory per node the methods had to be adjusted to accommodate the hardware, even at the expense of performance. The second method leaves the two PDEs coupled. This means that the system is larger than for the decoupled method. This idea of solving larger systems to speed up the simulation seems counter-intuitive. Nevertheless, we shall demonstrate that it is, in fact, more efficient than the decoupled scheme.

1) *Operator Splitting and Explicit Time Stepping*: In the first solution method the bidomain equations (Eqns. (1), (2)) are re-written as follows [3], [15]:

$$\frac{\partial V_m}{\partial t} = \frac{\nabla \cdot (\sigma_i \nabla (V_m + \phi_e))}{\chi C_m} - \frac{I_{\text{ion}}(\mathbf{u}, V_m)}{C_m}, \quad (8)$$

$$\nabla \cdot ((\sigma_i + \sigma_e) \nabla \phi_e) = -\sigma_i \nabla V_m, \quad (9)$$

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}, V_m). \quad (10)$$

In this formulation, the first of these equations can be considered as a parabolic PDE for  $V_m$  (assuming a known  $\phi_e$ ), the second considered as an elliptic PDE for  $\phi_e$  (assuming a known  $V_m$ ) and the third remains a system of ODEs.

On discretizing in time using the forward Euler method the following system is obtained at each mesh node:

$$\frac{V_m^{k+1} - V_m^k}{\Delta t} = \frac{\nabla \cdot (\sigma_i \nabla (V_m^k + \phi_e^k))}{\chi C_m} - \frac{I_{\text{ion}}(\mathbf{u}^k, V_m^k)}{C_m}, \quad (11)$$

$$\nabla \cdot ((\sigma_i + \sigma_e) \nabla \phi_e^{k+1}) = -\sigma_i \nabla V_m^{k+1}, \quad (12)$$

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\Delta t} = \mathbf{f}(\mathbf{u}^k, V_m^{k+1}), \quad (13)$$

where  $\Delta t$  is the time step and  $V_m^k$ ,  $\phi_e^k$  and  $\mathbf{u}^k$  are the discretizations in time of  $V_m$ ,  $\phi_e$  and  $\mathbf{u}$  respectively at time  $k\Delta t$ . Note that on using this formulation it is possible to solve first for  $V_m^{k+1}$  with no knowledge of the numerical approximation for  $\phi_e$  at the new time step ( $k+1$ ) and then solve for  $\phi_e^{k+1}$  using the solution that has already been computed for  $V_m^{k+1}$ , i.e. the equations are now decoupled at each time step.

The uncoupled equations are solved at each time step via the following lumped mass method [7], [14]:

$$\mathbf{V}^{k+1} = (I + \Delta t A_i) \mathbf{V}^k + \Delta t A_i \Phi^k - \frac{\Delta t}{C_m} \mathbf{I}_{\text{ion}}(\mathbf{u}^k, \mathbf{V}^k), \quad (14)$$

$$(A_i + A_e) \Phi^{k+1} = -A_i \mathbf{V}^{k+1} - \frac{1}{\chi C_m} \mathbf{I}_e^{k+1}, \quad (15)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \Delta t \mathbf{f}(\mathbf{u}^k, \mathbf{V}^{k+1}), \quad (16)$$

where  $\mathbf{V}^k$ ,  $\Phi^k$ ,  $\mathbf{u}^k$ , and  $\mathbf{I}_e^k$  are the spatial discretizations of  $V_m$ ,  $\phi_e$ ,  $\mathbf{u}$ , and  $I_e$  respectively at time  $k\Delta t$ ,  $\mathbf{I}_{\text{ion}}$  is the vector-valued equivalent of the scalar function  $I_{\text{ion}}$ ,  $A_\xi$ ,  $\xi = i, e$ , are the

discretized versions of the  $\nabla \cdot (\sigma_\xi \nabla) / (\chi C_m)$  operators and  $I$  is the identity matrix. The entries of  $\mathbf{I}_e^k$  are always zero for nodes that are not located on the stimulating electrode and this vector is used solely to impose the current injection boundary condition given by Eqn. (7).

This method is conditionally stable, with the maximum time step for which it is stable depending on the size of the *smallest* element in the FE mesh [11].

Note that since the equations for  $V_m$  and  $\phi_e$  have been decoupled it is not possible to impose the boundary condition given in Eqn. (4) exactly when solving Eqn. (11) — this boundary condition couples both  $V_m^{k+1}$  and  $\phi_e^{k+1}$ . The best that can be done is to set

$$\mathbf{n} \cdot (\sigma_i \nabla \mathbf{V}_m^{k+1}) = -\mathbf{n} \cdot (\sigma_i \nabla \Phi_e^k), \quad (17)$$

on the surface of the heart, where here  $\nabla$  represents the discretized gradient operator, and observe that since stability requirements mean that  $\Delta t$  must be chosen to be very small for an explicit scheme this is unlikely to undermine the numerical accuracy critically [11].

Solving the bidomain equations via Eqns. (14) – (16) is the default technique used in the CARP simulator . This simulator was used to generate the results of the uncoupled explicit system presented in this work.

2) *Operator Splitting and the Crank-Nicolson Method:* An alternative to solving the decoupled method using the forward Euler method is to use the second-order accurate Crank-Nicolson method for time-stepping for  $V_m$  in the PDEs (Eqns. (15) and (16) are still used for  $\phi_e$  and to solve the system of ODEs given by Eqn. (3) respectively). On discretizing Eqns. (8) and (10) in time using this method the system is the same as the explicit time stepping but with Eqn. (11) replaced with

$$\frac{V_m^{k+1} - V_m^k}{\Delta t} = \frac{\nabla \cdot (\sigma_i \nabla (\frac{1}{2}(V_m^{k+1} + V_m^k) + \phi_e^k))}{\chi C_m} - \frac{I_{\text{ion}}(\mathbf{u}^k, V_m^k)}{C_m}. \quad (18)$$

Using this time discretization gives rise to the following equation to update  $\mathbf{V}$ :

$$(I - \frac{1}{2}\Delta t A_i) \mathbf{V}^{k+1} = (I + \frac{1}{2}\Delta t A_i) \mathbf{V}^k + \Delta t A_e \Phi^k - \frac{\Delta t}{C_m} \mathbf{I}_{\text{ion}}(\mathbf{u}^k, \mathbf{V}^k), \quad (19)$$

where again a lumped mass matrix is used.

The slight extra complexity of implementing this scheme is justified for the stability properties that it offers [11]. As with the explicit scheme, the solution of the bidomain equations using Eqns. (12), (13), (19) is implemented as a user-specified option in the CARP simulator. This implementation was used to generate the results of the Crank-Nicolson scheme presented below.

As for the explicit method, the decoupled nature of the Crank-Nicolson scheme means that the boundary condition defined by Eqn. (4) on the surface of the heart cannot be imposed exactly, since again this would require recoupling the solution for  $\mathbf{V}^{k+1}$  and  $\Phi^{k+1}$ . It is shown in [11] that the natural boundary conditions for the FE solution of Eqns. (12), (13), (19) is a good numerical approximation of the exact boundary conditions, so this will not affect the accuracy of the solution. What is less clear, however, is whether by solving alternately for  $\mathbf{V}^{k+1}$  and  $\Phi^{k+1}$  results in a slower convergence to the solution than would be possible if the boundary condition could be applied to both  $\mathbf{V}^{k+1}$  and  $\Phi^{k+1}$  simultaneously.

3) *Coupled Solve with Semi-Implicit Time Stepping*: An alternative to the method described above is to solve the two bidomain Eqns. (1), (2) as a coupled system [16], [17]. The system of ODEs is still decoupled from the PDEs. Then, discretizing in time using the forward Euler method for the ODEs and the implicit backward Euler method for the PDEs (except for the  $I_{\text{ion}}$  term, which is treated explicitly to preserve linearity of the discretized system) gives:

$$\chi \left( C_m \frac{V_m^{k+1} - V_m^k}{\Delta t} + I_{\text{ion}}(\mathbf{u}^k, V_m^k) \right) - \nabla \cdot (\sigma_i \nabla (V_m^{k+1} + \phi_e^{k+1})) = 0, \quad (20)$$

$$\nabla \cdot ((\sigma_i + \sigma_e) \nabla \phi_e^{k+1} + \sigma_i \nabla V_m^{k+1}) = 0, \quad (21)$$

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\Delta t} = \mathbf{f}(\mathbf{u}^k, V_m^{k+1}), \quad (22)$$

where the variables are identical to those used in the time-discretized version of the uncoupled system.

Then using a lumped mass scheme (equivalent to those for the operator splitting methods) results in the following method:

$$\begin{pmatrix} A_i - \frac{1}{\Delta t} I & A_i \\ A_i & A_i + A_e \end{pmatrix} \begin{pmatrix} \mathbf{V}^{k+1} \\ \Phi^{k+1} \end{pmatrix} = \mathbf{b}^k, \quad (23)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \Delta t \mathbf{f}(\mathbf{u}^k, \mathbf{V}^{k+1}), \quad (24)$$

where

$$\mathbf{b}^k = \begin{pmatrix} -\frac{1}{\Delta t} \mathbf{V}^k + \frac{1}{C_m} \mathbf{I}_{\text{ion}}(\mathbf{u}^k, \mathbf{V}^k) \\ -\frac{1}{\chi C_m} \mathbf{I}_e^{k+1} \end{pmatrix}, \quad (25)$$

and all other variables are identical to those used with operator splitting. Since the equations for  $V_m$  and  $\phi_e$  remain coupled in this scheme it is possible to impose the boundary condition given by Eqn. (4) for the surface of the heart. In fact, it turns out that this is a ‘natural’ boundary condition for the FE scheme, so no additional terms need to be added to Eqn. (24) in order to implement it.

The stability properties of the backward time derivative allow long time steps of the order of 0.1 ms to be used for a mesh with nodal spacing of 0.1 mm that would otherwise require much smaller time steps for stability [10]. This stability condition is certain to be less restrictive than that for the explicit uncoupled method. So, it is more likely that physiological requirements (rather than the need for stability) can be used to determine the size of the time step. This will be especially useful if adaptive techniques are to be used in future. For example, [18] describes simulations that use an adaptive FE method with a maximum timestep of 1 ms and no significant loss of accuracy. This is far larger than the timesteps that can be used for the explicit method.

The method is *semi*-implicit since the ionic current terms are treated explicitly but the conduction terms are treated implicitly. The ODE system resulting from the cardiac cell model is solved explicitly at each time step. As many cell models give rise to stiff systems of ODEs it would be better to use an implicit method. However, state-of-the-art cell models consist of large, coupled systems of nonlinear equations that are generally expensive to solve implicitly (although, see [10] for a more efficient implicit scheme for a cardiac cell model that could be used to make the coupled solve fully implicit).

In order to facilitate a direct comparison between the two methods, a solver for Eqns. (23) – (25) was implemented in CARP, using the same assemblers, solvers and mesh handlers as for the explicit, operator splitting method. This ensures that the differences in timings for the two methods are independent of their implementation and as accurate a reflection of their numerical efficiency as possible.

### C. Benchmarks

Two benchmarks were run for each of the solution methods. These benchmarks are identical to those used in [14] to compare the performance of various preconditioners within the CARP software package. In each benchmark the bidomain equations were solved over a period of 200 ms on a full rabbit ventricle mesh consisting of 862,515 extra-cellular and 547,680 intra-cellular nodes and based on published geometrical data [19] with smooth epicardial and endocardial surfaces and realistic fibre orientation. The data was discretized using an unstructured grid with an average spatial resolution of  $250 \mu\text{m}$  and linear tetrahedral elements. All simulations were performed with  $C_m = 1 \mu\text{F}/\text{cm}^2$ ,  $\chi = 1400 \text{ cm}^{-1}$ . Intra-cellular and extra-cellular conductivity were defined to be  $\sigma_{ii} = 1.74 \text{ mS}/\text{cm}$  and  $\sigma_{ei} = 6.25 \text{ mS}/\text{cm}$  respectively in the direction of the fibres,  $\sigma_{it} = 0.19 \text{ mS}/\text{cm}$  and  $\sigma_{et} = 2.36 \text{ mS}/\text{cm}$  respectively transverse to the fibres [20] and  $\sigma_b = 1.0 \text{ mS}/\text{cm}$  (isotropic) in the surrounding fluid. A time step of  $8 \mu\text{s}$  was used and the Puglisi rabbit ventricular cell model [21] with an additional electroporation current [22] and a hypothetical outward current that activates strongly at positive potentials (outside the physiological range) [23] used to model  $I_{ion}$ . These two currents help to reproduce some experimental observations that cannot be accounted for by the standard Puglisi model.

The rabbit ventricle is embedded in a cubic bath, with two electrodes (one stimulating, one grounding) located on opposite faces of the cube. The stimulating electrode is used to deliver a train of ten pulses of current at 200 ms intervals, each lasting for 5 ms and with  $I_e = 5 \times 10^5 \mu\text{A}/\text{cm}^2$ . After delivery of the final pulse the simulation continues and re-entry is observed. Two representative subsequences of this simulation are chosen to analyze the properties of the two solution methods both before and after the onset of re-entry. The first is a 200 ms interval beginning with the delivery of the first pulse and the second a 200 ms interval containing a figure-of-eight re-entry sequence following the end of the final pulse (i.e. 2000 ms after the start of the first pulse).

### D. Linear Solve

The PETSc library [24] was used both to precondition and to solve the linear systems of the general form

$$Ax = b \tag{26}$$

that result from each of the two solution methods. The iterative conjugate gradient (CG) solver [25] was used for both the decoupled Eqn. (15) and the coupled Eqn. (23). In order to speed up the iterative CG solver a preconditioner was applied to each of the two systems. It has been shown in [14] that the BoomerAMG (BAMG) preconditioner [26] works very well in combination with the CG method for the decoupled system and so that was used for both the coupled system and the determination of  $\Phi^{k+1}$  in the decoupled systems in this work. For the Crank-Nicolson method it is also necessary to solve a linear system in the solve for  $V^{k+1}$  at each time step. An incomplete Cholesky preconditioner is used in this linear solve (which is much less computationally expensive than the solve for  $\Phi^{k+1}$ ).

By default PETSc distributes the problem amongst the  $p$  available processors by the first  $N/p$  rows on processor 0 (where  $A$  is an  $N$ -by- $N$  matrix), then the next  $N/p$  rows on processor 1 and so on. This means that for the coupled system as defined by Eqns. (23) – (25) the values of  $V_m$  and  $\phi_e$  at each node of the FE mesh would be stored on different processors (when  $p \geq 2$ ), necessitating a significant amount of communication time between the two processors storing each of these values. So, it would be better to store the two values on the same processor. At the same time, it is desirable to retain the block structure of the system matrix as much as possible (as the CG method is most efficient on sparse, diagonally dominant matrices) and to distribute the values of  $\phi_e$  at bath nodes evenly amongst the processors (to aid load balancing). In order to do this, rows of the matrix  $A$  and vectors  $x$  and  $b$  in Eqn. (23) are perturbed and then partitioned into  $p$  equal sections:

$$V^k = (V_1^k, \dots, V_p^k), \quad (27)$$

$$\Phi^k = (\Phi_1^k, \dots, \Phi_p^k), \quad (28)$$

where for any  $i$ , the vector segments  $V_i^k$  and  $\Phi_i^k$  contain the values of  $V_m$  and  $\phi_e$  respectively for some subset of the internal heart nodes, plus  $\Phi_i^k$  also contains further values of  $\phi_e$  at several of the bath nodes. The partitioned sections are then arranged in the overall solution matrix  $x^k$  so that:

$$x^k = (V_1^k, \Phi_1^k, \dots, V_p^k, \Phi_p^k), \quad (29)$$

with the rows of  $A$  and  $b$  being re-ordered in the same way. By leaving large sections of  $V^k$  and  $\Phi^k$  intact, the structure of the system matrix is preserved as much as possible, minimizing any possible loss of performance in the CG method.

All simulations were run on a cluster at Fujitsu Laboratories of Europe consisting of 44 compute nodes, each consisting of two dual core Xeon 5160 processors (3.0 GHz) with an InfiniBand interconnect.

### *E. Performance Measures*

Direct comparison of the performance of the two methods is difficult as each of them solves a different linear system. Ideally the computed solutions would be compared with an exact solution to the governing equations. However, for a model and geometry as complex as the whole ventricle used here it is impossible to determine such a solution. Instead, the solutions are compared with a good estimate of the exact solution, obtained by solving the benchmark cases using the coupled method, a very small time step and very small absolute and relative tolerances for the PETSc solver (both set to  $10^{-50}$ ) [6]. The root-mean-square (RMS) errors of the approximations to  $V_m$  and  $\phi_e$  for each of the solution methods relative to the good estimate are calculated at selected time steps and used as a measure of accuracy. The total computation time required to reach an equivalent level of accuracy for each method is used as a measure of efficiency.

In addition to the total computation time and the RMS of the residual, the amount of time spent computing the solution to the PDEs, the number of iterations needed for the iterative solver to converge and the time per iteration are also recorded. Computing the time spent solving the PDEs allows the relative performance of the different parts of the two methods to be compared, while calculating the iteration statistics allows analysis of the causes of any difference in efficiency between the two methods, i.e. determining whether the improved performance is due to the quicker method needing fewer iterations to converge or because each iteration can be completed in a much reduced time.

## III. RESULTS

The results of running the first benchmark for the semi-implicit coupled and for both of the uncoupled methods using a BAMG-preconditioned CG solver are summarized in Table I and Fig. 1. Table II shows the comparison between the three methods for the second benchmark. It can be seen that the coupled method outperforms both of the uncoupled methods on both benchmarks.

### A. Pacing

The pacing benchmark was run for  $p = 1, 2, 4, 8, 16, 32$ , where  $p$  is the number of nodes of the cluster that were used. For each simulation, the PETSc iterative CG solver was deemed to have converged when the relative tolerance (i.e. difference between two successive iterations of the solution) is less than  $10^{-5}$ . Calculation of the RMS errors (see the fifth and sixth columns of Table I) shows that the coupled semi-implicit method is slightly more accurate than the uncoupled CN method for both  $||V_m||$  and  $||\phi_e||$  (and for all  $p$ ). In turn, both are approximately twice as accurate as the uncoupled explicit method.

Fig. 1 shows the times required to run the entire benchmark using each numerical method. It can be seen that the coupled method runs between 1.31 and 1.61 times as fast as the explicit uncoupled method and between 1.47 and 1.76 times as fast as the Crank-Nicolson method, despite needing to solve a larger linear system at each time step. These values are increased to between 1.40 and 1.67 for the explicit method and between 1.50 and 1.80 for the Crank-Nicolson method if only the time taken solving the PDEs is considered. This is a more accurate indication of the speedup in the section of the code that differs between the various methods and also provides an indication of how the whole code will perform over longer simulations (since it excludes the time taken to assemble the solution matrices, which only has to be done once and, hence, is constant no matter how long the simulation is then run for).

Note, however, that the time per iteration (given in the fourth column of Table I) is approximately twice as large (for all  $p$ ) for the coupled method as it is for either of the two uncoupled methods. This is in line with what would be expected as the linear system that is solved for in the coupled method is roughly twice as large as that for the uncoupled systems (since it includes the values of both  $V_m$  and  $\phi_e$  at each node while the linear solves in the uncoupled methods are for the values of  $\phi_e$  only). The reason for the coupled method being faster overall is that it requires far fewer iterations per time step to converge to the solution of the bidomain equations (see the third column of Table I for the total number of iterations). Note also, however, that the iteration count for the CN method includes the iterations from *both* linear solves (for each of the extracellular and transmembrane potentials).

The parallel performance of each of the three solution methods is shown in Fig. 2. The scaling for all three methods is broadly similar. This is to be expected as they are each implemented

using similar code in the CARP simulator and use identical parallel libraries for the linear solve. The coupled method scales very slightly better than the two uncoupled methods. Again this is in line with what we would expect to see as of the three tasks that have to be performed at each time step, two (calculating the RHS vector  $\mathbf{b}$  and solving the ODEs) are trivially parallelizable, while the third (solving the linear system) is faster for the coupled model. So, when using the coupled method a higher proportion of time is spent doing the trivially parallelizable tasks.

Finally, note that the timings for running the benchmark using the uncoupled explicit method are considerably faster than for the similar benchmarks reported previously on the HPCx supercomputer [14], even though the processors in both machines have almost identical clock speeds (both approximately 3.0 GHz). There are at least three reasons for this. Firstly, only one processor per node was used for the simulations at FLE described in this work, compared to two processors per node on HPCx, with shared memory used for message passing within a node. Conflicts between the two processors in a single node will act to slow the simulation. Secondly, the tolerances used for this work are different from those used previously (in order to enable direct comparison with the coupled method) and this may lead to convergence in slightly fewer iterations. Finally, more recent versions of the numerical libraries were used in this work (in particular, PETSc 2.3.2 rather than 2.2.1). These observations are supported by comparing the timings in this work with those reported on a Linux cluster using only one processor per node in [14]. There, a time per iteration of 315 ms was reported for the re-entry benchmark on 16 processors with clock speeds of 2.0 GHz. This is 35% *faster* than the equivalent timing on HPCx (493 ms), even though the clock speed of each processor is only two-thirds as fast. The equivalent time per iteration in this work was 102 ms, which, while still twice as fast as would be accounted for by the difference in clock speeds, is still close enough to be accounted for by the improved numerical libraries.

### B. Re-Entry

The re-entry benchmark was run on the cluster with  $p = 32$  only, in order to verify that the numerical speedup seen in the pacing benchmark was still present in a simulation of a more complex wavefront (i.e. to check that the speedup was not dependent on the solution evolving very slowly). The results of these simulations are shown in Table II. It can be seen that the coupled method is 1.30 times as fast as the explicit method and 1.39 times as fast as the Crank-

Nicolson method. These results are consistent with those seen in the pacing benchmark. In terms of accuracy, the errors for the Crank-Nicolson method are again about half the size of those for the explicit method. However, unlike the pacing benchmark, the errors seen when using the coupled method are substantially smaller than those for either of the decoupled methods. It is not entirely clear why this is, but the size of the errors for the two decoupled methods are larger for the re-entry benchmark than for the pacing, so it may be that one more CG iteration for the two coupled methods would be desirable.

#### IV. DISCUSSION

The relative efficiency of three numerical methods for solving the bidomain equations have been compared in this paper. The first method was to decouple the bidomain equations and then solve one of the resulting equations using an FE scheme with an explicit forward Euler method and treat the other (which contains no time derivatives) as a steady state problem, which is also solved using FE methods and a BAMG preconditioned CG linear solver. The second method was to solve the same decoupled system, but using a Crank-Nicolson time step instead of the explicit Euler method in the FE scheme. The third solves the fully coupled bidomain equations using a semi-implicit FE scheme and the same preconditioner and linear solver as used for the decoupled methods. The method of decoupling and using an explicit scheme is very commonly used. This has been justified by the intuitive claim that two can be solved faster than one big linear system.

In this work, however, we have demonstrated that the decoupled solution method is, in fact, not as efficient as a coupled scheme for the same level of accuracy. The coupled scheme implemented in this work is both faster and more accurate than the explicit decoupled scheme on any number of processors that the algorithms have been tested on. In comparison to the Crank-Nicolson decoupled scheme, the coupled scheme shows significantly improved efficiency and a small improvement in accuracy, again for all  $p$ .

The speedup seen using the coupled method is due to the iterative solver (BAMG preconditioned CG) converging in fewer iterations than for the uncoupled methods, outweighing the increased time per iteration resulting from solving a larger linear system. Fig. 3 shows the convergence histories of the coupled solve and the elliptic solve in the coupled Crank-Nicolson method for a representative time step taken from the simulation of the re-entry benchmark (note

that the elliptic solve is identical whichever of forward Euler and Crank-Nicolson time-stepping is used). It can be seen that the rate at which the residuals decrease is monotonic for both methods and slightly faster for the coupled method.

However, this only tells part of the story. The linear system for the coupled method is larger than for the uncoupled and the values of  $V_m$  (which must be solved for in the coupled method, but not as part of the elliptic solve in the uncoupled method) are typically much larger than those of  $\phi_e$ . Both of these factors mean that the same relative errors in the solutions for each of the two methods would generate a larger residual for the coupled method than for the uncoupled. Hence, the value of  $\|Ae\|/\|Ax\|$  (where  $e$  is the vector of errors in the solution) is also plotted, to give an indication of how the norm of the errors relates to the norm of the solution vector (i.e. the relative error). It can be seen that using this relative method the convergence for the coupled method is dramatically faster than for the elliptic solve of the uncoupled method. Hence fewer iterations are required for convergence of the coupled method to a solution which is at least as accurate as the equivalent using an uncoupled method.

Clearly the choice of convergence criterion is also of critical importance. The coupled method must converge in fewer than half as many iterations as the decoupled methods in order that the reduction in the number of iterations outweighs the additional time spent on each iteration. So, some justification for the choice of a relative tolerance of  $10^{-5}$  is required. Here we note again that comparing residuals for the two methods is not necessarily indicative of their relative accuracy (since they solve different linear systems). The quantities that are actually of interest are the RMS errors in  $V_m$  and  $\phi_e$  compared to the ‘good estimate’ coupled with very small time steps and tolerances referred to in Sec. II-E. The values of these RMS errors for solution of the coupled method with various different choices of relative tolerance as the convergence criterion are shown in Fig. 4. It can be seen that decreasing the relative tolerance below  $10^{-5}$  does not greatly increase the accuracy of the solution compared to the good estimate, whereas increasing it to more than  $10^{-4}$  leads to a significant increase in the RMS errors. Note also that in each result reported the RMS errors resulting from the coupled method were *smaller* than those for the equivalent simulation using either of the decoupled methods. So, accuracy is not being sacrificed for the greater speed of the coupled method.

Note, however, that results of [27] (comparing a coupled method to a decoupled method with an outer block Gauss-Seidel (GS) loop) suggest that a decoupled method with only one

GS iteration (equivalent to the decoupled method used here) should be faster than a coupled method. This discrepancy will require further investigation in the future. However, the results of [27] were generated for a much smaller test case (70,699 nodes) using a structured FE mesh (leading to a very much more structured system matrix) for a much simpler geometry, so the differences between the two studies are sufficient that this is not a major concern at this stage.

In addition to the speedup seen by implementing the coupled method, the increased stability that is obtained by using an implicit method to advance in time means that the time step can be chosen to be significantly larger than for the explicit uncoupled method. In particular, the time step can be determined by physiological considerations rather than being constrained by the size of the smallest element of the FE mesh. This is likely to be very advantageous both when very fine meshes are used in order to increase the resolution of the solution (as there would be no need to reduce the time step to preserve stability) and if adaptive methods are used in order to speed up the solution of the bidomain equations in regions away from the electrical potential wave front (as the allowable increase in time step — and hence potential increase in efficiency — would be much larger).

Finally, observe that an improvement of 50% to 80% in efficiency, while not necessarily sounding like a significant leap forward, can be very useful in practice. For example, a simulation that previously lasted one day using the Crank-Nicolson method could be run overnight using the coupled method. This makes for more flexible working, with the results from a simulation that is started at the end of one day being available for analysis at the start of the next day.

#### ACKNOWLEDGEMENTS

This publication is based on work supported by Award No. KUK-C1-013-04 , made by King Abdullah University of Science and Technology (KAUST).

#### REFERENCES

- [1] R. Plonsey, “Bioelectric sources arising in excitable fibres (ALZA Lecture),” *Ann. Biomed. Eng.*, vol. 16, pp. 519–546, 1988.
- [2] N. G. Sepulveda, B. J. Roth, and J. P. Wisko Jr., “Current injection into a two-dimensional anisotropic bidomain,” *Biophys. J.*, vol. 55, pp. 987–999, 1989.
- [3] J. P. Keener and J. Sneyd, *Mathematical Physiology*. New York: Springer, 1998, ch. 11.
- [4] D. M. Harrild and C. S. Henriquez, “A finite volume model of cardiac propagation,” *Ann. Biomed. Eng.*, vol. 25, pp. 315–334, 1997.

- [5] J. P. Keener and K. Bogar, "A numerical method for the solution of the bidomain equations in cardiac tissue," *Chaos*, vol. 8, pp. 234–241, 1998.
- [6] E. J. Vigmond, F. Aguel, and N. A. Trayanova, "Computational techniques for solving the bidomain equations in three dimensions," *IEEE Trans. Biomed. Eng.*, vol. 49, pp. 1260–1269, 2002.
- [7] E. Vigmond, M. Hughes, G. Plank, and L. Leon, "Computational tools for modeling electrical activation in cardiac tissues," *J. Electrocardiol.*, vol. 36 (Suppl.), pp. 69–74, 2003.
- [8] M. Trew, I. LeGrice, B. Smaill, and A. Pullan, "A finite volume method for modeling discontinuous electrical activation in cardiac tissue," *Ann. Biomed. Eng.*, vol. 33, pp. 590–602, 2005.
- [9] M. L. Trew, B. H. Smaill, D. P. Bullivant, P. Hunter, and A. J. Pullan, "A generalized finite difference method for modeling cardiac electrical activation on arbitrary, irregular computational meshes," *Math. Biosci.*, vol. 198, pp. 169–189, 2005.
- [10] J. P. Whiteley, "An efficient numerical technique for the solution of the monodomain and bidomain equations," *IEEE Trans. Biomed. Eng.*, vol. 53, pp. 2139–2147, 2006.
- [11] E. J. Vigmond, R. Weber dos Santos, A. J. Prassl, M. Deo, and G. Plank, "Solvers for the cardiac bidomain equations," *Prog. Biophys. Mol. Biol.*, vol. 96, pp. 3–18, 2007.
- [12] G. W. Beeler and H. Reuter, "Reconstruction of the action potential of ventricular myocardial fibres," *J. Physiol.*, vol. 268, pp. 177–210, 1977.
- [13] K. Skouibine, N. A. Trayanova, and P. K. Moore, "Anode/cathode make and break phenomena in a model of defibrillation," *IEEE Trans. Biomed. Eng.*, vol. 46, pp. 769–777, 1999.
- [14] G. Plank, M. Liebmann, R. Weber dos Santos, and E. J. Vigmond, "Algebraic multigrid preconditioner for the cardiac bidomain model," *IEEE Trans. Biomed. Eng.*, vol. 54, pp. 584–597, 2007.
- [15] G. Strang, "On the construction and comparison of difference schemes," *SIAM J. Numer. Anal.*, vol. 5, pp. 506–517, 1968.
- [16] J. Sundnes, G. T. Lines, and A. Tveito, "An operator splitting method for solving the bidomain equations coupled to a volume conductor model for the torso," *Math. Biosci.*, vol. 194, pp. 233–248, 2005.
- [17] J. Sundnes, B. F. Nielsen, K. A. Mardal, X. Cai, G. T. Lines, and A. Tveito, "On the computational complexity of the bidomain and the monodomain models of electrophysiology," *Ann. Biomed. Eng.*, vol. 34, pp. 1088–1097, 2006.
- [18] J. P. Whiteley, "Physiology driven adaptivity for the numerical solution of the bidomain equations," *Ann. Biomed. Eng.*, vol. 35, pp. 1510–1520, 2007.
- [19] F. Vetter and A. McCulloch, "Three-dimensional analysis of regional cardiac function: a model of rabbit ventricular anatomy," *Prog. Biophys. Mol. Biol.*, vol. 69, pp. 157–183, 1998.
- [20] L. Clerc, "Directional differences of impulse spread in trabecular muscle from mammalian heart," *J. Physiol.*, vol. 255, pp. 335–346, 1976.
- [21] J. Puglisi and D. Bers, "LabHEART: an interactive computer model of rabbit ventricular myocyte ion channels and Ca transport," *Am. J. Physiol. Cell Physiol.*, vol. 281, pp. C2049–C2060, 2001.
- [22] K. DeBruin and W. Krassowska, "Electroporation and shock-induced transmembrane potential in a cardiac fiber during defibrillation strength shocks," *Ann. Biomed. Eng.*, vol. 26, pp. 584–596, 1998.
- [23] D. Cheng, L. Tung, and E. Sobie, "Nonuniform responses of transmembrane potential during electric field stimulation of single cardiac cells," *Am. J. Physiol.*, vol. 277, pp. H351–H362, 1999.
- [24] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, "PETSc users manual," Argonne National Laboratory, Tech. Rep. ANL-95/11 – Revision 2.3.2, 2006.
- [25] L. N. Trefethen and D. Bau, III, *Numerical Linear Algebra*. Philadelphia, PA: SIAM, 1997.

- [26] V. Henson and U. Yang, “BoomerAMG: a parallel algebraic multigrid solver and preconditioner,” *Appl. Numer. Math.*, vol. 41, pp. 155–177, 2002.
- [27] M. Pennacchio and V. Simoncini, “Efficient algebraic solution of reaction-diffusion systems for the cardiac excitation process,” *J. Comput. Appl. Math.*, vol. 145, pp. 49–70, 2002.

# LIST OF FIGURES

1	Performance comparison of the different numerical methods on the pacing benchmark. The accuracies of the coupled and CN methods are approximately equal, while the errors obtained using the explicit method are twice as large as those resulting from the coupled method. The left panel shows the total time required to run the benchmark on 1, 2, 4, 8, 16 and 32 processors using the coupled method (black bars), the CN decoupled method (grey) and the explicit decoupled method (white). The right panel shows the relative speedup for the solution of the PDEs in the pacing benchmark obtained by using the coupled solution method compared to the uncoupled explicit (dashed line) and uncoupled CN (solid line) methods. . . . .	20
2	Parallel speedup (left panel) and efficiency (right panel) of the coupled (solid line), uncoupled CN (dashed line) and uncoupled explicit (dot-dashed line) solution methods for the bidomain equations. For comparison, the dotted line in the plot of parallel speedup represents a theoretical linear speedup (equivalent to an efficiency of 1.0 for any number of processors in the efficiency plot). . . . .	21
3	Convergence history of the solution methods. Residuals (left panel) and $\ Ae\ /\ Ax\ $ (right panel) are plotted semi-logarithmically against the iteration number for both the coupled solve (solid line) and the elliptic solve in the uncoupled Crank-Nicolson method (dashed line). . . . .	22
4	Effect of varying the convergence criterion on the RMS errors in the solutions. The RMS errors in $V_m$ (solid line) and $\phi_e$ (dotted line) after 100 ms simulation of the re-entry benchmark, using the coupled method are shown, plotted against the relative tolerances (between $10^{-1}$ and $10^{-10}$ ) used to determine convergence of the iterative solver at each time step. . . . .	23

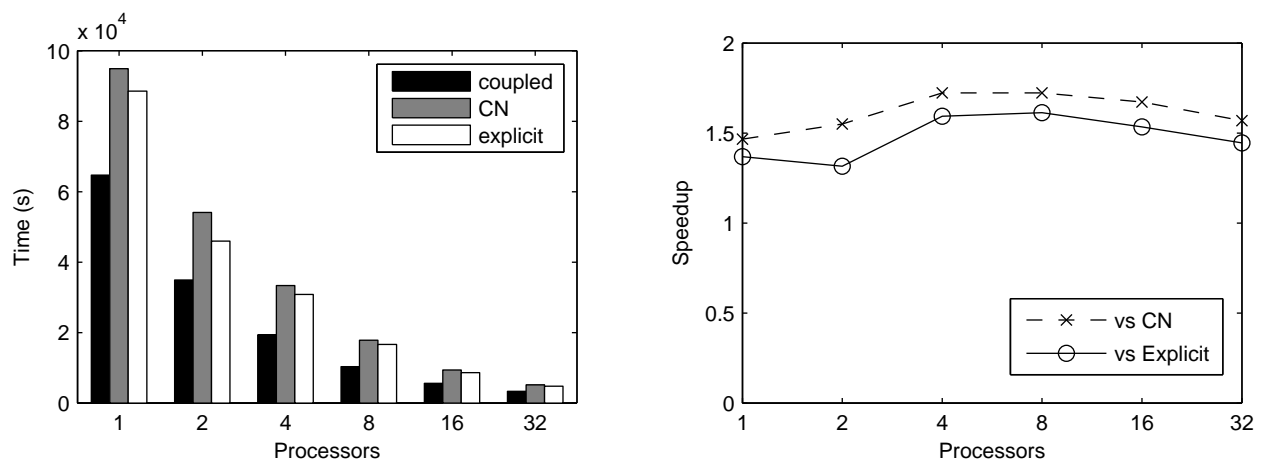


Fig. 1. Performance comparison of the different numerical methods on the pacing benchmark. The accuracies of the coupled and CN methods are approximately equal, while the errors obtained using the explicit method are twice as large as those resulting from the coupled method. The left panel shows the total time required to run the benchmark on 1, 2, 4, 8, 16 and 32 processors using the coupled method (black bars), the CN decoupled method (grey) and the explicit decoupled method (white). The right panel shows the relative speedup for the solution of the PDEs in the pacing benchmark obtained by using the coupled solution method compared to the uncoupled explicit (dashed line) and uncoupled CN (solid line) methods.

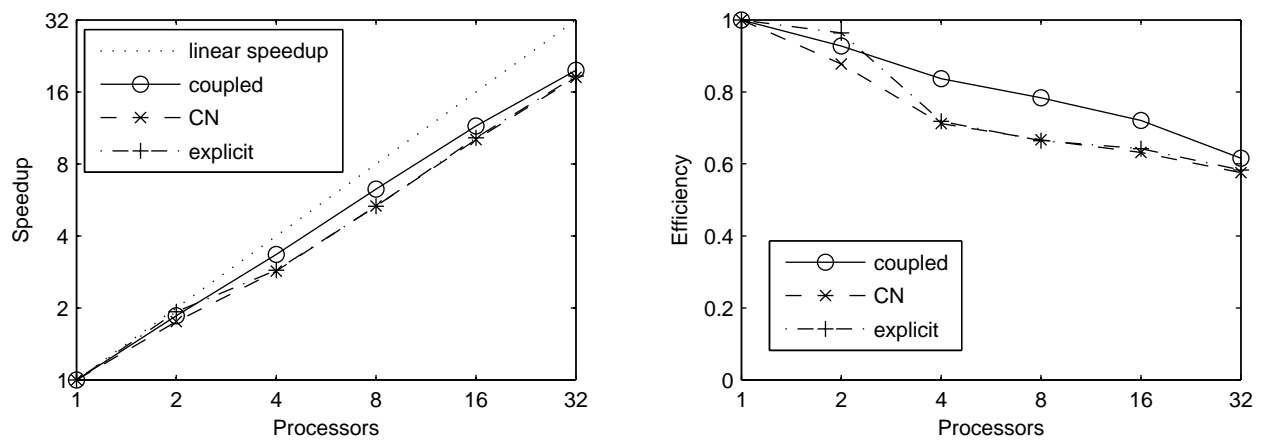


Fig. 2. Parallel speedup (left panel) and efficiency (right panel) of the coupled (solid line), uncoupled CN (dashed line) and uncoupled explicit (dot-dashed line) solution methods for the bidomain equations. For comparison, the dotted line in the plot of parallel speedup represents a theoretical linear speedup (equivalent to an efficiency of 1.0 for any number of processors in the efficiency plot).

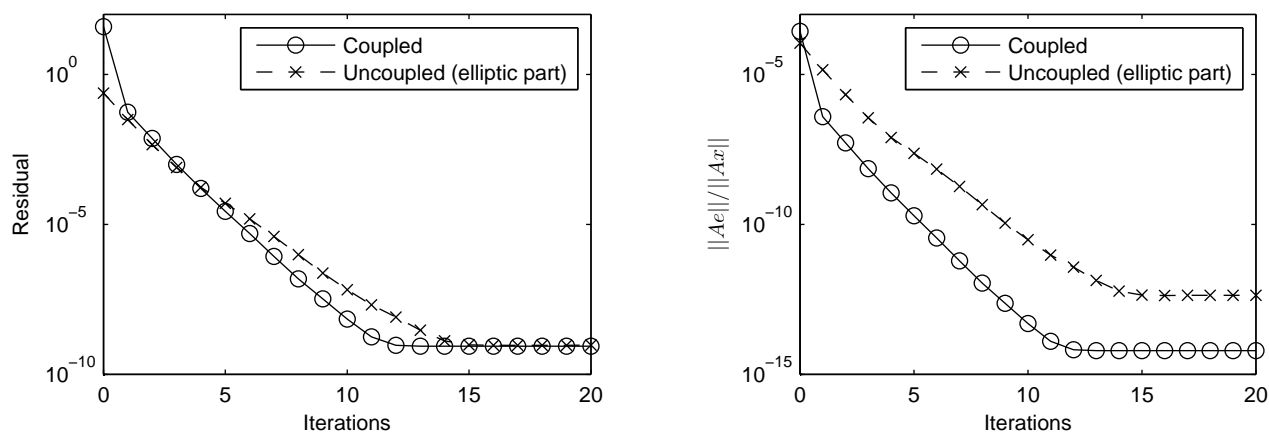


Fig. 3. Convergence history of the solution methods. Residuals (left panel) and  $\|Ae\|/\|Ax\|$  (right panel) are plotted semi-logarithmically against the iteration number for both the coupled solve (solid line) and the elliptic solve in the uncoupled Crank-Nicolson method (dashed line).

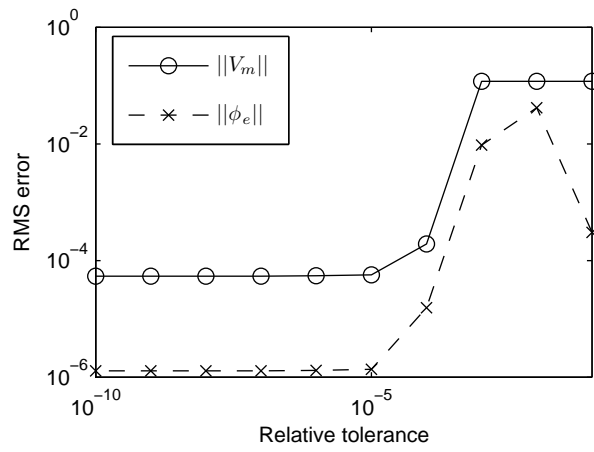


Fig. 4. Effect of varying the convergence criterion on the RMS errors in the solutions. The RMS errors in  $V_m$  (solid line) and  $\phi_e$  (dotted line) after 100 ms simulation of the re-entry benchmark, using the coupled method are shown, plotted against the relative tolerances (between  $10^{-1}$  and  $10^{-10}$ ) used to determine convergence of the iterative solver at each time step.

## LIST OF TABLES

I	Summary of the results obtained from simulating the pacing benchmark using the different numerical schemes. All simulations use the BAMG preconditioner. Shown in the table are number of processors used in the simulation ( $p$ ), total number of iterations ( $I$ ), mean time per iteration ( $\bar{T}_i$ ) and the RMS errors in $V_m$ and $\phi_e$ after 100 ms of the simulation ( $\ V_m\ $ and $\ \phi_e\ $ respectively). . . . .	25
II	Summary of the results obtained from simulating the re-entry benchmark. All simulations use the BAMG preconditioner and were run on 32 processors. Shown in the table are the total computational time required to run the simulation $T$ , total number of iterations ( $I$ ), mean time per iteration ( $\bar{T}_i$ ) and the RMS errors in $V_m$ and $\phi_e$ after 100 ms of the simulation ( $\ V_m\ $ and $\ \phi_e\ $ respectively). . . . .	26

TABLE I

SUMMARY OF THE RESULTS OBTAINED FROM SIMULATING THE PACING BENCHMARK USING THE DIFFERENT NUMERICAL SCHEMES. ALL SIMULATIONS USE THE BAMG PRECONDITIONER. SHOWN IN THE TABLE ARE NUMBER OF PROCESSORS USED IN THE SIMULATION ( $p$ ), TOTAL NUMBER OF ITERATIONS ( $I$ ), MEAN TIME PER ITERATION ( $\bar{T}_i$ ) AND THE RMS ERRORS IN  $V_m$  AND  $\phi_e$  AFTER 100 ms OF THE SIMULATION ( $\|V_m\|$  AND  $\|\phi_e\|$  RESPECTIVELY).

<b>Method</b>	$p$	$I$	$\bar{T}_i$ ms	$\ V_m\ $ $\times 10^{-6}$ mV	$\ \phi_e\ $ $\times 10^{-6}$ mV
<b>Coupled</b>	1	23276	2513	38.34	0.73
	2	23277	1355	38.71	0.74
	4	23277	756	39.19	0.75
	8	23277	403	39.60	0.77
	16	23279	219	40.04	0.78
	32	23276	129	41.10	0.81
<b>Explicit</b>	1	64469	1271	78.88	1.92
	2	68517	674	78.88	1.92
	4	78132	371	78.88	1.92
	8	77839	201	78.88	1.92
	16	78973	102	78.88	1.92
	32	80736	59	78.88	1.92
<b>CN</b>	1	64722	1363	42.85	0.87
	2	69255	732	42.85	0.87
	4	79338	398	42.85	0.87
	8	79061	213	42.85	0.87
	16	80067	111	42.85	0.87
	32	81629	59	42.85	0.87

TABLE II

SUMMARY OF THE RESULTS OBTAINED FROM SIMULATING THE RE-ENTRY BENCHMARK. ALL SIMULATIONS USE THE BAMG PRECONDITIONER AND WERE RUN ON 32 PROCESSORS. SHOWN IN THE TABLE ARE THE TOTAL COMPUTATIONAL TIME REQUIRED TO RUN THE SIMULATION  $T$ , TOTAL NUMBER OF ITERATIONS ( $I$ ), MEAN TIME PER ITERATION ( $\bar{T}_i$ ) AND THE RMS ERRORS IN  $V_m$  AND  $\phi_e$  AFTER 100 ms OF THE SIMULATION ( $\|V_m\|$  AND  $\|\phi_e\|$  RESPECTIVELY).

<b>Method</b>	$T$ s	$I$	$\bar{T}_i$ ms	$\ V_m\ $ $\times 10^{-6}$ mV	$\ \phi_e\ $ $\times 10^{-6}$ mV
<b>Coupled</b>	3327	25009	133	10.53	0.62
<b>Explicit</b>	4649	73604	63	98.30	2.60
<b>CN</b>	4339	73677	59	59.94	1.54



## RECENT REPORTS

### 2009

01/09	A Mass and Solute Balance Model for Tear Volume and Osmolarity in The Normal And The Dry Eye	Gaffney Tiffany Yokoi Bron
02/09	Diffusion and permeation in binary solutions	Peppin
03/09	On the modelling of biological patterns with mechanochemical models: insights from analysis and computation	Moreo Gaffney Garcia-Aznar Doblare
04/09	Stability analysis of reaction-diffusion systems with time-dependent coefficients on growing domains	Madzvamuse Gaffney Maini
05/09	Onsager reciprocity in premelting solids	Peppin Spannuth Wettlaufer
06/09	Inherent noise can facilitate coherence in collective swarm motion	Yates <i>et al.</i>
07/09	Solving the Coupled System Improves Computational Efficiency of the Bidomain Equations	Southern Plank Vigmond Whiteley
08/09	Model reduction using a posteriori analysis	Whiteley
09/09	Equilibrium Order Parameters of Liquid Crystals in the Landau-De Gennes Theory	Majumdar
10/09	Landau-De Gennes theory of nematic liquid crystals: the Oseen-Frank limit and beyond	Majumdar Zarnescu
11/09	A Comparison of Numerical Methods used for Finite Element Modelling of Soft Tissue Deformation	Pathmanathan Gavaghan Whiteley
12/09	From Individual to Collective Behaviour of Unicellular Organisms: Recent Results and Open Problems	Xue Othmer Erban
13/09	Stochastic modelling of reaction-diffusion processes: algorithms for bimolecular reactions	Erban Chapman
14/09	Chaste: a test-driven approach to software development for physiological modelling	Pitt-Francis <i>et al.</i>

15/09	Block triangular preconditioners for PDE constrained optimization	Rees Stoll
16/09	From microscopic to macroscopic descriptions of cell migration on growing domains	Baker Yates Erban

**Copies of these, and any other OCCAM reports can be obtained from:**

**Oxford Centre for Collaborative Applied Mathematics  
Mathematical Institute  
24 - 29 St Giles'  
Oxford  
OX1 3LB  
England  
[www.maths.ox.ac.uk/occam](http://www.maths.ox.ac.uk/occam)**