FAST STRUCTURED AMG PRECONDITIONING FOR THE BIDOMAIN MODEL IN ELECTROCARDIOLOGY.

MICOL PENNACCHIO* AND VALERIA SIMONCINI†

Abstract. The electrical activity of the heart may be modeled by a non-linear system of partial differential equations known as the bidomain model. Due to the rapid variations in the electrical field, accurate simulations require a fine-scale discretization of the equations and consequently the solution of large severely ill-conditioned linear systems at each time step. Solving these systems is a major bottleneck of the whole simulation. We propose a highly effective preconditioning strategy for a general and popular 3D formulation of the problem. A theoretical analysis of the preconditioned matrix ensuring mesh independence of the spectrum is also described. Numerical comparisons with state of the art approaches confirm the effectiveness of our preconditioning technique. Finally, we show that an equivalent but less exercised formulation provides the best performance, in terms of CPU time.

1. Introduction. Computer simulations of cardiac electrophisiology are a helpful tool in the study of the bioelectric activity of the heart both in normal and pathological conditions. These simulations are based on a system of two PDEs, named the *bidomain system*, and are coupled to a set of ODEs describing the membrane kinetics of cardiac cells. Indeed the excitation process in the myocardium is a complex phenomenon characterized by rapid ionic fluxes through the cellular membrane separating the intracellular and the interstitial fluid in the myocardium, see [16, 21, 38] for details.

Due to its time and space multiscale nature, the numerical solution of the bidomain system represents a very intensive computational task. More specifically, because of the rapid variations in the electrical field, sufficiently accurate simulations require a fine-scale discretization of the equations. For realistic geometries this leads to a large number of grid points, at least $O(10^7)$, and consequently large linear systems have to be solved each time step. To limit the computational cost, adaptive techniques and domain decomposition methods have been developed [10, 22, 30, 33, 53, 59]. Adaptivity in space *and* time may represent a valid solution to reduce the computational cost of the bidomain system, see, e.g., [6], although until now successful investigations have only been reported for moderate size problems. Earlier bidomain investigations based on finite difference discretizations can be found in [5, 39, 44].

Fully implicit methods in time have been considered in few studies (see e.g. [26, 24, 25]) and require the solution of non-linear systems at each time step. However, most numerical studies now employ semi-implicit methods in time: these allow for wider time steps than explicit schemes, at the cost of dealing with a large algebraic linear system at each time step, whose conditioning considerably worsens as the problem dimension increases, resulting in an unacceptable increase in the computational costs of the whole simulation. In this context, preconditioning is therefore mandatory. The design of computationally effective iterative solvers for such linear systems calls for the construction of efficient preconditioners, see [57] for a detailed overview of the methods. Attempts in the literature range from diagonal preconditioners [46], Symmetric Successive Over Relaxation [32], Block Jacobi preconditioners with incomplete LU factorization (ILU) for each block [10, 56], to multigrid [51, 1, 37, 58, 45].

*Istituto di Matematica Applicata e Tecnologie Informatiche del C.N.R., via Ferrata, 1 - 27100 Pavia, Italy (micol@imati.cnr.it)

†Dipartimento di Matematica, Università di Bologna, Piazza di Porta S. Donato, 5, I-40127 Bologna, Italy; CIRSA, Ravenna and IMATI-CNR, Pavia, Italy (valeria@dm.unibo.it).

In this work we build upon our former experience with structured preconditioning techniques in the 2D setting (cf. [34, 35]), to propose a highly effective preconditioning strategy for a general and popular 3D formulation of the problem. The approach does not require any parameter tuning and relies on a public-domain, recently developed algebraic multigrid solver as building block. The resulting preconditioner is *non-symmetric*, however the spectral properties are so favourable that a cheap short-term recurrence method such as BICGSTAB can be effectively employed. We describe a novel theoretical analysis of the preconditioned matrix ensuring mesh independence of the spectrum if appropriate multigrid strategies are used. Numerical comparisons with state of the art approaches confirm the effectiveness of our preconditioning strategy. In addition, we show that an equivalent but less exercised formulation provides the best performance, in terms of CPU time, with a saving of 50% of the total computational time with respect to the more common formulation. These results envision the possibility to simulate the whole excitation process in a very affordable time frame.

General geometric and algebraic multigrid preconditioning have been already applied to the bidomain system in one of the two formulations we shall adopt, and their effectiveness when compared to other classical methods has been reported; see, e.g., [51, 1, 37, 58, 59, 57]. However, most of these studies rely on the application of multigrid methods only on the elliptic equation of the bidomain system related to the extracellular potential. It was recently shown in [48] that, for the same level of accuracy, solving the coupled system of elliptic-parabolic equations is more efficient than dealing with the two equations separately.

The outline of the paper is as follows. In section 2 we introduce the bidomain system and two of its mainly used formulations. Space and time discretizations are presented in the same section, leading to the description of the algebraic linear systems to be solved at each time step. By using a unified form for the linear system stemming from the discretization of both formulations, in section 3 we introduce the class of structured preconditioners that we wish to analyze. Section 4 is devoted to the convergence analysis of the employed structured preconditioners. To the best of our knowledge, this provides the most realistic spectrum analysis in the literature for this general preconditioning strategy. Finally, Section 6 reports on numerical experiments showing the effectiveness of the block structured algebraic multigrid approach for one formulation, and the effectiveness of using the less common formulation.

Throughout the paper we shall often deal with singular matrices. With a little abuse of notation, we shall denote by A^{-1} the pseudoinverse of the symmetric and singular matrix A. Nonetheless, the singularity of the matrix is fully taken into account when dealing with A^{-1}.

2. The Bidomain Model. The Bidomain model is the most complete model to simulate the bioelectric activity of the heart (see [17, 31, 55]), and it consists of a nonlinear Reaction-Diffusion (R-D) system of equations for the intra- and extracellular potentials u_i and u_e, coupled through the transmembrane potential $v := u_i - u_e$. In this model the cardiac muscle is viewed as two superimposed anisotropic continuous media, intra (i) and extracellular (e), occupying the same volume and separated from each other by the cell membrane. The non linearity arises through the current-voltage relationship across the membrane which is described by a set of nonlinear ODEs (see [23] for more details).

Cardiac tissue model with fiber architecture and anisotropy. The domain $\Omega \subset \mathbb{R}^3$, chosen in this work to represent the left ventricle, is modeled by a family of truncated

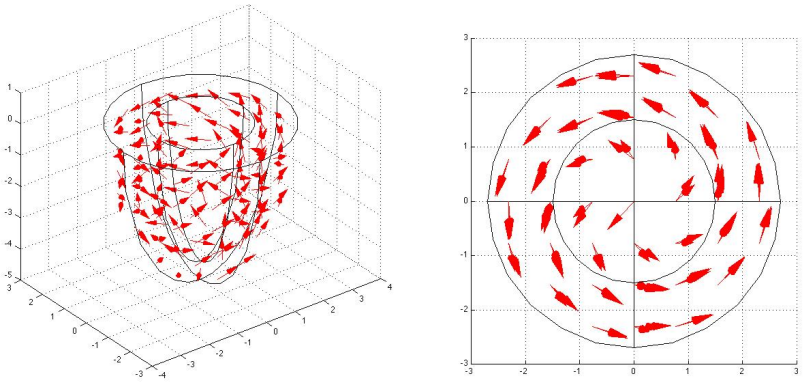

FIG. 2.1. *Fiber direction on the truncated ellipsoidal geometry modeling the left ventricle (left) and on an horizontal section of it (right).*

ellipsoids described by the parametric equations:

$$\begin{array}{ll} x = a(r)\ \cos\theta\ \cos\varphi & \theta_1 \le \theta \le \theta_2, \\ y = a(r)\ \cos\theta\ \sin\varphi & 0 \le \varphi \le 2\pi, \\ z = c(r)\ \sin\theta & 0 \le r \le 1, \end{array}$$

where $a(r) = a_1 + r\ (a_2 - a_1)$, $c(r) = c_1 + r\ (c_2 - c_1)$ and $a_i, c_i, i = 1, 2$ are given coefficients determining the main axes of the ellipsoid, see Fig. 2.1. The surfaces corresponding to $r = 0$ and $r = 1$ describe the endo- and epicardium respectively.

Anatomic studies showed that the fiber direction rotates counterclockwise from epicardium to endocardium and that fibers are arranged in sheets, running through the myocardial wall [49, 36]. The fiber structure and the anisotropic properties of the cardiac tissue are modeled by the intra- and extracellular conductivity tensors $M_i = M_i(\boldsymbol{x})$ and $M_e = M_e(\boldsymbol{x})$ defined as:

$$M_s(\boldsymbol{x}) = \sigma_l^s\ \mathbf{a}_l(\boldsymbol{x})\mathbf{a}_l(\boldsymbol{x})^T + \sigma_n^s\ \mathbf{a}_n(\boldsymbol{x})\mathbf{a}_n(\boldsymbol{x})^T + \sigma_t^s\ \mathbf{a}_t(\boldsymbol{x})\mathbf{a}_t(\boldsymbol{x})^T \qquad s = i, e, \quad (2.1)$$

with $\mathbf{a}_l(\boldsymbol{x})$ unit vector tangent to the cardiac fiber at a point $\boldsymbol{x} \in \Omega$, $\mathbf{a}_t$ orthogonal to the fiber direction in the fiber sheet and $\mathbf{a}_n$ orthogonal to the sheet. The conductivity coefficients measured along the corresponding directions are $\sigma_l^s, \sigma_l^n, \sigma_t^s$, $s = i, e$ respectively. If we have axial isotropy, i.e. same conductivity in both tangential and normal directions, the tensors can be written as:

$$M_s(\boldsymbol{x}) = \sigma_t^s I + (\sigma_l^s - \sigma_t^s)\ \mathbf{a}_l(\boldsymbol{x})\mathbf{a}_l(\boldsymbol{x})^T, \qquad s = i, e, \quad (2.2)$$

where I is the identity matrix and $\sigma_l^s,\ \sigma_t^s$ for $s = i, e$ are the conductivity coefficients along and across fiber, in the (i) and (e) media, assumed constant with $\sigma_l^s > \sigma_t^s > 0$. In both cases it can be easily verified that $M_{i,e}$ satisfy the following uniform ellipticity condition:

$$\exists \lambda_m^s, \lambda_M^s > 0, \qquad \lambda_m^s |\boldsymbol{\xi}|^2 \le \boldsymbol{\xi}^T M_s(\boldsymbol{x}) \boldsymbol{\xi} \le \lambda_M^s |\boldsymbol{\xi}|^2 \qquad \forall \boldsymbol{\xi} \in \mathbb{R}^3, \boldsymbol{x} \in \Omega,\ s = i, e. \quad (2.3)$$

The fibers rotate intramurally linearly with the depth for a total amount of 120^o proceeding counterclockwise from epicardium and endocardium. More precisely, in a local ellipsoidal reference system $(\mathbf{e}_\phi, \mathbf{e}_\theta, \mathbf{e}_r)$,the fiber direction $\mathbf{a}(\boldsymbol{x})$ at a point $\boldsymbol{x}$ is given by:

$$\mathbf{a}(\boldsymbol{x}) = \mathbf{e}_\phi \cos(\alpha(r)) + \mathbf{e}_\theta \sin(\alpha(r)), \qquad \text{with } \alpha = \frac{2}{3}\pi(1-r) - \frac{\pi}{4}, \qquad 0 \le r \le 1.$$

The obliqueness of $\mathbf{a}$ with respect to the ellipsoidal surfaces is modeled by introducing the "oblique" angle β (called *imbrication angle*) which describes the deviation of $\mathbf{a}$ from the tangent position, for more details see [9]. Figure 2.1 shows the fiber directions on the truncated ellipsoidal geometry modeling the left ventricle (left plot) and on an horizontal section of it (right plot).

The R-D system governing the cardiac electric activity may be written in various forms involving different combinations of the variables u_i, u_e, v; see, e.g., [18, 32]. Here we consider two different formulations, the (u_e, v) and the (u_i, u_e) formulations. For both formulations we will deal with a FEM discretization in space and a semi–implicit scheme in time: implicit for the diffusion term and explicit for the reaction term. In spite of the close relationship between the resulting formulations, the associated linear systems provide different challenges, so that the same general preconditioning strategy may lead to surprisingly different performance on the two formulations. We wish to stress that while the (v, u_e) formulation is well accounted for in the current literature, the (u_i, u_e) formulation has been somehow overlooked. This is mostly due to the fact that v is the quantity of chief interest as well as being directly observable from the single cell to the tissue level through optical mapping [13]. Moreover, formulation (v, u_e) has also been preferred for computational reasons: the parabolic and elliptic equations of the formulation can be solved one after the other by means of a block Gauss–Seidel method, see [1, 32, 57]. Nonetheless, we believe the (u_i, u_e) formulation deserves higher consideration due to its very favourable computational performance, also taking into account that v can be readily recovered as $v = u_i - u_e$.

(u_i, u_e) *formulation.* The evolution of the intra and extracellular potentials $u_i(\boldsymbol{x}, t)$ and $u_e(\boldsymbol{x}, t)$ and transmembrane potential $v(\boldsymbol{x}, t) = u_i(\boldsymbol{x}, t) - u_e(\boldsymbol{x}, t)$ is described by the following reaction–diffusion system, characterized by two nonlinear parabolic equations coupled through the reaction term I_{ion} with a system of ODE's describing the evolution of the gating variables and ion concentration:

given $I_{app} : \Omega\times]0,T[\to \mathbb{R}$ *and* $u_e^0, u_i^0 : \Omega \to \mathbb{R}$, *find* $u_i, u_e : \Omega\times]0,T[\to \mathbb{R}$ *and* $v = u_i - u_e$ *such that*:

$$\begin{cases} c_m \partial_t v - \text{div } M_i \nabla u_i + I_{ion}(v, w) = I_{app} & \text{in } \Omega\times]0,T[\\ c_m \partial_t v + \text{div } M_e \nabla u_e + I_{ion}(v, w) = I_{app} & \text{in } \Omega\times]0,T[\\ \mathbf{n}^T M_{i,e} \nabla u_{i,e} = 0 & \text{on } \Gamma\times]0,T[\\ v(\boldsymbol{x}, 0) = 0 & \text{in } \Omega \end{cases} \tag{2.4}$$

where $\Omega \subset \mathbb{R}^3$ models the heart tissue, $\Gamma = \partial\Omega$, $\mathbf{n}$ denotes the outward unit normal to the boundary Γ and I_{app} is an applied current used to initiate the process. The function I_{ion} is a nonlinear function of the transmembrane potential v and of suitable ionic variables denoted by w related to the ionic model chosen. There are additional ordinary differential equations governing the evolution of v, the analysis of which is beyond the scope of this work. Problem (2.4) admits the following variational formulation:

find $u_i, u_e :]0,T[\to H^1(\Omega)$ such that:

$$\begin{cases} c_m \dfrac{d}{dt}\left(v(t),\varphi\right) + a_i\left(u_i(t),\varphi\right) + \left(I_{ion}(v(t),w(t)),\varphi\right) = \left(I_{app}(t),\varphi\right) & \forall \varphi \in V \\ c_m \dfrac{d}{dt}\left(v(t),\varphi\right) - a_e\left(u_e(t),\varphi\right) + \left(I_{ion}(v(t),w(t)),\varphi\right) = \left(I_{app}(t),\varphi\right) & \forall \varphi \in V \end{cases} \tag{2.5}$$

where $v(t) = u_i(t) - u_e(t)$, $V = H^1(\Omega)$, $(\psi,\phi) = \int_\Omega \psi\ \phi\ dx \quad \forall \psi,\ \phi \in L^2(\Omega)$ and $a_s(\cdot,\cdot) : V \times V \to \mathbb{R}$, $s = i, e$, are the bilinear symmetric continuous forms defined as

$$a_s(\psi,\phi) = \int_\Omega (\nabla\psi)^T M_s \nabla\phi\ dx, \qquad \forall \psi,\phi \in V \qquad s = i, e. \tag{2.6}$$

Well posedness analysis of the Bidomain system with FitzHugh-Nagumo ionic model and more general ionic kinetics can be found in [11] and [55] respectively. For applied current I_{app} satisfying the compatibility condition $\int_\Omega I_{app}\, dx = 0$, system (2.4) uniquely determines v, while the potentials u_i and u_e are defined only up to the same additive time-dependent constant related to the reference potential, see [11]. To fix such arbitrary constant we impose that u_e has zero average on Ω.

The preconditioners proposed in this paper are mainly related to the structure of the partial differential equations of the bidomain system and are independent of membrane kinetics considered. We do not deal with a specific choice of the ionic model, however, in the numerical results presented in Section 6 we will use the Rogers-McCulloch model, that is:

$$I_{ion}(v,w) = Gv\left(1 - \frac{v}{v_{th}}\right)\left(1 - \frac{v}{v_p}\right) + \eta_1 v w, \qquad \frac{\partial w}{\partial t} = \eta_2\left(\frac{v}{v_p} - \eta_3 w\right),$$

where $G, \eta_1, \eta_2, \eta_3$ are positive real coefficients, v_{th} is the threshold potential and v_p the peak potential.

(v, u_e) *formulation.* The R-D system governing the cardiac electric activity may be written in various forms involving different combinations of the variables u_i, u_e, v; see, e.g., [32, 34]. Here we deal with the formulation generally used for the numerical simulations, i.e. with a parabolic equation for the transmembrane potential v coupled with an elliptic equation for the extracellular potential u_e:

find $(v(\boldsymbol{x},t), u_e(\boldsymbol{x},t))$, $\boldsymbol{x} \in \Omega$, $t \in [0,T]$ *such that*

$$\begin{cases} c_m \partial_t v - \text{div}\ M_i \nabla v + I_{ion}(v,w) = \text{div}\ M_i \nabla u_e + I_{app} & \text{in } \Omega\times]0,T[\\ -\text{div}\ M \nabla u_e = \text{div}\ M_i \nabla v & \text{in } \Omega\times]0,T[\\ \mathbf{n}^T M_i \nabla v = 0, \qquad \mathbf{n}^T M \nabla u_e = 0 & \text{on } \Gamma\times]0,T[\\ v(\boldsymbol{x},0) = 0 & \text{in } \Omega, \end{cases} \tag{2.7}$$

with $M = M_i + M_e$ bulk conductivity tensor. We will refer to (2.7) as the (v, u_e) formulation of the bidomain model.

The above system differs from problem (2.4) in that the change of variable allows us to replace the second parabolic equation with an elliptic equation, thus loosing the degenerate temporal structure of system (2.4). This approach was usually preferred because the two equations can be solved one after the other.

Space and Time discretization. Let $\mathcal{T}^h$ be a uniform unstructured tetrahedral mesh of Ω and V^h the associated space of P^1 finite elements. We obtain a semi-discrete problem by applying a standard Galerkin procedure. Choosing a finite element basis

$\{\varphi_i\}$ for V^h, we denote by

$$C = \left\{ c_{rk} = \int_\Omega \varphi_r \varphi_k \, dx \right\}, \quad A_s = \left\{ a_{r,k} = \sum_{K \in \mathcal{T}_h} \int_K (\nabla \phi_r)^T M_s \nabla \phi_k \, dx \right\} \quad s = i, e,$$

the symmetric mass matrix and stiffness matrices and by I_{ion}^h and I_{app}^h the finite element interpolants of I_{ion} and I_{app}, respectively. By using (2.3) and standard finite element arguments, we can assert that A_s, $s = i, e$ are symmetric and positive semidefinite with $A_s \mathbf{e} = 0$; in particular, $a_{k,k}$ for $k = 1, n$ are positive and $a_{k,k} = -\sum_{j \neq k} a_{j,k}$.

The time discretization is performed by a semi–implicit scheme (backward Euler method): the diffusion term is treated implicitly while the nonlinear reaction term I_{ion} explicitly. Then the following general algebraic system can be obtained:

$$\mathcal{M} \boldsymbol{\xi}^{k+1} = \mathbf{b}^k \qquad \text{with} \qquad \mathcal{M} = \begin{bmatrix} C_t + A_i & -C_t \\ -C_t & C_t + A_e \end{bmatrix}, \tag{2.8}$$

with $\mathbf{b}^k = \left[C_t \, \mathbf{v}^k - I_{ion}^h(\mathbf{v}^k, \boldsymbol{w}^{k+1}) + I_{app}^h; -C_t \, \mathbf{v}^k + I_{ion}^h(\mathbf{v}^k) - I_{app}^h \right]$, $C_t = \frac{c_m}{\tau} \, C$ diagonal with positive diagonal entries, τ the time step, $\mathbf{v}^k = \mathbf{u_i}^k - \mathbf{u_e}^k$ and $\boldsymbol{\xi}^{k+1} = \left[\mathbf{u_i}^{k+1}; \mathbf{u_e}^{k+1} \right]$. It may be easily verified that $\mathcal{M}$ is positive semidefinite (cf, e.g., [32]). Moreover, for $\mathbf{e}$ the vector of all ones, $\mathcal{M}[\mathbf{e}; \mathbf{e}] = \mathbf{0}$ and the system is consistent, in that $\mathbf{b}$ has zero mean, that is $\mathbf{e}^T \mathbf{b} = \mathbf{0}$. As in the continuous model, $\mathbf{v}^k$ is uniquely determined, while $\mathbf{u_i}^k$ and $\mathbf{u_e}^k$ are determined only up to the same additive time-dependent constant.

When considering the (v, u_e) formulation and using again a finite element discretization in space and a semi–implicit scheme in time, we get:

$$\mathcal{M} \boldsymbol{\xi}^{k+1} = \mathbf{b}^k \qquad \text{with} \qquad \mathcal{M} = \begin{bmatrix} C_t + A_i & A_i \\ A_i & A_i + A_e \end{bmatrix}, \tag{2.9}$$

with $\mathbf{b}^k = \left[C_t \, \mathbf{v}^k - I_{ion}^h(\mathbf{v}^k) + I_{app}^h; \mathbf{0} \right]$, $\mathbf{v}^k = \mathbf{u_i}^k - \mathbf{u_e}^k$, $\boldsymbol{\xi}^{k+1} = \left[\mathbf{v}^{k+1}; \mathbf{u_e}^{k+1} \right]$. It can be easily proven that the symmetric matrix $\mathcal{M}$ is positive semidefinite, see [34].

Whichever formulation is chosen for the bidomain system, at each time step a large linear system $\mathcal{M} \boldsymbol{\xi}^{k+1} = \mathbf{b}^k$ of size $2n \times 2n$ needs to be solved, whose conditioning considerably worsens as the problem dimension increases, resulting in an unacceptable increase in the computational costs of the whole simulation. In this context, preconditioning is therefore mandatory.

REMARK 2.1. Classically system (2.9) is solved by a block Gauss–Seidel method involving the two diagonal blocks [8]. To allow for accurate solutions the two equations should not be decoupled: an outer iteration has to be employed to reach accurate approximations of the exact solution of the coupled system, see [8, 32]. The overall workload is dominated by the solution of the elliptic equation and typically convergence is reached in less than 10 outer iterations. Nonetheless, to increase efficiency, most studies employing this strategy consider the decoupled components independently, and no outer iteration is carried out. On the other hand, it was recently shown in [48] that the decoupled solution method is less efficient than a coupled scheme for the same level of accuracy. In particular, it was experimentally verified in [48] that the time per iteration is approximately twice as large for the coupled system than it is for the uncoupled one. However, the coupled method is faster overall since it requires fewer iterations per time step to converge with the same accuracy to the solution of the bidomain system.

3. Block Preconditioners. Both formulations are characterized by a 2×2 block form, in which all blocks are square and symmetric, while the diagonal blocks are positive (semi)definite. It is therefore natural to derive preconditioners that exploit this structure, as is the case in various multilevel methods (see, e.g., [47, 52]) as well as in saddle point problems [3]. On the other hand, our previous numerical experience, see, e.g., [34, 35], showed that the preconditioning strategies for the two considered formulations, (u_i, u_e) and (v, u_e), need to be very different to achieve best performance. In particular, in the (u_i, u_e) formulation best timings were observed when an appropriate preconditioner was used for the whole matrix, that is the block structure was not taken into account. On the contrary, block structured preconditioning was absolutely required for the (v, u_e) formulation. The latter setting is described next in detail.

Symmetric preconditioners. Effective structured symmetric strategies include block diagonal and block factorized preconditioners:

$$\mathcal{P}_d = \text{blockdiag}(K, D), \qquad \mathcal{P}_f = \begin{bmatrix} I & O \\ A_i K^{-1} & I \end{bmatrix} \begin{bmatrix} K & A_i \\ O & D \end{bmatrix}, \tag{3.1}$$

where K is a symmetric and positive definite approximation to the (1,1) block, while D is a symmetric and positive definite approximation either to the Schur complement or to the (2,2) block [2]; we refer to [34] for computational experience with these preconditioners in the 2D setting. Note that if K and D exactly reproduce the (1,1) block and the Schur complement, respectively, then $\mathcal{P}_{f,ideal} = \mathcal{M}$. This ideal choice turns out to be computationally unfeasible because too expensive for large problems.

The application of the factorized preconditioner $\mathcal{P}_f$ corresponds to the solution of two block triangular systems, that is

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} = \mathcal{P}_f^{-1} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = \begin{bmatrix} K^{-1} & -K^{-1} A_i D^{-1} \\ O & D^{-1} \end{bmatrix} \begin{bmatrix} I & O \\ -A_i K^{-1} & I \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix},$$

showing that the application of $\mathcal{P}_f^{-1}$ requires two solves with K and one solve with D, together with two (cheap) multiplications with the sparse matrix B_{12}.

In [34] we discussed a-priori convergence results on the convergence of CG when $\mathcal{P}_d$ or $\mathcal{P}_f$ are used as preconditioners, for various selections of K and D. These algebraic results also apply to our 3D setting (cf. section 4).

Non-symmetric preconditioners. The following nonsymmetric "one-sided" version of $\mathcal{P}_f$ has been used for nonsymmetric and also for symmetric (indefinite) saddle point problems (see, e.g., [3]):

$$\mathcal{P}_M = \begin{bmatrix} K & A_i \\ O & D \end{bmatrix}.$$

If K and D exactly reproduced the (1,1) block and the Schur complement, respectively, the (nonsymmetric) preconditioned matrix would be the matrix

$$\mathcal{M}\mathcal{P}_M^{-1} = \begin{bmatrix} I & O \\ A_i (C_t + A_i)^{-1} & I \end{bmatrix},$$

whose spectrum consists of the single unit eigenvalue, so that a minimal residual method such as GMRES ([42]) would converge in at most two iterations. In general, the behavior of the approximate versions of K and D is less predictable. The

performance of $\mathcal{P}_M$ within the indefinite saddle point context highly overcomes its nonsymmetric nature. The situation is considerably different in our context, where the original matrix is positive (semi)definite. Applying $\mathcal{P}_M$ destroys symmetry. Remarkably, however, $\mathcal{P}_M$ appears to be very competitive with respect to the symmetric choices. First numerical evidence was reported in [34], while a preliminary convergence analysis under somewhat restrictive hypotheses was proposed in [35]. Here we provide an analysis of the spectrum location of the preconditioned problem when K and D are spectrally equivalent approximations to the ideal cases. In particular, if the equivalence scalars (cf. (4.2)-(4.3)) do not depend on the mesh parameters, then we show that the spectral region of $\mathcal{P}_M$ is also bounded independently of the mesh parameter.

4. Spectral analysis. In this section we analyze more in detail the spectral properties of the block upper triangular preconditioner described in the previous section. To this end we focus on the (v, u_e) formulation. The analysis of nonsymmetric structured preconditioners with approximation matrices K and D is very hard in general, usually providing only large overestimates of the actual spectrum. Due to the connection between $\mathcal{P}_M$ and $\mathcal{P}_f$, we are able to more accurately describe the spectral region of the matrix $\mathcal{M}\mathcal{P}_M^{-1}$.

In the following, the notation $B \leq A$ for two square matrices B, A means that $A - B$ is positive definite; if A and/or B are singular, the notation silently assumes that the relation holds only in the matrix ranges. Moreover, whenever the inverse of a singular matrix is used, this inverse is meant to be a pseudo-inverse, as all computation is assumed to be performed in the range of the given matrices.

For $K = A_i + C_t$ and $D = A_i + A_e$ (here and in the following the *ideal* choices), the matrix $\mathcal{M}\mathcal{P}_M^{-1}$ can be written as:

$$\mathcal{M}\mathcal{P}_M^{-1} = \begin{bmatrix} I & O \\ A_i(A_i + C_t)^{-1} & I \end{bmatrix} \begin{bmatrix} I & O \\ O & I - \mathcal{S} \end{bmatrix}, \; \mathcal{S} = A_i(A_i + C_t)^{-1} A_i (A_i + A_e)^{-1}, \tag{4.1}$$

whose *real* spectrum is bounded independently of the mesh parameter [35].

In the "non-ideal" case, we assume that K and D are spectrally equivalent to the matrices they approximate. More precisely, there exist positive scalars α_j and β_j, $j = 1, 2$ such that

$$\alpha_1(A_i + C_t) \leq K \leq \alpha_2(A_i + C_t), \tag{4.2}$$

$$\beta_1(A_i + A_e) \leq D \leq \beta_2(A_i + A_e). \tag{4.3}$$

The bounds above are assumed to hold on the range of the given matrices. Table 4.1 reports rough estimates of these parameters for the example in section 6 for various discretization refinements, when $A_i + A_e$ is approximated by an Algebraic Multigrid method (AMG), while $A_i + C_t$ is approximated by its diagonal. We refer to section 6 for additional information on the actual implementation. The numbers clearly show independence of the mesh size in all cases except for β_2, where a mild dependence is observed. We should add that such dependence may be due to a severe inaccuracy in the rather expensive estimation of this parameter.

To proceed with a spectral analysis when K and D are not ideal, we start by summarizing some relevant results from [34] with the current notation. To this end, we recall that the C.B.S. constant γ is defined as (see, e.g., [2])

$$\gamma^2 = \sup_{\mathbf{v} \in \mathbb{R}^n \setminus N(A_i + A_e)} \frac{\mathbf{v}^T A_i (A_i + C_t)^{-1} A_i \mathbf{v}}{\mathbf{v}^T (A_i + A_e) \mathbf{v}}, \tag{4.4}$$

n	α_1	α_2	β_1	β_2
22,782	1.000000331762804	1.000000419186690	0.9979	49.60
52,131	1.000000000037860	1.000004561309334	0.9981	39.80
172,785	1.000000001108531	1.000110488680402	0.9982	65.77
333,373	1.000000004725318	1.000294287694825	0.9972	70.45

TABLE 4.1
(v, u_e) formulation. Values of $\alpha_1, \alpha_2, \beta_1, \beta_2$ for different problem sizes.

and it holds that ([34, Lemma 4.1]) $\gamma^2 \leq (1+\lambda_{\min}(A_e, A_i))^{-1} < 1$, where $\lambda_{min}(A_e, A_i)$ denotes the smallest nonzero eigenvalue of $A_e A_i^{-1}$. Due to the properties of the matrices A_e, A_i, we obtain

$$\lambda_{\min}(A_e, A_i) \geq \ell, \qquad \ell := \lambda_m^e / \lambda_M^i, \tag{4.5}$$

and λ_m^e, λ_M^i defined in (2.3). If we consider axial isotropy, i.e. $M_{i,e}$ defined in (2.2) and for our choice of data (cf. Table 6.1), we obtain $\ell = (\sigma_l^e - \sigma_t^e)/(\sigma_l^i - \sigma_t^i) = 0.2416$. Therefore, γ^2 can be easily bounded, independently of the mesh parameter.

Next proposition collects some spectral bounds obtained for $\mathcal{P}_f^{-1}\mathcal{M}$ in [34, Proposition 4.2] (see also [2, Th.9.5]) and [34, Lemma 4.2].

PROPOSITION 4.1. *Assume that K and D satisfy (4.2) and (4.3), respectively, with $\alpha_2 \geq 1 \geq \alpha_1 > \gamma^2$. Let $\delta_1 = \beta_1$, $\delta_2 = \beta_2 + \alpha_1^{-1}$ and assume that $\delta_2 \geq 1 \geq \delta_1 > \gamma^2$. Moreover, let $\phi(\tau) = \frac{1}{2}(1+\tau) + \sqrt{\frac{1}{4}(1-\tau)^2 + \tau\gamma^2}$. Then*

$$\left(1 + \frac{\max\{\alpha_2, \delta_2\} - 1}{1-\gamma^2}\phi(r_2)\right)^{-1} \leq \lambda(\mathcal{P}_f^{-1}\mathcal{M}) \leq \left(1 - \frac{1 - \min\{\alpha_1, \delta_1\}}{1-\gamma^2}\phi(r_1)\right)^{-1},$$

where $r_2 = \min\{\frac{\alpha_2 - 1}{\delta_2 - 1}, \frac{\delta_2 - 1}{\alpha_2 - 1}\}$ and $\alpha_2 > 1$ and/or $\delta_2 > 1$; and $r_1 = \min\{\frac{1-\alpha_1}{1-\delta_1}, \frac{1-\delta_1}{1-\alpha_1}\}$ and $\alpha_1 < 1$ and/or $\delta_1 < 1$.

The following result proved in [35] shows that the spectrum of $\mathcal{M}\mathcal{P}_M^{-1}$ is bounded independently of the mesh parameter for the ideal choice $K = A_i + C_t$ and for judiciously chosen D. This result was described in the 2D context and it was a first step towards a spectral analysis of the fully approximated preconditioning matrix $\mathcal{P}_M$. In fact the result still holds in the three-dimensional case. In this section we complete this analysis for the general case $K \neq A_i + C_t$. To the best of our knowledge this represents the most exhaustive and realistic spectral analysis available in the literature for this general preconditioning strategy.

THEOREM 4.2. [35, Th. 1] *With the previous notation, let $K = A_i + C_t$.*

i) If $D = A_i + A_e$, then

$$\lambda_{\min}(\mathcal{M}\mathcal{P}_{M,ideal}^{-1}) = 1 - \mu, \quad \lambda_{\max}(\mathcal{M}\mathcal{P}_{M,ideal}^{-1}) = 1,$$

with $\mu \leq (1 + \lambda_{\min}(A_e, A_i))^{-1}$, μ constant independent of h.

ii) If D satisfies (4.3) for some positive β_1, β_2, then either $\lambda(\mathcal{M}\mathcal{P}_M^{-1}) = 1$ or $\beta_1(1-\mu) \leq \lambda(\mathcal{M}\mathcal{P}_M^{-1}) \leq \beta_2$, with μ satisfying the requirement of (i).

We next provide general and insightful bounds for the eigenvalues of the preconditioned matrix resulting from using the preconditioner $\mathcal{P}_M$ with spectrally equivalent matrices K and D. Our starting point is that the "one-sided" preconditioner can be written as

$$\mathcal{M}P_M^{-1} = \mathcal{M}P_f^{-1} \begin{bmatrix} I & 0 \\ A_i K^{-1} & I \end{bmatrix}. \tag{4.6}$$

LEMMA 4.3. *For any symmetric and positive semi-definite K and D we let $P_f = R^T R$, $K = R_K^T R_K$ and $D = R_D^T R_D$ be the Cholesky decompositions of P_f, K and D, respectively. Setting $\mathcal{H} = R^{-T}\mathcal{M}R^{-1}$ it holds*

$$R^{-T}\mathcal{M}P_M^{-1}R^T = \mathcal{H} + \mathcal{H}\begin{bmatrix} 0 & 0 \\ R_D^{-T}A_i R_K^{-1} & 0 \end{bmatrix}. \tag{4.7}$$

Proof. The equality follows from multiplying (4.6) by R^{-T} and R^T from the left and right, respectively. ∎

Lemma 4.3 shows a tight connection between the preconditioned matrix $\mathcal{M}P_f^{-1}$ with P_f symmetric and positive semidefinite, and the fully nonsymmetric preconditioned matrix $\mathcal{M}P_M^{-1}$. Keeping in mind that similarity transformations with R do not modify the spectrum, (4.7) reveals that $R^{-T}\mathcal{M}P_M^{-1}R^T$ is a modification of the symmetric matrix $\mathcal{H}$, therefore it is possible to bound its eigenvalues by means of the real eigenvalues of $\mathcal{H}$ or equivalently, of $\mathcal{M}P_f^{-1}$. In the following, $\Im(X) = (X - X^T)/(2\imath)$ is the skew-Hermitian part of a square matrix X.

THEOREM 4.4. *Assume that K and D satisfy (4.2) and (4.3), respectively, for some positive constants α_j, β_j. Let λ_M be an eigenvalue of $\mathcal{M}P_M^{-1}$. Then there exists an eigenvalue λ of $\mathcal{M}P_f^{-1}$ such that*

$$|\lambda - \lambda_M| \le \lambda_{\max}(\mathcal{M}P_f^{-1})\,\|R_D^{-T}A_i R_K^{-1}\|, \qquad \textit{with} \quad \|R_D^{-T}A_i R_K^{-1}\| < \sqrt{\frac{\alpha_1}{\beta_1}},$$

and

$$|\Im(\lambda_M)| \le \left\| \Im\left(\mathcal{H}\begin{bmatrix} 0 & 0 \\ R_D^{-T}A_i R_K^{-1} & 0 \end{bmatrix}\right)\right\|.$$

Proof. Using Lemma 4.3, the result is an application of Kahan's theorem [20]; see also [50, sec. IV.5.1]. We are only left to show the bound for $\|R_D^{-T}A_i R_K^{-1}\|$. We have $\|R_D^{-T}A_i R_K^{-1}\|^2 = \lambda_{\max}((R_D^{-T}A_i R_K^{-1})^T R_D^{-T}A_i R_K^{-1})$. Therefore, for any vector $\boldsymbol{x} \neq 0$,

$$\begin{aligned}
\frac{\boldsymbol{x}^T R_K^{-T}A_i D^{-1}A_i R_K^{-1}\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}} &= \frac{\boldsymbol{y}^T A_i D^{-1}A_i \boldsymbol{y}}{\boldsymbol{y}^T K \boldsymbol{y}} = \frac{\boldsymbol{z}^T A_i^{\frac{1}{2}} D^{-1} A_i^{\frac{1}{2}}\boldsymbol{z}}{\boldsymbol{z}^T A_i^{-\frac{1}{2}} K A_i^{-\frac{1}{2}}\boldsymbol{z}} \\
&= \frac{\boldsymbol{z}^T A_i^{\frac{1}{2}} D^{-1} A_i^{\frac{1}{2}}\boldsymbol{z}}{\boldsymbol{z}^T\boldsymbol{z}} \frac{\boldsymbol{z}^T\boldsymbol{z}}{\boldsymbol{z}^T A_i^{-\frac{1}{2}} K A_i^{-\frac{1}{2}}\boldsymbol{z}} \\
&\le \frac{1}{\beta_1}\frac{\boldsymbol{z}^T A_i^{\frac{1}{2}}(A_i + A_e)^{-1}A_i^{\frac{1}{2}}\boldsymbol{z}}{\boldsymbol{z}^T\boldsymbol{z}}\alpha_1 \frac{\boldsymbol{z}^T\boldsymbol{z}}{\boldsymbol{z}^T A_i^{-\frac{1}{2}}(A_i + C_t)A_i^{-\frac{1}{2}}\boldsymbol{z}} < \frac{\alpha_1}{\beta_1},
\end{aligned}$$

where we used $\boldsymbol{y} = R_K^{-1}\boldsymbol{x}$ and $\boldsymbol{z} = A_i^{\frac{1}{2}}\boldsymbol{y}$, while the last inequality follows from noticing that $A_i^{\frac{1}{2}}(A_i + A_e)^{-1}A_i^{\frac{1}{2}} \le 1$ and $A_i^{-\frac{1}{2}}(A_i + C_t)A_i^{-\frac{1}{2}} > 1$. ∎

n	22,782	52,131	172,785	333,373
norm	0.2336	0.2512	0.4250	0.4431

TABLE 4.2
Estimated values of $\|R_D^{-T}A_i R_K^{-1}\|$ for various refinements.

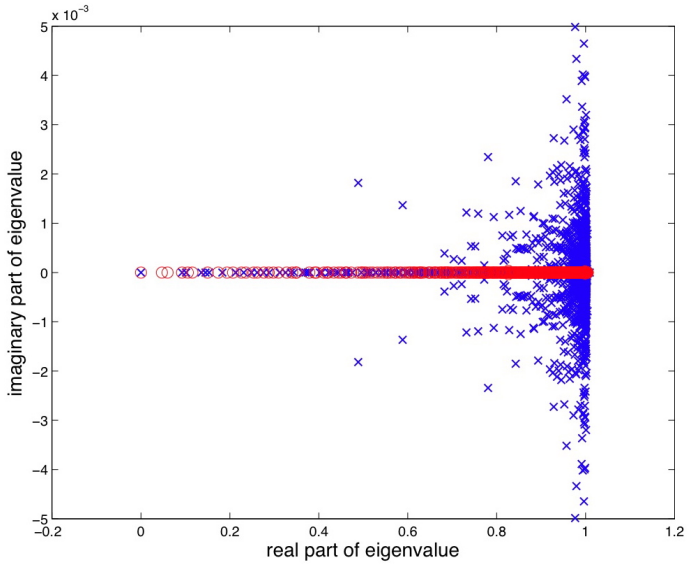

FIG. 4.1. *Eigenvalues of* $\mathcal{M}P_f^{-1}$ *('∘') and of* $\mathcal{M}P_M^{-1}$ *('×'). The imaginary scale is* $[-5\,10^{-3}, 5\,10^{-3}]$.

We remark that if the constants α_1, β_1 are independent of h, then $\|R_D^{-T} A_i R_K^{-1}\|$ is bounded independently of h. Since all eigenvalues of $\mathcal{M}\mathcal{P}_f^{-1}$ are bounded independently of h, the result of Theorem 4.4 combined with that of Theorem 4.2 implies that the spectrum of $\mathcal{M}\mathcal{P}_M^{-1}$ is also bounded independently of h. The numbers in Table 4.2 report an estimate of the norm in Theorem 4.4 for various discretizations of the (v, u_e) formulation. For a much coarser discretization ($n = 1676$) we were able to explicitly compute the spectra of both $\mathcal{M}\mathcal{P}_f^{-1}$ and $\mathcal{M}\mathcal{P}_M^{-1}$, and these are reported in Figure 4.1. Notice that the spectrum of $\mathcal{M}\mathcal{P}_M^{-1}$ ($\times$ symbol) remains very close to that of the symmetric counterpart, with an imaginary part that is significantly lower than even $\lambda_{\max}(\mathcal{M}P_f^{-1})\,\|R_D^{-T} A_i R_K^{-1}\| \approx 8 \cdot 10^{-2}$.

We conclude this section with an important consideration on the actual expected convergence behavior for a residual minimizing method such as GMRES. It is known that the eigenvalues alone may not give a complete picture of the actual behavior of the iterative solver in the nonsymmetric case; information on the associated eigenvectors or invariant subspaces would provide a more reliable picture. In [35] explicit eigenvector bases were given for the ideal and quasi-ideal cases, which could be used for bounding the residual norm of GMRES [42]. In the much more general case analyzed here an explicit expression of the eigenvectors appears cumbersome. A perturbation analysis would allow us to present a more sophisticated though very technical analysis, which is however far beyond the aim of this paper. Nonetheless, we found the eigenvalue distribution to satisfactory adhere to the actual convergence history of the methods, suggesting that the eigenvectors do not play a significant role.

5. Implementation considerations. If the positive definite symmetric preconditioners $\mathcal{P}_d$ or $\mathcal{P}_f$ were employed, the Preconditioned Conjugate Gradient (PCG) algorithm would be the method of choice; this was extensively used, for instance, in

the 2D case in [34]. In there, however, we also experimentally observed that $\mathcal{P}_M$ is more effective. Therefore, here we opt for this block triangular preconditioner, although it requires the use of a nonsymmetric solver. In this section we describe some important aspects associated with the use of iterative methods and $\mathcal{P}_M$, for various choices of its diagonal blocks.

All block preconditioners mentioned in previous sections require solves with K and D, once the selection of appropriate matrices or operators is made. In the case of $\boldsymbol{y} = \mathcal{P}_M^{-1}\mathbf{v}$, $\boldsymbol{y}^T = [\boldsymbol{y}_1^T, \boldsymbol{y}_2^T]$ for instance, this entails performing the following two steps:

$\boldsymbol{y}_2 = D^{-1}\mathbf{v}_2$

$\boldsymbol{y}_1 = K^{-1}(\mathbf{v}_1 - A_i\boldsymbol{y}_2)$

We recall that $\mathcal{P}_M$ is a one-sided version of $\mathcal{P}_f$, and thus it avoids the extra solve with K one would have at each application of $\mathcal{P}_f$, lowering the computational cost per iteration. Therefore, given that using $\mathcal{P}_M$ is cheaper than using $\mathcal{P}_f$, the overall effectiveness of the nonsymmetric approach fully relies on the total cost of the employed nonsymmetric solver. Unless the number of iterations is very low, the optimal GMRES method ([43]) may be expensive, with a cost that grows nonlinearly with the number of iterations. On the other hand, due to the very favorable spectral properties of our preconditioned problem (cf. Theorem 4.4), we found the much cheaper BICGSTAB method to perform very satisfactorily [54], with a cost per iteration comparable to that of PCG.

As for the choice of matrices K and D, we employ an algebraic multigrid preconditioner for K: a preliminary call is made to set up the preconditioner before the actual iterative solution has started; then a call to an AMG routine at each application of the preconditioner is performed. As of D, we found that a simple Jacobi preconditioner, that is $D = \mathrm{diag}(C_t + A_i)$, was sufficient to ensure mesh independence. We experimentally observed that the constants α_i, β_i in (4.2), (4.3) were mesh independent, so that we expect these choices to provide optimal performance of a method like GMRES. In fact, the performance of BICGSTAB did not significantly deviate from that of GMRES, yielding much lower timings. Not unexpectedly, other choices for K and D such as Incomplete Cholesky factorizations did not provide us with mesh independent performance, and in fact their performance significantly degraded with the problem size, therefore these options were soon abandoned.

5.1. Inner-outer procedure. In this section we discuss a different strategy for approximating the ideal preconditioner $\mathcal{P}_{M,ideal}$ that will be compared in section 6 with our approach to solve for the (v, u_e) formulation. An alternative to Multigrid, Multilevel or Incomplete Decompositions consists of approximating the ideal choices $K = A_i + A_e$ and $D = C_t + A_i$ by means of a preconditioned iterative solver. This is the approach taken in [14], giving rise to a so called inner-outer procedure: at each iteration of the (outer) solver, two (inner) iterative solves, one with $A_i + A_e$ and one with $C_t + A_i$ are performed. The accuracy at which the inner solvers can be stopped is in general a major concern. In [14] the authors state that a very loose stopping tolerance can be used for both inner preconditioned iterations, without significantly influencing the performance of the outer solver. This consideration is particularly crucial for systems with $A_i + A_e$, whose accurate solution would require very many PCG iterations without heavy preconditioning. Instead, solving with $C_t + A_i$ is extremely cheap and the authors report 1-2 inner iterations with it; this performance confirms that replacing $C_t + A_i$ with its diagonal (Jacobi preconditioning) would be equally effective.

The inner-outer strategy requires tuning an extra parameter, the inner stopping tolerance. In addition, since the operator that approximates, say, $(A_i + A_e)^{-1}$ changes at each iteration, a flexible outer algorithm needs to be employed to ensure that the unpreconditioned solution be correctly recovered. The authors in [14] opt for the flexible version of the residual minimizing GMRES method, here and later denoted by FGMRES [41]. As already mentioned, the computational costs of this method grow superlinearly with the number of iterations while the memory allocations grow linearly with the number of iterations; in addition, the flexible variant requires storing twice the number of long vectors of GMRES [41].

The use of an inner-outer procedure for this special problem necessitates of a careful analysis of the final solution accuracy, which seems to be influenced by the interplay between inner and outer stopping tolerances. In particular, since a stricter inner tolerance may lead to a more expensive overall method, one is usually tempted to employ looser values for the inner tolerance. We proceed with an experimental analysis of the error norm; for the description of the employed data we refer to section 6. The exact solution to the given problem was not available, therefore here and in the following we use a reference solution obtained with the iterative solver by using a very small stopping tolerance (tol=10^{-12}); we will denote this solution by u^{ref}. The solution uniqueness is ensured by the deflation of the null space vector.

The stopping criterion was based on the relative residual norm; experiments with the backward error yielded very similar results. Here and in the sequel, we employ the relative error Euclidean norm, which is commonly restricted to the extracellular potential u_e, namely

$$\text{error} = \frac{\|u_e^{ref} - u_e\|}{\|u_e^{ref}\|}, \tag{5.1}$$

to measure the accuracy of the obtained solution for a-priori selected inner and outer tolerances (data corresponding to 10 msec after stimulation). The computation time to reach an equivalent level of accuracy for each method is used as a level of efficiency.

In Table 5.1 we report the error norms for different levels of mesh refinement. We used FGMRES as outer solver, and PCG+ILU(0) as inner solvers, as proposed in [14]. We refer to section 6 for additional information on the implementation. The digits reported in Table 5.1 show that in general the final accuracy depends on the outer tolerance, as expected. However, there are cases where a smaller inner tolerance yields a more accurate outer solution: compare, e.g., the choices tols=$[10^{-5}, 0.12]$ (the numbers denote outer and inner tolerances, respectively) and tols=$[10^{-5}, 10^{-2}]$ for $n = 52\,132, 172\,785, 333\,373$. Moreover, requiring a more accurate inner solution may allow for a looser outer tolerance, compare, e.g., tols=$[10^{-8}, 0.12]$ and tols=$[10^{-7}, 10^{-2}]$ with $n \leq 780\,333$, for comparable timings. Overall, it is clear that some trial-and-error procedure is required whenever an inner-outer method is chosen for this application problem. Further experiments are reported in the next section.

6. Numerical results. Computational experiments were carried out to compare the different formulations and solvers introduced in the previous sections. The considered domain Ω is a truncated ellipsoidal region representing the left ventricle, see the left top plot of Figure 6.1. The domain Ω was modeled and discretized using COMSOL [12].

The propagation was elicited by applying a current pulse of $1.A/cm^3$ lasting 1 msec; hence a value of $I_{app} = 1$ is applied to each grid node of the stimulated region. We considered the same parameter calibration used in [8] (cf. Table 6.1).

TABLE 5.1
Inner-outer solver. outer: FGMRES*; inner:* PCG+ILU(0)*.* $tols = [tol, tol_{inner}]$

	$tols = [10^{-5}, 0.12]$			$tols = [10^{-7}, 0.12]$			$tols = [10^{-8}, 0.12]$		
n	Time	It	error	Time	it	error	Time	it	error
22,782	0.15	5	9.00e-3	0.32	7	7.88e-5	0.32	7	7.88e-5
52,132	0.36	4	7.35e-2	0.97	6	6.11e-4	1.61	8	4.77e-6
172,785	1.62	4	7.71e-2	5.95	7	1.39e-4	7.66	8	1.14e-5
333,373	8.04	5	9.53e-3	15.96	7	9.09e-5	18.96	8	2.57e-5
780,333	23.04	5	2.21e-2	60.92	8	4.75e-5	71.93	9	8.80e-5
1,346,895	49.29	5	1.74e-2	121.85	8	1.02e-4	152.74	9	7.00e-6

	$tols = [10^{-5}, 10^{-2}]$			$tols = [10^{-7}, 10^{-2}]$		
n	Time	It	error	Time	It	error
22,782	0.14	3	5.94e-3	0.29	4	9.03e-5
52,132	0.56	3	7.27e-3	1.08	4	1.62e-4
172,785	2.67	3	1.67e-2	8.19	5	1.85e-5
333,373	12.02	4	1.67e-3	18.75	5	7.88e-5
780,333	35.65	4	5.38e-3	72.80	6	7.51e-5
1,346,895	66.58	4	9.22e-3	139.27	6	7.23e-5

Meshes were built on Ω by using the mesh generator of COMSOL that creates unstructured meshes made up of tetrahedral mesh elements. We fixed a starting mesh and built the subsequent meshes by decreasing the mesh diameter. The number of considered mesh nodes is $2n$ where $n \in \{22\,782, 52\,132, 172\,785, 333\,373, 780\,333, 1\,346\,895\}$, with corresponding mesh sizes $h \in \{0.2, 0.15, 0.1, 0.08, 0.06, 0.05\}$, whereas the time step τ was chosen to be equal to $5 \cdot 10^{-2}$ msec. For these choices of time and space steps it is possible to obtain stable and accurate results as shown by the validation carried out in [8]. In Figure 6.1 proceeding clockwise from the left top we display the coarsest mesh (to allow for a good visualization of the mesh) and typical extracellular potential maps (on a finer mesh) on the boundary of the domain at 10, 30 and 50 msec after stimulation. To show the propagation through the cardiac wall, in Figure 6.2 we plot extracellular potential maps on a two dimensional section of the ellipsoid for the same time instants, i.e. at 10, 30 and 50 msec after stimulation.

$\chi = 1000$ cm^{-1}	$G = 1.5 \times 10^{-3}$ Ω^{-1} cm^{-2}	$c_m = 1.\mu F$ cm^{-2}
$i_{app} = 1.$ A cm^{-3}	$v_p = 100$ mV	$v_{th} = 13$ mV
$\sigma_l^e = 2. \times 10^{-3}$,	$\sigma_t^e = 1.3514 \times 10^{-3}$	Ω^{-1} cm^{-1}
$\sigma_l^i = 3. \times 10^{-3}$,	$\sigma_t^i = 3.1525 \times 10^{-4}$	Ω^{-1} cm^{-1}

TABLE 6.1
Parameter calibration.

Most reported experiments below correspond to a typical temporal instant in the time step evolution (10 msec after stimulation), thus the right–hand side includes information generated during the previous time steps.

All computations were performed in FORTRAN 90 on a 8GB ram, AMD Phenom(tm) II X4 955 Quad-Core 64 bit at 3.2 GHz with 2MB L2 cache.

The stopping tolerance for all methods was chosen a posteriori so as to obtain a comparable error norm of 10^{-5}, as defined in (5.1). This strategy allowed us to make fair comparisons among methods that employ different stopping criteria. The CPU

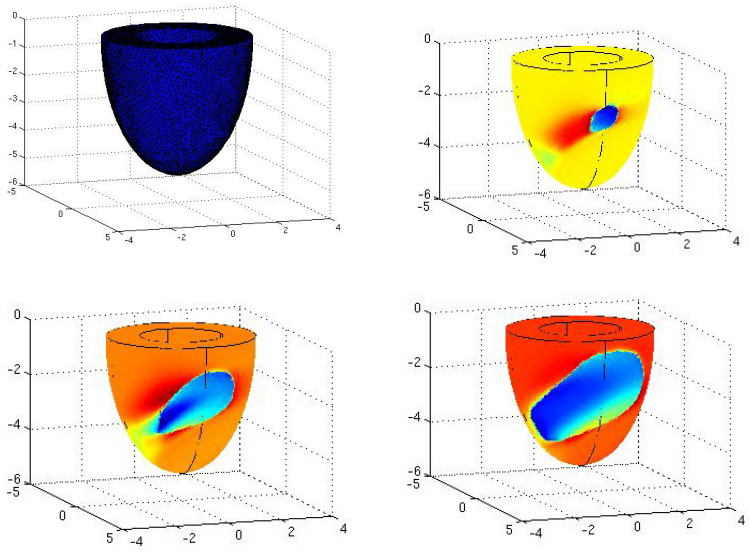

FIG. 6.1. *Truncated ellipsoid modeling the left ventricle: starting mesh (top-left) and extracellular potential map at 10, 30 and 50 msec after stimulation.*

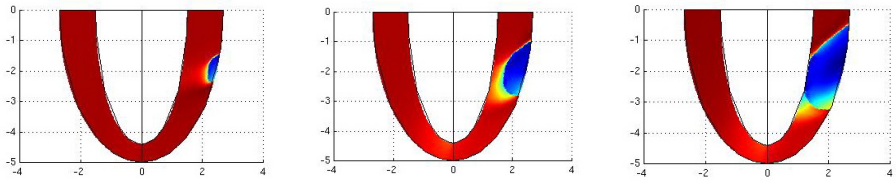

FIG. 6.2. *Extracellular potential map on a vertical section of the ellipsoid at 10, 30 and 50 msec after stimulation.*

time to reach the required level of accuracy is used as an efficiency measure.

Except where explicitly stated, we used the Aggregation-based Algebraic Multigrid (AGMG) Method of Y. Notay ([27]) as preconditioner in all instances when an Algebraic Multigrid method is used; we refer to [28] for the description of the method and its general performance. Default values of the method were used in all instances.

Formulation (u_i, u_e). The Conjugate Gradient (CG) algorithm was used to solve the large linear system (2.8). The singularity of $\mathcal{M}$ is not harmful for CG, as long as the system is consistent. The CG algorithm available within the AGMG code was used. As already mentioned, for this formulation an AGMG preconditioner on the whole matrix, hereafter $\mathcal{P}_{whole}$, appeared to be by far the best performing approach among the choices we have tested. This behavior fully confirms our findings in the 2D case, see, e.g., [34, 35], and thus we omit reporting results with other preconditioners. We emphasize that the good performance of this simple unstructured approach is apparently strongly related to the particular sparsity structure of the blocks of $\mathcal{M}$ in this formulation (cf. (2.8)). In particular, the presence of negative off-diagonal elements, due to the matrix $-C_t$, and the larger weight of the diagonal entries, seem to make the coarsening procedure highly effective, compared to the (v, u_e) case, where

the nondiagonal block A_i includes positive entries (cf. also Table 6.5). The CPU times, number of iterations and associated error norms are displayed in Table 6.2. We start by emphasizing the very low CPU time associated with the solution of the finest discretization, which involves a matrix $\mathcal{M}$ of size $2n = 2,693,790$, envisioning the possibility to simulate the whole excitation process in a very affordable time frame. We also notice that the number of iterations *decreases* as mesh refinement takes place, providing a favorable "mesh-dependent" performance. We also remark that a substantial mesh independence was also observed in the original AGMG paper for other 3D experiments [28]. Finally, we observe that the true relative error norm also decreases with the mesh refinement.

n \ $\mathcal{P}_{whole}$	tol= 10^{-7}			tol= 10^{-8}		
	Time	Iter	err	Time	Iter	err
22,782	0.83	34	3.25e-4	1.27	52	3.68e-5
52,132	1.55	26	1.53e-4	2.55	42	1.93e-5
172,785	5.51	24	1.47e-4	8.24	35	1.18e-5
333,373	10.30	22	6.48e-5	14.52	30	9.14e-6
780,333	25.52	20	3.64e-5	33.88	26	5.09e-6
1,346,895	38.49	17	1.57e-5	51.02	22	2.29e-6

TABLE 6.2

(u_i, u_e) formulation. CPU time and number of CG iterations to obtain a reduction of the residual norm by a factor of 10^{-7} and 10^{-8} for P_{whole} at 10 msec after stimulation.

To increase our understanding of the preconditioner performance, we provide estimates for the associated spectral interval, as the problem size increases. The table below reports estimates for the parameters χ_1, χ_2 in the inequalities $\chi_1\mathcal{M} \leq \mathcal{P}_{whole} \leq \chi_2\mathcal{M}$.

n	χ_1	χ_2
22,782	0.9368	169.52
52,132	0.9377	114.01
172,785	0.9479	87.18
333,373	0.9634	85.14

We use the following well known estimate for the error $\mathcal{B}$-norm after j CG iterations

$$\|\boldsymbol{\xi}_j - \boldsymbol{\xi}\|_{\mathcal{B}} \leq \left(\frac{\sqrt{\text{cond}(\mathcal{B})} - 1}{\sqrt{\text{cond}(\mathcal{B})} + 1} \right)^j , \tag{6.1}$$

where $\text{cond}(\mathcal{B}) = \lambda_{\max}(\mathcal{B})/\lambda_{\min}(\mathcal{B})$, in which $\lambda_{\max}(\mathcal{B})$, $\lambda_{\min}(\mathcal{B})$ are the largest and smallest nonzero eigenvalues of $\mathcal{B}$. For $\mathcal{B} = \mathcal{P}_{whole}^{-1}\mathcal{M}$, we can estimate $\text{cond}(\mathcal{B})$ as $\text{cond}(\mathcal{B}) \approx \chi_2/\chi_1$ to obtain that the error in the energy norm must be below 10^{-5} after at most $j = 73$ iterations on the coarsest grid. Faster convergence is observed in practice.

For the sake of completeness in Table 6.3 we report some comparisons with a different Algebraic Multigrid preconditioner, namely the AMG code available in the HSL library, the HSL_MI20 routine [4]. This function implements the classical (Ruge-Stüben) AMG method, as described in [40]. All default preconditioning parameters were used, except for `one_pass_coarsen=.true., st_method=1, testing=0.`

	MI20				AGMG			
n	Setup	Time	Iter	err	Setup	Time	Iter	err
22,782	0.16	2.47	20	5.29e-5	0.06	0.83	34	3.25e-4
52,132	0.42	4.32	20	1.55e-5	0.15	1.55	26	1.53e-4
172,785	1.48	15.04	25	2.36e-5	0.56	5.51	24	1.47e-4
333,373	2.99	29.58	26	1.59e-5	1.13	10.30	22	6.48e-5
780,333	7.43	81.86	31	1.19e-5	2.77	25.52	20	3.64e-5
1,346,895	13.82	134.97	29	1.57e-5	4.96	38.49	17	1.57e-5

TABLE 6.3
$P_{whole}(u_i, u_e)$ at 10 msec after stimulation with MI20 *tol=1.e-8 and* AGMG *tol=1.e-7.*

	$P_{whole}(u_i, u_e)$ tol= 10^{-7}			$\mathcal{P}_M$ AGMG BICGSTAB tol= 10^{-8}			$\mathcal{P}_M$ ILU(0) FGMRES tols= $[10^{-8}, 0.12]$		
n	Time	It	error	Time	It	error	Time	It	error
22,782	0.83	34	3.25e-4	0.38	13	5.56e-4	0.32	7	7.88e-5
52,132	1.55	26	1.53e-4	1.45	17	1.61e-4	1.61	8	4.77e-6
172,785	5.51	24	1.47e-4	6.69	20	6.01e-5	7.66	8	1.14e-5
333,373	10.30	22	6.48e-5	11.25	17	5.24e-5	18.96	8	2.57e-5
780,333	25.52	20	3.64e-5	41.72	24	9.93e-5	71.93	9	8.80e-5
1,346,895	38.49	17	1.57e-5	67.56	22	2.09e-5	152.74	9	7.00e-5

TABLE 6.4
Performance evaluation for $P_{whole}(u_i, u_e)$, $\mathcal{P}_M$ BICGSTAB *and $\mathcal{P}_M$* FGMRES*. CPU time, number of iterations and final error norms.*

The preconditioner was employed within the HSL_MI21 routine implementing the CG method [19]. The table reports the preconditioner setup time, the total elapsed time for the CG relative residual norm to pass the stopping tolerance, the number of CG iterations and the corresponding relative error norm (cf. (5.1)). Stopping tolerances were tuned so as to obtain similar final error norms for the finer grid. To ease the numerical comparison the table also repeats some of the numbers from Table 6.2. For all mesh discretizations, results are largely in favor of the AGMG package of Notay [27]. As an example, for $n = 780,333$ and the same level of accuracy of the computed solution, setup time and time per iterations of HSL_MI20 are about three times greater than those required by AGMG. These results are consistent with those obtained in [28] for 3D problems. In light of these results and of similar ones obtained with the other formulation, in the sequel we only report on the numerical performance of AGMG and not of MI20.

Formulation (v, u_e). We report on solving system (2.9) with the non-symmetric preconditioner $\mathcal{P}_M$. Numerical experiments not reported here showed that $\mathcal{P}_M$ is more effective than $\mathcal{P}_f$ and $\mathcal{P}_d$ for the same choices of blocks D and K (cf. section 3). As already mentioned we used BICGSTAB as solver (routine HSL_MI26, see [19]), the diagonal of $A_i + C_t$ as matrix K, and the multigrid preconditioner AGMG as D. CPU times (in seconds), number of BICGSTAB iterations and relative error norm, to obtain a reduction of the residual norm by a factor of 10^{-8} and an accuracy of order 10^{-5} are reported in the second set of columns of Table 6.4.

For the sake of comparison, in the right portion of columns of the same table we also report the performance of an inner-outer method (cf. section 5.1), similar to that suggested in [14]. The algorithm was implemented as follows: the outer method is FGMRES (routine HSL_MI15), the inner method is CG (routine HSL_MI21), and the preconditioner for both inner solvers is ILU(0) (routine HSL_MI11), all from the HSL package [19]. The inner and outer tolerances were selected so as to obtain the best overall performance for a comparable error norm (cf. Table 5.1). We can see that the computational cost for solving with the inner-outer method is, on the finest grid, more than twice that of the solver with the structured AMG preconditioner, while requiring the tuning of some extra parameter. These experiments suggest that the structured Algebraic Multigrid is superior and more reliable than the presented inner-outer procedure, at least for the proposed setting. To help the reader in an overall comparison of the two formulations, we have included (once again) in Table 6.4 the digits associated with the performance of $\mathcal{P}_{whole}$ in the (u_e, u_i) formulation from Table 6.2. We readily observe that for a comparable final accuracy, the (u_i, u_e) formulation provides the fastest solution. Finally, we observe that all approaches are essentially mesh independent, as the number of iterations does not appreciably increase with the mesh refinement.

We remark that AMG on the whole matrix (preconditioner $\mathcal{P}_{whole}$) is not competitive on this formulation: the number of iterations significantly increases with the problem size, leading to excessive computational costs; a typical performance is reported in Table 6.5. This feature can be related to the particular structure of the matrix in the (v, u_e) formulation, characterized by many rows with non-negative off-diagonal elements associated with the (1,2) and (2,1) blocks (cf. (2.9)).

n \ $\mathcal{P}_{whole}$	tol= 10^{-7}		
	Time	Iter	err
22,782	0.23	6	3.77e-4
52,132	1.34	14	7.30e-4
172,785	11.48	28	9.94e-4
333,373	45.89	54	8.96e-4
780,333	207.27	100	5.25e-4
1,346,895	438.40	120	5.01e-4

TABLE 6.5

(v, u_e) formulation. CPU time and number of CG iterations to obtain a reduction of the residual norm by a factor of 10^{-7} for P_{whole} at 10 msec after stimulation.

To complete the set of comparison tests, we also analyze the setup costs and memory requirements for the leading approaches. The setup time for generating the preconditioner is often a bottleneck of multilevel methods. Fortunately, AGMG turns out to be very effective, compared to other multigrid methods, as reported in Table 6.6; see also the relevant discussion in [28]. A closer look at the table shows that the setup time is not a discriminant among the various methods. Memory requirements seem to differ more remarkably. In Table 6.7 we report the major memory requirements of the used preconditioners: for AGMG this is the allocation reported by the code, while for ILU(0) the computation is performed on the basis of the nonzero elements of the involved matrices. We also recall that FGMRES also requires a number of long vectors

n	P_{whole} (u_i, u_e)	$\mathcal{P}_M$ AGMG BICGSTAB	$\mathcal{P}_M$ ILU(0) FGMRES
22,782	0.06	0.03	0.024+ 0.02
52,132	0.15	0.08	0.056+0.055
172,785	0.56	0.34	0.19+0.19
333,373	1.13	0.69	0.37+0.37
780,333	2.77	1.70	0.90+0.90
1,346,895	4.96	3.00	1.56+1.55

TABLE 6.6

Preconditioning setup time of $P_{whole}(u_i, u_e)$ and of $\mathcal{P}_M$ AGMG and $\mathcal{P}_M$ ILU(0) for the (v, u_e) formulation.

n	P_{whole} (u_i, u_e)	$\mathcal{P}_M$ AGMG BICGSTAB	$\mathcal{P}_M$ ILU(0) FGMRES
22,782	6.25	3.50	2×2.54
52,132	14	6.21	2×5.94
172,785	47.55	20.14	2×19.98
333,373	90.76	39.55	2×38.84
780,333	214.20	91.77	2×91.59
1,346,895	368.79	159.32	2×158.48

TABLE 6.7

Memory requirements (in Mbytes) for the preconditioners all both formulations and strategies.

that is at least twice as the number of performed iterations. As expected, the AGMG preconditioner on the whole matrix (in the (u_i, u_e) formulation) requires the largest amount of memory, followed by that of FGMRES, if inner ILU(0) preconditioning is performed on both blocks K and D, as suggested in [14]. Table 6.7 shows that AGMG preconditioning requires roughly the same amount of storage as an ILU(0) preconditioner, but with a much higher performance.

6.1. Overall computational costs in typical simulations. To assess the performance of the various approaches in a more realistic setting, we next report on the results of a longer simulation of the myocardium excitation process. In Tables 6.6-6.8 we display the setup times and the total execution times for a simulation of $50\,msec$, i.e. 1000 time steps, with both formulations. In each case, we only show the best performing setting.

The AGMG based preconditioner with BICGSTAB is more efficient than $\mathcal{P}_M$ FGMRES, for all considered discretizations. It is interesting that the performance of all methods is somewhat sensible to the time instant, as the system right-hand side changes significantly throughout the simulation. Indeed, the average time is lower than that obtained for the single time step chosen in previous experiments.

For $n = 1,346,895$ and the (v, u_e) formulation, we observe a CPU time reduction of 55% of $\mathcal{P}_M$ AGMG BICGSTAB when compared with $\mathcal{P}_M$ FGMRES. However, the fastest time is obtained for $\mathcal{P}_{whole}$ with formulation (u_i, u_e), which provides a further 20% CPU time reduction. As already verified, the setup phase time is irrelevant for all methods, compared to the overall cost of the computation.

	$\mathcal{P}_{whole}$ (u_i, u_e)		$\mathcal{P}_M$ AGMG BICGSTAB		$\mathcal{P}_M$ ILU(0) FGMRES	
n	Time	Iter	Time	Iter	Time	Iter
22,782	0.52	20.94	0.39	13.73	0.35	7.12
52,132	1.60	26.36	1.34	16.06	1.44	7.35
172,785	5.16	22.66	5.83	17.60	8.04	8.00
333,373	9.31	19.38	12.70	18.77	20.15	8.01
780,333	22.53	17.66	40.82	23.70	70.27	8.57
1,346,895	35.51	15.47	60.91	19.52	145.54	9.02

TABLE 6.8

Averaged execution time (in sec) per time step and average iteration count per time step for 50 msec (i.e. 1000 time steps) for P_{whole} and the (u_i, u_e) formulation and $\mathcal{P}_M$ AGMG BICGSTAB and $\mathcal{P}_M$ FGMRES for the (v, u_e) formulation .

6.2. Comparisons with an alternative formulation. An alternative formulation was recently proposed in [14]. If we linearly combine the two equations of (2.4) with coefficients $\frac{\lambda}{1+\lambda}$ and $\frac{-1}{1+\lambda}$, $\lambda > 0$, we obtain the following parameter-dependent non-symmetric formulation of the bidomain system:

find $(v(\boldsymbol{x}, t), u_e(\boldsymbol{x}, t))$, $\boldsymbol{x} \in \Omega$, $t \in [0, T]$ *such that*

$$\begin{cases} c_m \partial_t v - \operatorname{div} \frac{\lambda}{1+\lambda} M_i \nabla v - \operatorname{div}\Big(\frac{\lambda M_i - M_e}{1+\lambda}\Big) \nabla u_e + I_{ion}(v, w) = I_{app} & \text{in } \Omega_T \\ -\operatorname{div}\, M \nabla u_e = \operatorname{div}\, M_i \nabla v & \text{in } \Omega_T \\ \mathbf{n}^T M_i \nabla v = 0, \qquad \mathbf{n}^T M \nabla u_e = 0 & \text{on } \Gamma_T \\ v(\boldsymbol{x}, 0) = 0 & \text{in } \Omega. \end{cases} \tag{6.2}$$

Discretization leads to a linear system with coefficient matrix

$$\mathcal{M} = \begin{bmatrix} C_t + \frac{\lambda}{1+\lambda} A_i & \frac{\lambda}{1+\lambda} A_i - \frac{1}{1+\lambda} A_e \\ A_i & A_i + A_e \end{bmatrix}.$$

This formulation was introduced in [14] to build a non-symmetric preconditioner similar to the one studied here and in [35] for the (v, u_e) formulation.

Table 6.9 reports the number of iterations for two sets of tolerances required by the preconditioner $\mathcal{P}_M$ proposed in [14] applied to the (v, u_e) formulation (2.9), and to the $(\lambda v, u_e)$ formulation (6.2), with two different values of λ as suggested in [15]. We considered the same preconditioner proposed in [14, 15], i.e. a lower triangular preconditioner instead of the upper triangular[1] one chosen in this paper. The preconditioner is implemented by means of an inner-outer method for the diagonal blocks, as described in earlier sections. Since the cost per outer iteration is the same for all methods in the table, the number of iterations represents a good measure of the relative performance.

From the reported digits we can see that the number of iterations even slightly increases when the alternative $(\lambda v, u_e)$ formulation in considered, for both values of λ. Since the changes are not really in favor of this parameter-dependent formulation, these experiments strongly suggest that the classical (v, u_e) formulation should be preferred, not last the consideration that no extra tuning of parameters is required.

[1] Note that in this context, the performance of the upper and lower triangular preconditioners is expected to be very similar.

	$tols = [10^{-5}, 0.12]$			$tols = [10^{-8}, 0.12]$		
n	$\mathcal{P}_M$ (v, u_e)	$\mathcal{P}_M$ $(\lambda v, u_e)$ $\lambda = 1.3$	$\mathcal{P}_M$ $(\lambda v, u_e)$ $\lambda = 1.5344$	$\mathcal{P}_M$ (v, u_e)	$\mathcal{P}_M$ $(\lambda v, u_e)$ $\lambda = 1.3$	$\mathcal{P}_M$ $(\lambda v, u_e)$ $\lambda = 1.5344$
22,782	4	5	5	7	8	8
52,132	4	5	5	7	8	8
172,785	5	5	5	8	9	9
333,373	5	6	6	8	10	10
780,333	6	7	7	10	10	10
1,346,895	6	7	7	10	10	10

TABLE 6.9

Number of iterations required by $\mathcal{P}_M$ at 10 msec after stimulation when applied to formulations (v, u_e) and $\lambda(v, u_e)$ for two different values of λ and of tolerances tols: $\lambda = 1.3, 1.5344$, tols $= [10^{-5}, 0.12], [10^{-7}, 10^{-2}]$.

7. Conclusions. We have shown that (nonsymmetric) block triangular preconditioning provides an efficient tool for solving the large symmetric linear system arising at each time-step in the semi-implicit discretization of a popular formulation of the bidomain model. Optimality is obtained by using the recently developed algebraic multigrid method AGMG, which experimentally ensures mesh independence of the solver, in terms of number of iterations. The aggregation-based AGMG method is a building block within our structured preconditioner; alternative - public domain - AMG strategies may lead to even better timings of our preconditioning technique. We showed that alternative inner-outer methods are not competitive and may suffer from parameter tuning. We theoretically justified the good performance of the structured preconditioner by deriving spectral bounds for the preconditioned matrix, which ensure that short-term recurrence system solvers can be efficiently applied. Finally, our best numerical results were obtained with a less popular but equivalent formulation, for which AGMG preconditioning alone is particularly effective. We thus encourage the use of this formulation, from which the numerical solutions of the more exercised formulation can be easily recovered.

Acknowledgments. We wish to thank Jennifer Scott and Jonathan Hogg for their help with the use of several HSL routines. We thank Yvan Notay for sharing with us his computational experience with the matlab and fortran versions of his AGMG software. All reported experiments were carried out on the machines of the SINCEM Laboratory at CIRSA in Ravenna. We are indebted with Luca Giacomelli for technical support.

REFERENCES

[1] T. M. Austin, M. L. Trew, and A. Pullan. Solving the cardiac bidomain equations for discontinuous coefficients. *IEEE Trans. Biomed. Eng.*, 53(7):1265–1272, 2006.

[2] O. Axelsson. *Iterative Solution Methods.* Cambridge University Press, New York, 1994.

[3] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.

[4] J. Boyle, M. D. Mihajlović and J.A. Scott. HSL_MI20: an efficient AMG preconditioner. *RAL Tech.Rep. RAL-TR-2007-021*, 2007.

[5] E. Cherry, H. Greenside, and C. S. Henriquez. Efficient simulation of three-dimensional anisotropic cardiac tissue using an adaptive mesh refinement method. *Chaos*, 3, 2003.

[6] P. Colli Franzone, P. Deuflhard, B. Erdmann, J. Lang, and L. F. Pavarino. Adaptivity in space and time for reaction-diffusion systems in electrocardiology. *SIAM J. Sci. Comput.*, 28(3):942–962, 2006.

[7] P. Colli Franzone, M. Pennacchio, L. Guerri. Accurate computation of electrograms in the left ventricular wall. *Math. Mod. Meth. Appl. Sci.* 10:507-538 (2000)

[8] P. Colli Franzone and L. Guerri. Spreading of excitation in 3-D models of the anisotropic cardiac tissue. I: Validation of the eikonal model. *Math. Biosci.*, 113:145–209, 1993.

[9] P. Colli Franzone, L. Guerri, M. Pennacchio and B. Taccardi. Spread of Excitation in 3-D Models of the Anisotropic Cardiac Tissue. Part II: Effects of Fiber Architecture and Ventricular Geometry *Math. Biosci.*, 147:131-171, 1998.

[10] P. Colli Franzone and L.F. Pavarino. A parallel solver for reaction-diffusion systems in computational electrocardiology. *Math. Mod. Meth. Appl. Sci.*, 14:883–911, 2004.

[11] P. Colli Franzone and G. Savaré. *Evolution equations, Semigroups and Functional Analysis, Progr. Nonlinear Differential Equations Appl.*, volume 50, chapter Degenerate evolution systems modeling the cardiac electric field at micro and macroscopic level., pages 49–78. Birkhauser, Basel, 2002.

[12] *Comsol Multiphysics User's Guide, Version 3.4. 2009.*

[13] I. R Efimov, V. P. Nikolski, and G. Salama. Optical imaging of the heart. *Circ. Res.*, 95(1):21–33, 2004.

[14] L. Gerardo Giorda, L. Mirabella, F. Nobile, M. Perego and A. Veneziani. A model-based block-triangular preconditioner for the Bidomain system in electrocardiology, *J. Comput. Phys.* 228, 3625-3639 (2009)

[15] L. Gerardo Giorda, L. Mirabella. Spectral analysis of a block-triangular preconditioner of the Bidomain system in electrocardiology, *Tech. Report* TR-2010-006- Mathematics and Computer Science, Emory University (2010)

[16] A. C. Guyton and J. C. Hall. *Textbook of Medical Physiology* W. B. Sanders, Philadelphia, 1996.

[17] C. S. Henriquez. Simulating the electrical behavior of cardiac tissue using the bidomain model. *Crit. Rev. Biomed. Engr*, 21(1):1–77, 1993.

[18] N. Hooke, C. S. Henriquez, P. Lanzkron, and D. Rose. Linear algebraic transformations of the bidomain equations: implications for numerical methods. *Math. Biosci.*, 120:127–145, 1994.

[19] HSL, A Collection of Fortran codes for large-scale scientific computation. See http://www.hsl.rl.ac.uk, (2007)

[20] W. Kahan. Spectra of nearly Hermitian matrices. *Proceedings of the American Mathematical Society*, 48:11–17, 1975.

[21] J. Keener and J. Sneyd. *Mathematical Physiology.* Vol. I-II. Second Edition Springer, New York, 2009.

[22] G Lines, P Grøttum, and A Tveito. Modeling the electrical activity of the heart. A bidomain model of the ventricles embedded in a torso. *Computing and Visualization in Science*, 5(4):195–213, 2003.

[23] C. Luo and Y. Rudy. A model of ventricular cardiac action potential: depolarization, repolarization, and their interaction. *Circ. Res.*, 68:1501–1526, 1991.

[24] M. Munteanu and L. F. Pavarino, Decoupled Schwarz algorithms for implicit discretizations of nonlinear Monodomain and Bidomain systems *Math. Mod. Meth. Appl. Sci.*, 19 (7): 1065 - 1097, 2009.

[25] M. Munteanu, L. F. Pavarino, S. Scacchi A scalable Newton-Krylov-Schwarz method for the Bidomain reaction-diffusion system. *SIAM Journal on Scientic Computing,* 31 (5): 3861–3883, 2009.

[26] M. Murillo and X. C. Cai. A fully implicit parallel algorithm for simulating the non-linear electrical activity of the heart. 1 *Numer. Linear Algebr. Appl.*, 11:261–277, 2004.

[27] Y. Notay. Users Guide to AGMG, third Ed, March 2010. `http://homepages.ulb.ac.be/ ynotay/`.

[28] Y. Notay. An aggregation-based algebraic multigrid method. *ETNA*, 37: 123-146, 2010.

[29] L. F. Pavarino and S. Scacchi, Multilevel additive Schwarz preconditioners for the Bidomain reaction-diusion system, *SIAM Journal on Scientic Computing*, 31 (1): 420 - 443, 2008.

[30] M. Pennacchio. The mortar finite element method for the cardiac "bidomain" model of extracellular potential. *J. Sci. Comput.*, 20:191–210, 2004.

[31] M. Pennacchio, G. Savaré, and P. Colli Franzone. Multiscale modeling for the bioelectric activity of the heart. *SIAM J. Math. Anal.*,, 37:1333–1370, 2006.

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

[33] M. Pennacchio and V. Simoncini. Substructuring preconditioners for mortar discretization of a degenerate evolution problem. *J. Sci. Comput.*, 36:391–419, 2008.

[34] M. Pennacchio and V. Simoncini. Algebraic Multigrid Preconditioners for the Bidomain Reaction–Diffusion system. *Appl. Numer. Math.*, vol. 59, n. 12, 3033–3050 (2009)

[35] M. Pennacchio and V. Simoncini Non-symmetric Algebraic Multigrid Preconditioners for the Bidomain Reaction-Diffusion System. Proceedings ENUMATH 2009, to appear.

[36] C.S. Peskin Fiber architecture of the left ventricular wall: an asymptotic analysis *Comm. Pure Appl. Math.*, vol. 62, 79–113 (1989)

[37] G. Plank, M. Liebmann, R. Weber dos Santos, E. J. Vigmond, and G. Haase. Algebraic multigrid preconditioner for the cardiac bidomain model. *IEEE Trans. Biomed. Eng.*, 54:585–596, 2007.

[38] A. J. Pullan, L. K. Cheng and M. L. Buist. *Mathematically Modeling the Electrical Activity of the Heart. From Cell to Body Surface and Back* World Scientific, Singapore, 2005.

[39] W. Quan, S. T. Evans, and H. M. Hastings. Efficient integration of a realistic two–dimensional cardiac tissue model by domain decomposition. *IEEE Trans. Biomed. Eng.*, 45(3):372–385, 1998.

[40] J.W. Ruge and K. Stüben. Algebraic multigrid. In S.F. McCormick, editor, *Multigrid Methods*, v.3 of *Frontiers in Applied Mathematics*, pages 73–130. SIAM, 1987.

[41] Yousef Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing*, 14:461–469, 1993.

[42] Y. Saad. *Iterative methods for sparse linear systems.* The PWS Publishing Company, 1996.

[43] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[44] H. I. Saleheen and K. T. Ng. A new three–dimensional finite difference bidomain formulation for the inhomogeneous anisotropic cardiac tissues. *IEEE Trans. Biomed. Eng.*, 45(1):15–25, 1998.

[45] S. Scacchi. A hybrid multilevel Schwarz method for the Bidomain model. *Comput. Meth. Appl. Mech. Eng.,* 197 (4548): 40514061, 2008.

[46] K. Skouibine and W. Krassowska. Increasing the computational efficiency of a bidomain model of defibrillation using a time–dependent activating function. *Ann. Biomed. Eng.*, 28:772–780, 2000.

[47] B. F. Smith, P. E. Björstad, and W. D. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, 1996.

[48] J. Southern, G. Plank, E.J. Vigmond, J. Whiteley, Solving the coupled system improves computational efficiency of the bidomain equations, *IEEE Trans. Biomed. Eng.* 56(10):2404-12 (2009).

[49] D.D. Streeter, H.M. Spotnitz, D.P. Patel, J. Ross and E.H. Sonnenblick Fiber Orientation in the Canine Left Ventricle during Diastole and Systole *Circ. Res.*, 24(1):339–347 (1969)

[50] G.W. Stewart, J-G. Sun. Matrix Perturbation Theory. Academic Press (1990)

[51] J. Sundnes, G. Lines, K .A. Mardal, and A Tveito Multigrid block preconditioning for a coupled system of partial differential equations modeling the electrical activity in the heart *Comput. Methods Biomech. Biom. Engrg.*, 5(6): 397–409 (2002)

[52] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics.* Springer, 2004.

[53] J. Trangestein and C. Kim. Operator splitting and adaptive mesh refinement for the luo–rudy i model. *J. Comput. Phys.*, 196:645–679, 2004.

[54] Henk A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific Computing*, 13:631–644, 1992.

[55] M. Veneroni. Reaction-diffusion systems for the macroscopic bidomain model of the cardiac electric field. *Math. Meth. Appl. Sci.* 29(14) 1631–1661 (2006).

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

[57] E. J. Vigmond, R. Weber dos Santos, A. J. Prassl, M. Deo, and G. Plank. Solvers for the cardiac bidomain equations. *Progress in Biophysics & Molecular Biology*, 96:3–18, 2008.

[58] R. Weber dos Santos, G. Plank, S. Bauer, and E. J. Vigmond. Parallel multigrid preconditioner for the cardiac bidomain model. *IEEE Trans. Biomed. Eng.*, 51(11):1960–1968, 2004.

[59] R. Weber dos Santos, G. Plank, S. Bauer, and E. J. Vigmond. Preconditioning techniques for the bidomain equations. In *Lecture Notes In Computational Science And Engineering*, volume 40, pages 571–580. Springer, New York, 2004.