Computational modelling of cardiac electro-physiology: explanation of the variability of results from different numerical solvers

P. Pathmanathan[1], M. O. Bernabeu[1], S. A. Niederer[2], D. J. Gavaghan[1], D. Kay[1]

November 14, 2011

[1] Oxford University Department of Computer Science, Parks Road, Oxford, OX1 3QD, UK.

[2] Department of Biomedical Engineering, Division of Imaging Sciences and Biomedical Engineering, Kings College, London, UK.

Address for correspondance: Pras Pathmanathan, Oxford University Department of Computer Science, Wolfson Building, Parks Road, Oxford, OX1 3QD, UK.

Phone number: +44 (0)1865 273858

Fax number: +44 (0)1865 273839

E-mail address: pras@cs.ox.ac.uk

Abstract

A recent verification study [Niederer et al., Phil. Trans. Royal Soc. A, 2011] compared eleven large-scale cardiac electro-physiology solvers on an unambiguously-defined common problem. An unexpected amount of variation was observed between the codes, including significant error in conduction velocity in the majority of the codes at certain spatial resolutions. In particular, the results of the six finite element codes varied considerably, despite each using the same order of interpolation. In this present study, we compare various algorithms for cardiac electro-physiological simulation, which allows us to fully explain the differences between the solvers. We identify the use of mass lumping as the fundamental cause of the largest variations—specifically, the combination of the commonly-used techniques of mass lumping and operator splitting, which results in a slightly different form of mass lumping to that supported by theory and leads to increased numerical error. Other variations are explained through the manner in which the ionic current is interpolated. We also investigate the effect of different forms of mass lumping in various types of simulation.

Keywords: numerical methods, finite element, operator splitting, mass lumping, verification

1 Introduction

For more than half a century, computational models of cardiac electro-physiology have been used to contribute to the study of cardiac activity, and many codes have been developed in the last few decades which simulate cardiac electrical activity at the tissue or whole-organ level. Up until very recently, however, there had been no attempt made to systematically compare different codes on a common problem, in order to verify their accuracy against one another. This was rectified in [1], in which eleven different cardiac electro-physiology codes were used to compute the solution of a carefully-chosen and unambiguously-defined benchmark problem. The benchmark involved determining the electrical activity within a cuboid region of tissue using the monodomain equation, with particular choices of transversely isotropic conductivities, cell model and parameter values. The tissue was stimulated in a region at one corner of the mesh and node-wise activation times recorded for each of the eleven simulators. For each code, the benchmark was solved with three choices of spatial stepsize, $h = 0.01$, 0.02 and 0.05cm, and three choices of timestep, $\Delta t = 0.005$, 0.01 and 0.05ms. The results were somewhat surprising.

Firstly, even on the finest mesh and with the smallest timestep, there was a significant range of results between the codes. For example, the activation time of the node at the opposite corner of the tissue to that stimulated—essentially, the time for total depolarisation of the tissue—varied between 37.8ms and 48.7ms between the codes. Rather concerningly, it was not possible to confidently predict the correct value of this quantity, even to the nearest millisecond, using the combination of results of all the codes. This is particularly worrying since this spatial resolution, 0.01cm, is smaller than that commonly accepted in cardiac

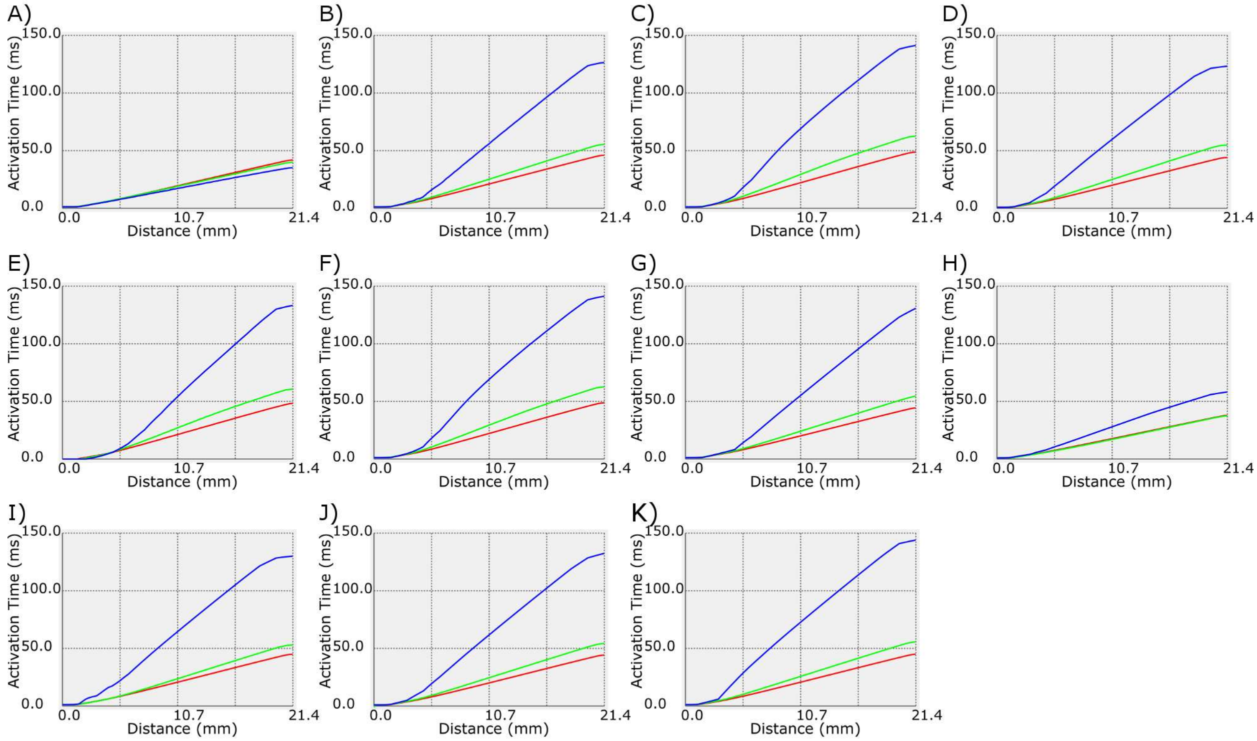

Figure 1: Activation time across a diagonal line through a cuboidal tissue, using the benchmark problem, for a number of different electro-physiology solvers. This figure is reproduced from [1], and compares a number of solvers, in contrast to the remainder of the paper, which compares algorithms written in one solver. Red lines represent solutions using a spatial stepsize of $h = 0.01$cm, green lines solutions with $h = 0.02$cm, and blue lines $h = 0.05$cm. Code A is Chaste, Code H is Fenics, and codes A, B, C, E, F and H are finite element codes. Codes D, G, I, J, and K use finite differences. *Figure reproduced with permission.*

modelling as 'suitably fine'. Significantly larger spatial stepsizes are employed in simulation studies—for example, studies have used average edge-lengths of 0.03cm [2], 0.046cm [3], and 0.03–0.05cm [4]. Even a highly-detailed rabbit mesh made up of 24.2 million elements [5] has a spatial resolution of 0.0125cm.

The second notable result was that all but two of the eleven codes displayed a very large reduction of conduction velocity (CV) on the lowest resolution meshes, as illustrated in Figure 1, which is reproduced from the benchmark paper [1]. Specifically, these nine codes reported conduction velocities that were approximately 2.5-3 times slower on the 0.05cm meshes compared to the 0.01cm meshes, i.e. errors of the order of 200% or more. Whilst some error is naturally expected at this resolution, the magnitude of the error is large, and the fact that two codes did not exhibit this error indicates that it can be avoided (even without using higher-order schemes).

The eleven codes between them made use of a wide range of numerical methods. Five codes used the finite difference method, six the finite element (FE) method. Of the FE codes, five used operator splitting of the reaction term from the diffusion term in the monodomain equations, while the other code employed a traditional FE discretisation. (All of these techniques will be discussed more completely in Section 2). Three of the FE codes used tetrahedral meshes, the other three hexahedral. All the FE codes used low-order schemes (linear basis functions).

The two codes that did not exhibit the large conduction velocity decrease on the coarsest meshes are Chaste [6, 7], developed in the University of Oxford, and Fenics [8], developed at Simula, Norway. Both are FE codes. Chaste uses a traditional FE approach, whereas Fenics uses operator splitting. The main aim of this paper is to identify the fundamental differences in the algorithms employed by Chaste/Fenics and the other FE codes that caused the large difference in results on the benchmark problem. This was initially very unclear, but after some preliminary analysis we (the Chaste development team) suggested that it was *mass lumping* that was causing the spurious slow-down of conduction velocity in the other finite element codes. Mass lumping, which will be described in more detail in Section 2, involves replacing the mass matrix, a matrix which arises in the FE discretisation of the equations, with a diagonal matrix whose entries are the sums of the rows of the original matrix. It was used in all four finite element codes that displayed CV slow-down. However, as we shall see, the answer is more subtle than just 'the use of mass lumping'.

In this paper, we compare a number of FE-based numerical algorithms for cardiac electrophysiology. The choice of algorithms to be compared is motivated by the aim of understanding the results in [1]. All the algorithms are based on a common time and space discretisation, and are all implemented in the same code, Chaste. By comparing the results of these algorithms on a set of test problems (including the benchmark problem defined in [1]), we are able to gauge the error induced by making certain algorithmic changes, such as introducing mass lumping. Note that we do not attempt to fully reproduce the methods employed by the other solvers. Instead, by making small changes to a common algorithm run using a single code, we are able to determine the impact of these different FE methods. We then use the insight obtained to interpret the results in [1].

We do not include any finite difference methods in our comparison. Whilst this paper is mostly concerned with explaining the FE results in the benchmark paper [1], the link between the finite difference and finite element results will also be discussed in Section 5.2.

Note that we will only touch upon the efficiency of different schemes in this paper. Finite difference methods will generally be as fast or faster than FE schemes, and mass lumping will usually lead to reduced computational costs. In particular, no linear system solve is required when mass lumping is employed with explicit time-discretisations, which can be a huge computational saving. This paper pertains instead to accuracy of given schemes at given mesh resolutions. This can be used to inform simulation studies of appropriate stepsizes based on scheme and allowable numerical error. The question of which scheme is most efficient given accuracy constraints is outside the scope of this paper.

We begin in Section 2 where we write down the governing equations and describe the numerical methods compared in this paper, which include operator splitting and two forms of mass lumping: (i) *full-lumping*, where the lumped mass matrix is used on both the time-derivative and reaction terms of the parabolic PDE; and (ii) *half-lumping*, where mass lumping is used only on the time-derivative term. We shall note that operator splitting can only correspond to full-lumping. In Section 3 we survey the use of mass lumping in the cardiac simulation literature as well as stating a theorem which guarantees $\mathcal{O}(h^2)$ convergence of a *half-lumped* scheme. Next, in Section 4, we test various schemes on the benchmark problem, before also investigating the isotropic and cardiac geometry cases, and looking at convergence of a 1D problem. We then fully interpret the results of [1] in Section 5. We conclude by discussing our results in Section 6.

2 Methods

2.1 Governing equations and finite element schemes

Let $\Omega \subset \mathbb{R}^3$ denote the region occupied by the cardiac tissue. In the monodomain equations [9], the transmembrane voltage, V, is governed by

$$\chi \left(\mathcal{C} \frac{\partial V}{\partial t} + I_{\text{ion}}(\mathbf{u}, V) \right) - \nabla \cdot (\sigma \boldsymbol{\nabla} V) \;=\; I_{\text{stim}}, \tag{1a}$$

$$\frac{\partial \mathbf{u}}{\partial t} \;=\; \mathbf{f}\left(\mathbf{u}, V\right), \tag{1b}$$

where σ is the conductivity tensor, $\mathcal{C}$ the capacitance across the membrane, χ the surface-area-to-volume-ratio, and I_{stim} a stimulus current. $\mathbf{u}$ are a set of cell-level variables whose dynamical behaviour is governed by the ordinary differential equations (ODEs) given by $\mathbf{f}$ and which couple back to the partial differential equation (PDE) through the ionic current I_{ion}.

Numerical solution of the monodomain equations has been carried out using both the finite difference method (see [10] for a discussion) and the finite element method [11]. The FE method has the advantage of handling derivative boundary conditions on irregular geometries such as the heart in a straightforward, systematic manner. In the finite difference approach it is natural to solve cell model ODEs at the nodes. In the finite element method however, the ionic current is technically required in the interior of elements, and one option is to use a cell model at each quadrature point of each element. This can be prohibitively expensive, and most finite element simulators instead use cell models at nodes.

Chaste, which is used for all simulations in this paper, uses a semi-implicit time-discretisation where the diffusion term is treated implicitly and the reaction (ionic current) term treated explicitly (more details are given in [12]). Although other options are used by the other ten codes surveyed the benchmark paper [1], we are more concerned with errors as the spatial stepsize h is varied, and since the results of all the FE codes on the benchmark problem were highly insensitive to the chosen timesteps, we do not expect the choice of time-discretisation to affect our results.

Let the standard piecewise-linear basis functions be $\psi_1, \dots, \psi_N$, corresponding to the nodes $\mathbf{x}_1, \dots, \mathbf{x}_N$ and satisfying $\psi_j(\mathbf{x}_k) = \delta_{jk}$. Letting $\mathbf{V}^n = (V_1^n, \dots, V_N^n)$ be a vector of the nodal values of the voltage at time t_n, a standard finite element spatial discretisation of the semi-implicit time-discretisation of (1a) is [7] (for the rest of this section we drop the stimulus current, which would add a term to the definition of $\mathbf{b}^n$ below which we assume is integrated exactly):

$$\left(\frac{\chi\mathcal{C}}{\Delta t}M + K\right)\mathbf{V}^{n+1} = \frac{\chi\mathcal{C}}{\Delta t}M\mathbf{V}^n + \mathbf{b}^n, \tag{2}$$

where $M_{jk} = \int \psi_j \psi_k \, \mathrm{d}^3\mathbf{x}$ is the mass matrix, $K_{jk} = \int \boldsymbol{\nabla}\psi_j \cdot \sigma \boldsymbol{\nabla}\psi_k \, \mathrm{d}^3\mathbf{x}$ is the stiffness matrix, and

$$b_j^n = -\int_\Omega \chi I_{\text{ion}}(\mathbf{u}^{n+1}, V^n)\psi_j \, \mathrm{d}^3\mathbf{x}. \tag{3}$$

Ionic current and state-variable interpolation

To evaluate $\mathbf{b}^n$ numerical integration is used, and the ionic current is therefore required at quadrature points. We have in previous work [12] evaluated two ways of doing this: (i) ionic current interpolation (ICI), in which the nodal values of the ionic current are interpolated linearly onto the quadrature point, or (ii) state-variable interpolation (SVI), where the state-variables $\mathbf{u}$ are interpolated instead and the ionic current evaluated using this. The options lead to the following linear systems

$$\left(\frac{\chi\mathcal{C}}{\Delta t}M + K\right)\mathbf{V}^{n+1} = M\left(\frac{\chi\mathcal{C}}{\Delta t}\mathbf{V}^n - \chi\mathbf{F}^n\right), \qquad \text{for ICI} \tag{4}$$

$$\left(\frac{\chi\mathcal{C}}{\Delta t}M + K\right)\mathbf{V}^{n+1} = M\left(\frac{\chi\mathcal{C}}{\Delta t}\mathbf{V}^n - \chi\mathbf{F}^n\right) + \mathbf{c}^n, \qquad \text{for SVI} \tag{5}$$

Here $\mathbf{F}^n$ is the vector of nodal ionic currents, and

$$c_j^n = \chi \int_{\mathcal{T}} \left(I_{\text{ion}}^{\text{ICI}} - I_{\text{ion}}^{\text{SVI}}\right)\psi_j \, \mathrm{d}^3\mathbf{x}$$

is a ‘correction term’ that involves removing, *on a selection of elements,* $\mathcal{T}$, the linearly interpolated ionic current (coming from the $-\chi M\mathbf{F}^n$ term in (5)) and replacing it with an ionic current computed through state-variable interpolation. This is only done on a selection of elements (specifically elements on which the ionic current is not small, i.e. the wavefront), to maintain efficiency—using state-variable interpolation is in general expensive. For full details see the hybrid scheme discussed in [12]. We include SVI in this study as it was used to produce the Chaste results in [1].

Operator splitting (OS)

The alternative approach that was taken by all the other finite element codes is reaction-diffusion operator splitting [13], in which the PDE (1a) is broken down into a set of nodal ODEs, $\mathcal{C}\frac{\mathrm{d}V}{\mathrm{d}t} = -I_{\text{ion}}(\mathbf{u}, V)$; and a diffusion (not reaction-diffusion) PDE, $\chi\mathcal{C}\frac{\partial V}{\partial t} = \nabla\cdot(\sigma\boldsymbol{\nabla}V)$. These two equations are solved in series, and this can be done in a manner that, for example, is second-order in time. This is very common in cardiac electro-physiology [14, 15, 13, 16, 17].

OS is very similar to ICI. In particular, in the case of simple (first-order) Godunov splitting [13], with forward Euler for the ODEs and backward Euler for the PDEs, it is formally identical to ICI (4). This can be trivially seen by writing this case as follows (using an intermediate variable $\mathbf{V}^{n+*}$):

$$\begin{aligned} \mathcal{C}\mathbf{V}^{n+*} &= \mathcal{C}\mathbf{V}^{n} - \Delta t\mathbf{F}^{n}, && \text{(6a)}\\ \left(\frac{\chi\mathcal{C}}{\Delta t}M + K\right)\mathbf{V}^{n+1} &= M\left(\frac{\chi\mathcal{C}}{\Delta t}\mathbf{V}^{n+*}\right), && \text{(6b)} \end{aligned}$$

and noting that substituting (6a) in (6b) produces ICI (4). When other splitting schemes such as second-order Strang splitting, and/or alternative ODE solvers are used, the method will not be identical to ICI (4), but the differences will only be in the time-discretisation part of the scheme. The method will always have, in common with ICI, a vector of nodal values of ionic current, multiplied by the mass matrix.

Mass Lumping

Equations (4)–(6b) illustrate the manner in which the mass matrix arises in the linear systems. Mass lumping involves replacing the mass matrix M with a diagonal matrix M_L satisfying

$$(M_L)_{ij} = \begin{cases} \sum_k M_{ik} & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

This is in fact related to using the trapezoidal rule to compute integrals, rather than say Gaussian quadrature [18], an $\mathcal{O}(h)$ integration error.

When using mass lumping with ICI we can distinguish between two alternatives, which we will refer to as *full-lumping* and *half-lumping.* For full-lumping the lumped mass matrix is used on both the voltage and the reaction terms, whereas for half-lumping it is only used on the voltage terms:

$$\left(\frac{\chi\mathcal{C}}{\Delta t}M_L + K\right)\mathbf{V}^{n+1} = \frac{\chi\mathcal{C}}{\Delta t}M_L\mathbf{V}^n - \chi M_L\mathbf{F}^n \qquad \text{for full-lumping} \tag{7}$$

$$\left(\frac{\chi\mathcal{C}}{\Delta t}M_L + K\right)\mathbf{V}^{n+1} = \frac{\chi\mathcal{C}}{\Delta t}M_L\mathbf{V}^n - \chi M\mathbf{F}^n \qquad \text{for half-lumping} \tag{8}$$

Note that when mass lumping is used together with OS (lumping the two mass matrices in (6b)), it automatically means that the lumped mass matrix has multiplied the reaction terms. In other words, *mass lumping with operator splitting necessarily means full-lumping.*

2.2 Test problems

We will consider four problems when comparing the schemes:

The anisotropic benchmark - this is precisely the benchmark problem [1]. The region is cuboid and of size 2cm by 0.7cm by 0.3cm, it has fibres defined to be parallel to the long (2cm) direction, and transversely isotropic conductivities. The fibre conductivity is 1.334mS/cm, the cross-fibre conductivity is 0.176mS/cm, and the other parameters are $\chi = 1400$/cm and $\mathcal{C} = 1\mu$F/cm^2. The ten Tusscher 2006 (epicardial variant) cell model [19] is used. A cubic region of size 0.15cm at one corner of the tissue is stimulated, for 2ms at -50000μA/cm^3.

The isotropic benchmark - this is identical to the anisotropic benchmark except we set the fibre and cross-fibres conductivities to each equal 1.334mS/cm.

1D convergence - to study convergence using more general spatial-temporal norms, we will also run simulations on a 1D fibre. This is the benchmark definition restricted to one dimension (the X-direction), taking the tissue to be of length 0.5cm. The stimulus needs to be continuous to obtain expected convergence rates, so the stimulus used here increases linearly, from -200000μA/cm^3 at $x = 0$, to 0 at $x = 0.05$cm.

Realistic geometry - here we use a detailed rabbit bi-ventricular mesh [5], which is comprised of 4.3m nodes and 24.2m elements and has an average edge-length of 0.0125cm. Fibre directions are provided, and transversely isotropic conductivities are used. All parameter values are the same as in the anisotropic benchmark. The Luo-Rudy 1991 cell model [20] is used, and a small region at the apex of the tissue is stimulated.

3 Mass lumping in cardiac electro-physiology

The use of mass lumping in the context of the FE solution of the mono/bidomain equations is common practice, with various authors having independently derived FE discretisations of the mono/bidomain equations including mass lumping. The choice is often justified by the savings in computational cost that the technique provides. However, we can find no comprehensive analysis of the error introduced. To the best of our knowledge, Vigmond et al. [21] and Pennacchio et al. [22] are the first works to mention the use of mass lumping for the FE solution of the bidomain equations. The first article chooses the parabolic-elliptic formulation of the bidomain equations, decoupling the PDEs and uses an explicit time discretisation on the parabolic PDE, resulting in a linear system $A\mathbf{x} = \mathbf{b}$ with A being a mass matrix. Mass lumping then avoids the need to solve a linear system, since a lumped mass matrix is trivially invertible. The second article chooses the parabolic-parabolic formulation and solves both PDEs together, which leads to a linear system with a 2×2 block structure with mass matrices in the $(0, 1)$ and $(1, 0)$ blocks. Here, the use of mass lumping reduces matrix bandwidth, storage requirements and, in general, the computational cost of linear system solution algorithms involving matrix-vector products (e.g. Krylov subspace methods), or updates based on the coefficients of the matrix (e.g. stationary iterative methods). However, neither of these two publications analyse the error introduced, or provide references to justify the decision to introduce the mass lumping approximation. We believe that the justification for mass lumping in these works is implicit, and based on classic results in FE analysis proving that the order of convergence of the FE solution of parabolic problems is unaffected by mass lumping. Furthermore, mass lumping is often used in several fields of scientific simulation and can be therefore considered “common practice”.

The analytical proof that mass lumping is justifiable is given in the works of Thomée and Chen [23, 18]. Theorem 15.1 of [18] gives the following bound: for the heat equation $u_t = \nabla^2 u + f$, with boundary condition $u = 0$ on $\partial\Omega$ and initial condition $u(0, x) = u_0(x)$ (we assume u_0 lies in the FE solution space for the purposes of stating this theorem), and under further assumptions, the solution, $u_h(t)$, of the semi-discrete problem in which the mass matrix multiplying the time-derivative has been lumped, satisfies

$$\|u_h(t) - u(t)\|_{L^2(\Omega)} \leq Ch^2 \left(\|u_0\|_{H^2(\Omega)} + \|u(t)\|_{H^2(\Omega)} + \left(\int_0^t \|u_t\|^2_{H^2(\Omega)} \, \mathrm{d}s \right)^{1/2} \right) \tag{9}$$

where C is a constant dependent only on Ω. A similar theorem states that the error in the $H^1(\Omega)$ norm is $\mathcal{O}(h)$. (Here $H^p(\Omega)$ denote standard Sobolev spaces). The conclusion is that mass lumping does not affect rates of convergence. However, this assumes that the $\int f\psi_j \, \mathrm{d}V$ terms in the FE discretisation (i.e. the terms corresponding to $\mathbf{b}^n$ in (2)) are integrated exactly, and therefore *only applies to half-lumping*.

Wendlend et al. [24] provide a comprehensive empirical survey on the use of mass lumping for the FE solution of advection-diffusion equations. Here, the authors point out the lack of agreement regarding the suitability of the technique and identify publications both favourable and contrary to the use of mass lumping. They evaluate the impact of mass lumping on two simple benchmarks of diffusion-dominated and advection-dominated transport.

To the best of our knowledge, only two papers discuss the effect of mass lumping in cardiac electro-physiological simulations. In [25], it is stated that FE convergence accuracy is unchanged by lumping but that errors may be increased by a constant factor, which can be observed as wave slow-down. No precise results are presented however, and the effects of anisotropy, or the difference between full-lumping and half-lumping, is not discussed. A very recent paper [26] includes a study of the effect of lumping, and reaches some similar conclusions to this paper. The present article differs from [26] by studying numerical convergence with decreasing h, and examining more test cases.

4 Results

4.1 Anisotropic benchmark

In this section, we repeat the anisotropic benchmark simulation described in Section 2.2, with five numerical schemes:

1. SVI without mass lumping
2. ICI without mass lumping
3. Strang operator splitting (OS) without mass lumping
4. ICI with full-lumping
5. ICI with half-lumping.

The results of the schemes are given in Figure 2.

Firstly, we observe that the results for OS and ICI are almost identical, which is as expected given the discussion on their spatial similarities. For this reason, for the remainder of this paper we will often refer to ICI and OS together. Hence, for OS *with* mass lumping, see the ICI with full-lumping results.

The results of SVI compared with ICI/OS are noticeably different, even on the finest mesh—the final node depolarises after 42ms using SVI compared to 38ms for ICI/OS. That ICI leads to overly large conduction velocities, even at mesh resolutions of 0.01cm, was observed in [12], where it is discussed in detail.

The ICI with full-lumping results in Figure 2 show significantly increased activation times (i.e. significantly decreased conduction velocities) over ICI with no lumping. Even on the finest mesh, the activation time of the final node is noticeably larger for ICI with full-lumping. On the coarsest mesh, the relative error in the conduction velocity is huge, of the order to 250%. Finally, consider the ICI with half-lumping results. Here, the large slow-down of CV on the coarse meshes is not observed; in fact CV increases with coarsening.

4.2 Isotropic benchmark

Next we run the schemes on the isotropic benchmark simulation described in Section 2.2, in order to investigate the extent to which it is small cross-fibre conductivities that causes the large errors observed in the anisotropic benchmark. Since OS has been shown to be virtually identical to ICI, we now consider only: SVI; ICI; ICI with full-lumping; and ICI with half-lumping.

The results are given in Figure 3. Here the coarse mesh conduction velocity errors for full-lumping are substantially smaller, although still significant (the activation time of the

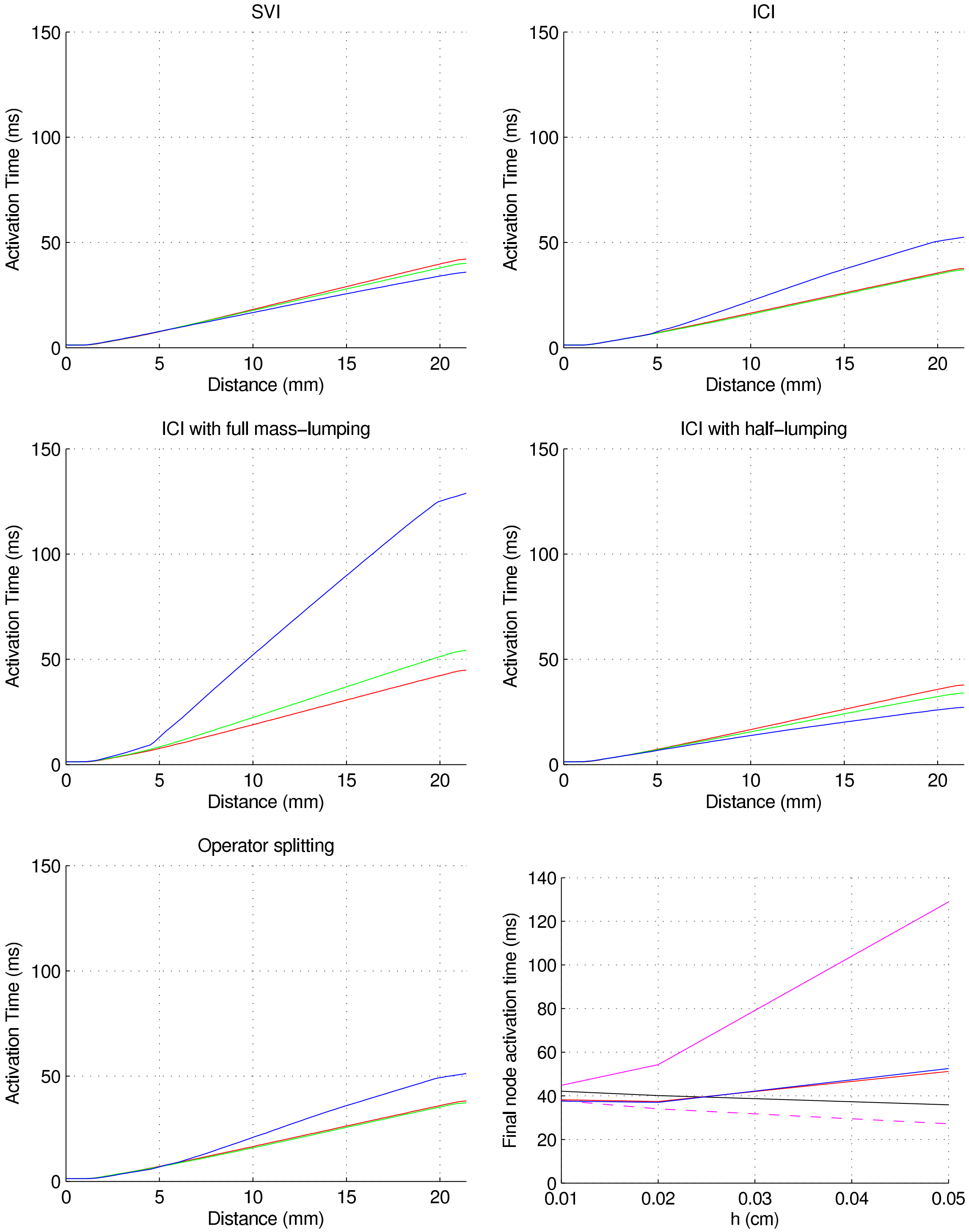

Figure 2: All figures except bottom right: Benchmark simulation results using various methods. Activation time across a diagonal line in the mesh, using stepsizes of: $h = 0.01$cm (red), $h = 0.02$cm (green) and $h = 0.05$cm (blue). This figure should be compared with Figure 1. Bottom right: Depolarisation time of the opposite-corner-node, against h, for: SVI (black line), ICI (blue), operator splitting (red), ICI with full-lumping (purple), ICI with half-lumping (dashed purple).

final node for ICI with full-lumping on the coarsest mesh is about 49ms, compared to about 35ms for simulations on the finest mesh). The half-lumping results are again an improvement over full-lumping, although the errors introduced are not particularly small—about half the magnitude of the full-lumping errors.

4.3 1D spatio-temporal convergence analysis

The benchmark survey paper used a very simple metric—nodal activation times—to compare results, as comparing spatio-temporal results would have required multiple international groups providing many gigabytes of data, probably in a variety of file formats. More appropriate metrics, infeasible for the benchmark paper but applicable here, are the following spatio-temporal norms on $[0,T]\times\Omega$:

$$\|u_h\|_{L^\infty(0,T;L^2(\Omega))} = \max_{t\in[0,T]} \|u_h(t)\|_{L^2(\Omega)}, \qquad \|u_h\|_{L^2(0,T;H^1(\Omega))} = \int_0^T \|u_h(t)\|_{H^1(\Omega)}\,\mathrm{d}t. \tag{10}$$

(Or more precisely, essential supremum instead of maximum in the definition of $L^\infty(0,T;L^2(\Omega))$, although these are equivalent for discrete solutions). We refer to these as $L^\infty(L^2)$ and $L^2(H^1)$ norms.

Since exact solutions are not available for the monodomain equation with physiological cell models, we consider a one-dimensional analogue of the benchmark problem, as described in Section 2.2, for which we can compare against a solution on a very fine mesh. We also consider the heat equation with a reaction term $f(x) = \cos(\pi x)$ (equivalent to the monodomain equation with non-physiological parameters and cell model), for which an exact solution is easily computed, so that precise numerical error can be determined. In both these experiments we let $h\to 0$ with $\Delta t/h$ fixed.

Figure 4(a) displays the exact error in these two norms for the heat equation problem using: (i) ICI; (ii) ICI-with-full-lumping; (iii) and ICI-with-half-lumping. In all cases $\mathcal{O}(h^2)$ convergence is observed in the $L^\infty(L^2)$ norm and $\mathcal{O}(h)$ convergence in the $L^2(H^1)$ norm. These are the rates predicted assuming exact integration of the reaction term, with or without half-lumping. We see that (for this example) the approximations of ICI, and of full-lumping, do not affect convergence rate.

The results for the 1D benchmark problem, using physiological parameter values and cell model, are given in Figure 4(b). The results suggest that ICI, full- and half-lumping all produce nearly identical convergence results for small enough h, but that there are differences at the higher spatial range, $h\in[0.01,0.05]$cm. At these stepsizes, either form of lumping introduces error in the $L^\infty(L^2)$ norm, and full-lumping is slightly worse than half-lumping and ICI in the $L^2(H^1)$ norm. The main point is that the spatial stepsizes used in simulation studies are certainly greater than those required to obtain the assymptoptic rates of convergence predicted for the schemes.

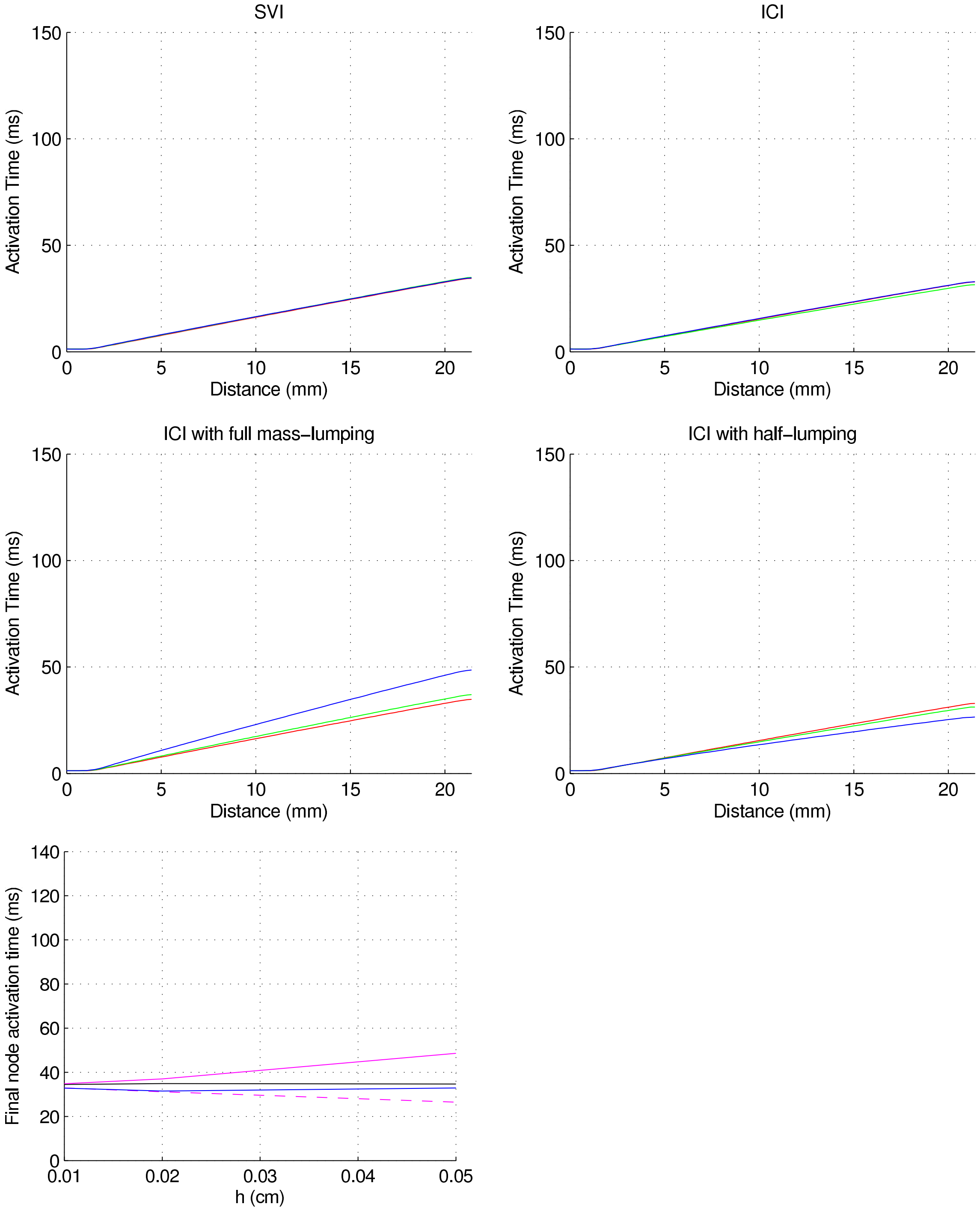

Figure 3: All figures except bottom left: *Isotropic* version of the benchmark simulation, using various methods. Activation time across a diagonal line in the mesh, using stepsizes of: $h = 0.01$cm (red), $h = 0.02$cm (green) and $h = 0.05$cm (blue). Bottom left: Depolarisation time of the opposite-corner-node, against h, for: SVI (black line), ICI (blue), ICI with full-lumping (purple), ICI with half-lumping (dashed purple).

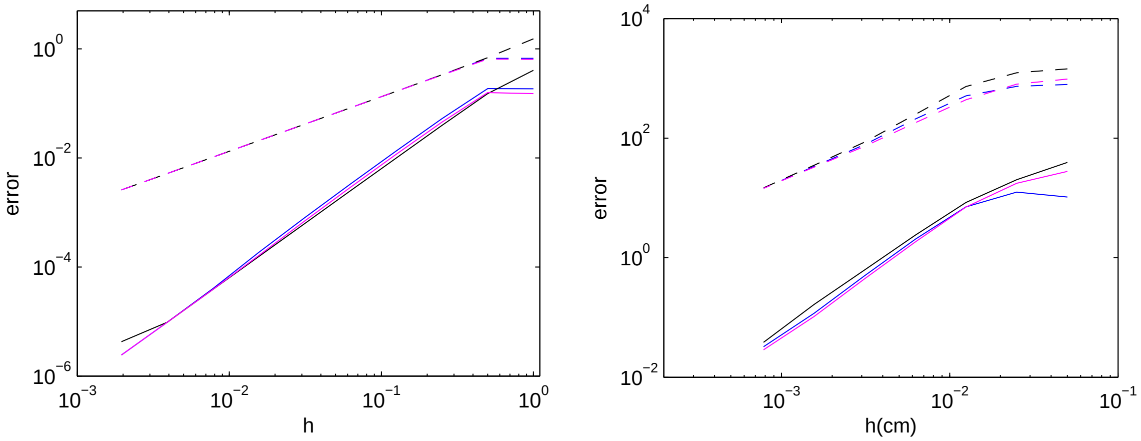

Figure 4: Errors in different norms and using different schemes on (a) the heat equation with reaction term $f(x) = \cos(\pi x)$, and (b) the monodomain equation with a physiological cell model. The results in (a) are exact errors, the results in (b) are estimates based on comparing against a solution on a more refined mesh. In both plots the dotted lines are the error in the $L^\infty(L^2)$ norm and the solid lines are the error in the $L^2(H^1)$ norm, and the schemes are: ICI (blue); ICI with full-lumping (black); and ICI with half-lumping (purple).

4.4 Cardiac geometry

Finally, we consider ICI, ICI with full-lumping, and ICI with half-lumping on a realistic rabbit geometry, as described in Section 2.2. On a realistic geometry with realistic fibre orientations, errors in fibre direction CV and errors in cross-fibre CV will interact in a complicated manner which is difficult to predict.

The results are given in Figures 5 and 6. This mesh is very fine compared to many to have been used in simulation studies. We observe that the effect of full-lumping cannot be neglected even on this mesh: despite the high-resolution, full-lumping very noticeably decreases total conduction velocity. There is little difference between the results using no lumping and half-lumping however—it is very difficult to distinguish differences in the images in Figure 5, although there are some very small variations, in particular just behind the wavefront.

The location of the full-lumped wave after 80ms is approximately equal to the location of the non-lumped wave after about 74ms, so we can estimate the change in CV due to full-lumping to be about 8% at this resolution (0.0125cm). We expect the disparities to increase on coarser meshes.

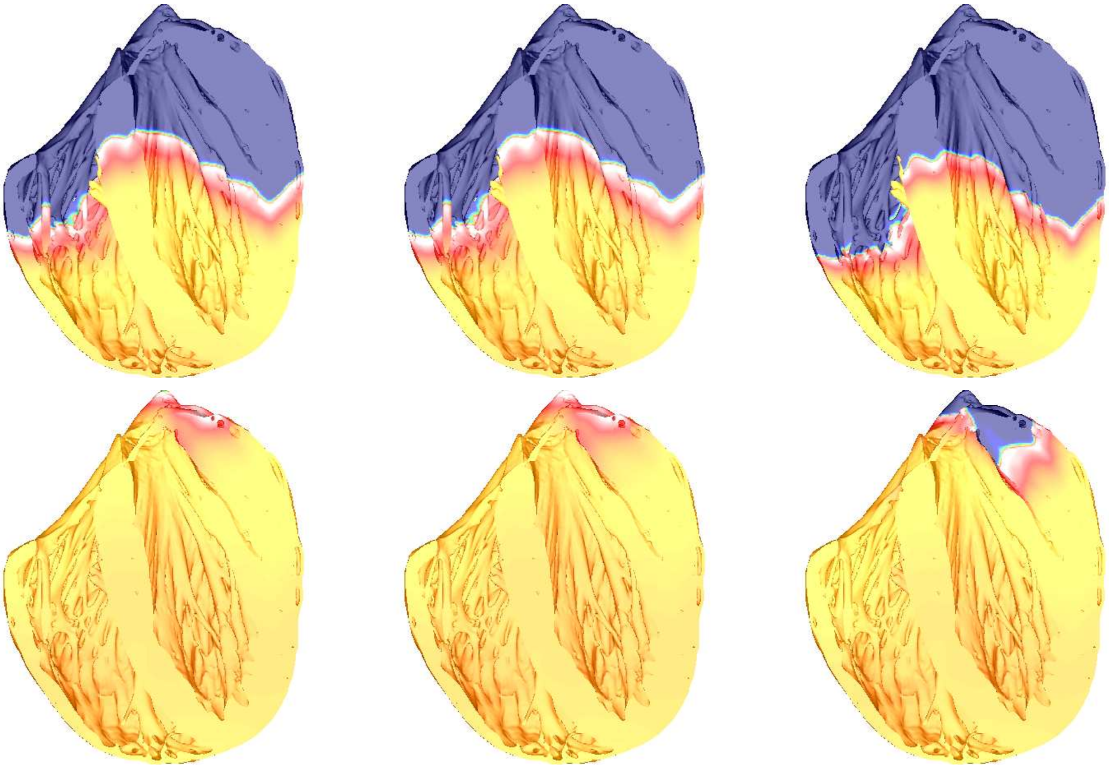

Figure 5: Voltage after 50ms (top row) and 80ms (bottom row) on a realistic rabbit geometry (average edge-length is 0.0125cm), using no lumping (left); half-lumping (middle); and full-lumping (right) (all ICI).

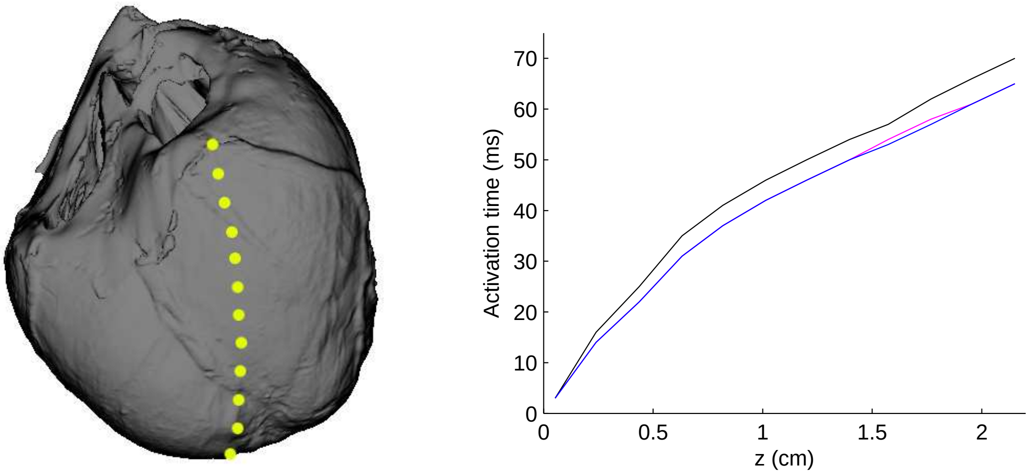

Figure 6: On a fine rabbit heart mesh (average edge-length of 0.0125cm), for the 12 highlighted nodes in the left figure, the right figure plots the activation time versus z (distance along the long axis), for no mass lumping (blue), half-lumping (purple) and full-lumping (black) (all ICI)

5 Interpretation of the benchmark comparison paper [1]

5.1 FE results

We now use the results in the previous section to interpret the results in [1]. To explain the differences between the various FE codes in the benchmark paper, compare Figures 1 and 2.

Firstly, the SVI results in Figure 2 plot the same data as the Chaste results in Figure 1 (code A). Now consider Fenics (code H in Figure 1), which used OS without mass lumping, but would have differed in other respects to the simulations we carried out. We see that the ICI/OS without mass lumping results (Figure 2, top right) are very closely matched to the Fenics results in Figure 1, suggesting that the difference between the Chaste and Fenics results can be mainly attributed to the use of SVI.

The remaining finite element codes in Figure 1 (codes B, C, E and F) all used operator splitting and mass lumping (full-lumping), but they differed in other aspects of their schemes. The ICI with full-lumping results in Figure 2 (which we expect to be very similar to OS with full-lumping) are very similar to the results for these codes in Figure 1 (compare Figure 2, middle-left, with codes B, C, E and F in Figure 1). Since we know that half-lumping does not lead to a large slow-down of CV on coarse meshes, this is extremely strong evidence that it is the use of full mass lumping, arising from the combination of operator splitting and lumped mass matrices, that is the fundamental cause of the differences between these FE codes and Chaste/Fenics.

5.2 FE with full-lumping interpreted as finite differences

We see from Figure 1 that the four FE codes which used operator splitting with mass lumping (full-lumping) produced very similar results to the five finite difference codes. This is unsurprising. Consider the ICI scheme, and suppose we have a 1D regular mesh, and that the conductivity is constant in space. It is well-known that on a regular 1D grid the stiffness matrix arising in the FE method is tri-diagonal with entries $\frac{-\sigma}{h}$, $\frac{2\sigma}{h}$, $\frac{-\sigma}{h}$ on each row which corresponds to an internal node—i.e. the standard second-order finite difference operator multiplied by $-h\sigma$. Also, the mass matrix is tri-diagonal with entries $\frac{h}{6}$, $\frac{2h}{3}$, $\frac{h}{6}$ on each (internal) row, which becomes upon lumping the identity matrix multiplied by h. The use of full-lumping therefore converts (4) into $\frac{h\chi\mathcal{C}}{\Delta t}\mathbf{V}^{n+1} + K\mathbf{V}^{n+1} = \frac{h\chi\mathcal{C}}{\Delta t}\mathbf{V}^{n} - h\chi\mathbf{F}^{n}$, which, alternatively stated, is

$$\chi\mathcal{C}\frac{V_i^{n+1} - V_i^n}{\Delta t} - \sigma\frac{V_{i-1}^{n+1} - 2V_i^{n+1} + V_{i+1}^{n+1}}{h^2} = -\chi F_i^n,$$

a standard finite difference approximation. The point is that full-lumping causes the reaction term to be treated locally in the FE method (rather than being integrated), as with finite differences. In higher dimensions, or with operator splitting, or other time-discretisations, the FE discretisation will not be identical to this finite difference scheme, but full-lumping will still lead to the reaction terms being treated in this localised manner, again, as with finite differences.

Note that it follows that all comments in this paper on the accuracy of ICI with full-lumping (in the cardiac electro-physiological setting) can also be applied to finite differences for cardiac electro-physiology.

6 Discussion

The aim of this paper has been to explain reported differences between solutions obtained using several cardiac electro-physiology solvers on an unambiguously-defined common problem. To do this, we have compared a selection of algorithms for cardiac electro-physiological simulation. These were all based on a common time and space-discretisation but differed in: (i) the manner in which the ionic current was computed—interpolation of the nodal ionic currents onto quadrature points (ICI), which is related to operator splitting methods, versus interpolation of cell state-variables onto quadrature points (SVI)—and (ii) how much mass lumping is applied. The choice of algorithms studied was motivated by the aim of explaining [1], and therefore there are various possibilities that have not been considered, such as lumping only the mass matrix multiplying reaction terms, or lumping the time-derivative mass matrices in the SVI scheme. These are left for further work.

By observing the impact of these approximations, we were able to interpret the results of the benchmark paper. Differences between Chaste and Fenics were attributed to the use of SVI, for which more details can be found in [12]. (Users of Chaste should be aware that Chaste uses ICI by default, and that SVI must be turned on if it is required—release 2.2

onwards, see documentation). For the remaining codes, which comprise both finite element and finite difference solvers, conduction velocity decreased significantly as h increased. In the finite element context, we have explained how mass lumping together with operator splitting leads to effectively lumping a mass matrix which multiplies the nodal ionic currents (full-lumping), which differs from standard mass lumping for which the reaction terms are unaltered (half-lumping). We have shown how full-lumping introduces significant error over no lumping or half-lumping, and have argued that this is the fundamental cause for the decreased conduction velocities in the other FE codes. We have also explained why ICI or operator splitting finite element schemes with full-lumping provide similar results to finite difference schemes. In fact, the results can be interpreted in terms of the way in which the ionic current (reaction term) is assimilated into the scheme: (i) for SVI the current is evaluated by linear state-variable interpolation and integrated; (ii) for ICI with no or half-lumping, or OS with no lumping, the current is (effectively) linearly interpolated and integrated; and (iii) for ICI with full-lumping, OS with lumping, or finite differences, the current is only used locally. All the schemes are convergent, but unfortunately, for commonly-used mesh resolutions the differences are pronounced.

For most of the simulations in this paper, we used the ten Tusscher cell model (epicardial variant) [19], since our test problems are directly derived from the benchmark problem defined in [1]. The choice of cell model will also have an impact on numerical error—the choice of cell model affects the steepness of the action potential, which affects the conduction velocity. The importance of the cell model and of integration of the ionic current on conduction velocity is considered in more detail in [12]. We expect that this choice of physiological cell model, with a relatively high upstroke velocity, will have the effect of magnifying numerical errors. In particular, it should be noted that if Figure 1 was repeated with a simpler cell model, such as the Fitz-Hugh-Nagumo model [27], the discrepancies between the results on different meshes would be expected to be smaller, and similarly with the simulations in this paper.

Theory which guarantees that mass lumping does not affect rates of convergence assumes reaction terms in the FE discretisations are integrated exactly, and therefore only applies to half-lumping methods. In the computational experiments in this paper, assymptotic rates of convergence were identical for no lumping, half-lumping and full-lumping; however it was shown that full-lumping generally leads to significantly greater errors than half-lumping for practical values of spatial stepsize ($h \in [0.01, 0.05]$cm). In particular, full-lumping leads to very large reductions in cross-fibre CV in anisotropic problems. Half-lumping leads to less (although still appreciable) cross-fibre CV error. On a high-resolution rabbit mesh with prescribed fibre directions and anisotropic conductivities, full-lumping conduction velocities were about 8% smaller than those with no lumping or half-lumping. Even 0.01cm spatial resolutions (corresponding to state-of-the-art anatomical detail in realistic geometry meshes [5]) are not sufficiently small for the impact of these different implementation choices to be neglected. (It is worth noting that this re-raises questions on the appropriateness of the bidomain equations. Although it is not invalid to use fine meshes with edge-lengths smaller than individual myocytes to obtain highly accurate solutions of the continuum equations, at this resolution it may be more appropriate, and indeed more efficient, to solve a non-continuum model.)

This paper has only focussed on the accuracy at particular mesh resolutions of the selected numerical methods, we have not considered the computational cost of the schemes—certainly, the less accurate schemes will often be significantly less expensive at given resolutions (for example, explicit time-discretisations and FD or FE with half-lumping), and it may be that the best choice is such a scheme on a suitably fine mesh. Overall, we conclude that numerical scheme, mesh resolution, and acceptable numerical error ought to be considered carefully together when simulation studies are performed.

Acknowledgements

PP is supported by the EPSRC-funded OxMOS project (EP/D048400/1), MOB is supported by the European Commission preDiCT project (224381), and SAN is supported on EPSRC grants EP/F043929/1 and EP/F059361/1.

References

[1] Niederer S, Kerfoot E, Benson A, Bernabeu M, Bernus O, Bradley C, Cherry E, Clayton R, Fenton F, Garny A, Heidenreich E, Land L, Malecker M, Pathmanathan P, Plank G, Rodriguez J, Roy I, Sachse F, Seemann G, Shavhaug O, Smith N. Verification of cardiac tissue electrophysiology simulators using an N-version benchmark. *Royal Society Philosophical Transactions A*, 2011; **369**(1954):4331–4351.

[2] Vigmond E, Vadakkumpadan F, Gurev V, Arevalo H, Deo M, Plank G, Trayanova N. Towards predictive modelling of the electrophysiology of the heart. *Experimental Physiology* 2009; **94**(5):563.

[3] Byrd I, Rogers J, Smith W, Pollard A. Comparison of conventional and biventricular antitachycardia pacing in a geometrically realistic model of the rabbit ventricle. *Journal of Cardiovascular Electrophysiology* 2004; **15**(9):1066–1077.

[4] Trayanova N. Defibrillation of the heart: insights into mechanisms from modelling studies. *Experimental Physiology* 2006; **91**(2):323–337.

[5] Bishop M, Plank G, Burton R, Schneider J, Gavaghan D, Grau V, Kohl P. Development of an anatomically detailed MRI-derived rabbit ventricular model and assessment of its impact on simulations of electrophysiological function. *American Journal of Physiology - Heart and Circulatory Physiology* 2010; **298**(2):H699.

[6] Pitt-Francis J, Pathmanathan P, Bernabeu M, Bordas R, Cooper J, Fletcher A, Mirams G, Murray P, Osbourne J, Walter A, *et al.*. Chaste: a test-driven approach to software development for biological modelling. *Computer Physics Communications* 2009; **180**:2452–2471.

[7] Pathmanathan P, Bernabeu M, Bordas R, Cooper J, Garny A, Pitt-Francis J, Whiteley J, Gavaghan D. A numerical guide to the solution of the bidomain equations of cardiac electrophysiology. *Progress in Biophysics and Molecular Biology* 2010; **102**:136–155.

[8] Mardal K, Skavhaug O, Lines G, Staff G, Ødegård Å. Using python to solve partial differential equations. *Computing in Science & Engineering* 2007; :48–51.

[9] Keener J, Sneyd J. *Mathematical Physiology*, *Interdisciplinary Applied Mathematics*, vol. 8. Springer, 1998.

[10] Clayton R, Panfilov A. A guide to modelling cardiac electrical activity in anatomically detailed ventricles. *Progress in Biophysics and Molecular Biology* 2008; **96**(1-3):19–43.

[11] Reddy J. *An Introduction to the Finite Element Method*. McGraw–Hill, 1993.

[12] Pathmanathan P, Mirams G, Southern J, Whiteley J. The significant effect of the choice of ionic current integration method in cardiac electro-physiological simulations. *International Journal for Numerical Methods in Biomedical Engineering (in press)* 2011.

[13] Sundnes J, Lines G, Cai X, aand K Mardal BN, Tveito A. *Computing the electrical activity in the heart*. Springer Verlag, 2006.

[14] Austin T, Trew M, Pullan A. Solving the cardiac bidomain equations for discontinuous conductivities. *IEEE Transactions on Biomedical Engineering* 2006; **53**(7):1265–1272.

[15] dos Santos R, Plank G, Bauer S, Vigmond E. Parallel multigrid preconditioner for the cardiac bidomain model. *Biomedical Engineering, IEEE Transactions on* 2004; **51**(11):1960–1968.

[16] Trangenstein J, Kim C. Operator splitting and adaptive mesh refinement for the Luo-Rudy I model. *Journal of Computational Physics* 2004; **196**(2):645–679.

[17] Vigmond E, dos Santos RW, Prassl A, Deo M, Plank G. Solvers for the cardiac bidomain equations. *Progress in Biophysics and Molecular Biology* 2008; **96**(1–3):3–18.

[18] Thomée V. *Galerkin Finite Element Methods for Parabolic Problems*, *Springer Series in Computational Mathematics*, vol. 25. Springer - Verlag: Berlin - Heidelberg - New York, 1997.

[19] ten Tusscher KHWJ, Panfilov AV. Alternans and spiral breakup in a human ventricular tissue model. *Am J Physiol Heart Circ Physiol* 2006; **291**(3):H1088–1100.

[20] Luo C, Rudy Y. A model of the ventricular cardiac action potential. Depolarization, repolarization, and their interaction. *Circulation Research* 1991; **68**(6):1501.

[21] Vigmond E, Aguel F, Trayanova N. Computational techniques for solving the bidomain equations in three dimensions. *IEEE Transactions on Biomedical Engineering* 2002; **49**(11):1260–1269.

[22] Pennacchio M, Simoncini V. Efficient algebraic solution of reaction-diffusion systems for the cardiac excitation process. *J. Comput. Appl. Math.* August 2002; **145**:49–70, doi:10.1016/S0377-0427(01)00535-0.

[23] Chen C, Thomée V. The lumped mass finite element method for a parabolic problem. *The Journal of the Australian Mathematical Society. Series B. Applied Mathematics* 1985; **26**(03):329–354.

[24] Wendland E, Schulz HE. Numerical experiments on mass lumping for the advection-diffusion equation. *Revista Minerva* 2005; **2**(2):227–233.

[25] Weiser M, Erdmann B, Deuflhard P. On efficiency and accuracy in cardioelectric simulation. *Progress in Industrial Mathematics at ECMI 2008*, *Mathematics in Industry*, vol. 15. Springer Berlin Heidelberg, 2010; 371–376.

[26] Vázquez M, Arís R, Houzeaux G, Aubry R, Villar P, Garcia-Barnés J, Gil D, Carreras F. A massively parallel computational electrophysiology model of the heart. *International Journal for Numerical Methods in Biomedical Engineering* 2011 (online ahead of print).

[27] Nagumo J, Arimoto S, Yoshizawa S. An active pulse transmission line simulating nerve axon. *Proc IRE* 1962; **50**:2061–2070.