Journal of Computational Physics 228 (2009) 3625–3639

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A model-based block-triangular preconditioner for the Bidomain system in electrocardiology

L. Gerardo-Giorda [a], L. Mirabella [a,b,*], F. Nobile [b], M. Perego [a,b], A. Veneziani [a]

[a] Department of Mathematics and Computer Science, Emory University, 201 Dowman Drive, Atlanta, GA 30322, USA
[b] MOX, Department of Mathematics, Politecnico di Milano, piazza L. da Vinci 32, 20133 Milano, Italy

ARTICLE INFO

Article history:
Received 31 May 2008
Received in revised form 27 January 2009
Accepted 30 January 2009
Available online 12 February 2009

Keywords:
Preconditioning
Computational electrocardiology
Bidomain and Monodomain models

ABSTRACT

We introduce a preconditioner for the solution of the Bidomain system governing the propagation of action potentials in the myocardial tissue. The Bidomain model is a degenerate parabolic set of nonlinear reaction-diffusion equations. The nonlinear term describes the ion flux at the cellular level. The degenerate nature of the problem results in a severe ill conditioning of its discretization. Our preconditioning strategy is based on a suitable adaptation of the Monodomain model, a simplified version of the Bidomain one, which is by far simpler to solve, nevertheless is unable to capture significant features of the action potential propagation. The Monodomain preconditioner application to a non-symmetric formulation of the Bidomain system results at the algebraic level in a lower block-triangular preconditioner. We prove optimality of the preconditioner with respect to the mesh size, and corroborate our theoretical results with 3D numerical simulations both on idealized and real ventricle geometries.

1. Introduction

The *Bidomain model* is currently considered one of the most complete models for the description of electrical potential in the cardiac tissue (see e.g. [27,33,30,18]). It consists of a system of nonlinear unsteady partial differential equations including the dynamics of intra and extracellular potentials. The degenerate parabolic nature of this system implies high computational costs in the numerical solution. For this reason in many applications a simplified formulation of the problem, called *Monodomain model*, has been preferred [26,19]. However, the Monodomain model is derived under the assumption of proportionality between the intra and extracellular conductivity tensors. This assumption is in general quite unrealistic and some relevant patterns in the potential propagation can be missed by this model. Several studies have been therefore devoted to devise effective preconditioners for the solution of the Bidomain system (see e.g. [12,24,38,25,39,9,36,35]). Discretization of the problem is usually carried out by resorting to finite elements for the space variables and semi-implicit time advancing schemes allowing one to skip the solution of computationally expensive nonlinear systems. Nevertheless, the algebraic system obtained after the discretization is intrinsically ill-conditioned. Preconditioning strategies available in the literature are often based on a proper decomposition of the computational domain for setting up parallel preconditioners, or on suitable multigrid schemes still coupled with parallel architectures.

In this work, we present a different approach. As a matter of fact, we propose to use the *Monodomain model as a preconditioner* in solving an appropriate reformulation of the Bidomain system. At the discrete level, still assuming to exploit a

[*] Corresponding author. Address: MOX, Department of Mathematics, Politecnico di Milano, piazza L. da Vinci 32, 20133 Milano, Italy.
E-mail addresses: luca@mathcs.emory.edu (L. Gerardo-Giorda), lucia.mirabella@mail.polimi.it (L. Mirabella), fabio.nobile@polimi.it (F. Nobile), mauro.perego@polimi.it (M. Perego), ale@mathcs.emory.edu (A. Veneziani).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.01.034

finite-element/semi-implicit discretization, our Monodomain-based preconditioner can be reinterpreted as a block-triangular preconditioner of the Bidomain system. Block-triangular preconditioners are used in many fields of scientific computing ranging from fluid-dynamics to Maxwell equations. Convergence estimates have been obtained, however, only for specific applications and in particular saddle point problems (see [6] and the bibliography quoted therein). Here, we provide a rigorous proof of optimality of our preconditioner based on a frequency analysis, similar to the one carried out in [17,4] for advection diffusion and Maxwell problems, respectively. Optimality of the preconditioner was expected as a consequence of two circumstances. On the one hand, the non-symmetric formulation of the Bidomain problem we propose is such that the block of the original Bidomain matrix dropped in the preconditioner is made quantitatively small by an appropriate selection of a parameter. On the other hand, in parabolic problems the most unfavorable part in terms of dependence of the condition number on the mesh size is the elliptic one (stiffness matrix) in comparison with the mass matrices (see [16]). The proposed preconditioner actually retains the elliptic core of the Bidomain system.

Optimality of the preconditioner is particularly significant for 3D simulations on real geometries retrieved by medical data such as SPECT or MRI (see e.g. [3,8]). In view of this, numerical results presented here will refer to both simplified and real ventricular geometries as well.

The outline of the paper is as follows. In Section 2 we introduce the Bidomain and the Monodomain models. In Section 3 we introduce the preconditioned problem and its relevant features. Moreover, we introduce the flexible GMRES (right) preconditioned iterations of the problem at hand. In Section 4 we prove the optimality of the preconditioner. Numerical results of Section 5 refer to 3D simulations, carried out with `LifeV` [1], a finite-element solver whose linear algebra solver is based on `Trilinos` packages [2]. More precisely, we present performance comparison between our preconditioner and the algebraic ILU preconditioner, which is one of the best general purpose preconditioners for serial architectures. Optimality of the preconditioner is confirmed through numerical tests, the number of iterations being essentially independent of the mesh size. Comparison in terms of CPU time is favorable too.

Finally, we point out that we assume to work here on a serial architecture. However, it is worth observing that a proper implementation of our preconditioner exploiting parallel architectures following guidelines similar to the papers mentioned above is expected to be an important development of the present work.

2. The Bidomain and Monodomain models

Bidomain model. The myocardial tissue is composed of elongated cells, the *cardiac fibers*, connected to each other by gap junctions and surrounded by an extracellular medium. From a mathematical point of view, this structure can be modeled as a continuum in which the electrical variables are obtained as the average of the single cell properties, after a homogenization process [14,28,5]. The cardiac tissue can be represented as a superposition of intra and extracellular media connected by a cell membrane dislocated in the domain. The *Bidomain* model should take into account the direction of the cardiac fibers. Anatomical studies show that the fibers direction rotates counterclockwise from epicardium to endocardium and that they are arranged in sheets, running across the myocardial wall [12,21,33]. We set the problem in a bounded region $\Omega \subset \mathbb{R}^3$, and we assume that the cardiac tissue is characterized at each point by three directions: $\boldsymbol{a}_l$ along the fiber, $\boldsymbol{a}_t$ orthogonal to the fiber direction and in the fiber sheet and $\boldsymbol{a}_n$ orthogonal to the sheet. The intra and extracellular media present different conductivity values in each direction. We denote by $\sigma_i^l(\boldsymbol{x})$ (resp. $\sigma_e^l(\boldsymbol{x})$) the intracellular (resp. extracellular) conductivity in $\boldsymbol{a}_l(\boldsymbol{x})$ direction at point $\boldsymbol{x} \in \Omega$, and similarly by $\sigma_i^t(\boldsymbol{x})(\sigma_e^t(\boldsymbol{x}))$ and $\sigma_i^n(\boldsymbol{x})(\sigma_e^n(\boldsymbol{x}))$ the conductivities along $\boldsymbol{a}_t(\boldsymbol{x})$ and $\boldsymbol{a}_n(\boldsymbol{x})$. We will use throughout the paper the notation $\sigma_\tau^l(\boldsymbol{x}), \sigma_\tau^n(\boldsymbol{x}), \sigma_\tau^t(\boldsymbol{x})$ where $\tau = i, e$ for indicating intra and extracellular conductivity in a compact form.

The intra and extracellular local anisotropic conductivity tensors read therefore

$$\mathbf{D}_\tau(\boldsymbol{x}) = \sigma_\tau^l(\boldsymbol{x})\boldsymbol{a}_l(\boldsymbol{x})\boldsymbol{a}_l^T(\boldsymbol{x}) + \sigma_\tau^t(\boldsymbol{x})\boldsymbol{a}_t(\boldsymbol{x})\boldsymbol{a}_t^T(\boldsymbol{x}) + \sigma_\tau^n(\boldsymbol{x})\boldsymbol{a}_n(\boldsymbol{x})\boldsymbol{a}_n^T(\boldsymbol{x}) \tag{1}$$

for $\tau = i, e$. Should the myocardium show the same conductivity in both the tangential and normal direction (*axial isotropy*), the tensors simplify in

$$\mathbf{D}_\tau(\boldsymbol{x}) = \sigma_\tau^t\mathbf{I} + (\sigma_\tau^l - \sigma_\tau^t)\boldsymbol{a}_l(\boldsymbol{x})\boldsymbol{a}_l^T(\boldsymbol{x}) \tag{2}$$

for $\tau = i, e$. In the present work, following [12], we assume (2) to hold. Moreover, we assume that $\mathbf{D}_\tau$ fulfills in Ω a uniform elliptic condition.

Let $u_\tau(\tau = i, e)$ be the *intra and extracellular potentials* and $u = u_i - u_e$ be the *transmembrane potential*. The density current in each domain can be computed as $\mathbf{J}_\tau = -\mathbf{D}_\tau\nabla u_\tau$. The net current flux between the intra and the extracellular domain is assumed to be zero as a consequence of the charge conservation in an arbitrary portion of tissue. Let us denote by I_m the ingoing membrane current flow and by χ the ratio of membrane area per tissue volume. We get therefore

$$\nabla \cdot (\mathbf{D}_i\nabla u_i) = \chi I_m = -\nabla \cdot (\mathbf{D}_e\nabla u_e). \tag{3}$$

Here I_m can be further expressed as $I_m = C_m du/dt + I_{ion}(u, \boldsymbol{w})$ where C_m denotes a capacitance and I_{ion} the ionic current, depending on the potential u and on suitable ionic variables that we denote with $\boldsymbol{w}$. The dependence of I_{ion} on u and $\boldsymbol{w}$ has been described in two different ways in the literature. One approach is based on a precise description of ionic channels, see [31,40], and, in particular, we cite the Luo–Rudy phase I model [22]. In the latter case $\boldsymbol{w}$ represents a vector composed of six gate variables and the calcium concentration in the cell. The second approach is based on a purely phenomenological evidence. We mention in particular the FitzHugh–Nagumo [15] and the Rogers–McCulloch [29] models. In this case $\boldsymbol{w}$ represents a scalar variable called *recovery variable*. We do not dwell here upon a specific selection of ionic models, since the preconditioner proposed here is independent of it. In the numerical results presented in Section 5 we will employ either the Luo–Rudy phase I or the Rogers–McCulloch model.

The complete Bidomain model reads

$$\chi C_m \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \frac{\partial}{\partial t} \begin{bmatrix} u_i \\ u_e \end{bmatrix} - \begin{bmatrix} \nabla \cdot \mathbf{D}_i \nabla u_i \\ \nabla \cdot \mathbf{D}_e \nabla u_e \end{bmatrix} + \chi \begin{bmatrix} I_{\text{ion}}(u, \boldsymbol{w}) \\ -I_{\text{ion}}(u, \boldsymbol{w}) \end{bmatrix} = \begin{bmatrix} I_i^{\text{app}} \\ -I_e^{\text{app}} \end{bmatrix}, \tag{4}$$

where I_{ion} depends on the chosen ionic model, and $I_\tau^{\text{app}}(\tau = i, e)$ represent applied external stimuli. The problem is completed by an initial condition $u(\boldsymbol{x}, 0) = u_0$ and boundary conditions on $\partial\Omega$. In particular we prescribe homogeneous Neumann boundary conditions

$$\mathbf{n}^T \mathbf{D}_i \nabla u_i(\boldsymbol{x}, t) = 0 \quad \text{and} \quad \mathbf{n}^T \mathbf{D}_e \nabla u_e(\boldsymbol{x}, t) = 0, \quad \text{on } \partial\Omega \times (0, T), \tag{5}$$

where $\mathbf{n}$ is the unit normal outward-pointing vector on the surface. Conditions (5) correspond to an insulated myocardium. As a consequence of the Gauss theorem, the applied external stimuli must fulfill the compatibility condition

$$\int_\Omega I_i^{\text{app}}\, d\boldsymbol{x} = \int_\Omega I_e^{\text{app}}\, d\boldsymbol{x}. \tag{6}$$

System (4) consists of two parabolic reaction-diffusion equations for u_i and u_e where the vector of time derivatives is multiplied by a singular matrix. The system is thus said to be *degenerate*. The transmembrane potential u is uniquely determined, while the intra and extracellular potentials u_i and u_e are determined up to the same function of time. To fix such arbitrary function we require that u_e has zero average on Ω. Let us define $\boldsymbol{V} = H^1(\Omega) \times H^1(\Omega) \setminus \{[c, c] : c \in \mathbb{R}\}$ and denote by $(\cdot, \cdot)$ the scalar product in L^2. The variational form of the Bidomain problem reads as follows: given I_τ^{app}, find $[u_i, u_e] \in L^2(0, T; \boldsymbol{V})$ such that

$$\chi C_m \left(\frac{\partial u}{\partial t}, \phi \right) + a_i(u_i, \phi_i) + a_e(u_e, \phi_e) + (I_{\text{ion}}(u, \boldsymbol{w}), \phi) = (I_i^{\text{app}}, \phi_i) + (I_e^{\text{app}}, \phi_e) \tag{7}$$

for each $[\phi_i, \phi_e] \in \boldsymbol{V}$, where $\phi = \phi_i - \phi_e$. The forms $a_\tau(v, \phi)$ are defined as $a_\tau(v, \phi) = \int_\Omega \nabla v^T D_\tau \nabla \phi\, d\boldsymbol{x}$. For well-posedness analysis of the Bidomain problem coupled with the FitzHugh–Nagumo ionic model we refer to [14], while for Luo–Rudy I and more general ionic models to [37].

Monodomain model. To overcome high computational costs associated with the Bidomain problem a simplified model has been proposed, the so called *Monodomain* problem. Its derivation can be obtained in two different ways. One way consists in assuming $\mathbf{D}_e = \lambda \mathbf{D}_i$, where λ is a constant to be properly chosen.

Thanks to this assumption, a linear combination of the Bidomain equations with coefficients $\frac{\lambda}{1+\lambda}$ and $-\frac{1}{1+\lambda}$ yields the Monodomain model

$$\begin{cases} \chi C_m \frac{\partial u}{\partial t} - \nabla \cdot (\mathbf{D}^M \nabla u) + \chi I_{\text{ion}}(u, \boldsymbol{w}) = I^{\text{app}} & \text{in } \Omega \times (0, T), \\ u(\boldsymbol{x}, t = 0) = u_0 & \text{in } \Omega, \\ \mathbf{n}^T \mathbf{D}^M \nabla u = 0 & \text{on } \partial\Omega \times (0, T), \end{cases} \tag{8}$$

where $\mathbf{D}^M = \frac{\lambda \mathbf{D}_i}{1+\lambda}$ and $I^{\text{app}} = \frac{\lambda I_i^{\text{app}} + I_e^{\text{app}}}{1+\lambda}$.

Under assumption (2), the parameter λ can be chosen, for instance, by minimizing the functional

$$J = \left(\sigma_e^l - \lambda \sigma_i^l\right)^2 + 2\left(\sigma_e^t - \lambda \sigma_i^t\right)^2$$

for given values of the conductivities. Another way of selecting λ has been proposed in [23]. In general, if we define

$$\lambda_m = \min\left\{ \frac{\sigma_e^l}{\sigma_i^l}, \frac{\sigma_e^t}{\sigma_i^t} \right\}, \quad \lambda_M = \max\left\{ \frac{\sigma_e^l}{\sigma_i^l}, \frac{\sigma_e^t}{\sigma_i^t} \right\}, \tag{9}$$

it is reasonable to choose $\lambda_m \leqslant \lambda \leqslant \lambda_M$.

Another way to derive a Monodomain model, that we will not use in what follows, can be found in [20,10], where the authors mediate the contribution of the intra and extracellular media.

Using the notation introduced in the previous section, the variational form of the Monodomain problem reads: given I^{app}, find $u \in L^2(0, T; H^1(\Omega))$ such that

$$\chi C_m \left(\frac{\partial u}{\partial t}, \phi \right) + a_M(u, \phi) + (I_{\text{ion}}(u, \boldsymbol{w}), \phi) = (I^{\text{app}}, \phi) \tag{10}$$

for each $\phi \in H^1(\Omega)$. The form $a_M(v, \phi) := \int_\Omega \nabla v^T \mathbf{D}^M \nabla \phi d\boldsymbol{x}$ is bilinear, continuous and weakly coercive on $H^1(\Omega) \times H^1(\Omega)$. For well-posedness analysis of this problem, we still refer to [14].

The Monodomain model is a single parabolic reaction-diffusion PDE for the transmembrane potential, replacing the two equations of the original model. However, this model is not able to capture some physiological and pathological patterns of the action potential propagation (see [11]).

3. The preconditioned Bidomain model

As a consequence of the degenerate nature of the Bidomain model that entails a severe ill conditioning of the matrix associated with its fully discrete approximation, the numerical solution of (4) requires significant computational effort. On the contrary, though relying on assumptions that prove quite often to be unrealistic, system (8) is by far more affordable. Our approach now is to use the Monodomain model as a preconditioner for the Bidomain system. To this end we properly reformulate both systems. More precisely, we consider the non-symmetric form of the Bidomain problem in terms of the transmembrane and the extracellular potentials u and u_e, setting $I^{\text{app}} = \frac{\lambda I_i^{\text{app}} + I_e^{\text{app}}}{1+\lambda}$ and $\widetilde{I}^{\text{app}} = I_i^{\text{app}} - I_e^{\text{app}}$:

$$\begin{cases} \chi C_m \frac{\partial u}{\partial t} - \nabla \cdot \left(\frac{\lambda \mathbf{D}_i}{1+\lambda} \nabla u \right) - \nabla \cdot \left(\frac{\lambda \mathbf{D}_i - \mathbf{D}_e}{1+\lambda} \nabla u_e \right) + \chi I_{ion}(u, \boldsymbol{w}) = I^{\text{app}}, \\ -\nabla \cdot [\mathbf{D}_i \nabla u + (\mathbf{D}_i + \mathbf{D}_e) \nabla u_e] = \widetilde{I}^{\text{app}}. \end{cases} \tag{11}$$

Hereafter we will refer to (11) as "non-symmetric formulation" of Bidomain model. The first equation in (11) is obtained by linearly combining the two equations in (4), with coefficients $\frac{\lambda}{1+\lambda}$ and $-\frac{1}{1+\lambda}$. The second equation is obtained summing up the two equations in (4). We point out that formulation (11) is non-standard and has been introduced in view of our preconditioning technique. In order to match the dimension of the Bidomain problem, the Monodomain model needs to be extended. The same linear combination leading to (11), combined with the assumption $\mathbf{D}_e = \lambda \mathbf{D}_i$ yields the extended Monodomain formulation in terms of the variables u and u_e

$$\begin{cases} \chi C_m \frac{\partial u}{\partial t} - \nabla \cdot \left(\frac{\lambda \mathbf{D}_i}{1+\lambda} \nabla u \right) + \chi I_{ion}(u, \boldsymbol{w}) = I^{\text{app}}, \\ -\nabla \cdot [\mathbf{D}_i \nabla u + (1+\lambda) \mathbf{D}_i \nabla u_e] = \widetilde{I}^{\text{app}}. \end{cases} \tag{12}$$

System (12) is lower triangular, where the first equation (the "genuine" Monodomain model) is independent of u_e. In view of its use as a preconditioner, however, there is no reason for retaining the simplifying Monodomain assumption $\lambda \mathbf{D}_i = \mathbf{D}_e$ in the second equation so we finally resort to

$$\begin{cases} \chi C_m \frac{\partial u}{\partial t} - \nabla \cdot \left(\frac{\lambda \mathbf{D}_i}{1+\lambda} \nabla u \right) + \chi I_{ion}(u, \boldsymbol{w}) = I^{\text{app}}, \\ -\nabla \cdot [\mathbf{D}_i \nabla u + (\mathbf{D}_i + \mathbf{D}_e) \nabla u_e] = \widetilde{I}^{\text{app}}. \end{cases} \tag{13}$$

Observe that our formulation of the Monodomain model comes immediately from the non-symmetric Bidomain (11) when the differential term in u_e in the first equation is dropped. As for the Bidomain model, also in the extended Monodomain model (12) (or 13) the extra cellular potential u_e is defined only up to a function of time. Again, we will fix such function by requiring that u_e has zero average.

3.1. Numerical discretization

Let Δt be the (constant) time step of the discretization. We denote with superscript n the variables computed at time $t^n = n\Delta t$. Moving from time step t^n to t^{n+1} the semi-implicit time-discretization of Bidomain Eq. (11) reads

$$\begin{cases} \chi C_m \frac{u^{n+1} - u^n}{\Delta t} - \nabla \cdot \left(\frac{\lambda \mathbf{D}_i}{1+\lambda} \nabla u^{n+1} + \frac{\lambda \mathbf{D}_i - \mathbf{D}_e}{1+\lambda} \nabla u_e^{n+1} \right) = I^n & \text{in } \Omega, \\ -\nabla \cdot \left[\mathbf{D}_i \nabla u^{n+1} + (\mathbf{D}_i + \mathbf{D}_e) \nabla u_e^{n+1} \right] = \widetilde{I}^{\text{app}} & \text{in } \Omega, \\ u^0(\boldsymbol{x}) = u_0(\boldsymbol{x}) & \text{in } \Omega, \\ \mathbf{n}^T \mathbf{D}_i (\nabla u^{n+1} + \nabla u_e^{n+1}) = 0 \quad \mathbf{n}^T \mathbf{D}_e \nabla u_e^{n+1} = 0 & \text{on } \partial\Omega, \end{cases} \tag{14}$$

where we have set $I^n = I^{\text{app}} - \chi I_{ion}(u^n, \boldsymbol{w}^{n+1})$, the latter term including the selected model for ionic current. Concerning the spatial approximation, we discretize the domain Ω with a triangulation $\mathcal{T}_h$ and we build a finite-element space V_h approximating $H^1(\Omega)$ on $\mathcal{T}_h$, in which we will look for the approximate solution u^h and u_e^h. In this work V_h is the space of piecewise linear continuous functions on $\mathcal{T}_h$, and we denote by $\Phi = \{\varphi_j\}_{j=1}^{N_h}$ a basis for V_h. Well-posedness of the discrete problem and convergence analysis for the Rogers–McCulloch model are carried out in [34].

Let us denote by $\boldsymbol{M}$ the mass matrix with entries $\boldsymbol{M}^{ij} = \sum_{K \in \boldsymbol{T}_h} (\varphi_j, \varphi_i)|_K$, and by $\boldsymbol{K}_\tau$ the stiffness matrices with $\boldsymbol{K}_\tau^{ij} = \sum_{K \in \mathcal{T}_h} (\mathbf{D}_\tau \nabla \varphi_j, \nabla \varphi_i)|_K, \varphi_i, \ \varphi_j \in \Phi$. The unknowns of the fully discrete problem are represented by vectors $\mathbf{u}$ and $\mathbf{u}_e$, storing the nodal values of u^h and u_e^h, respectively, we let $\mathbf{f}$ and $\mathbf{g}$ denote the discretization of the forcing terms, and we set

$$\mathbf{B}_{uu} = \frac{\boldsymbol{M}}{\Delta t} + \frac{\lambda \boldsymbol{K}_i}{1+\lambda}, \quad \mathbf{B}_{ue} = \frac{\lambda \boldsymbol{K}_i}{1+\lambda} - \frac{\boldsymbol{K}_e}{1+\lambda}, \quad \mathbf{B}_{eu} = \boldsymbol{K}_i, \quad \mathbf{B}_{ee} = \boldsymbol{K}_i + \boldsymbol{K}_e.$$

At step t^{n+1} we solve

$$\mathbf{B}_{\text{NS}}\mathbf{u}_{\text{NS}}^{n+1} = \mathbf{f}_{\text{NS}}^{n}, \tag{15}$$

where

$$\mathbf{B}_{\text{NS}} = \begin{bmatrix} \mathbf{B}_{uu} & \mathbf{B}_{ue} \\ \mathbf{B}_{eu} & \mathbf{B}_{ee} \end{bmatrix}, \quad \mathbf{u}_{\text{NS}} = \begin{bmatrix} \mathbf{u} \\ \mathbf{u}_e \end{bmatrix}, \quad \mathbf{f}_{\text{NS}} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}.$$

As a preconditioner for (15) we select

$$\mathbf{M}_{\text{NS}} = \begin{bmatrix} \mathbf{B}_{uu} & \mathbf{0} \\ \mathbf{B}_{eu} & \mathbf{B}_{ee} \end{bmatrix} \tag{16}$$

which corresponds to the discrete operator associated to the extended Monodomain problem (13), suitably discretized in time. This form highlights that our model-based preconditioner results in a block Gauss–Seidel preconditioner for the non-symmetric Bidomain system, featuring u and u_e. However, we will still refer to the Monodomain-based interpretation, which allows us to gain insight on its effectiveness. In principle, the same approach based on the block-triangular preconditioning could be applied also to the symmetric formulation of the Bidomain system in terms of u_i and u_e. However, this choice proves to be ineffective, as we show in Section 5.

Linear solver. Since the matrix in (15) is not symmetric, we resort to a Krylov iterative solver. More precisely, in order to reduce the CPU time, we resort to a *flexible* strategy, corresponding to solve inaccurately the preconditioner by an iterative method with a coarse tolerance. In this case, the actual preconditioner depends on the current iteration. A Flexible GMRES (FGMRES) with a right preconditioner (see [32]) needs then to be used accordingly. By extension, we use the same right preconditioning approach also for non-flexible GMRES with an accurate solution of the preconditioner.

The implementation of the Monodomain preconditioner requires to solve system $\mathbf{M}_{\text{NS}}\mathbf{z} = \mathbf{v}$, where $\mathbf{z} = [\mathbf{z}^1, \mathbf{z}^2]^T$ and $\mathbf{v} = [\mathbf{v}^1, \mathbf{v}^2]^T$. To this aim, we exploit the lower triangular structure of $\mathbf{M}_{\text{NS}}$, solving the sequence of the following systems

$$\mathbf{B}_{uu}\mathbf{z}^1 = \mathbf{v}^1 \quad \mathbf{b} = \mathbf{v}^2 - \mathbf{B}_{eu}\mathbf{z}^1 \quad \mathbf{B}_{ee}\mathbf{z}^2 = \mathbf{b}. \tag{17}$$

Since the systems in $\mathbf{B}_{uu}$ and $\mathbf{B}_{ee}$ are symmetric, we solve them using ILU preconditioned conjugate gradient method. Notice that both matrices $\mathbf{B}_{NS}$ and $\mathbf{M}_{NS}$ defined in (15) and (16), respectively, are singular, their kernel being given by $span\left\{[\mathbf{0}, \mathbf{1}]^T\right\}$. In particular, we solve the singular systems by an iterative method, as this is a reliable strategy for elliptic problem with homogeneous Neumann boundary conditions [7]. After the solution of non-symmetric Bidomain system (15), we force the average of u_e to be zero at each time iteration.

4. Fourier analysis of the preconditioner

In this section we analyze the performances of the proposed preconditioner at each time step by means of Fourier analysis. For the sake of notation, we drop hereafter the time index. We assume, without loss of generality, the reference frame to have the first component aligned with the longitudinal axis of the fibers, so that, the diffusion tensors are diagonal, thanks to (2). We work in an unbounded domain $\Omega \equiv \mathbb{R}^3$ and we introduce the continuous operators $\mathcal{B} : [H^1(\Omega)]^2 \to [H^{-1}(\Omega)]^2$ and $\mathcal{M} : [H^1(\Omega)]^2 \to [H^{-1}(\Omega)]^2$, associated with problems (14) and with the semi-discrete counterpart of (13). Observe that in this case, asymptotic requirements for the Fourier transformability of the extracellular potential automatically fix the arbitrary function of time, so no arbitrariness is anymore present. Denoting by k_1, k_2 and k_3 the dual frequency variables, the Fourier transform of $w(x,y,z) = u(x,y,z), u_e(x,y,z)$ reads

$$\mathcal{F} : w(x,y,z) \mapsto \widehat{w}(k_1,k_2,k_3) = \int\int\int_{\mathbb{R}^3} e^{-i(k_1x+k_2y+k_3z)} w(x,y,z)\,dxdydz.$$

Action of operators $\mathcal{B}$ and $\mathcal{M}$ can now be expressed for any $u \in [H^1(\Omega)]^2$ by means of the inverse Fourier transform, namely

$$\mathcal{B}u = \mathcal{F}^{-1}\left(B\widehat{u}\right), \quad \mathcal{M}u = \mathcal{F}^{-1}\left(M\widehat{u}\right)$$

where B and M represent the operators $\mathcal{B}$ and $\mathcal{M}$, respectively. We denote by $[f,g]^T$ the right hand side in (14), and we let $\boldsymbol{k}^2 = k_2^2 + k_3^2$ as a consequence of assumption (2). The transformed (linearized) Bidomain problem reads

$$\begin{aligned} &\chi C_m \widehat{u} + \frac{\Delta t}{1+\lambda}\left(\lambda\left[\sigma_i^l k_1^2 + \sigma_i^t \boldsymbol{k}^2\right]\widehat{u} + \left[(\lambda\sigma_i^l - \sigma_e^l)k_1^2 + (\lambda\sigma_i^t - \sigma_e^t)\boldsymbol{k}^2\right]\widehat{u}_e\right) = \Delta t\widehat{f} \\ &\left[\sigma_i^l k_1^2 + \sigma_i^t \boldsymbol{k}^2\right]\widehat{u} + \left[(\sigma_i^l + \sigma_e^l)k_1^2 + (\sigma_i^t + \sigma_e^t)\boldsymbol{k}^2\right]\widehat{u}_e = \widehat{g}. \end{aligned} \tag{18}$$

The first equation of the expanded Monodomain problem reads

$$\chi C_m \widehat{u} + \Delta t\frac{\lambda}{1+\lambda}\left[\sigma_i^l k_1^2 + \sigma_i^t \boldsymbol{k}^2\right]\widehat{u} = \Delta t\widehat{f},$$

the second transformed equation coinciding with the second equation in (18). We set $\xi = \sigma_i^l k_1^2 + \sigma_i^t \boldsymbol{k}^2$, and $\eta = \sigma_e^l k_1^2 + \sigma_e^t \boldsymbol{k}^2$. Bidomain and Monodomain problems in the frequency domain can be rewritten in matrix form as

$$B(k_1,\boldsymbol{k})\begin{bmatrix}\widehat{u}\\ \widehat{u}_e\end{bmatrix}=\begin{bmatrix}\Delta t\widehat{f}\\ \widehat{g}\end{bmatrix},\quad M(k_1,\boldsymbol{k})\begin{bmatrix}\widehat{u}\\ \widehat{u}_e\end{bmatrix}=\begin{bmatrix}\Delta t\widehat{f}\\ \widehat{g}\end{bmatrix}$$

where

$$B(k_1,\boldsymbol{k})=\begin{bmatrix}\chi C_m+\Delta t\frac{\lambda}{1+\lambda}\,\xi(k_1,\boldsymbol{k}) & \Delta t\left[\frac{\lambda}{1+\lambda}\,\xi(k_1,\boldsymbol{k})-\frac{1}{1+\lambda}\,\eta(k_1,\boldsymbol{k})\right]\\ \xi(k_1,\boldsymbol{k}) & \xi(k_1,\boldsymbol{k})+\eta(k_1,\boldsymbol{k})\end{bmatrix}\tag{19}$$

and

$$M(k_1,\boldsymbol{k})=\begin{bmatrix}\chi C_m+\Delta t\frac{\lambda}{1+\lambda}\,\xi(k_1,\boldsymbol{k}) & 0\\ \xi(k_1,\boldsymbol{k}) & \xi(k_1,\boldsymbol{k})+\eta(k_1,\boldsymbol{k})\end{bmatrix}.\tag{20}$$

For $(k_1,\boldsymbol{k})\neq(0,0)$ the matrix $M(k_1,\boldsymbol{k})$ is invertible. From now on we set $\chi C_m=1$, as this is the standard assumption in the applications (see [13]).

Considering $|k_1|<k_1^M$ and $|\boldsymbol{k}|<\boldsymbol{k}^M$, where k_1^M and $\boldsymbol{k}^M$ represent the maximal frequencies supported by the numerical grid (of order π/h), we analyze the effectiveness of the preconditioned operator over the domain

$$T=\{\lambda_M\xi-c_1\leqslant\eta\leqslant\lambda_M\xi,\ \lambda_m\xi\leqslant\eta\leqslant\lambda_m\xi+c_2\}\setminus\{(0,0)\},$$

shown in Fig. 1, where c_1 and c_2 are positive constants depending on $k_1^M,\boldsymbol{k}^M$ and on the conductivity values. As k_1^M and $\boldsymbol{k}^M$ tend to infinity, the domain T covers the angular sector $S=\{\lambda_m\xi\leqslant\eta\leqslant\lambda_M\xi\}\setminus\{(0,0)\}$. With these notations, the preconditioned operator reads

$$[M(\xi,\eta)]^{-1}B(\xi,\eta)=\begin{bmatrix}1 & \alpha(\xi,\eta)\\ 0 & 1-\frac{\xi}{\xi+\eta}\alpha(\xi,\eta)\end{bmatrix},\tag{21}$$

where

$$\alpha(\xi,\eta)=\frac{\Delta t}{1+\lambda}\,\frac{[\lambda\,\xi-\eta]}{1+\frac{\lambda}{1+\lambda}\,\Delta t\,\xi}.\tag{22}$$

The eigenvalues of $M^{-1}B$ are clearly given by

$$\gamma_1(\xi,\eta)=1,\quad \gamma_2(\xi,\eta)=1-\frac{\xi}{\xi+\eta}\alpha(\xi,\eta)=\frac{1+\Delta t\xi\,\frac{1}{\frac{\xi}{\eta}+1}}{1+\Delta t\xi\,\frac{1}{\frac{1}{\lambda}+1}}.\tag{23}$$

Since they are both real and positive, the conditioning of the continuous preconditioned problem can be estimated by $\max(\gamma_1,\gamma_2)/\min(\gamma_1,\gamma_2)$ (see [17]). From (22) and (23), we have $\gamma_2(\xi,\eta)<1$ for $\eta<\lambda\xi$ and $\gamma_2(\xi,\eta)>1$ for $\eta>\lambda\xi$, thus, for any $\lambda_m\leqslant\lambda\leqslant\lambda_M$

$$\mathcal{K}(\mathcal{M}^{-1}\mathcal{B})\simeq\frac{\max\left[1,\max\limits_{(\xi,\eta)\in T}\gamma_2(\xi,\eta)\right]}{\min\left[1,\min\limits_{(\xi,\eta)\in T}\gamma_2(\xi,\eta)\right]}=\frac{\max\limits_{(\xi,\eta)\in T}\gamma_2(\xi,\eta)}{\min\limits_{(\xi,\eta)\in T}\gamma_2(\xi,\eta)}.\tag{24}$$

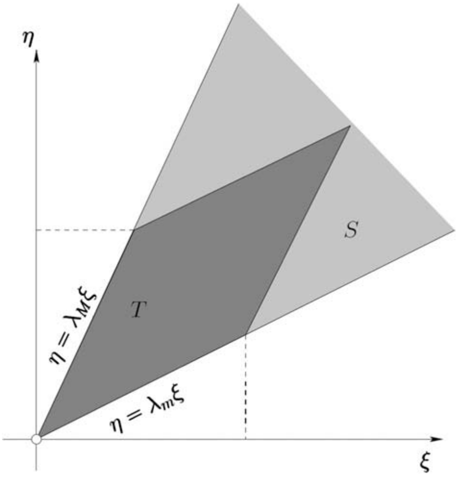

Fig. 1. The domains T and S.

Since $\frac{1}{\lambda_M} \leqslant \frac{\xi}{\eta} \leqslant \frac{1}{\lambda_m}$, we have

$$\Gamma_m(\xi) = \frac{1 + \Delta t \xi \frac{1}{\frac{1}{\lambda_m}+1}}{1 + \Delta t \xi \frac{1}{\frac{1}{\lambda}+1}} \leqslant \gamma_2(\xi, \eta) \leqslant \frac{1 + \Delta t \xi \frac{1}{\frac{1}{\lambda_M}+1}}{1 + \Delta t \xi \frac{1}{\frac{1}{\lambda}+1}} = \Gamma_M(\xi), \tag{25}$$

namely, $\gamma_2(\xi, \eta)$ is bounded independently of η. For any $\lambda_m \leqslant \lambda \leqslant \lambda_M, \Gamma_m(\xi)$ is nonincreasing and $\Gamma_M(\xi)$ is nondecreasing. Taking the limit for $\xi, \eta \to \infty$ in (25), corresponding to the mesh size h tending to 0, domain T does coincide with S and we get

$$\min_{(\xi,\eta)\in T} \gamma_2(\xi, \eta) \geqslant \frac{\frac{1}{\lambda}+1}{\frac{1}{\lambda_m}+1}, \quad \max_{(\xi,\eta)\in T} \gamma_2(\xi, \eta) \leqslant \frac{\frac{1}{\lambda}+1}{\frac{1}{\lambda_M}+1}. \tag{26}$$

Gathering together (24) and (26) we obtain that for all $\lambda_m \leqslant \lambda \leqslant \lambda_M$

$$\mathcal{K}(\mathcal{M}^{-1}\mathcal{B}) \lesssim \left(1 + \frac{1}{\lambda_M}\right)^{-1} \left(1 + \frac{1}{\lambda_m}\right). \tag{27}$$

We conclude that the preconditioner is optimal with respect to the mesh size, since the stability of the continuous problem $\mathcal{K}(\mathcal{M}^{-1}\mathcal{B})$ is bounded by a constant depending only on the anisotropy ratio in the coefficients of the Bidomain problem. A rigorous estimate of the conditioning would be provided by the ratio between the maximum and the minimum singular values of the preconditioned operator (21). This quite tedious computation, not reported for the sake of brevity, leads to the same conclusion.

The previous analysis suggests some further considerations on λ. Beyond physical meaning, λ can be considered here as a parameter to be selected for enhancing the convergence of the preconditioned iterations. We plot in Fig. 2 the distribution of the generalized eigenvalues ω of the matrices $\mathbf{B}_{\mathrm{NS}}$ and $\mathbf{M}_{\mathrm{NS}}$($\mathbf{B}_{\mathrm{NS}}\boldsymbol{v} = \omega \mathbf{M}_{\mathrm{NS}}\boldsymbol{v}$), computed with Matlab®, for two different mesh sizes. Taking the values for σ_τ^l and σ_τ^t proposed in [12], we have $\lambda_m = 0.6667$ and $\lambda_M = 4.2868$. We consider the three cases, namely $\lambda = \lambda_m, \lambda = \lambda_M$, and $\lambda = 1.3$, which is the value used in the numerical simulations of Section 5. Corresponding bounds for γ_2 are: $1 \leqslant \gamma_2 \leqslant 2.02706$ when $\lambda = \lambda_m$, $0.49332 \leqslant \gamma_2 \leqslant 1$ when $\lambda = \lambda_M$ and $0.70771 \leqslant \gamma_2 \leqslant 1.43458$ when $\lambda = 1.3$. Notice that the spectrum spreads out as the mesh parameter h decreases, however predicted bounds for the eigenvalues are fulfilled in both the cases. The choice of $\lambda = 1.3$, that we empirically tuned for minimizing computational costs, leads to a good clustering of the spectrum around 1.

5. Numerical results

Numerical results presented hereafter refer to the 3D Bidomain problem on different geometries: a slab, a truncated ellipsoid, representing a simplified ventricular geometry, and a real geometry reconstructed from SPECT images (see Fig. 3). All the geometries are completed with an analytical representation of the fiber orientation as detailed in [12]. Simulations on the slab and on the real ventricle use the Luo–Rudy Phase I model, while in the case of the truncated ellipsoid the Bidomain model is coupled with either the Rogers–McCulloch or the Luo–Rudy Phase I ionic models. We consider the parameters listed in [22] for the Luo–Rudy ionic model and the parameters in [12] to set the Rogers–McCulloch model and the Bidomain one. Throughout this section, unless differently stated, we use a time step $\Delta t = 0.5$ ms for the Rogers–McCulloch ionic model, while $\Delta t = 0.1$ ms is needed for the Luo–Rudy one, in order to solve the ionic problem accurately enough. Numerical simulations are carried out with `LifeV` [1], using the `Trilinos` packages `BELOS` and `IFPACK` [2]. The details of the implementation of the Monodomain preconditioning strategy are explained hereafter: system (15) is solved by the Flexible Block-GMRES (with block size set to 1) implemented in `BELOS`. Stopping criterion is based on the control of 2-norm of the current residual, normalized with respect to the 2-norm of the initial residual and the tolerance (denoted *outer tolerance*) is set to 10^{-5}. Both linear systems in (17) are solved by a Block-CG (with block size equal to 1), implemented in `BELOS` as well, with an ILU left preconditioner, implemented in `IFPACK`, with the same stopping criterion as for system (15). The tolerance chosen for the solution of these systems (denoted *inner tolerance*) will be discussed in Section 5.1.

All the computations are carried out on a workstation equipped with a 2.2 GHz AMD Dual-Core Opteron processor and 4 GB RAM.

To compare the performances of the proposed preconditioning strategy with a reference solver, we consider at first two possible alternative reference solvers, namely the GMRES method applied to the non-symmetric formulation (15) that originates our preconditioner and the CG method applied to the symmetric Bidomain system, both preconditioned with an ILU factorization. The latter at each time t^{n+1} reads

$$\mathbf{B}_{\mathrm{S}}\mathbf{u}_{\mathrm{S}}^{n+1} = \mathbf{f}_{\mathrm{S}}^{n}, \tag{28}$$

where

$$\mathbf{B}_{\mathrm{S}} = \begin{bmatrix} \mathbf{B}_{ii}^S & \mathbf{B}_{ie}^S \\ \mathbf{B}_{ei}^S & \mathbf{B}_{ee}^S \end{bmatrix}, \quad \mathbf{u}_{\mathrm{S}} = \begin{bmatrix} \mathbf{u}_i \\ \mathbf{u}_e \end{bmatrix}, \quad \mathbf{f}_{\mathrm{S}} = \begin{bmatrix} \mathbf{f}_i \\ \mathbf{f}_e \end{bmatrix}$$

and

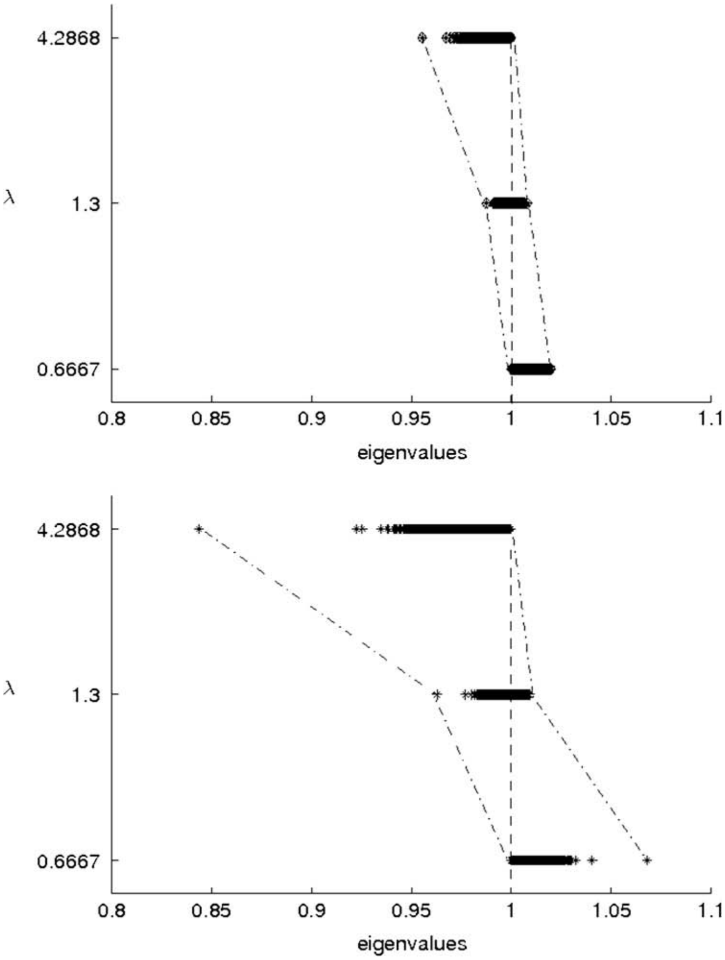

Fig. 2. Spectra of the preconditioned problem for different mesh sizes: 5272 nodes (top) and 12,586 nodes (bottom). The dashed-dotted lines highlight clustering of the eigenvalues around 1 for different values of λ.

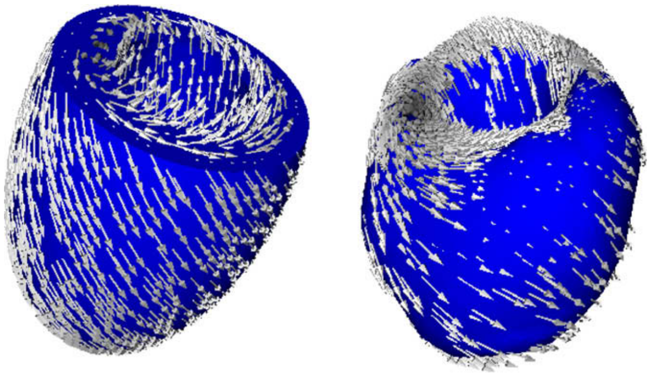

Fig. 3. Left: truncated ellipsoidal geometry representing an idealized left ventricle. Right: real ventricular geometry reconstructed from SPECT (courtesy of Dr. E.V. Garcia) images. White arrows represent myocardial fiber orientation used in our numerical simulations (see [12] for their analytical description).

$$\mathbf{B}^S_{ii} = \frac{\boldsymbol{M}}{\Delta t} + \boldsymbol{K}_i, \quad \mathbf{B}^S_{ie} = \mathbf{B}^S_{ei} = -\frac{\boldsymbol{M}}{\Delta t}, \quad \mathbf{B}^S_{ee} = \frac{\boldsymbol{M}}{\Delta t} + \boldsymbol{K}_e.$$

Entries of vectors $\mathbf{u}_i$ and $\mathbf{u}_e$ are the nodal values of u_i^h and u_e^h, while vectors $\mathbf{f}_i$ and $\mathbf{f}_e$ represent the discretization of the forcing terms in the two equations. Again, system (28) is solved by a Block-CG algorithm implemented in BELOS package, with an

Table 1
Comparison of the performances of ILU-CG applied to the symmetric formulation (28) and ILU-GMRES applied to the non-symmetric formulation (15) of the Bidomain linear system coupled with the Rogers–McCulloch ionic model, for different mesh sizes and an ellipsoidal geometry: average execution time per time step, and average iteration counts per time step for the solution of the Bidomain linear systems with the ILU-CG and ILU-GMRES solver, respectively.

# Nodes	ILU-CG		ILU-GMRES	
	Time	Iter	Time	Iter
29,560	4.32848	29.85	4.91394	32.16
62,566	13.4555	37.32	17.1016	41.89
172,878	61.561	48.16	88.8527	54.54

ILU left preconditioner (with level of fill 1 - see next paragraph), with stopping criterion based on the normalized residual and tolerance 10^{-5}. The ILU-GMRES is used for the same formulation to which our preconditioner is applied so it is the natural candidates for our comparisons. On the other hand, ILU-CG exploits the symmetry of the original formulation of the problem, so it is supposed to be more effective. Preliminary computations on a truncated ellipsoidal geometry reported in Table 1 show that the two approaches feature similar performances. The ionic model used in this test is the Rogers–McCulloch, and the simulations are run for 50 ms. In the first column we report the number of nodes of the computational meshes used in the simulations. In the second and fourth columns we report the average execution time for the solution of the Bidomain linear system with ILU-CG and ILU-GMRES, respectively. The average is computed over all the time iterations of the simulations, with the exception of the first one, which is the most expensive one, as the ILU factorizations are computed at this stage. In the third and fifth columns we show the average iteration counts for the solution of the Bidomain linear systems obtained with the mentioned solvers. In this case the average is computed over all the time iterations of the simulations.

Since the performances of the two alternative solvers are comparable, in the sequel, we compare our results with the ILU-CG solver, which is based on the most popular formulation of the problem.

Level of sparsity. In this paragraph we test the effect of the level of sparsity of the ILU preconditioner on the performances of the ILU-CG method and the Monodomain preconditioning strategy, where the incomplete LU factorization is used to precondition systems (17). The level of sparsity selection in the ILU preconditioner is driven by the *level of fill* (see [32]). Lower is the level, and closer the pattern of the L and U factors is to the pattern of the original matrix.

In Table 2 we compare the iteration counts and the execution time of the Bidomain linear system solutions with different levels of fill, using ILU-CG solver. In particular, since the first time step is by far the most expensive, as the ILU factorizations are carried out at this stage, we separate the contribution of the execution time of the first time step from the average execution time over the remaining time steps. The average iteration counts of the conjugate gradient algorithm are computed on the overall simulation (the same choice will be applied in Tables 5–7). As expected, the average number of iterations is affected by the different level of fill, being higher with a lower level of fill, but the CPU time does not significantly change, except for the first iteration.

In Table 3 we show the results obtained applying the Monodomain preconditioner to the FGMRES solver, with 2 different levels of fill. Again we separate the execution time of the first time step from the average execution time over the remaining time steps. The average iteration counts, computed on the overall simulation, refer to the outer iterations of the Flexible GMRES algorithm. Since the preconditioner is solved up to the same tolerance for both the levels of fill chosen for the ILU factorization, the outer iteration counts are expected to be substantially independent of the level of fill. The slight increase is likely consequence of small changes in matrix conditioning for this specific case.

Table 2
Impact of different levels of fill for ILU-CG solver: execution time (in s) for the first time step, average execution time (in s) per time step (excluding the first one), average iteration counts per time step and number of non-zero entries of the incomplete LU factorization of $\mathbf{B}_S$.

Level of fill	1st step	Time	Iter	NNZ
0	107.75	74.462	72.088	11,653,944
1	267.42	78.2781	39.61	40,363,506

Table 3
Impact of different levels of fill for the Monodomain preconditioner: execution time (in s) for the first time step, average execution time (in s) per time step (excluding the first one), average number of outer FGMRES iterations per time step and sum of the number of non-zero entries of the incomplete LU factorization of $\mathbf{B}_{uu}$ and $\mathbf{B}_{ee}$.

Level of fill	1st step	Time	Iter	NNZ
0	49.45	42.881	6.294	5,618,124
1	78.79	45.3721	6.986	14,540,464

In conclusion, the ILU-CG exhibits a significant reduction in the number of iterations (not in the computing time) with higher level of fill, that is an advantage in parallel implementations. The difference of the iteration numbers and computing time in the two preconditioning cases is however still remarkable even with level 1 of fill. A more extensive comparison in a parallel framework is therefore in order.

Simulations described in this paragraph are performed using the Luo–Rudy I ionic model on a slab 1 × 1 × 0.2 cm geometry with 208,848 nodes, over a time interval of 50 ms, with $\lambda = 1.3$ and inner tolerance set to 0.12.

5.1. Influence of the inner tolerance and of λ

This set of numerical experiments aims at investigating the robustness of the preconditioner with respect to the accuracy in the solution of systems (17). We performed numerical simulations over an idealized ventricular geometry represented by the truncated ellipsoid reported in Fig. 3(left).

We set $\lambda = 1.3$, and the simulation is run for 50 ms with the Luo–Rudy phase I model. We solve here systems (17) with a tolerance $tol. = 10^{-5}$, which is the same tolerance used as a stopping criterion in the outer iterations. Then we solve the problem with a coarse tolerance $tol. = 0.12$ for solving systems (17), which is the result of a fine tuning for finding a trade-off between the number of outer iterations and CPU time to solve (17).

In Table 4 we report the average CPU time and the average FGMRES iteration counts over the entire simulation with different mesh sizes. The two solutions of the Bidomain systems are computed up to the fulfillment of the same outer tolerance on the residual. Table 4 highlights the relevant CPU time reduction with the use of a coarse inner tolerance, while the outer iteration counts are almost insensitive to the different accuracy required to the solution of the preconditioned systems.

A test comparing three different inner tolerances (0.12, 0.01 and 10^{-5}) on a wide range of mesh size, from 22,470 to 677,000 nodes for the real geometry introduced above has been performed as well. We report in Fig. 4 the results obtained. As expected, as the inner tolerance decreases the average CPU time increases and the average number of iterations decreases. However, the iteration count reduction is small in comparison with the increase of CPU time. This results suggest to use an inner tolerance of 0.12 in performing the subsequent tests.

In order to choose the value of λ used in our numerical tests, we have performed some computations on real geometry meshes with different size, trying different values of λ within the range $[\lambda_m, \lambda_M]$. In particular we selected $\lambda = 0.6667, 0.9833, 1.3, 2.7934, 4.2868$. We compared the performances obtained using three different meshes with 22,470, 276,578, 677,000 nodes, with respect to the average CPU time and the average iteration counts. Results are reported in Fig. 5.

Table 4
Comparison of the performances of the preconditioner with a fine vs coarse inner resolution.

Nodes	$tol. = 10^{-5}$		$tol. = 0.12$	
	Time	Iter	Time	Iter
29,560	3.89158	2.988	1.08365	3.068
62,566	10.6337	3	3.82078	3.972
172,878	53.0898	3.922	15.4344	3.99

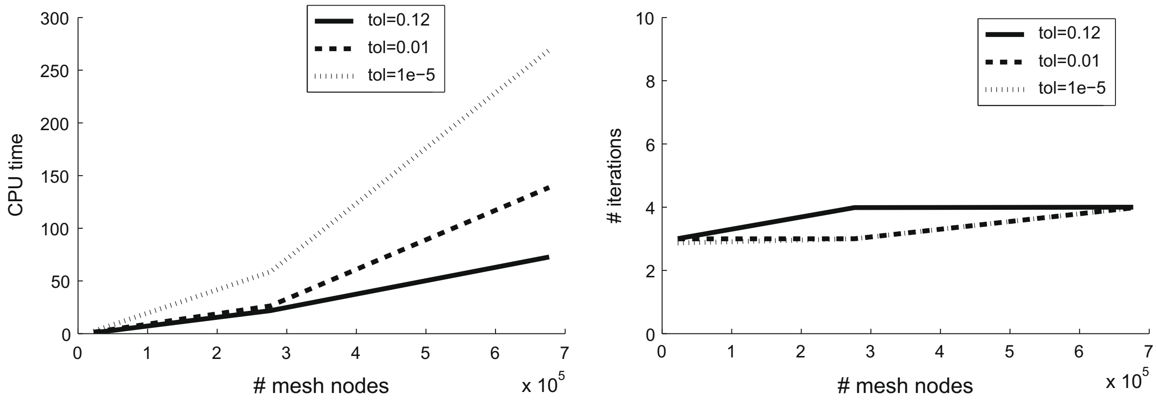

Fig. 4. Comparison of the performances of the preconditioner with different values of inner tolerance on different meshes. Inner tolerance is set to 10^{-5} (dotted line), 0.01 (dashed line) and 0.12 (solid line). Left: average CPU time. Right: average iteration counts.

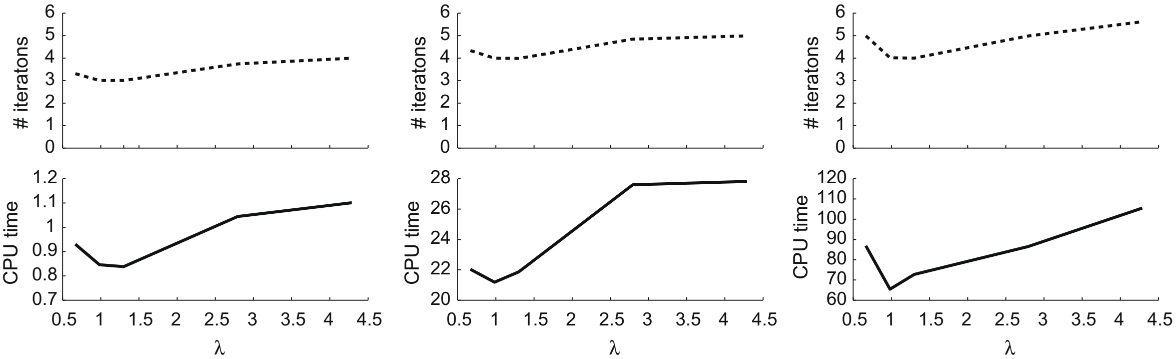

Fig. 5. Comparison of the performances of the Monodomain preconditioner with different values of λ on meshes with different size. Dashed line: average number of iterations. Solid line: average computational time. Results are obtained with a real ventricle mesh with 22,470 (left), 276,578 (center) and 677,000 (right) nodes.

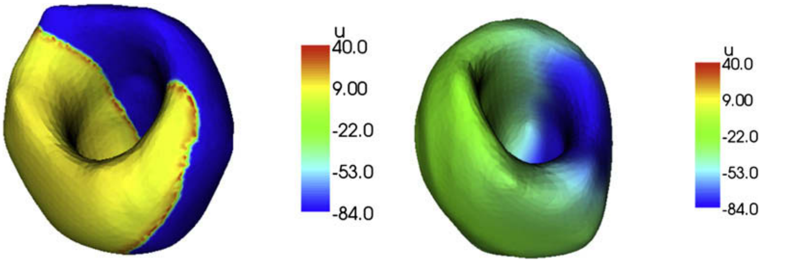

Fig. 6. Screenshots of the action potential propagation at $t = 70$ ms (left) and $t = 400$ ms (right), computed on a real left ventricular geometry.

Screenshots of the solution are reported in Fig. 6.

The average number of iterations is quite insensitive to the choice of λ for all the mesh tested, while the average computational time features an “optimal” value that depends on the mesh size. A more accurate analysis of this dependence of the performances on λ will be considered elsewhere. However, we can observe that for very different mesh sizes, the best value of λ lays between 0.9833 and 1.3. Therefore we suggest to select λ in this range.

5.2. Heartbeat simulation

In this test we analyze the effectiveness of the Monodomain preconditioner from the depolarization to the repolarization (500 ms) of the ventricle tissue in one cardiac cycle on a fine grid. In particular we choose $h = 0.02$ cm on a computational domain given by a slab geometry of size 1x1x0.2 cm, that can be handled on a single processor computer. The resulting tetrahedral grid counts 208,848 vertices.

The Bidomain system is coupled with the Luo–Rudy Phase I model. We set again $\lambda = 1.3$, and we solve the systems in (17) with an inner tolerance $tol. = 0.12$. We plot in Fig. 7(top) the evolution of the iteration counts for both the ILU preconditioned problem and the Monodomain preconditioned problem (denoted *MPrec*). ILU preconditioner simulation shows a remarkable variation in the iteration counts at the beginning and at the end of the simulation, as already observed in [24]. Correspondingly, CPU time at the beginning and at the end of simulation is increased. This phenomenon at the moment does not have a clear justification. A possible heuristic explanation is that it is a consequence of the discretization errors due to large variations in the ionic variables, in correspondence to the opening and closing of gating channels, occurring at the beginning and the end of APD. These errors could be amplified by the ill conditioning of the Bidomain system. On the other hand the Monodomain preconditioner has a fairly constant performance along the whole simulation in terms of number of iterations per time step, which is remarkably smaller than in the ILU preconditioner case. CPU time plot (Fig. 7(bottom)) shows that, also for our preconditioner, the CPU time slightly increases, as should be expected because of the large variations of the ionic current. This effect is however less evident for our preconditioner versus the ILU one. Fig. 8 shows transmembrane and extracellular potentials computed with ILU preconditioner (dashed line) and Monodomain preconditioner (solid line). Computed solutions are clearly the same.

5.3. Influence of the mesh size

In this test we run the Bidomain simulations on the truncated ellipsoid for 50 ms. We set again $\lambda = 1.3$, while the tolerance for systems (17) is $tol. = 0.12$. We compare the iteration counts and the execution time of the Bidomain linear system

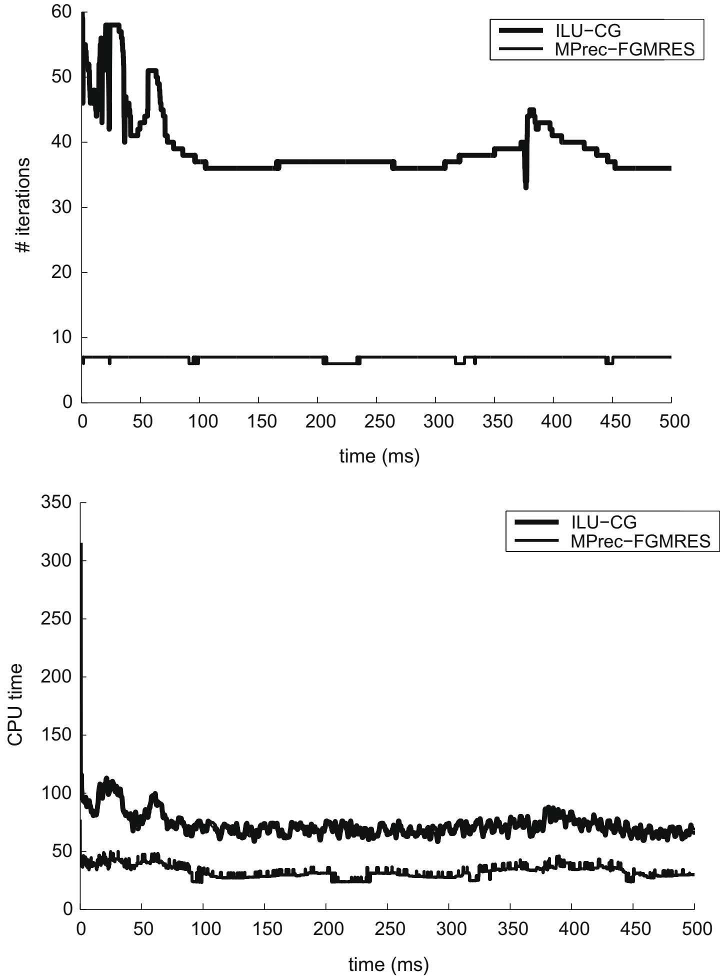

Fig. 7. Test of Section 5.2: Top: number of iterations at each time step. Bottom: CPU time for the solution of the linear system at each time step. Thick line: conjugate gradient method with ILU preconditioner. Thin line: flexible GMRES with Monodomain preconditioner.

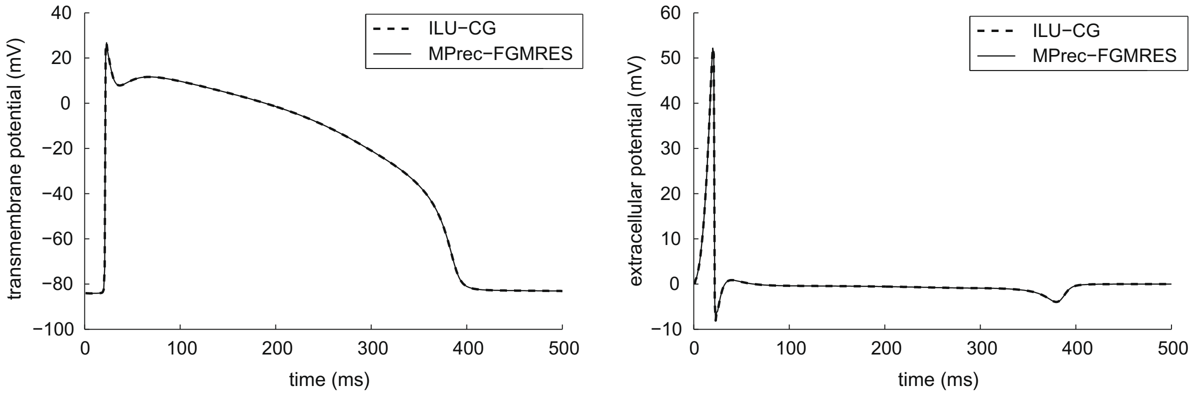

Fig. 8. Time evolutions of transmembrane and extracellular potentials at a fixed spatial point in the slab. Solutions obtained with conjugate gradient method and ILU preconditioner (dashed line), and with flexible GMRES and Monodomain preconditioner (solid line).

solution, for both solvers, as explained for Tables 2 and 3. Results in Table 5 refer to Rogers–McCulloch model, while results in Table 6 refer to Luo–Rudy phase I model. The iteration counts of the Monodomain preconditioner appear to be essentially

Table 5
Rogers–McCulloch model: execution time (in s) for the first time step, average execution time (in s) per time step (excluding the first one), and average iteration counts per time step. Columns 2–4: symmetric Bidomain with ILU preconditioned CG. Columns 5–7: non-symmetric Bidomain with Monodomain preconditioned flexible GMRES.

# Nodes	ILU-CG			MPrec-FGMRES		
	1st step	Time	Iter	1st step	Time	Iter
12,586	9.66	1.31869	22.75	2.77	0.885859	5.03
29,560	25.53	4.32848	29.85	7.53	2.87768	6
62,566	61.34	13.4555	37.32	18.12	7.76394	6.01
127,401	137.03	40.6704	46.29	41.35	20.6259	6.01
172,878	211.91	61.561	48.16	61.28	31.6269	6.02
508,383	1043.32	392.806	78	295.95	181.893	7
841,413	1939.73	779.183	92.56	514.2	329.209	7

Table 6
Luo–Rudy phase I model: execution time (in s) for the first time step, average execution time (in s) per time step (excluding the first one), and average iteration counts per time step. Columns 2–4: symmetric Bidomain with ILU preconditioned CG. Columns 5–7: non-symmetric Bidomain with Monodomain preconditioned flexible GMRES.

# Nodes	ILU-CG			MPrec-FGMRES		
	1st step	Time	Iter	1st step	Time	Iter
12,586	9.27	0.659178	10.358	2.37	0.425992	3.006
29,560	23.42	1.80633	11.268	5.96	1.08365	3.068
62,566	53.49	4.8288	12.136	13.58	3.82078	3.972
127,401	118.06	13.779	14.478	33.47	9.28469	3.992
172,878	170.27	22.0005	16.076	46.55	15.4344	3.99
508,383	767.96	134.46	26.894	198.45	58.1504	4.04
841,413	1509.6	294.201	33.268	333.65	149.989	4.9

Table 7
Ratio of the CPU times and ratio of iteration counts between conjugate gradient method with ILU preconditioner and flexible GMRES with Monodomain preconditioner. Rogers–McCulloch model (columns 2–4) and Luo–Rudy phase I model (columns 5–7).

# Nodes	ILU/MPrec (RMC)			ILU/MPrec (LR1)		
	1st step	Time	Iter	1st step	Time	Iter
12,586	3.4874	1.4886	4.5229	3.9114	1.5474	3.4458
29,560	3.3904	1.5042	4.9750	3.9295	1.6669	3.6728
62,566	3.3852	1.7331	6.2097	3.9389	1.2638	3.0554
127,401	3.3139	1.9718	7.7022	3.5273	1.4841	3.6268
172,878	3.4581	1.9465	8.0000	3.6578	1.4254	4.0291
508,383	3.5253	2.1595	11.1428	3.8698	2.3123	6.6569
841,413	3.7723	2.3668	13.2229	4.5245	1.9615	6.7894

insensitive to the mesh size for both ionic models. Execution time of the Monodomain preconditioned system remains significantly lower than the one of the symmetric Bidomain problem (see Table 7), the differences becoming more pronounced when we use finer meshes. The difference is particularly evident in the execution time of the first time step when the incomplete LU factorization is carried out. This feature is likely relevant when the LU factorization needs to be frequently repeated during the simulations, like, for instance, when the movement of the cardiac tissue is included in the model. It is worth noticing that in order to accurately describe the sharp fronts of potentials and the propagation velocity, finer mesh and smaller time steps should be used. Mesh sizes used in this work are basically limited by the use of serial architectures. In this respect, our preconditioner not only reduces the CPU time in comparison with the ILU-CG, but demands for less storage resources as the NNZ columns indicate in Tables 2 and 3. On the other hand, the optimality of the Monodomain preconditioner is a promising feature in view of parallel implementations that allow the use of finer meshes.

5.4. A symmetric block Gauss–Seidel preconditioner

The idea of using a block-triangular preconditioner could be applied also to the symmetric formulation (28) of the Bidomain problem, by simply dropping the block $\mathbf{B}_{ie}^{S}$ in the preconditioner. Numerical results show that this choice is not effective. As a matter of fact, for a mesh of 29,560 nodes (with the Luo–Rudy Phase I ionic model), for instance, the execution time for the first time step is 31.02 s, the average iteration count is 87.114, and the average execution time is 24.4803 s, showing that this choice is more expensive than both the Monodomain preconditioner based on the non-symmetric formulation and the ILU-CG preconditioner (compare these execution times with those reported in Table 6). This can be explained by observing that to drop block $\mathbf{B}_{ie}^{S}$ amounts to neglect a significant part of the Bidomain system (28), in particular when $\Delta t \to 0$. On the contrary, in the non-symmetric formulation (15) the effect of dropping block $\mathbf{B}_{ue}$ is less relevant since the neglected block is the difference of two terms that can be made "small" for a suitable choice of the Monodomain parameter λ.

6. Conclusions

We introduced a preconditioner for the Bidomain problem in electrocardiology, based on a non-symmetric formulation and on a suitable extension of the Monodomain model. We proved its optimality, assessed both theoretically by Fourier analysis and numerically by 3D numerical tests. At the algebraic level, this is a lower block-triangular preconditioner including part of the elliptic core of the problem and dropping a block that can be made quantitatively small by an appropriate selection of the parameters.

The preconditioner seems to be insensitive to both the size of the system and the time interval considered. As the size of the problem increases, the better performances of Monodomain preconditioner applied to the non-symmetric Bidomain with respect to ILU preconditioner applied to the symmetric Bidomain become even more evident. No parallelism has been included in the preconditioner solution, and we expect that its adoption could provide a strong improvement in terms of CPU times. For this reasons, one of the future development of this work will be the extension to parallel implementations of this preconditioner.

Another aspect that deserves to be investigated is the analysis of the role of parameter λ and the identification of a possible "optimal" value.

Acknowledgments

The authors thank Michele Benzi (Emory University) for many fruitful discussions and suggestions in preparing this work. Fabio Nobile and Alessandro Veneziani have been partially supported by the INDAM Project "Mathematical and Numerical Modelling of the Electro-Fluid-Mechanics of the Heart". Fabio Nobile has been partially supported by Italian project PRIN 2005 "Numerical Modeling for Scientific Computing and Advanced Applications".

References

[1] LifeV Software, <http://www.LifeV.org>.
[2] Trilinos Software, <http://trilinos.sandia.gov>.
[3] N. Ajmone Marsan, M. Henneman, J. Chen, C. Ypenburg, P. Dibbets, S. Ghio, G. Bleeker, M. Stokkel, E.E. van der Wall, L. Tavazzi, E. Garcia, J. Bax, Left ventricular dyssyncrony assessed by two three-dimensional imaging modalities: phase analysis of gated myocardial perfusion SPECT and tri-plane tissue Doppler imaging, Eur. J. Nucl. Med. Mol. Imaging 35 (2008) 166–173.
[4] A. Alonso-Rodriguez, L. Gerardo-Giorda, New non-overlapping domain decomposition methods for the time-harmonic Maxwell system, SIAM J. Sci. Comput. 28 (1) (2006) 102–122.
[5] M.E. Belik, T.P. Usyk, A.D. McCulloch, Computational methods fro cardiac electrophysiology, in: N. Ayache (Ed.), Computational Models for the Human Body, Handbook of Numerical Analysis, Elsevier, North Holland, 2004.
[6] M. Benzi, G. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. (2005) 1–137.
[7] P. Bochev, R. Lehouc, On the finite element solution of the pure Neumann problem, SIAM Rev. 47 (1) (2005) 50–66.
[8] J. Chen, E.V. Garcia, R. Folks, C. Cooke, T. Faber, E. Tauxe, A. Iskandrian, Onset of left ventricular mechanical contraction as determined by phase analysis of ECG-gated myocardial perfusion SPECT imaging: development of a diagnostic tool for assessment of cardiac mechanical dyssyncrony, J. Nucl. Cardiol. 12 (6) (2005) 687–695.
[9] R. Clayton, A. Panfilov, A guide to modelling cardiac electrical activity in anatomically detailed ventricles, Progr. Biophys. Mol. Biol. 96 (1–3) (2008) 19–43.
[10] J. Clements, J. Nenonen, P. Li, M. Horacek, Activation dynamics in anisotropic cardiac tissue via decoupling, Ann. Biomed. Eng. 32 (7) (2004) 984–990.
[11] P. Colli Franzone, L. Guerri, M. Pennacchio, B. Taccardi, Spread of excitation in 3-d models of the anisotropic cardiac tissue. iii: Effects of ventricular geometry and fiber structure on the potential distribution, Math. Biosci. 151 (1998) 51–98.
[12] P. Colli Franzone, L.F. Pavarino, A parallel solver for reaction-diffusion systems in computational electrocardiology, Math. Mod. Meth. Appl. Sci. 14 (6) (2004) 883–911.
[13] P. Colli Franzone, L.F. Pavarino, B. Taccardi, Simulating patterns of excitation, repolarization and action potential duration with cardiac Bidomain and Monodomain models, Math. Biosci. 197 (2005) 35–66.
[14] P. Colli Franzone, G. Savarè, Degenerate evolution systems modeling the cardiac electric field at micro and macroscopic level, In: A. Lorenzi, B. Ruf (Eds.), Evolution Equations, Semigroups and Functional Analysis: in memory of Brunello Terreni, vol. 50, Progress in Nonlinear Differential Equations and their Applications, Birkhäuser Verlag, 2002, pp. 49–78.
[15] R. FitzHugh, Impulses and physiological states in theoretical models of nerve membrane, Biophys. J. 1 (1961) 445–466.
[16] I. Fried, Bounds on the extremal eigenvalues of the finite element stiffness and mass matrices and their spectral condition number, J. Sound Vib. 22 (1972) 407–418.
[17] L. Gerardo-Giorda, P. Le Tallec, F. Nataf, A Robin–Robin preconditioner for advection-diffusion equations with discontinuous coefficients, Comput. Meth. Appl. Mech. Eng. 193 (9–11) (2004) 745–764.
[18] C. Henriquez, Simulating the electrical behavior of cardiac tissue using the Bidomain model, Crit. Rev. Biomed. Eng. 21 (1993) 1–77.
[19] G. Huiskamp, Simulation of depolarization in a membrane-equation-based model of the anisotropic ventricle, IEEE Trans. Biomed. Eng. 45 (7) (1998) 847–855.
[20] J.P. Keener, Direct activation and defibrillation of cardiac tissue, J. Theor. Biol. 178 (1996) 313–324.
[21] J. Le Grice, B. Smaill, P. Hunter, Laminar structure of the heart: a mathematical model, Am. J. Physiol. 272 (41) (1995) H2466–H2476. Heart Circ. Physiol.

[22] L. Luo, Y. Rudy, A model of the ventricular cardiac action potential: depolarization, repolarization and their interaction, Circ. Res. 68 (1991).

[23] B.F. Nielsen, T.S. Ruud, G.T. Lines, A. Tveito, Optimal Monodomain approximation of the Bidomain equations, Appl. Math. Comput. 184 (2007) 276–290.

[24] L.F. Pavarino, S. Scacchi, Multilevel additive Schwarz preconditioners for the Bidomain reaction-diffusion system, SIAM J. Sci. Comput. 31 (1) (2008) 420–443.

[25] M. Pennacchio, V. Simoncini, Efficient algebraic solution of reaction-diffusion systems for the cardiac excitation process, J. Comput. Appl. Math. 145 (1) (2002) 49–70.

[26] M. Potse, B. Dubé, J. Richer, A. Vinet, A comparison of Monodomain and Bidomain reaction-diffusion models for action potential propagation in the human heart, IEEE Trans. Biomed. Eng. 53 (12) (2006) 2425–2435.

[27] A. Pullan, M. Buist, L. Cheng, Mathematical Modelling the Electrical Activity of the Heart, World Scientific, Singapore, 2005.

[28] P. Colli Franzone, L. Pavarino, G. Savaré, Computational electrocardiology: mathematical and numerical modeling, in: A. Quarteroni, L. Formaggia, A. Veneziani (Eds.), Complex Systems in Biomedicine, Springer, Milan, 2006.

[29] J. Rogers, A. McCulloch, A collocation-Galerkin finite element model of cardiac action potential propagation, IEEE Trans. Biomed. Eng. 41 (1994) 743–757.

[30] B. Roth, Action potential propagation in a thick strand of cardiac muscle, Circ. Res. 68 (1991) 162–173.

[31] Y. Rudy, J. Silva, Computational biology in the study of cardiac ion channels and cell electrophysiology, Quart. Rev. Biophys. 39 (1) (2006) 57–116.

[32] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS, Boston, 1996.

[33] F.B. Sachse, Computational Cardiology, Springer, Berlin, 2004.

[34] S. Sanfelici, Convergence of the Galerkin approximation of a degenerate evolution problem in electrocardiology, Numer. Meth. Part. Diff. Eqn. 18 (2002) 218–240.

[35] S. Scacchi, A hybrid multilevel Schwarz method for the Bidomain model, Comput. Meth. Appl. Mech. Eng. 197 (45–48) (2008) 4051–4061.

[36] S. Scacchi, Multilevel Schwarz preconditioner for the Bidomain system and applications in Electrocardiology, Ph.D. Thesis, University of Pavia, 2008.

[37] M. Veneroni, Reaction-diffusion systems for the microscopic cellular model of the cardiac electric field, Math. Meth. Appl. Sci. 29 (14) (2006) 1631–1661.

[38] E. Vigmond, R. Weber dos Santos, A. Prassl, M. Deo, G. Plank, Solvers for the cardiac Bidomain equations, Progr. Biophys. Mol. Biol. 96 (1–3) (2008) 3–18.

[39] E.J. Vigmond, F. Aguel, N.A. Trayanova, Computational techniques for solving the Bidomain equations in three dimensions, IEEE Trans. Biomed. Eng. 49 (11) (2002) 1260–1269.

[40] R.L. Winslow, J.J. Rice, S. Jafri, E. Marban, B. O'Rourke, Mechanisms of altered excitation–contraction coupling in canine tachycardia-induced heart failure, ii: model studies, Circ. Res. 84 (5) (1999) 571–586.