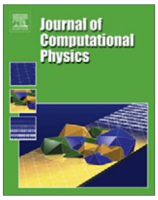
Journal of Computational Physics 231 (2012) 82–97

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

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



Preconditioning the bidomain model with almost linear complexity

Charles Pierre

Laboratoire de Mathématiques et de leurs Applications, UMR CNRS 5142, Université de Pau et des Pays de l'Adour, av. de l'Université BP 1155, 64013 PAU Cédex, France

ARTICLE INFO

Article history:
Received 13 December 2010
Received in revised form 25 August 2011
Accepted 28 August 2011
Available online 14 September 2011

Keywords:
Preconditioning
Electro-cardiology
Bidomain and monodomain models
Hierarchical matrices
Reaction diffusion equations
Numerical simulations

ABSTRACT

The bidomain model is widely used in electro-cardiology to simulate spreading of excitation in the myocardium and electrocardiograms. It consists of a system of two parabolic reaction diffusion equations coupled with an ODE system. Its discretisation displays an ill-conditioned system matrix to be inverted at each time step: simulations based on the bidomain model therefore are associated with high computational costs. In this paper we propose a preconditioning for the bidomain model either for an isolated heart or in an extended framework including a coupling with the surrounding tissues (the torso). The preconditioning is based on a formulation of the discrete problem that is shown to be symmetric positive semi-definite. A block *LU* decomposition of the system together with a heuristic approximation (referred to as the monodomain approximation) are the key ingredients for the preconditioning definition. Numerical results are provided for two test cases: a 2D test case on a realistic slice of the thorax based on a segmented heart medical image geometry, a 3D test case involving a small cubic slab of tissue with orthotropic anisotropy. The analysis of the resulting computational cost (both in terms of CPU time and of iteration number) shows an almost linear complexity with the problem size, i.e. of type $n\log^{\alpha}(n)$ (for some constant α) which is optimal complexity for such problems.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

The bidomain model [1–7] is up to now the most physiologically founded model to describe the heart electrical activity. The bidomain model is here considered in an extended version referred to as the *coupled heart and torso bidomain model*. It includes a coupling of the cardiac electrical activity with the surrounding tissue electrical activity, allowing in particular electrocardiogram simulations.

The bidomain model mathematical formulation is composed of a system of two PDEs (parabolic reaction diffusion equations) describing the evolution of two potentials: the *intra-* and *extra-cellular potentials* within the myocardium. This system is coupled with a set of ODEs modelling the kinetic of ionic transfer across the cellular membrane.

The discretisation of the bidomain model displays an ill conditioned system matrix to be inverted at each time step. This is essentially due to the nature of the model. Two reasons are raised for this. The bidomain model can be formulated as a degenerate system of two coupled parabolic equations [4], which degeneracy causes ill-conditioning. Another formulation of the bidomain model, made up of a single scalar semi-linear parabolic equation, is studied in [8]. This formulation involves a non-local operator of second order in space, referred to as the *bidomain operator*. The bidomain operator is defined as the harmonic mean between two elliptic operators. The non-locality of the bidomain operator generates high computational costs.

On top of this structural ill-conditioning, the physical features of the modelled phenomena (because of fast and sharp space and time variations of potential: namely transmembrane potential wave fronts) necessitates to resort to fine space

E-mail address: charles.pierre@univ-pau.fr
http://www.web.univ-pau.fr/cpierre1/~cpierre1/

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.08.025

and time grids. Ill conditioning together with fine meshes imply very high computational costs for the bidomain model simulations that remain challenging for 3D realistic settings. For this, many efforts were devoted to the reduction of this cost, see e.g. [9–14].

Few papers are dealing with the preconditioning of the bidomain model. In [15] Pavarino and Scacchi proposed a preconditioner designed to a parallel implementation of the bidomain model. In [16] Gerardo-Giorda et al. introduced a very interesting preconditioning strategy discussed deeper on at the end of this section.

The aim of this paper is the definition of a general preconditioning for the bidomain system of equations. Its implementation only requires in practise to define two local block preconditioners for two matrices: obtained by discretising an elliptic and a parabolic type equations respectively. A wide class of preconditioners for such problems already has been developed, either sequential or parallel, with available implemented versions (see e.g. [17–21], details follow). We actually can resort to any of these preconditioners to embed it into the bidomain model preconditioning here presented. In this sense, our preconditioning framework provides a lifting from preconditioners for elliptic problems to preconditioners for the bidomain model.

The natural question raised by this is: "can we recover the (already available) high performances of elliptic problem preconditioners for the bidomain equations ?". This question is here addressed from the point of view of complexity. Let A denote a sparse matrix with size n obtained by discretising an elliptic equation. Optimal complexity to perform $X \mapsto A^{-1}X$ is in $O(n\log(n)^{\alpha})$ (α constant) referred to as almost linear complexity (developments on complexity matters are given in Section 5.3). Optimal complexity has been obtained for elliptic problems for instance using multi-grid approaches [17,19] or hierarchical matrix factorisations [21–24]. In this paper we numerically prove that almost linear complexity can be reached for the bidomain model embedding a hierarchical Cholesky decomposition into our general bidomain model preconditioning.

Several (equivalent) mathematical formulations of the bidomain model have been proposed: we refer to [7] for a comprehensive review. The bidomain model can be set as a system of two coupled degenerate parabolic equations: this formulation has been used to prove existence of solutions in [4,25] and numerically used e.g. in [26,27,15]. A second formulation involves a coupled parabolic-elliptic system of two equations. This formulation has been widely studied either for theoretical or numerical purposes: either using non-symmetric versions (see for instance [16,28]) or a self-adjoint positive semi-definite version studied in [8]. We consider here a general discretisation of the self-adjoint formulation. This discrete formulation of the bidomain model is here shown to be symmetric positive semi-definite: this property holds including the coupling of the heart with the surrounding tissues. This discrete formulation of the bidomain model has already been used e.g. in [29,30].

Embedding the strong structural properties of the bidomain model (i.e. symmetry and positivity) at the discrete level is quite natural and should provide an efficient implementation. We personally experienced the difference between the symmetric positive formulation here adopted and the non-symmetric one in [28]. A gain in CPU time of factor more than 5 was made with the symmetric positive version and for a similar resolution strategy.

Let us now detail the general preconditioning strategy. It relies on the symmetric positive semi-definite formulation of the coupled heart and torso bidomain model. Various space discretisations (including classical Lagrange P^k finite elements or various finite volume techniques) can be considered. For simplicity we adopted here an Euler semi-implicit time discretisation but the technique generalises to more sophisticated time schemes. Once discretised, this formulation involves the inversion of one system matrix (symmetric positive semi-definite) per time step. The two following points are used to precondition the system matrix.

1- *LU* **factorisation**. The system matrix displays a 2×2 block structure that can be factorised into a block-*LU* form.

2- **Monodomain model heuristic**. Among the blocks of the *LU* factorisation, all blocks have a simple definition (they are sparse and do not lead to computational difficulties) except one block. This block is of *Schur component* type. It is shown to be symmetric positive definite and to be the sum of a mass matrix and of a *discrete bidomain operator* (discrete analogue of the bidomain operator mentioned earlier on) that is shown to be the harmonic mean between two stiffness matrices.

The monodomain model approximation basically consists in approximating the *bidomain operator* in [8] (the harmonic mean between two diffusion operators) by a simple diffusion operator. The monodomain model can provide an accurate approximation of the bidomain model [31,27,32,33]. It has been shown in [33] that a monodomain model could provide activation time mappings in complex situations with 1% of relative error as compared to the bidomain model predictions. The diffusivity tensor for the monodomain model approximation will here be set to the harmonic mean of the intra-and extra-cellular conductivity tensors.This approximation is heuristic, it is exact in dimension 1 and in case of equal anisotropy ratio between the intra-and extra-cellular media.

In a recent paper [16], Gerardo-Giorda et al. introduced a preconditioner for the bidomain model also based on a monodomain model heuristic approximation and on a lower block triangular approximation. Let us point out the differences between these two papers. The *LU* factorisation presented here should provide more efficient algorithms than the lower bock-triangular approximation since this factorisation is exact. The formulation in [16] is based on a non-symmetric formulation whereas we here considered a symmetric positive semi-definite system matrix. We then can benefit from symmetry and positivity properties in terms of computational efficiency, for instance resorting to a conjugate gradient linear solver. A draft of quantitative comparison between these two preconditioning is made in the conclusion Section 6.3.

The paper is organised as follows. The coupled heart and torso bidomain model is stated in Section 2. Its numerical discretisation follows in Section 3. In Section 4 are stated and proved the mathematical properties of the discretised bidomain problem system matrix: it is shown to be symmetric positive semi-definite, its *LU* block factorisation is then analysed. The general preconditioning of the bidomain model is defined in Section 5, Section 5.3 is devoted to its practical implementation. Numerical results are in Section 6. The two test cases are presented in 6.1. The complexity of the preconditioned system matrix inversion is numerically studied in Section 6.2. Results are discussed in the conclusion Section 6.3.

2. Bidomain model of the heart embedded in the torso

Let us denote by Ω and H two bounded open subsets such that $H \subset \Omega \subset \mathbb{R}^d$ with $d = 2, 3$ and with smooth boundaries. We moreover assume that $\partial\Omega \cap \partial H = \emptyset$: Ω represents a thorax and H the region occupied by the heart (assumed fixed here). We also consider $T := \Omega - \overline{H}$ that will be referred to as the *torso*, see Fig. 1. We denote Q, Q_H and Q_T the time-space cylinders $\mathbb{R}^+ \times \Omega, \mathbb{R}^+ \times H$ and $\mathbb{R}^+ \times T$ respectively.

Two potential fields will be involved, the transmembrane potential $v : Q_H \mapsto \mathbb{R}$ and the potential $u : Q \mapsto \mathbb{R}$. When restricted to H (*resp.* to T), the potential u is referred to the extra-cellular potential (resp. extra-cardiac potential). The transmembrane potential $v = u_i - u_{|H}$ is the difference between an intra-cellular potential $u_i : Q_H \mapsto \mathbb{R}$ and the extra-cellular potential $u_{|H}$; the intra-cellular potential will not be considered in the following mathematical formulation of the problem.

The heart has a fibrous organisation implying anisotropic electrical conductivities. The cardiac fibres rotate around the ventricular cavities, see Fig. 1. The fibres remain tangent to the cardiac boundaries. This anisotropy is taken into account by introducing in H two tensors σ_i and σ_e. Introducing the 4 conductivity parameters $g^l_{i,e}, g^t_{i,e}$, they read as follows:

$$\sigma_i(x) = \mathrm{Diag}(g^l_i, g^t_i), \quad \sigma_e(x) = \mathrm{Diag}(g^l_e, g^t_e),$$

in a moving system of coordinates whose principal orientation is given by the fibre orientation at point x. Of course, when written in a fixed basis, these tensors no longer are diagonal. Physically, the parameters $g^l_{i,e}$ and $g^t_{i,e}$ are the electrical conductivities longitudinally and transversely to the fibre direction (subscript l and t) and relatively to the intra-or extra-cellular media (index i or e) respectively.

The torso region T is assumed to have an isotropic but heterogeneous electrical conductivity. We define in T the conductivity tensor $\sigma_T(x) = k(x)Id$ where the conductivity $k : T \mapsto \mathbb{R}$ basically is piecewise constant on the different organs considered in T.

The torso model consists in:

$$\begin{cases} \mathrm{div}(\sigma_T(x)\nabla u) = 0, & (t,x) \in \quad Q_T, \\ \nabla u \cdot \mathbf{n} = 0 & \text{on} \quad \partial\Omega, \end{cases} \tag{1}$$

where $\mathbf{n}$ denotes the outward unit normal to $\partial\Omega$.

In the heart region, the bidomain model is composed of the three following equations in H, for $(t,x) \in Q_H$:

$$\begin{cases} \mathrm{div}((\sigma_i(x) + \sigma_e(x))\nabla u) = -\mathrm{div}(\sigma_i(x)\nabla v), \\ \chi(c\partial_t v + I_{ion}(v, \mathbf{w}) - I_{st}(t,x)) = \mathrm{div}(\sigma_i(x)\nabla(u+v)), \\ \partial_t \mathbf{w} = g(v, \mathbf{w}). \end{cases} \tag{2}$$

In the second equation, c denotes the cell membrane surface capacitance, χ is the ratio of cell membrane surface per unit volume, $I_{st} : Q_H \mapsto \mathbb{R}$ is the stimulation current (source term). $I_{ion}(v, \mathbf{w})$ (reaction term) denotes the surface ionic current distribution on the membrane. The gating variable $\mathbf{w} : Q_H \mapsto \mathbb{R}^p$ characterises the state of the cell membrane, its evolution is ruled by the ODE system in the third equation. The definitions of I_{ion} and of g are fixed by the chosen ionic model in Section 6.1.

Eqs. (2) are coupled with the torso model (1) with the following coupling condition:

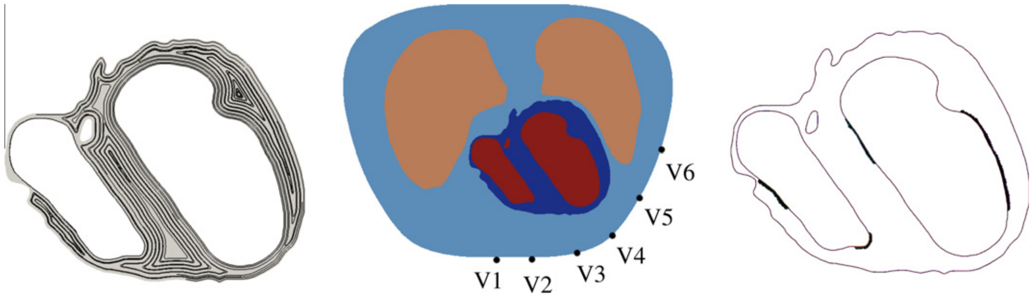


Fig. 1. 2D test case description. Left: fibrous anisotropic structure of the two ventricles. Middle: 2D geometry Ω and its sub-domains. body surface potential (ECG) are recorded at the vertices V1 to V6. Right: stimulation site locations.

$$\text{on } \partial H: \quad \begin{cases} u_{|H} = u_{|T}, \quad \sigma_e(x)\nabla u_{|H}\cdot\mathbf{n} = \sigma_T(x)\nabla u_{|T}\cdot\mathbf{n}, \\ \sigma_i\nabla u_{|H}\cdot\mathbf{n} + \sigma_i\nabla v\cdot\mathbf{n} = 0. \end{cases} \tag{3}$$

where $\mathbf{n}$ denotes the outward unit normal to ∂H.

The model is closed by imposing initial conditions on v and $\mathbf{w}$,

$$v(0,x) = v_0(x), \quad \mathbf{w}(0,x) = \mathbf{w}_0(x), \quad x\in H. \tag{4}$$

Clearly, the potential field u is defined up to an additive constant. We therefore impose the normalisation condition for all time $t > 0$:

$$\int_\Omega u(t,\cdot)dx = 0. \tag{5}$$

2.1. Weak formulation

We introduce the tensor $\bar\sigma_1$ on Ω:

$$\bar\sigma_1(x) = \begin{cases} \sigma_i(x)+\sigma_e(x), \quad x\in H \\ \sigma_T(x), \quad x\in T \end{cases}.$$

The weak formulation of the bidomain model (1)–(3) is the following: $\forall\ \psi\in H^1(\Omega)$, $\forall\phi\in H^1(H)$,

$$\begin{cases} \int_\Omega \bar\sigma_1\nabla u\cdot\nabla\psi dx + \int_H \sigma_i\nabla v\cdot\nabla\psi dx = 0, \\ \chi c\partial_t\int_H v\phi dx + \chi\int_H (I_{ion}(v,\mathbf{w}) - I_{st}(x,t))\phi dx = -\int_H \sigma_i\nabla(u+v)\cdot\nabla\phi dx, \end{cases} \tag{6}$$

The first equation in (6) is obtained by multiplying (1) and the first equation in (2) by a test function $\psi\in H^1(\Omega)$, by integrating on Ω and by using the coupling conditions (3) and the boundary condition (1). The second equation in (6) is obtained by multiplying the second equation in (2) by a test function $\phi\in H^1(H)$, by integrating on H together with (3).

2.2. Case of an isolated heart

We here address the case where the heart is considered as isolated from the surrounding tissues. In this case we have $H = \Omega$ and $T = \emptyset$. Eqs. (2) only are considered and the coupling conditions (3) are replaced by zero flux boundary conditions on ∂H for v and u.

3. Implementation

For simplicity, temporal discretisation is fixed to a semi implicit Euler scheme: implicit for the diffusion and explicit on the reaction. Extensions to other time schemes is possible as discussed in Remark 2. The implementation strategy is similar for various space discretisations including P^k Lagrange finite elements or finite volume scheme such as the CVFE scheme (Control Volume Finite Element, see e.g. [34]) or such as the DDFV scheme in [28]. Assumptions ($H1$) and ($H2$) on the space discretisation are detailed in Section 3.1 whereas the numerical scheme itself is presented in Section 3.2.

3.1. Settings

Let us consider a mesh $\mathfrak{M}$ of Ω and a mesh $\mathfrak{M}_H$ of the cardiac region H: we assume that $\mathfrak{M}_H$ is a sub mesh of $\mathfrak{M}$, that is to say that all elements (or cells or control volumes) of $\mathfrak{M}_H$ also are elements of $\mathfrak{M}$.

Relatively to the considered space discretisation, let us denote by $\mathbb{R}^{\mathfrak{M}}$, $\mathbb{R}^{\mathfrak{M}_H}$ the set of discrete functions attached to these two meshes. Their dimensions are denoted N and N_H respectively. A "*natural*" basis usually is provided for $\mathbb{R}^{\mathfrak{M}}$ and $\mathbb{R}^{\mathfrak{M}_H}$, denoted $(U_i)_{1\leqslant i\leqslant N}$ and $(U_i^H)_{1\leqslant i\leqslant N_H}$ respectively. In the case of P^k finite element methods, these functions simply are the standard P^k Lagrange basis functions. Considering these basis induces an isomorphism between $\mathbb{R}^{\mathfrak{M}}$ and $\mathbb{R}^N$ and between $\mathbb{R}^{\mathfrak{M}_H}$ and $\mathbb{R}^{N_H}$. A discrete function $U = \sum_{i=1}^N c_iU_i$ will be considered either as a real function or as the real vectors $(c_i)_{i\leqslant 1\leqslant N}$. Using these identifications, the canonical Euclidian structures on $\mathbb{R}^N$ and $\mathbb{R}^{N_H}$ extend to $\mathbb{R}^{\mathfrak{M}}$ and $\mathbb{R}^{\mathfrak{M}_H}$. We denote by $(\cdot,\cdot)_{\mathfrak{M}}$ and $(\cdot,\cdot)_{\mathfrak{M}_H}$ the associated scalar products.

We make the following first assumption on the space discretisation method:

($H1$) for all i, $1\leqslant i\leqslant N_H$: $U_{i|H} = U_i^H$ (where $U_{i|H}$ denotes the restriction of the function U_i to H).

In the case of the P^k finite element methods, this first assumption is true modulo a reordering of the basis functions $(U_i)_{1\leqslant i\leqslant N}$. Assumption ($H1$) allows us to define the restriction operation:

$$\Pi : U = \sum_{i=1}^{N} c_i U_i \in \mathbb{R}^{\mathfrak{M}} \mapsto U_{|H} = \sum_{i=1}^{N_H} c_i U_i^H \in \mathbb{R}^{\mathfrak{M}_H}. \tag{7}$$

Equivalently, Π can be seen as a simple truncation operation:

$$\Pi : U = (c_i)_{1\leqslant i\leqslant N} \in \mathbb{R}^{\mathfrak{M}} \mapsto U_{|H} = (c_i)_{1\leqslant i\leqslant N_H} \in \mathbb{R}^{\mathfrak{M}_H},$$

following the above described identification between $\mathbb{R}^{\mathfrak{M}}$ and $\mathbb{R}^N$ and between $\mathbb{R}^{\mathfrak{M}_H}$ and $\mathbb{R}^{N_H}$. The transpose mapping ${}^T\Pi$ for Π is:

$${}^T\Pi : U = \sum_{i=1}^{N_H} c_i U_i^H \in \mathbb{R}^{\mathfrak{M}_H} \mapsto \sum_{i=1}^{N_H} c_i U_i \in \mathbb{R}^{\mathfrak{M}}.$$

We point out that in this discrete setting ${}^T\Pi$ does not match the prolongation by zero outside H. The following property will be useful:

$$\Pi^T\Pi = id_{\mathbb{R}^{\mathfrak{M}_H}}. \tag{8}$$

Let us introduce the mass matrices M, M_H and the stiffness matrices S_1, S_i so that:

$$\forall U_1, U_2 \in \mathbb{R}^{\mathfrak{M}} : \int_\Omega U_1 U_2 dx = (MU_1, U_2)_{\mathfrak{M}}, \quad \int_\Omega \bar{\sigma}_1 \nabla U_1 \cdot \nabla U_2 dx = (S_1 U_1, U_2)_{\mathfrak{M}}$$
$$\forall V_1, V_2 \in \mathbb{R}^{\mathfrak{M}_H} : \int_H V_1 V_2 dx = (M_H V_1, V_2)_{\mathfrak{M}_H}, \quad \int_H \sigma_i \nabla V_1 \cdot \nabla V_2 dx = (S_i V_1, V_2)_{\mathfrak{M}_H}$$

The second assumption on the space discretisation is the following:

($H2$) Let us denote I_Ω and I_H the characteristic functions of Ω and H respectively (constant functions equal to one):

$$I_\Omega \in \mathbb{R}^{\mathfrak{M}}, \quad I_H \in \mathbb{R}^{\mathfrak{M}_H}. \tag{9}$$

Assumption ($H2$) is related with the considered boundary conditions here: homogeneous Neumann on $\partial\Omega$ and transmission conditions on ∂H. It implies that the stiffness matrices S_1, S_i (that are symmetric positive semi-definite) have for kernels the one dimensional spaces $I_\Omega\mathbb{R}$ and $I_H\mathbb{R}$ respectively.

3.2. Scheme statement

The three unknowns v, u and $\mathbf{w}$ of the (continuous) bidomain model are represented by the discrete functions $U \in \mathbb{R}^{\mathfrak{M}}, V \in \mathbb{R}^{\mathfrak{M}_H}$ and $W \in [\mathbb{R}^{\mathfrak{M}_H}]^p$.

We have for all test function $\Psi \in \mathbb{R}^{\mathfrak{M}}$:

$$\int_H \sigma_i \nabla V \cdot \nabla \Psi dx = (S_i V, \Pi\Psi)_{\mathfrak{M}_H} = ({}^T\Pi S_i V, \Psi)_{\mathfrak{M}}$$

Discretisation of (6) thus is:

$$\begin{cases} S_1 U^{n+1} + {}^T\Pi S_i V^{n+1} = 0, \\ \chi c M_H \frac{V^{n+1} - V^n}{\Delta t} + \chi M_H \left(I_{ion}(V^n, W^n) - I_{st}^n\right) = -S_i \Pi U^{n+1} - S_i V^{n+1} \end{cases}. \tag{10}$$

We introduce the positive parameter γ:

$$\gamma := \chi c / \Delta t.$$

Resolution algorithm 1. The complete bidomain model (1)–(3) is numerically solved applying the following three operations at each time step.

Being given $V^n \in \mathbb{R}^{\mathfrak{M}}$ and $W^n \in [\mathbb{R}^{\mathfrak{M}_H}]^p$:

Step 1. Compute the right hand side Y:

$$Y := \begin{bmatrix} 0 \\ M_H\left(\gamma V^n - \chi(I_{ion}(V^n, W^n) - I_{st}^n)\right) \end{bmatrix}.$$

Step 2. find the solution $X = {}^T[U^{n+1}, V^{n+1}]$ to $\Lambda X = Y$ with

$$\Lambda := \begin{bmatrix} S_1 & {}^T\Pi S_i \\ S_i\Pi & \gamma M_H + S_i \end{bmatrix} \quad \text{that satisfies} \quad \int_\Omega U^{n+1} dx = 0. \tag{11}$$

Step 3. Update the gating variable by computing W^{n+1} according to the third equation in Eq. (2). □

This paper is devoted to Step 2 only. Proposition 1 states that step 2 is well posed.

4. Properties and *LU* factorisation of the system matrix Λ

Let us precise that $S_1 : \mathbb{R}^{\mathfrak{M}} \mapsto \mathbb{R}^{\mathfrak{M}}$ and that $S_i : \mathbb{R}^{\mathfrak{M}_H} \mapsto \mathbb{R}^{\mathfrak{M}_H}$. Then, $\Lambda : \mathbb{R}^{\mathfrak{M}} \times \mathbb{R}^{\mathfrak{M}_H} \mapsto \mathbb{R}^{\mathfrak{M}} \times \mathbb{R}^{\mathfrak{M}_H}$.

Proposition 1. *The system matrix Λ is symmetric positive semi-definite with kernel* $\mathrm{Ker}(\Lambda) = I_\Omega\mathbb{R} \times \{0\}$. *By symmetry Λ has for range* $\mathrm{Ran}(\Lambda) = I_\Omega^\perp \times \mathbb{R}^{\mathfrak{M}_H}$. *For all $(Y_1, Y_2) \in I_\Omega^\perp \times \mathbb{R}^{\mathfrak{M}_H}$, there exists a unique $(U,V) \in \mathbb{R}^{\mathfrak{M}} \times \mathbb{R}^{\mathfrak{M}_H}$ such that*

$$\Lambda\begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} \quad \text{and} \quad \int_\Omega U dx = 0. \tag{12}$$

The resolution of step 2 in the resolution algorithm proceeds in two steps: first find a solution ${}^T[X_1,X_2]$, then normalise X_1. We now focus on the first step.

Definition 1 (*Pseudo-inverses $S_1^{\widetilde{-1}}$ and $S_i^{\widetilde{-1}}$*). The stiffness matrices S_1 and S_i are isomorphisms on $I_\Omega^\perp = \mathrm{Ran}(S_1)$ and on $I_H^\perp = \mathrm{Ran}(S_i)$ respectively. We introduce their pseudo inverses $S_1^{\widetilde{-1}}$ and $S_i^{\widetilde{-1}}$: they are equal to the inverse of S_1, S_i on $I_\Omega^\perp, I_H^\perp$ respectively and equal to 0 on $I_\Omega\mathbb{R}, I_H\mathbb{R}$ respectively.

Considering p_Ω (resp. p_H) the orthogonal projection of $\mathbb{R}^{\mathfrak{M}}$ on $I_\Omega^\perp$ (resp. of $\mathbb{R}^{\mathfrak{M}_H}$ on $I_H^\perp$), we have:

$$S_1^{\widetilde{-1}}S_1 = S_1S_1^{\widetilde{-1}} = p_\Omega, \quad S_i^{\widetilde{-1}}S_i = S_iS_i^{\widetilde{-1}} = p_H.$$

Proposition 2. *We have the block decomposition $\Lambda = LU$ with:*

$$L := \begin{bmatrix} S_1 & 0 \\ S_i\Pi & K \end{bmatrix}, \quad U := \begin{bmatrix} id_{\mathbb{R}^{\mathfrak{M}}} & S_1^{\widetilde{-1}T}\Pi S_i \\ 0 & id_{\mathbb{R}^{\mathfrak{M}_H}} \end{bmatrix}, \tag{13}$$

The matrix K is defined by $K := \gamma M_H + S_i - S_i\Pi S_1^{\widetilde{-1}T}\Pi S_i$, it is symmetric, positive definite. Let us consider the tensor

$$\bar{\sigma}_e(x) = \begin{cases} \sigma_e(x), & x \in H \\ \sigma_T(x), & x \in T \end{cases},$$

and denote S_e the associated stiffness matrix. Since S_1 and S_e have the same range $I_\Omega^\perp$, one can define the pseudo-inverse $S_e^{\widetilde{-1}}$ for S_e with the same meaning as for S_1. Then the matrix K can be rewritten as

$$K = \gamma M_H + \left(S_i^{-1} + \Pi S_e^{-1T}\Pi\right)^{-1}. \tag{14}$$

where all inverses are pseudo-inverses.

Remark 1. The matrix K basically is the Schur component of the system matrix Λ using S_1 as pivot, with the slight difference than S_1 is not invertible.

Eq. (14) is quite interesting: the second term appears as the "*harmonic mean*" between the stiffness matrices S_i and S_e. This second term is the discretisation of the "*bidomain operator*" as defined in [8] that was introduced as the harmonic mean between two diffusion operators.

Proposition 3. *L has a pseudo inverse $L^{\widetilde{-1}}$ in the following sense:*

$$LL^{\widetilde{-1}} = L^{\widetilde{-1}}L = \begin{bmatrix} p_\Omega & 0 \\ 0 & id_{\mathbb{R}^{\mathfrak{M}_H}} \end{bmatrix},$$

U is invertible, U^{-1} and $L^{\widetilde{-1}}$ are given by:

$$L^{\widetilde{-1}} = \begin{bmatrix} S_1^{\widetilde{-1}} & 0 \\ -K^{-1}S_i\Pi S_1^{\widetilde{-1}} & K^{-1} \end{bmatrix}, \quad U^{-1} = \begin{bmatrix} id_{\mathbb{R}^{\mathfrak{M}}} & -S_1^{\widetilde{-1}T}\Pi S_i \\ 0 & id_{\mathbb{R}^{\mathfrak{M}_H}} \end{bmatrix}. \tag{15}$$

For $Y \in \mathrm{Ran}(\Lambda)$, a solution to $\Lambda X = Y$ is provided by $X = U^{-1}L^{\widetilde{-1}}Y$.

Remark 2 (*About the time discretisation*). Choosing another time discretisation scheme will basically imply two changes: the computation of the right hand side (Step 1 in the resolution algorithm above) and the definition of K. In general the global structure of the system matrix Λ (which is symmetric positive semi-definite) as well as the positivity of K will not be affected by considering different time discretisation: this is for instance the case for the Crank–Nicolson scheme or for operator splitting schemes (Strang formula e.g.).

Proof of Proposition 1. For $X = {}^T(U,V) \in \mathbb{R}^{\mathfrak{M}} \times \mathbb{R}^{\mathfrak{M}_H}$, we have:

$$ {}^T X \Lambda X = (S_1 U, U)_{\mathfrak{M}} + 2(S_i \Pi U, V)_{\mathfrak{M}_H} + (S_i V, V)_{\mathfrak{M}_H} + \gamma (M_H V, V)_{\mathfrak{M}_H} $$

We consider S_e and $\bar{\sigma}_e$ defined in Remark 1. Since $\bar{\sigma}_1 - \bar{\sigma}_e$ is equal to 0 on T and to σ_i on H, $S_1 - S_e$ is positive semi-definite.

Eq. (8) says that $({}^T\Pi V)_{|\,H} = \Pi^T \Pi V = V$. Together with $\bar{\sigma}_1 - \bar{\sigma}_e = 0$ outside H one gets:

$$ (S_i V, V)_{\mathfrak{M}_H} = \int_H (\bar{\sigma}_1 - \bar{\sigma}_e) \nabla V \cdot V dx = \int_\Omega (\bar{\sigma}_1 - \bar{\sigma}_e) \nabla^T \Pi V \cdot \nabla^T \Pi V dx = \left((S_1 - S_e)^T \Pi V, {}^T \Pi V \right)_{\mathfrak{M}} $$

$$ (S_i \Pi U, V)_{\mathfrak{M}_H} = \int_H (\bar{\sigma}_1 - \bar{\sigma}_e) \nabla \Pi U \cdot \nabla V dx = \int_\Omega (\bar{\sigma}_1 - \bar{\sigma}_e) \nabla U \cdot \nabla^T \Pi V dx = \left((S_1 - S_e) U, {}^T \Pi V \right)_{\mathfrak{M}}. $$

From these two equalities we deduce that:

$$ {}^T X \Lambda X = (S_e U, U)_{\mathfrak{M}} + \left((S_1 - S_e)(U + {}^T \Pi V), (U + {}^T \Pi V) \right)_{\mathfrak{M}} + \gamma (M_H V, V)_{\mathfrak{M}_H} $$

so ensuring that Λ is positive semi-definite. Assuming that $\Lambda X = 0$ implies that all the terms on the right of the last equality are equal to zero. The mass matrix being definite this means $V = 0$ and so $S_1 U = 0$. Thus $U \in \mathrm{Ker}(S_1) = I_\Omega \mathbb{R}$ and we then have $\mathrm{Ker}(\Lambda) = I_\Omega \mathbb{R} \times \{0\}$.

Let $X = {}^T[U, V]$ be a solution to $\Lambda X = Y$ for $Y \in \mathrm{Ran}(\Lambda)$. A simple computation shows that $Z = {}^T[U - \alpha I_\Omega, V]$ is the unique solution to (12) iff $\alpha = (MU, I_\Omega)_{\mathfrak{M}} / (M I_\Omega, I_\Omega)_{\mathfrak{M}}$, so ending the proof. □

Proof of Proposition 2. We have:

$$ LU = \begin{bmatrix} S_1 & p_\Omega{}^T \Pi S_i \\ S_i \Pi & \gamma M_H + S_i \end{bmatrix}, $$

and so $LU = \Lambda$ *iif* $p_\Omega{}^T \Pi S_i = {}^T \Pi S_i$. This last equality holds since for all $V \in \mathbb{R}^{\mathfrak{M}_H}$,

$$ ({}^T \Pi S_i V, I_\Omega)_\Omega = (S_i V, \Pi I_\Omega)_H = (S_i V, I_H)_H = 0, $$

and so $\mathrm{Ran}({}^T \Pi S_i) \subset I_\Omega^\perp$.

The symmetry of K is obvious. Let us prove it is positive definite.

We decompose $K = \gamma M_H + K_0$ so with $K_0 := S_i - S_i \Pi S_1^{\tilde{-1}\,T} \Pi S_i$. We will prove that K_0 (which is symmetric) is positive semi-definite. This implies the positivity of K since γM_H is positive definite. Precisely: K_0 clearly vanishes on $I_H \mathbb{R}$. Then $I_H^\perp$ is stable by K_0. Let us prove that K_0 is positive definite on $I_H^\perp$.

We consider again S_e and $\bar{\sigma}_e$ defined in Remark 1. Let us first prove that:

$$ K_0 = \Pi S_e S_1^{\tilde{-1}\,T} \Pi S_i \tag{16} $$

Firstly, we have: $\forall U_1 . U_2 \in \mathbb{R}^{\mathfrak{M}}$,

$$ \int_\Omega (\bar{\sigma}_1 - \bar{\sigma}_e) \nabla U_1 \cdot \nabla U_2 dx = \int_H \sigma_i \nabla U_1 \cdot \nabla U_2 dx, $$

and so ${}^T \Pi S_i \Pi = S_1 - S_e$.

Secondly, multiplying K_0 by $\Pi^T \Pi = id_{\mathbb{R}^{\mathfrak{M}_H}}$ on the left gives:

$$ \begin{aligned} K_0 = \Pi^T \Pi K_0 &= S_i - \Pi^T \Pi S_i \Pi S_1^{\tilde{-1}\,T} \Pi S_i = S_i - \Pi (S_1 - S_e) S_1^{\tilde{-1}\,T} \Pi S_i = S_i - \Pi (p_\Omega - S_e S_1^{\tilde{-1}})^T \Pi S_i \\ &= \Pi S_e S_1^{\tilde{-1}\,T} \Pi S_i + S_i - \Pi p_\Omega{}^T \Pi S_i. \end{aligned} $$

One already showed in this proof that $p_\Omega{}^T \Pi S_i = {}^T \Pi S_i$ ensuring that $\Pi p_\Omega{}^T \Pi S_i = S_i$. This gives us (16).

Clearly $S_e^{\tilde{-1}}$ and $S_i^{\tilde{-1}}$ are positive definite on $I_\Omega^\perp$ and $I_H^\perp$ respectively. We moreover have ${}^T \Pi (I_H^\perp) \subset I_\Omega^\perp$ since for all $V \in I_H^\perp$:

$$ ({}^T \Pi V, I_\Omega)_\Omega = (V, \Pi I_\Omega)_H = (V, I_H)_H = 0. $$

Then $\Pi S_e^{\tilde{-1}\,T} \Pi$ is positive definite on $I_H^\perp$. Let us define $A := (S_i^{\tilde{-1}} + \Pi S_e^{\tilde{-1}\,T} \Pi) : I_H^\perp$ is stable by A. A is positive definite and so invertible on $I_H^\perp$. We will end this proof by showing that $K_0 = A^{-1}$ on $I_H^\perp$.

$$K_0A = (\Pi S_e S_1^{-1T}\Pi S_i)(S_i^{-1} + \Pi S_e^{-1T}\Pi) = \Pi S_e S_1^{-1T}\Pi p_H + \Pi S_e S_1^{-1T}\Pi S_i \Pi S_e^{-1T}\Pi = \Pi S_e S_1^{-1T}\Pi p_H + \Pi S_e S_1^{-1}(S_1 - S_e)S_e^{-1T}\Pi$$
$$= \Pi S_e S_1^{-1T}\Pi p_H + \Pi S_e (p_\Omega S_e^{-1} - S_i^{-1} p_\Omega)^T \Pi = \Pi S_e S_1^{-1T}\Pi p_H + \Pi S_e (S_e^{-1} - S_i^{-1})^T \Pi = \Pi p_\Omega{}^T \Pi + \Pi S_e S_1^{-1T}\Pi (p_H - id_{\mathbb{R}^{\mathfrak{M}_H}}).$$

Clearly, $p_H - id_{\mathbb{R}^{\mathfrak{M}_H}}$ vanishes on $I_H^\perp$. Moreover, since ${}^T\Pi(I_H^\perp) \subset I_\Omega^\perp$, $\Pi p_\Omega{}^T \Pi$ is the identity on $I_H^\perp$. Thus $K_0A\,V = V$ for all $V \in I_H^\perp$. □

5. Preconditioning

The previously studied algebraic properties of the system matrix Λ naturally suggest a block-*LU* designed preconditioner for Λ, here defined in Section 5.1. This general algebraic setting is the first key ingredient towards the preconditioning of the bidomain model. The second key ingredient is a heuristic approximation of the matrix *K*, presented in Section 5.2. The last layer to practically implement the subsequent preconditioning indeed is discussed in Section 5.3.

5.1. Preconditioner definition

The practical strategy to solve (11) will be to use an iterative solver for the left preconditioned system:

$$P_\Lambda^{-1}\Lambda X = P_\Lambda^{-1} Y,$$

for a global preconditioner P_Λ defined as follows.

Definition 2. Let us consider P_1 a preconditioner for S_1 and P_K a preconditioner for *K*. We define a global preconditioner P_Λ for Λ as:

$$P_\Lambda = L_P U_P, \quad L_P := \begin{bmatrix} P_1 & 0 \\ S_i\Pi & P_K \end{bmatrix}, \quad U_P := \begin{bmatrix} id_{\mathbb{R}^{\mathfrak{M}}} & P_1^{-1T}\Pi S_i \\ 0 & id_{\mathbb{R}^{\mathfrak{M}_H}} \end{bmatrix}. \tag{17}$$

The inversion of P_Λ is achieved as follows. The solution *X* to $P_\Lambda X = Y$ is given by $X = U_P^{-1} L_P^{-1} Y$ with:

$$L_P^{-1} := \begin{bmatrix} P_1^{-1} & 0 \\ -P_K^{-1} S_i \Pi P_1^{-1} & P_K^{-1} \end{bmatrix}, \quad U_P^{-1} := \begin{bmatrix} id_{\mathbb{R}^{\mathfrak{M}}} & -P_1^{-1T}\Pi S_i \\ 0 & id_{\mathbb{R}^{\mathfrak{M}_H}} \end{bmatrix}. \tag{18}$$

Neglecting the vector additions, the operational cost to compute $X = \Lambda Y$ is:

- 2 multiplications by S_i
- 1 multiplication by S_1
- 1 multiplication by M_H,

Whereas the operational cost to compute $X = P_\Lambda^{-1} Y$ is:

- 2 inversions of P_1,
- 1 inversion of P_K,
- 2 multiplications by S_i,

The symmetry and positivity properties of Λ allow to resort to a preconditioned conjugate gradient (PCG) algorithm to solve (11). The cost for this iterative solver (again neglecting scalar products and vector additions) is for each step: one multiplication by Λ and one inversion of $P_\Lambda^{-1} X = Y$.

5.2. Heuristic approximation of K

The hard task for the definition of P_Λ in (17) is the definition of P_K. As developed in Remark 1, *K* has a complex structure:

$$K = \gamma M_H + K_0,$$

where K_0 is a non-sparse matrix obtained by making the harmonic mean between S_i and S_e. Since *K* is a full matrix, it will never be computed and the alternative strategy to define P_K is to derive an approximation of *K* displaying a sparse pattern.

Let us consider the tensor σ_m:

$$\sigma_m(x) := (\sigma_e^{-1}(x) + \sigma_i^{-1}(x))^{-1}, \quad x \in H,$$

which is the harmonic mean between σ_i and σ_e. We introduce the stiffness matrix S_m associated to σ_m acting on $\mathbb{R}^{\mathfrak{M}_H}$. We make the following approximation:

$$K \simeq K_m := \gamma M_H + S_m.$$

This approximation is referred to as the *monodomain model approximation* [27].

The matrix K_m has a simple structure. It is the discretisation matrix of a parabolic equation. It is moreover symmetric, positive definite and sparse (with the same pattern as S_i).

5.3. Practical implementation of P_1 and P_K

The two preconditioners P_1 and P_K will be built from the matrices S_1 and K_m respectively. These matrices (sparse, symmetric positive semi-definite) have classical structures arising from the discretisation of elliptic and parabolic problems respectively. A wide literature has been devoted to the preconditioning of such matrices: among classical choices we not comprehensively quote incomplete decomposition methods (incomplete *LU* or incomplete Cholesky, see e.g. [18]) multi-grid or multi-level methods, see [17,19]. Fixing one of these classical possible choices actually provide a fully defined implementation of the here presented bidomain model preconditioning.

We insist on the versatility of this bidomain model preconditioning. This versatility relies on the freedom for the choice of P_1 and P_K.

Remark 3 (*Parallelisation*). At this stage, let us underline the consequences on parallelisation induced by this versatility characteristic of the bidomain model preconditioning. Once embedded into some iterative solver (e.g. CG or GMRes) the resolution of system (11) preconditioned by $P_\mathcal{A}$ only requires:

- Matrix vector multiplications by $\mathcal{A}$,
- Inversions of $P_\mathcal{A} X = Y$: as detailed in Section 5.3 this operation consists in matrix vector multiplication and inversions of $P_1 X = Y$ and of $P_K X = Y$,
- Various remaining operations, such as scalar products..

Except the inversions of $P_1 X = Y$ and of $P_K X = Y$, all these operations have trivial parallelisation. But since P_1 and P_K are preconditioners for classical elliptic or parabolic discretised PDEs, classical parallel versions for P_1 and P_K already are available. For instance a review of algebraic methods (such as parallel version of incomplete factorisations) is provided in [20,18]. Another wide class of parallelisation strategies based on domain decomposition is analysed in [35] and also described in [18]. For instance the multi-level additive Schwarz preconditioner, such as presented in [15] and applied to the bidomain model, also could be incorporated inside the here presented general preconditioning framework.

For this reason, the here presented preconditioning strategy for the bidomain model naturally fits with the constraints of parallelism.

Optimal complexity to solve a discretised elliptic problem $AX = Y$ is $O(n)$ with n the system size: since $X \mapsto AX$ has $O(n)$ complexity one cannot hope better for $Y \mapsto A^{-1}Y$ (A being sparse whereas A^{-1} is full). Although this optimality can be reached for some particular problems (for instance in case A is tri-diagonal), in practise the most efficient algorithms have *almost linear complexity*: that is $O(n\log(n)^\alpha)$ with α a constant.

Hierarchical matrices preconditioning strategy [21–24] provides such an almost linear complexity (among various possible choices such as multi-grid methods [17]). This method will be used for the numerical results in Section 6 to precondition S_1 and K_m. This method proceeds in two steps. Firstly compute an approximation of the considered matrix (here S_1 or K_m). This approximation is built using hierarchical matrices arithmetic (basically including block partition of the matrix and defining a blockwise approximation by low rank matrices), ensuring low storage cost. This approximation accuracy is controlled by the parameter ϵ: in matrix norm the error goes to 0 with ϵ. Secondly perform the exact decomposition (either *LU* or Cholesky) of this approximation. Hierarchical Cholesky decomposition has been used here to build P_1 and P_K. Taking advantage of the hierarchical arithmetic, both the construction, storage and inversion of the preconditioners are in $O(n\log(n)^\alpha)$, precisely with $\alpha = 2$ (resp. 4) for the decomposition and $\alpha = 1$ (resp. 2) for the storage/inversion in dimension 2 (resp. 3).

The setting of the accuracy parameter ϵ strongly impacts the preconditioning efficiency. Naturally the PCG convergence rate increases as ϵ goes to 0. A convergence in one single PCG iteration is expected provided a small enough value for ϵ. Meanwhile the preconditioner inversion cost increases as $\epsilon \mapsto 0$: thus the highest PCG convergence rate may not correspond to the most efficient setting of the preconditioner. An optimal value for ϵ (not too small but not too large) has to be searched. PCG convergence rate for such optimal value are shown in Section 6.2 for which 3 PCG iterations typically have to be performed.

In practise the construction of P_1 and P_K was made using the H-Lib library from L. Grasedyck and S. Börm.[1] The sequential version of the code has been used: a parallel version also is available.

6. Numerical results

The efficiency of the preconditioner presented in Section 5 is analysed in this section. The bidomain model has been implemented following Section 3 and using the CVFE finite volume spatial discretisation (see e.g. [34]). For this spatial discretisation the degrees of freedom are located at the mesh vertices and the mass matrices are diagonal. Two test cases are

[1] http://www.hlib.org/

considered, they are detailed in Section 6.1. For these two test cases a depolarisation potential wave is simulated. The spreading of depolarisation The cost for the inversion of the preconditioned system (11) is measured during the spreading of the depolarisation wave, that numerically is by far the stiffest part of the simulation. The dependence of this cost on the problem size is then analysed. For this a series of meshes $\mathfrak{M}_n$ is considered with an increasing number of vertices DOF(n). We here aim to validate an almost linear dependence of the cost on DOF(n).

The cost has been measured in two ways. Firstly in terms of CPU time. The averaged CPU time spent on the inversion of system (11) during the depolarisation sequence is denoted CPU(n). The logarithmic growth rate r_n of CPU(n) relatively to DOF(n) will be considered:

$$r_n = \frac{\log(\mathrm{CPU}(n)/\mathrm{CPU}(n-1))}{\log(\mathrm{DOF}(n)/\mathrm{DOF}(n-1))}. \tag{19}$$

The CPU time measurements however might be perturbed by cache effects and memory-access differences for large-scale problems. To cope with this, the cost also is evaluated in terms of number of iterations. The averaged number of iterations required by the PCG algorithm to invert (11) during the depolarisation sequence is denoted Iter(n). Each step of the PCG algorithm requires one multiplication by $\mathcal{A}$ and one inversion of $P_{\mathcal{A}}$. These operations are of linear and almost linear complexity with DOF(n) respectively. Thus a constant or logarithmic behaviour is expected for Iter(n) to validate an almost linear complexity of the preconditioning.

Numerical results for the preconditioning complexity are presented and discussed in Section 6.2and 6.3 respectively.

6.1. Test cases

For the two test cases, the reaction terms $I_{ion}(v,\mathbf{w})$ and $g(v,\mathbf{w})$ in (2) have been set to the Luo and Rudy ionic model of class II [36] designed for mammalian ventricular cells and for which the system of ODEs in (2) is of size 20 (i.e. $\mathbf{w} \in \mathbb{R}^{20}$). The model parameters χ, c as well as the conductivities are displayed in Table 1: these values are physiological values taken from [37,38].

2D test case. The domain Ω is an horizontal slice of a human thorax. This geometry has been obtained by segmentation of a medical image (CT-Scan, courtesy of the Ottawa Heart Institute) with resolution 0.5 mm. We refer to [39,40] for details on the segmentation procedure. The segmented image is depicted in Fig. 1. It includes 4 sub-domains: the two ventricles (H) and the torso (T) made of the ventricular cavities, the lungs and the remaining tissues.

Four meshes $(\mathfrak{M}_n)_{n=1\ldots4}$ of Ω will be considered: with DOF(1) = 143 053, DOF(2) = 344 408, DOF(3) = 684 112 and DOF(4) = 1 257 312. The associated time steps are Δt= 0.07, 0.05, 0.035 and 0.025 ms respectively. The associated mesh size are h= 0.15, 0.10, 0.07 and 0.05 mm.

The anisotropic structure of the two ventricles is displayed on Fig. 1: bundles of fibres rotating around the ventricular cavities have been considered. Inside the torso T, heterogeneous conductivities have been considered for each sub-domains: the lungs, ventricular cavities and the remaining tissues conductivities are given in Table 1.

With these settings, a depolarisation potential wave is simulated. For this a stimulation current $I_{st}(x,t)$ (see Eq. (2)) is applied during 1 ms at four locations (stimulation sites) on the ventricular cavities as depicted on Fig. 1; the right ventricle being stimulated 5 ms later than the left one.

The spreading of this potential wave across the myocardium is depicted on Fig. 2. The transmembrane potential v in the heart is depicted 15, 30 and 45 ms after stimulation on the left. Without entering the details: the region in blue is at rest potential ($v \simeq -90$ mV) whereas the region in red is excited ($v \simeq 50$ mV). Downward: the excitation wave starts at the stimulation site location and then spreads throughout the cardiac tissue. The activation time $\phi(x)$ is computed pointwise as the time $t = \phi(x)$ so that $v(\phi(x),x) = -20$ mV (the time instant when the depolarisation wave reaches the point x). Activation time are depicted on Fig. 3.

The modifications on the extra-cellular (and extra-cardiac) potential u on Ω (heart and torso) induced by the transmembrane depolarisation wave spreading also is depicted on Fig. 2. The body surface potential (ECG) is recorded at 6 points on

Table 1
Model parameters.

	Values	**Unit**
Cell membrane surface-to-volume ratio (2D)	χ = 1500	[cm^{-1}]
Cell membrane surface-to-volume ratio (3D)	χ = 500	”
Membrane surface capacitance	c = 1.0	[μF/cm^2]
Longitudinal intra-cellular conductivity	$g_i^l = 1.741$	[mS/cm]
Transverse intra-cellular conductivity	$g_i^t = 0.1934$	”
Longitudinal extra-cellular conductivity	$g_e^l = 3.906$	”
Transverse extra-cellular conductivity	$g_e^t = 1.970$	”
Lung conductivity	0.5	”
Blood conductivity (ventricular cavities)	6.7	”
Remaining tissues conductivity	2.2	”

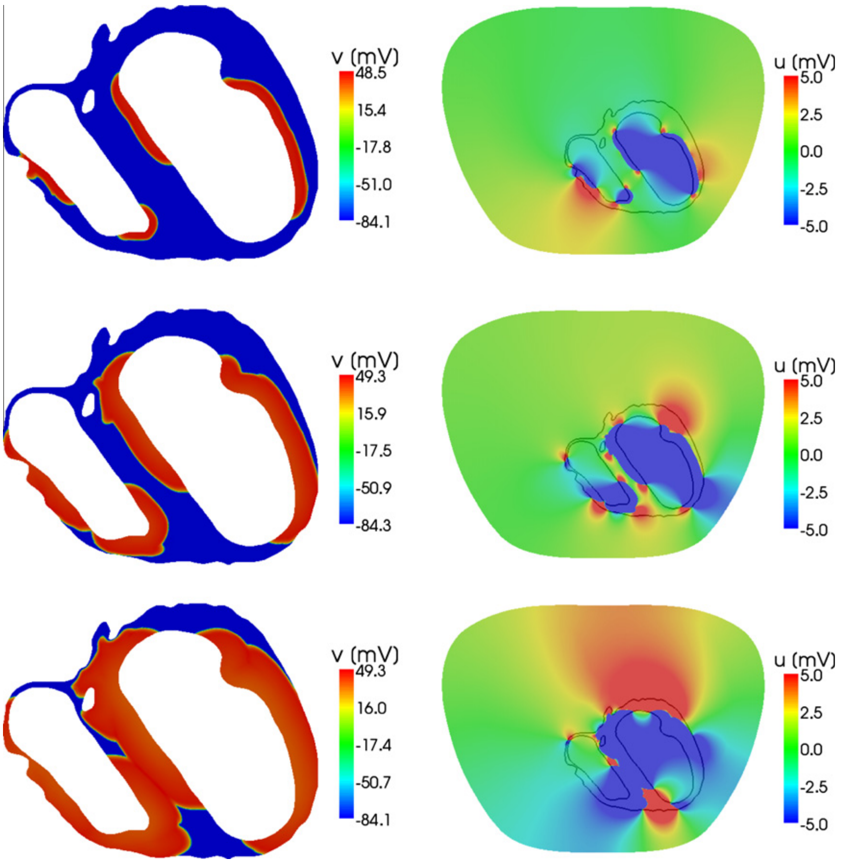


Fig. 2. 2D simulation. Left: depolarisation sequence of the heart, the transmembrane potential v is represented 15, 30 and 45 ms after stimulation. Right: associated potential u in the heart and in the extra cardiac region.

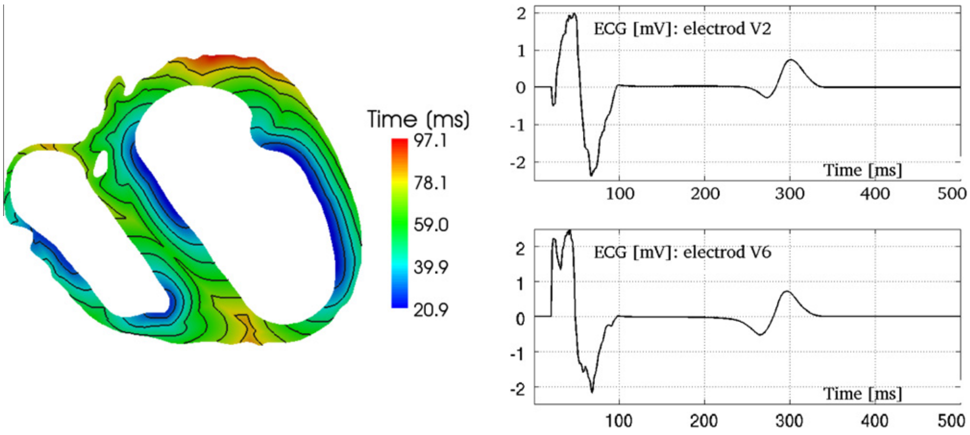


Fig. 3. 2D simulation. Left: activation time in the heart, isolines in black are separated by 10 ms. Right: ECG recordings, the extra-cardiac potential is recorded on the torso surface at two points located at electrodes V2 (above) and V6 (below), see Fig. 1 for the electrode location.

$\partial\Omega$, their location is depicted on Fig. 1 (points V1 to V6). These potentials $(u(t,Vi))_{i=1\dots6}$ are recorded at each time step along a complete cardiac cycle (including depolarisation and repolarisation). Results are depicted on Fig. 3 on the right for the two electrodes V2 and V6.

3D test case. We here consider a small slab of tissue: a cubic domain with one centimetre width ($\Omega = [0,1]^3$). A series of 5 meshes $(\mathfrak{M}_n)_{n=1\dots5}$ has been considered, from 500 to 1 250 000 vertices (see Table 2 for exact figures). The mesh size being

Table 2
Average number of iterations for one system inversion.

n	DOF(n)	Iter(n)		
		$\epsilon = 10^{-2}$	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$
(a) 2D case				
1	143 053	3.19	3.00	3.00
2	344 408	3.82	3.00	3.00
3	684 112	4.00	3.00	3.00
4	1 257 312	4.54	3.00	3.00
(b) 3D case				
		Iter(n)		
		$\epsilon = 10^{-1}$	$\epsilon = 10^{-2}$	$\epsilon = 10^{-3}$
1	497	2.40	2.00	2.00
2	3 220	4.03	2.79	2.76
3	22 256	5.14	3.00	3.00
4	162 981	7.43	3.24	3.00
5	1 253 910	11.20	3.96	2.00

divided by 2 from $\mathfrak{M}_n$ to $\mathfrak{M}_{n+1}$, the time stepping Δt also is divided by 2 and ranges from 0.2 to 0.0125 ms from the coarsest to the finest mesh. The heart is here considered as isolated: no torso T is involved as described in Section 2.2. The cardiac tissue anisotropy is set to be of orthotropic type, as defined in [27]. Muscular fibres are horizontal and independent of x and y. The fibre directions linearly rotate from $+\pi/4$ to $-\pi/4$ as z goes from 0 to 1. Orthotropic anisotropy represents the physiologically observed rotation of the cardiac fibres from $+\pi/4$ to $-\pi/4$ from the endo-cardium to the epi-cardium.

A depolarisation potential wave is simulated by applying a stimulation current at the centre of the domain during 1 ms.

The spreading of transmembrane depolarisation wave is depicted on Fig. 4. Activation time are here represented for three slices of the domain $\Omega = [0,1]^3$: $z = 0$, $z = 0.5$ and $z = 1$. Each slice corresponds to the endo-cardium, middle wall and epi-cardium respectively. The fibre angle with $\mathbf{e}_x$ is clearly visible on each slice: $+\pi/4$ for $z = 0$ (left), 0 for $z = 0.5$ (middle) and $-\pi/4$ for $z = 1$ (right).

6.2. Results

All figures and tables reported here have been obtained fixing a tolerance of 10^{-6} for the system (11) inversion; the residual being defined as $\|AX - Y\|/\|Y\|$in Euclidian vector norm. The hierarchical Cholesky decompositions for P_1 and P_K have been built for various values of the accuracy parameter ϵ introduced in Section 5.3. All computations were ran on a clustered platform with processor cores of type AMD Opteron, 2.3 GHz.

Number of iterations. We first investigate the cost for system (11) during the depolarisation sequence in terms of number of iterations Iter(n) for the PCG algorithm. As already developed in this section preamble, the global cost theoretically is in $O(\mathrm{Iter}(n)\mathrm{DOF}(n)\log(\mathrm{DOF}(n))^{\alpha})$.

The numerical results are reported in Table 2. In dimension 2, for $\epsilon = 10^{-2}$ Iter(n) globally is multiplied by 1.18 between the coarsest and the finest meshes when meanwhile the problem size is multiplied by almost 9. For $\epsilon \leqslant 10^{-3}$ Iter(n) remains constant. In dimension 3 Iter(n) increases very slowly: for $\epsilon = 10^{-2}$ (resp. 10^{-1}) it is multiplied by 2 (resp. 4.66) when the problem size is multiplied by more than 2,500; for $\epsilon = 10^{-2}$ it even decreases.

The very slow variation of Iter(n) with DOF(n) (when it is not constant) appears in good agreement with a $O(\log(\mathrm{DOF}(n))^{\beta})$ assumption ensuring almost linear complexity of the preconditioning global cost. It is unfortunately not possible to numerically estimate β from these results since log(log(DOF(n))) has a too small range of variation. To have a deeper insight on the behaviour of Iter(n) when it does not remain constant we instead consider the cost indicator DOF(n) × Iter(n). An almost lin-

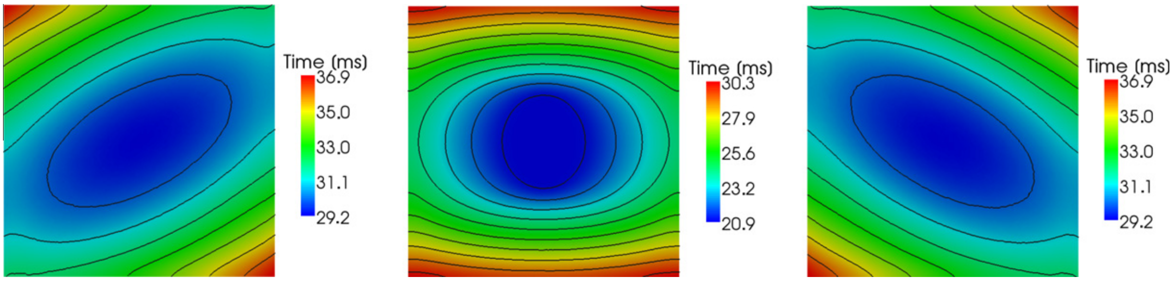


Fig. 4. 3D simulation. Activation times for three slices of the domain $\Omega = [0,1]^3$: $z = 0$, $z = 0.5$ and $z = 1$ from left to right. Isolines (in black) are separated by 1 ms.

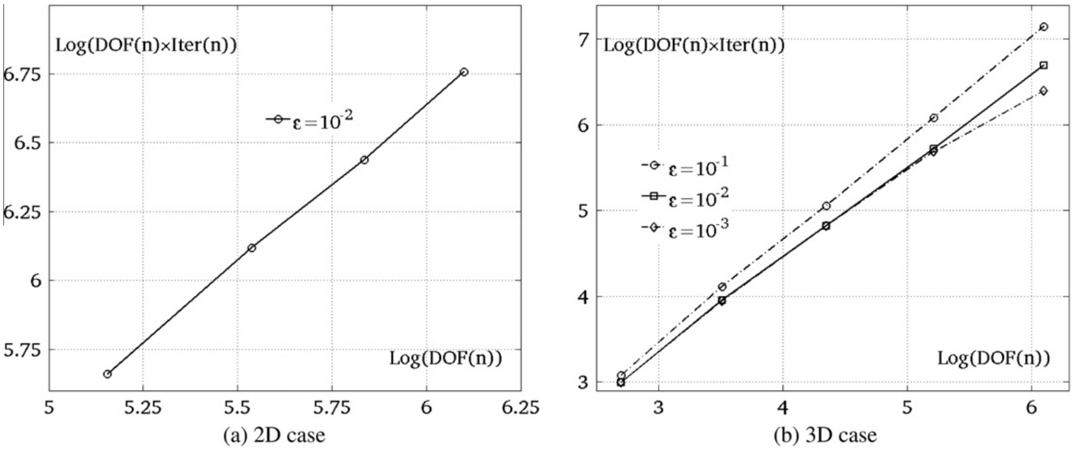
ear behaviour of this indicator is expected. It has been represented as a function of DOF(n) in decimal logarithmic scale on Fig. 5. In dimension 2 the curve has a global estimated slope of 1.15 using a linear least square best approximation. In dimension 3 the slopes have been estimated to 1.19, 1.07 and 1.0 for $\epsilon = 10^{-1}$, 10^{-2} and 10^{-3} respectively. Again, these results are in good agreement with the almost linear complexity assumption on the preconditioning.

CPU time consumption. The cost CPU(n) is reported in Table 3 (resp. Table 4) in dimension 2 (resp. 3) together with the logarithmic growth rate r_m of CPU(n) with respect to DOF(n) defined in (19). As for the iteration number, the behaviour of CPU(n) is clearer for the smallest values of ϵ. For $\epsilon \leqslant 10^{-3}$ (resp. $\epsilon \leqslant 10^{-2}$) in dimension 2 (resp. 3), r_n decreases with n and goes to 1 or even below 1.

The data in Tables 3 and 4 have been plotted on Fig. 6. The curve slopes have been estimated using a least square best linear approximation. In dimension 2 the slopes are of 1.17, 1.09 and 0.94 for $\epsilon = 10^{-2}$, 10^{-3} and 10^{-4} respectively. In dimension 3 they are of 1.27, 1.21 and 1.12 for $\epsilon = 10^{-1}$, 10^{-2} and 10^{-3} respectively (and neglecting the first data point).

Firstly, since r_n roughly decreases (starting with rates higher than 1.7 in dimension 3), these computed slopes indeed are upper-bounds on the complexity. Secondly CPU time is not a fully reliable cost measurement: because of cache effects memory-access differences for large-scale problems and because of the cluster load. For these two reasons we conclude that these CPU data are in good agreement with an almost linear complexity of the preconditioned system inversion, confirming the study of Iter(n).

PCG convergence rate. The convergence rate of the residual towards 0 for the preconditioned conjugate gradient algorithm has been measured in dimension 2 and 3 for the accuracy parameter set to $\epsilon = 10^{-3}$. The (decimal) logarithm of the



(a) 2D case

(b) 3D case

Fig. 5. Plot of DOF(n) × Iter(n) as a function of DOF(n) in (decimal) Log/Log Scale. Left: 2D case for $\epsilon = 10^{-2}$. Right: 3D case for the three values of $\epsilon = 10^{-2}$, 10^{-3} and 10^{-3}.

Table 3
CPU Time, 2D case. Left: averaged CPU time in seconds for one system inversion. Right: logarithmic growth of CPU(n)with respect to DOF(n).

n	DOF(n)	CPU(n)			r_n		
		$\epsilon = 10^{-2}$	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-2}$	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$
1	143 053	1.73	1.57	1.78	1.47	1.16	1.04
2	344 408	6.32	4.34	4.42	0.74	1.02	0.93
3	684 112	10.49	8.75	8.39	1.36	1.09	0.78
4	1 257 312	23.96	17.04	13.46			

Table 4
CPU Time, 3D case. Left: averaged CPU time in seconds for one system inversion. Right: logarithmic growth of CPU(n)with respect to DOF(n).

n	DOF(n)	CPU(n)			r_n		
		$\epsilon = 10^{-1}$	$\epsilon = 10^{-2}$	$\epsilon = 10^{-3}$	$\epsilon = 10^{-1}$	$\epsilon = 10^{-2}$	$\epsilon = 10^{-3}$
1	497	$2.0\ 10^{-3}$	$1.7\ 10^{-3}$	$1.8\ 10^{-3}$	1.75	1.70	1.70
2	3 220	$5.1\ 10^{-2}$	$4.1\ 10^{-2}$	$4.2\ 10^{-2}$	1.34	1.22	1.27
3	22 256	$6.9\ 10^{-1}$	$4.4\ 10^{-1}$	$4.9\ 10^{-1}$	1.27	1.19	1.21
4	162 981	8.6	4.6	5.5	1.22	1.25	0.86
5	1 253 910	102.96	59.8	32.2			

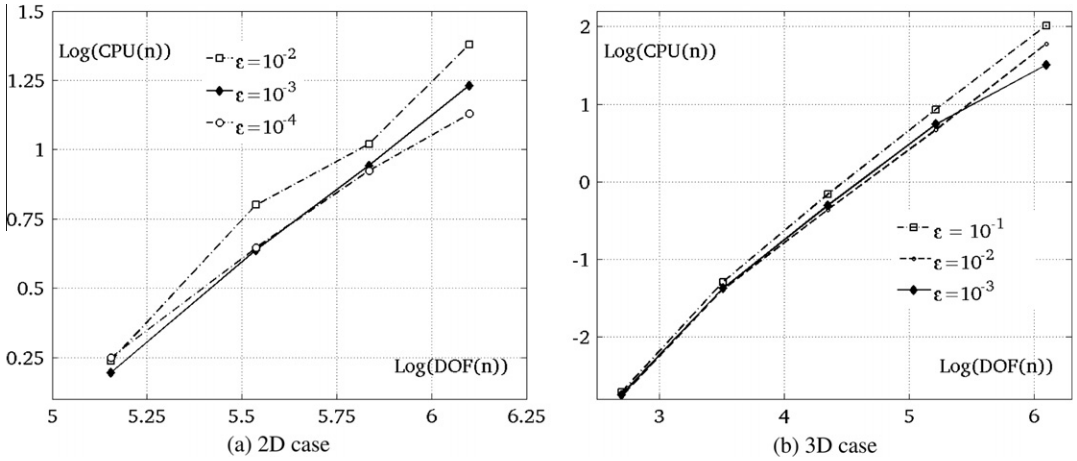
residual has been plotted as a function of the iteration number on Fig. 7 for the four considered meshes in dimension 2 and for 3 meshes in dimension 3. Due to the very small number of iterations needed, this convergence rate obviously is quite large.

In dimension 3, for the finest mesh $\mathfrak{M}_5$ with 1 250,000 vertices, the residual is divided by more than 150 at step one and by more than 75 at step 2. For the two other meshes, each PCG iteration divides the residual by at least 100.

In dimension 2, for all four meshes log(residual) displays the same global slope with respect to the number of iterations that is equal to 1.6. Globally the residual is divided by 40 at each time step. More precisely the residual is usually divided by 100 at the first step, by 30 at the second one and by 20 at the third one.

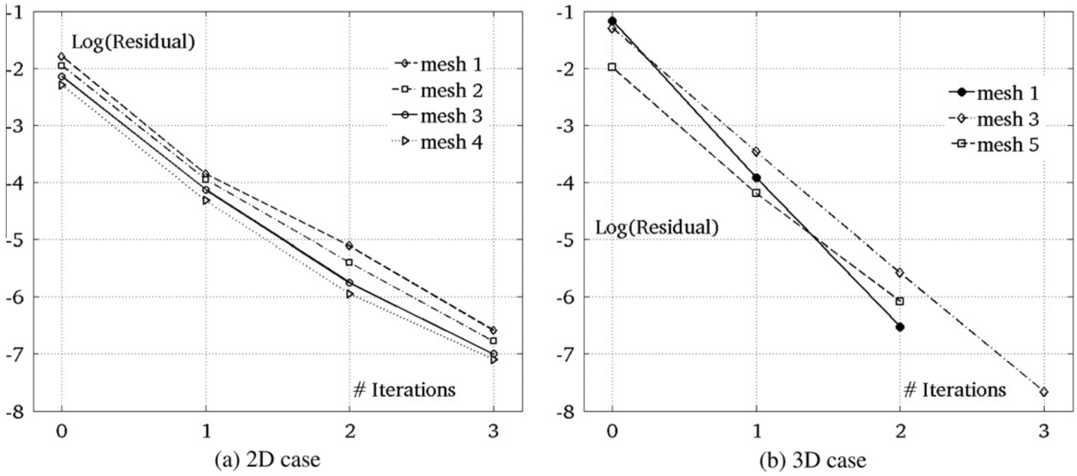
Cost calibration and profiling. Neither the CPU time nor the number of iterations actually provides an absolute evaluation for the preconditioning cost in the following sense. CPU time measurements are device dependent and the iteration number does not take into account the cost for the inversion of P_1 and P_K that may be large. These indicators are relevant and sufficient to evaluate the asymptotic complexity with DOF(n) but do not allow practical comparison with other techniques.

To address this question we proceed as follows. Firstly we consider the complete algorithm profiling: we measure the amount of time spent on each task (RHS computation, system inversion, normalisation ...) at each time step and average these durations along the depolarisation sequence. Secondly we compare the amount of time inside the PCG algorithm spent on the two predominant operations $X \mapsto P_{\Lambda}^{-1}X$ and $X \mapsto \Lambda X$. The ratio between these two times provides a calibration of the preconditioner P_{Λ} inversion cost in terms of matrix vector multiplication by Λ, which last operation has a fully established operational cost.



(a) 2D case
(b) 3D case

Fig. 6. Cost of one inversion of $\Lambda X = Y$ in terms of CPU Time as a function of the problem size in (decimal) Log/Log scale.



(a) 2D case
(b) 3D case

Fig. 7. PCG convergence rate. Convergence of the residual of the preconditioned system (11) as a function of the number of iterations. On both the 2D and the 3D cases, the preconditioner is set with $\epsilon = 10^{-3}$. Left, 2D case: convergence is shown for each of the four 2d meshes. Right, 3D case: convergence is depicted for the coarsest mesh (mesh 1), for the finest mesh (mesh 5) and on the intermediate mesh 3.

We point out that this ratio makes sense because of the almost linear complexity with DOF(n). Practically it varies sufficiently slowly with DOF(n) to derive a typical ratio for practically used problem size.

In dimension 2 (resp. 3), these typical figures are as follows:

- 70% (resp. 85%) of the whole computational effort is dedicated on the system (11) inversion,
- each operation $X \mapsto P_{\Lambda}^{-1}X$ has cost 15 (resp. 25) matrix-vector multiplication by Λ,
- considering an average number of iteration equal to 3, inverting $X \mapsto \Lambda^{-1}X$ has the same cost has 50 (resp. 80) matrix-vector multiplication by Λ.

6.3. Conclusion

We introduced in this paper a new preconditioning for the bidomain model based on an algebraic block-*LU* decomposition of its system matrix Λ and a heuristic approximation. The complexity for solving the preconditioned system $\Lambda X = Y$ with respect to the matrix size has been numerically analysed using both a 2D and a 3D test case and a hierarchical Cholesky preconditioning. This complexity has been numerically showed to be almost linear; which is optimal in this context (see discussion in Section 5.3).

We firstly would like to recall that the notion of complexity is not sufficient to compare algorithms in practise. The only certainty is that the resolution strategy presented here will *asymptotically* become more efficient than a second algorithm with worst complexity (as the problem size goes to infinity). Being fixed a problem, the second algorithm might be more efficient. The calibration and profiling provided in Section 6.2 might however help towards such comparisons and especially with the preconditioning developed in [16]. Firstly the data given in this paper do not indicate almost linear complexity. Precisely, CPU time data rather fit a complexity of 1.4 with the problem size. Despite the limitations on CPU time measurements we already mentioned, it is likely that this complexity is greater than 1. Iteration numbers also are reported (on a test case quite close to the 3D test case here on the mesh $\mathfrak{M}_4$) that are of order 6 with a flexible GMRes. Flexible GMRes performs m matrix-vector multiplications and preconditioner inversions per iteration with m the restart number, typically of order 25. This would mean 150 matrix-vector multiplications and preconditioner inversions. Each preconditioner inversion itself uses an *iLU*(0) PCG: thus one matrix-vector multiplication and one *iLU*(0) inversion per iteration. Even assuming a fast convergence of the PCG in a few steps, this may lead to a calibration of the cost in terms of matrix-vector multiplications several times larger than the one we obtained (equal to 80). The comparison of CPU times on the same case (almost the same processor has been used for the two papers) confirms this option.

We eventually would like to underline that almost linear complexity for the resolution of (11) does not mean almost linear complexity for the resolution of the bidomain model. Assuming for simplicity a linear dependence for the cost on the number of nodes, this still implies an h^{-d} dependence of the cost on the mesh size h and with d the dimension. Considering the global cost of the simulation and not only the cost of one inversion, this now leads to an $h^{-(d+1)}$ dependence of the cost on the mesh size. For instance, considering some precision criterion e based on the activation time, that is of order 1 with h as established in [28,33], the complexity for the bidomain model with respect to e also is of e^{-3} and e^{-4} in dimension 2 and 3 respectively.

Thus a linear dependence of one system inversion cost on the problem size still leads to really heavy global costs for this type of problems.

References

[1] L. Tung, A bidomain model for describing ischemic myocardial D-D properties, Ph.D. thesis, M.I.T. 1978.

[2] W. Krassowska, J. Neu, Homogenization of syncytial tissues, CRC Crit. Rev. Biomed. Eng. 21 (1993) 137–199.

[3] L. Ambrosio, P. Colli-Franzone, G. Savaré, On the asymptotic behaviour of anisotropic energies arising in the cardiac bidomain model, Interfaces Free Bound. 2 (2000) 213–266.

[4] P. Colli-Franzone, G. Savaré, Degenerate evolution systems modeling the cardiac electric field at micro- and macroscopic level, Evolution equations, semigroups and functional analysis 2002.

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

[6] A.J. Pullan, M.L. Buist, L.K. Cheng, Mathematically Modelling the Electrical Activity of the Heart, World Scientific Publishing, 2005.

[7] R. Clayton, O. Bernus, E. Cherry, H. Dierckx, F. Fenton, L. Mirabella, A. Panfilov, F. Sachse, G. Seemann, H. Zhang, Models of cardiac tissue electrophysiology: progress, challenges and open questions, Progress Biophys. Molecular Bio. 104 (2011) 22–48.

[8] Y. Bourgault, Y. Coudière, C. Pierre, Existence and uniqueness of the solution for the bidomain model used in cardiac electrophysiology, Nonlinear Anal.: Real World Appl. 10 (2009) 458–482.

[9] P. Colli Franzone, L. Pavarino, A parallel solver for reaction–diffusion systems in computational electrocardiology, Math. Models Methods Appl. Sci. 14 (2004) 883–911.

[10] M. Ethier, Y. Bourgault, Semi-implicit time discretization schemes for the bidomain model, SIAM J. Numer. Anal. 46 (2008) 2443–2468.

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

[12] J. Sundnes, B. Nielsen, K. Mardal, X. Cai, G. Lines, A. Tveito, On the computational complexity of the bidomain and the monodomain models of electrophysiology, Ann. Biomed. Eng. 34 (2006) 1088–1097.

[13] B. Nielsen, T. Ruud, G. Lines, A. Tveito, Optimal monodomain approximations of the bidomain equations, Appl. Math. Comput. 184 (2007) 276–290.

[14] L. Gerardo-Giorda, M. Perego, A. Veneziani, Optimized Schwarz coupling of bidomain and monodomain models in electrocardiology, M2AN 2010.

[15] L.F. Pavarino, S. Scacchi, Multilevel additive Schwarz preconditioners for the bidomain reaction–diffusion system, SIAM J. Sci. Comput. 31 (2008) 420–445.

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

[17] W. Hackbusch, Multigrid methods and applications, Springer Series in Computational Mathematics, vol. 4, Springer-Verlag, Berlin, 1985.

[18] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003, pp. xviii+528.

[19] V.V. Shaidurov, Some estimates of the rate of convergence for the cascadic conjugate-gradient method, Comput. Math. Appl. 31 (1996) 161–171.

[20] M. Benzi, Preconditioning techniques for large linear systems: a survey, J. Comput. Phys. 182 (2002) 418–477.

[21] W. Hackbusch, B.N. Khoromskij, Towards $\mathscr{H}$-matrix approximation of linear complexity 121 (2001) 194–220.

[22] S. Boerm, L. Grasedyck, W. Hackbusch, An introduction to hierarchical matrices with applications, Eng. Anal. Bound. 27 (2003) 405–422.

[23] L. Grasedyck, W. Hackbusch, Construction and arithmetics of $\mathscr{H}$-matrices, Computing 70 (2003) 295–334.

[24] L. Grasedyck, R. Kriemann, S. Le Borne, Parallel black box $\mathscr{H}$-LU preconditioning for elliptic boundary value problems, Comput. Vis. Sci. 11 (2008) 273–291.

[25] M. Boulakia, M.A. Fernández, J. Gerbeau, N. Zemzemi, A coupled system of PDEs and ODEs arising in electrocardiograms modeling, Appl. Math. Res. Express. AMRX 2 (2008) 28.

[26] S. Sanfelici, Convergence of the Galerkin approximation of a degenerate evolution problem in electrocardiology, Numer. Methods Partial Differ. Eq. 18 (2002) 218–240.

[27] P. Colli-Franzone, L. Pavarino, B. Taccardi, Simulating patterns of excitation, repolarization and action potential duration with cardiac Bidomain and Monodomain models, Math. Biosci. 197 (2005) 35–66.

[28] B. Andreianov, M. Bendahmane, K.H. Karlsen, C. Pierre, Convergence of DDFV schemes for the bidomain cardiac model, Networks and Heterogeneous Media 6 (2011) 195–240.

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

[30] M. Boulakia, S. Cazeau, M.A. Fernández, J. Gerbeau, N. Zemzemi, Mathematical modeling of electrocardiograms: a numerical study, Ann Biomed. Eng. 38 (2010) 1071–1097.

[31] J. Clements, J. Nenonen, P.K. Li, M. Horácek, Activation dynamics in anisotropic cardiac tissue via decoupling, Ann. Biomed. Eng. 32 (2004) 984–990.

[32] M. Potse, B. Dube, J. Richer, A. Vinet, R. Gulrajani, A comparison of monodomain and bidomain reaction–diffusion models for action potential propagation in the human heart, IEEE Trans. Biomed. Eng. 53 (2006) 2425–2435.

[33] C. Pierre, Y. Bourgault, Comparing the bidomain and monodomain models in electro-cardiology through convergence analysis, HAL Preprint <http://www.hal.archives-ouvertes.fr/hal-00545888/fr/> 2010.

[34] Z. Cai, J. Mandel, S. McCormick, The finite volume element method for diffusion equations on general triangulations, SIAM J. Numer. Anal. 28 (1991) 392–403.

[35] A. Quarteroni, A. Valli, Domain Decomposition Methods for Partial Differential Equations, Oxford Science Publications, 1999.

[36] C. Luo, Y. Rudy, A dynamic model of the cardiac ventricular action potential I. simulations of ionic currents and concentration changes, Circ. Res. 74 (1994) 1071–1096.

[37] P. Le Guyader, F. Trelles, P. Savard, Extracellular measurement of anisotropic bidomain myocardial conductivities. I. theoretical analysis, Annals Biomed. Eng. 29 (2001) 862–877.

[38] N. Smith, M. Buist, A. Pullan, Altered *t* wave dynamics in contracting cardiac model, J. Cardiovascular Electrophysio. 14 (2003) 5203–5209.

[39] O. Rousseau, Geometrical modeling of the heart., Ph.D thesis, University of Ottawa, 2010.

[40] O. Rousseau, Y. Bourgault, Heart segmentation with an iterative Chan-Vese algorithm, HAL Preprint <http://www.hal.archives-ouvertes.fr/hal-00403627/fr/> 2009.