

Parameter identification in cardiac electrophysiology using Proper Orthogonal Decomposition method

M. BOULAKIA¹, J-F. GERBEAU²

¹ Université Pierre et Marie Curie-Paris 6, LJLL, PARIS, F-75005 France

² INRIA Paris-Rocquencourt BP 105, F-78153 Le Chesnay Cedex, France

Abstract. We consider the problem of estimating some parameters (like ionic models or parameters involved in the initial stimulation) of a model of electrocardiograms (ECG) from the data of the Einthoven leads. This problem can be viewed as a first attempt to identify or to locate a pathology. The direct model is based on the bidomain equations in the heart and a Poisson equation in the torso and. To keep the computational time reasonable, the evaluation of the direct problem is approximated with a reduced order model based on Proper Orthogonal Decomposition (POD). The optimization problem is solved using a genetic algorithm. Numerical tests show that, with noisy synthetic data, the proposed procedure allows to recover ionic parameters and initial activation regions with a fair accuracy.

1 Introduction

The inverse problem of electrocardiology is usually addressed by reconstructing the epicardial potential from the body surface potential by solving the Cauchy problem for the Poisson equation in the torso with a suitable regularization (see [9] e.g.). The present study follows another route: assuming that a source model is available in the heart, we endeavor to estimate some parameters of this model from the body surface potential. The source model considered in this study is the one proposed in [2]. The inverse procedure is based on a genetic algorithm which evaluates a reduced order approximation of the direct problem. The reduced order approximation is built on the POD of the bidomain equations. We show that POD can successfully approximate the problem when some coefficients are perturbed, but fails for others. We propose a simple strategy to also handle those cases. The global strategy is illustrated through various numerical tests based on synthetic data.

2 Methods

2.1 Resolution of the direct problem

The electrical activity in the heart is modeled by the bidomain equations (see [8, 9] e.g.). We denote by Ω_H the heart domain, by u_e and V_m the extracellular

potential and the transmembrane potential and by σ_i and σ_e the intra- and extracellular conductivity tensors. Thus we have in Ω_H

$$\begin{aligned} A_m \left(C_m \frac{\partial V_m}{\partial t} + I_{\text{ion}}(V_m, w) \right) - \text{div}(\sigma_i \nabla V_m) - \text{div}(\sigma_i \nabla u_e) &= A_m I_{\text{app}} \\ - \text{div}((\sigma_i + \sigma_e) \nabla u_e) - \text{div}(\sigma_i \nabla V_m) &= 0, \end{aligned} \quad (1)$$

where A_m is the rate of membrane area per volume unit and C_m the membrane capacitance per area unit. The term $I_{\text{ion}}(V_m, w)$ represents the ionic current across the membrane which depends on V_m and w , and I_{app} is a given source function. The dynamics of the ionic variable w and the ionic current I_{ion} are described by the phenomenological two-variable model [5]. It is given by

$$\begin{aligned} I_{\text{ion}}(V_m, w) &= -\frac{w}{\tau_{\text{in}}} \frac{(V_m - V_{\text{min}})^2 (V_{\text{max}} - V_m)}{V_{\text{max}} - V_{\text{min}}} + \frac{1}{\tau_{\text{out}}} \frac{V_m - V_{\text{min}}}{V_{\text{max}} - V_{\text{min}}}, \\ \partial_t w = -g(V_m, w) &= -\begin{cases} \frac{w}{\tau_{\text{open}}} - \frac{1}{\tau_{\text{open}} (V_{\text{max}} - V_{\text{min}})^2} & \text{if } V_m < V_{\text{gate}}, \\ \frac{w}{\tau_{\text{close}}} & \text{if } V_m > V_{\text{gate}}, \end{cases} \end{aligned} \quad (2)$$

where τ_{in} , τ_{out} , τ_{open} , τ_{close} , V_{gate} , V_{min} , V_{max} are given parameters.

In the domain outside the heart region, denoted Ω_T , the electrical potential u_T is solution of the equation:

$$\text{div}(\sigma_T \nabla u_T) = 0, \quad \text{in } \Omega_T, \quad \sigma_T \nabla u_T \cdot \mathbf{n} = 0, \quad \text{on } \Gamma_{\text{ext}}, \quad (3)$$

where Γ_{ext} is the external boundary of the torso. The two problems are weakly coupled, following the procedure explained in [2]. Differences of potentials on standard points of Γ_{ext} define the ECG (Figure 1, left). For the sake of simplicity, another domain Ω_H , consisting of a parallelepiped, will also be considered in the following (Figure 1, right). The model is discretized in space with a finite element method and in time by combining a second order BDF implicit scheme with an explicit treatment of the ionic current.

2.2 Optimization method

Our objective is to identify the values of some parameters, like ionic parameters or parameters linked to the initial stimulation. To do so, we minimize a cost function corresponding to the norm of the difference between a reference ECG and the ECG obtained for a given value of the parameters. In this preliminary study, the reference ECGs are all ‘‘synthetic’’ which means that they are generated by the model itself. Let $n \in \mathbb{N}^*$ be the number of parameters and $\theta \in \mathbb{R}^n$ the vector of parameters we are looking for in a subset I of \mathbb{R}^n . The subset I is given by $I_1 \times \cdots \times I_n$ where I_j is an interval where the value θ_j is assumed to be. The following cost function is minimized

$$J(\theta) = \delta t \sum_{i=1}^{N_T} |V_I(t_i) - V_{I,\text{ref}}(t_i)|^2 + |V_{II}(t_i) - V_{II,\text{ref}}(t_i)|^2 + |V_{III}(t_i) - V_{III,\text{ref}}(t_i)|^2$$

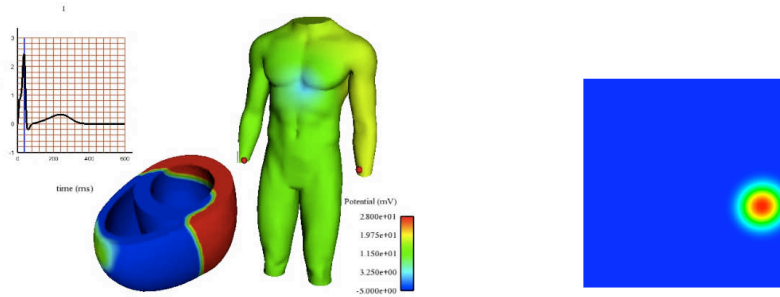


Fig. 1. Two representative pictures of the simulations run in this study. Left: heart-torso coupling and Einthoven I lead ECG. Right: schematic configuration used for the identification of the initial activation point

with respect to $\theta \in I$, where V_I , V_{II} and V_{III} are the three Einthoven leads given by the simulation for the value θ of the parameters and $V_{I,\text{ref}}$, $V_{II,\text{ref}}$ and $V_{III,\text{ref}}$ are the Einthoven leads of the reference ECG. An alternative in the definition of J is to consider the twelve standard leads instead of the three Einthoven leads. Numerical tests give similar results for these two choices of cost function.

The optimization problem is solved using a genetic algorithm (we refer for example to [3]). This algorithm is a global optimization method which consists of following the evolution of a population of N_p elements $(\theta_1, \dots, \theta_{N_p}) \in I^{N_p}$ corresponding to a set of values of the parameters. The population is regenerated N_g times, where N_g corresponds to the number of generations. At each generation, J is evaluated for each element of the population and the population evolves from a generation to another following three stochastic principles: selection (promote the elements of the population whose value by J is small), crossover (create from two elements of the population two new elements by doing a random barycentric combination of them), mutation (replace an element of the population by a new one randomly chosen in its neighborhood). To speed up this algorithm, many evaluations of J are performed using a surrogate model. This model consists of approximating the value of J by a Radial Basis Functions interpolation based on previously computed exact evaluations. The total number N_{ex} of exact evaluations is fixed and the number of exact evaluations decreases at each generation.

2.3 POD Reduced-Order Modelling

The genetic algorithm presented in the previous section offers many advantages: it can easily be run in parallel and it does not need the gradient of the cost function. Its main flaw is to require a large number of evaluations of the direct problem, even if many evaluations are avoided with the surrogate model strategy. To keep the computational time reasonable, we propose to use a reduced order model based on POD in the optimization loop.

Notions about the POD method. POD is a method to derive reduced models by projecting the system onto subspaces spanned by a basis of elements that contains the main features of the expected solution. We briefly recall this method here and refer the reader interested by more details to [4, 7] for example. To generate the POD basis associated with a precomputed solution u of an approximated Galerkin problem, we make a first numerical simulation (or set of simulations) and keep some snapshots $u(t_k)$, $1 \leq k \leq p$. Then a singular value decomposition (SVD) of the matrix $B = (u(t_1), \dots, u(t_p)) \in \mathbb{R}^{N,p}$ is performed: $B = USV'$, where $U \in \mathbb{R}^{N,N}$ and $V \in \mathbb{R}^{p,p}$ are orthogonal matrices, $S \in \mathbb{R}^{N,p}$ is the matrix of the singular values ordered by decreasing order, and $N \geq p$ is the dimension of the Galerkin basis of the finite element method.

The N_{modes} first POD basis functions $\{\Psi_i\}_{1 \leq i \leq N_{\text{modes}}}$ are then given by the N_{modes} first columns of U and the POD Galerkin problem is solved by looking

$$\text{for a solution of the type } u = \sum_{i=1}^{N_{\text{modes}}} \alpha_i(t) \Psi_i.$$

The $N \times N$ sparse system of the finite element method is thus replaced by a full system of size $N_{\text{modes}} \times N_{\text{modes}}$ with the POD method. To give a rough idea, it is generally possible to get a good accuracy for the problems at hand with $N_{\text{modes}} \approx 100$. With the time scheme used in this work, the matrix is constant over the time, since all the nonlinearities are treated explicitly. The matrix is therefore projected on the POD basis and factorized only once at the beginning of the computation. As a consequence, for the simulations presented in this paper, the reduced order model resolution is about one order of magnitude faster than the full order one.

Strategies to handle parameter perturbations To apply the POD method in parameter identification problems, a critical difficulty has to be faced: a POD basis generated from a solution obtained with a given set of parameters may be inaccurate to approximate a solution obtained from another set of parameters. The issue of the stability of a POD basis with respect to parameters perturbation is still the topic of active researches. The approaches proposed below are simple, but quite efficient for the considered problems. More sophisticated strategies, like interpolation of POD bases [1], could be considered in the future.

Assume that the parameters θ lie in $I \subset \mathbb{R}^n$. The simplest method is the following:

- **M1** : Only one POD basis is generated for a value $\theta_0 \in I$. This POD basis is then used for any $\theta \in I$.

As will be illustrated in section 3.1, approach **M1** proves to be satisfactory for some parameters (τ_{close} for example) but may be extremely inaccurate in other cases. To make the POD basis more robust with respect to parameter perturbation, two other methods are considered:

- **M2** : Many POD bases are computed “off-line” for different values of θ taken in a finite subset \mathcal{A} of I . Next, for an arbitrary value $\theta \in I$, the POD basis corresponding to the closest value of $\theta \in \mathcal{A}$ is used.

- **M3** : POD bases are computed from the combination of several simulations. More precisely, let us introduce $\mathcal{A} = \{\mathcal{A}_1, \dots, \mathcal{A}_m\}$ where \mathcal{A}_l , $1 \leq l \leq m$ is a finite subset of I of cardinal k_l . For each l , for each value $\theta_i \in \mathcal{A}_l$, $1 \leq i \leq k_l$, numerical simulations are run “off-line” and the snapshots $u_{l,i}(t_k)$, $1 \leq k \leq p$ are stored. Then, these snapshots are gathered into the matrix $B_l = (u_{l,1}(t_1), \dots, u_{l,1}(t_p), u_{l,2}(t_1), \dots, u_{l,2}(t_p), \dots, u_{l,k_l}(t_1), \dots, u_{l,k_l}(t_p))$ which is used to generate the POD basis as before. Then, for each $\theta \in I$, the POD basis obtained for the closest set of parameters is used.

Method **M1** is of course the cheapest, but the choice of θ_0 is somehow arbitrary and this approach is too crude in some configurations. In the two other methods **M2** and **M3**, the role of the finite set \mathcal{A} is to reasonably sample the whole set $I = I_1 \times \dots \times I_n$ in order to capture different behaviors of the solutions. In practice, a grid can be defined on the parameters space. Then, for **M2**, the POD bases are precomputed on the vertices of the grid, whereas for **M3**, the POD bases are precomputed by element using the snapshots corresponding the vertices of the elements of the parameter grid.

3 Results

3.1 Two illustrative examples of POD simulations

The accuracy of the POD approximation of the bidomain equations will only be evaluated on the ECG corresponding to the solution, since this is the only information used in the optimization loop.

Let us first consider the perturbation of parameter τ_{close} which corresponds to the characteristic closing time of ionic channels. In our model, the heart is divided in four regions where this parameter takes four different constant values. We focus on the value in the epicardium of the left ventricle $\tau_{\text{close}}^{\text{epi}}$, and in the right ventricle $\tau_{\text{close}}^{\text{RV}}$. A POD basis of 80 vectors is first constructed with $(\tau_{\text{close}}^{\text{epi}}, \tau_{\text{close}}^{\text{RV}}) = (80, 80)$. This basis is sufficient to get an excellent accuracy if the *same* experiment is run with the reduced order model. More interestingly, it still gives quite good results when $(\tau_{\text{close}}^{\text{epi}}, \tau_{\text{close}}^{\text{RV}})$ are significantly modified. For example, Figure 2 (left) shows a comparison of the ECG get with the full and the reduced models corresponding to $(\tau_{\text{close}}^{\text{epi}}, \tau_{\text{close}}^{\text{RV}}) = (90, 120)$. The QRS is in excellent agreement, the T-wave is slightly underestimated, which is not surprising since τ_{close} mainly affects the repolarization phase. It is interesting to note that for the values $(\tau_{\text{close}}^{\text{epi}}, \tau_{\text{close}}^{\text{RV}}) = (80, 80)$, the T-wave of the ECG is negative whereas it is positive with $(90, 120)$. It is therefore particularly satisfactory to obtain the correct T-wave orientation with the reduced order model after perturbing these coefficients.

Unfortunately, the accuracy obtained with a POD basis may get worse when other parameters are modified. This is the case for example with the parameter governing the initial activation: if we use a POD basis obtained from a reference simulation with an initial activation in the septum to run a reduced order simulation with an initial activation at the apex, the results are totally wrong, as shown in Figure 2 (right).

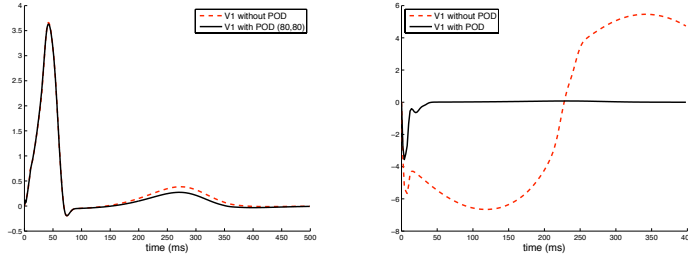


Fig. 2. Left: First leads of the ECGs with $(\tau_{\text{close}}^{\text{epi}}, \tau_{\text{close}}^{\text{RV}}) = (90, 120)$. Comparison of the ECGs obtained with the full model and with a POD basis generated with $(\tau_{\text{close}}^{\text{epi}}, \tau_{\text{close}}^{\text{RV}}) = (80, 80)$. Right: First leads of ECGs obtained for an initial stimulation in the apex. Comparison of the ECGs with the full model and with the POD basis obtained from a simulation with an initial activation on the septum.

3.2 Identification of two ionic parameters

Based on the first three leads of the ECG, we wish to estimate $\tau_{\text{close}}^{\text{epi}}$ and $\tau_{\text{close}}^{\text{RV}}$. In [2, Figure 24], it is shown that the ECG is quite sensitive to τ_{close} . The reference ECG used in the cost function is obtained from a first numerical simulation with $(\tau_{\text{close}}^{\text{epi}}, \tau_{\text{close}}^{\text{RV}}) = (90, 120)$ without using POD. Our goal is to test whether our identification algorithm based on POD and genetic algorithm is able to find these values out.

The parameters $\theta = (\tau_{\text{close}}^{\text{epi}}, \tau_{\text{close}}^{\text{RV}})$ are searched for in the set $[50, 150] \times [50, 150]$. The “exact” evaluations required by the optimization algorithm are based on the reduced order model using the 80 first POD modes. The results are summarized in Table 1 where the genetic algorithm is run with $N_p = 25$, $N_g = 15$ and $N_{\text{ex}} = 150$ (“Population 1”, top), and with $N_p = 90$, $N_g = 15$ and $N_{\text{ex}} = 600$ (“Population 2”, bottom).

For **M2**, we use $\mathcal{A}_1 = \{50; 60; 70; 80; 90; 100; 110; 120; 130; 140; 150\}$ and $\mathcal{A}_2 = \{50; 70; 90; 110; 130; 150\}$. Since the outcome of the genetic algorithm depends on the initial random population, the results presented correspond to a mean value of several runs. The relative error is defined by $\max\left(\frac{|\tau_{\text{close}}^{\text{epi}} - 90|}{90}, \frac{|\tau_{\text{close}}^{\text{RV}} - 120|}{120}\right)$. Table 2 shows the results obtained with Population 1 when the synthetic ECG used as reference is perturbed by a 1% or 4% noise.

3.3 Identification of the initial stimulation

In this test, the computational domain is a parallelepiped of size $[-2.5, 2.5] \times [-2.5, 2.5] \times [0, 0.2]$ (Figure 1, right). The initial activation is applied on $[c_1 - 0.2, c_1 + 0.2] \times [c_2 - 0.2, c_2 + 0.2] \times [0, 0.2]$. Our goal is to identify the location (c_1, c_2) in $[-2.5, 2.5] \times [-2.5, 2.5]$. The standard ECG leads are replaced by the differences of potential u_e evaluated in three points of the boundary. The reference pseudo-ECG has been obtained with $(c_1, c_2) = (1.8, -0.6)$. The error

| Population 1 | $(\tau_{\text{close}}^{\text{epi}}, \tau_{\text{close}}^{\text{RV}})$ | Relative error (in %) | Value of the cost function |
|---|---|--------------------------|-------------------------------|
| M1 with $\theta_0 = (100, 100)$ | (88.4, 122) | 1.8 | 0.23 |
| M1 with $\theta_0 = (150, 50)$ | (89.6, 127.1) | 5.9 | 6.4 |
| M2 with $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_1$ | (90.5, 120.5) | 0.6 | 0.08 |
| M2 with $\mathcal{A} = \mathcal{A}_2 \times \mathcal{A}_2$ | (86.9, 117.5) | 3.4 | 0.9 |

| Population 2 | $(\tau_{\text{close}}^{\text{epi}}, \tau_{\text{close}}^{\text{RV}})$ | Relative error (in %) | Value of the cost function |
|---|---|--------------------------|-------------------------------|
| M1 with $\theta_0 = (100, 100)$ | (87.7, 121.6) | 2.6 | 0.22 |
| M1 with $\theta_0 = (150, 50)$ | (85.7, 124.2) | 4.8 | 5.78 |
| M2 with $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_1$ | (90.2, 120.2) | 0.2 | 0.07 |
| M2 with $\mathcal{A} = \mathcal{A}_2 \times \mathcal{A}_2$ | (89.8, 119.7) | 0.2 | 0.36 |

Table 1. Identification of $(\tau_{\text{close}}^{\text{epi}}, \tau_{\text{close}}^{\text{RV}})$ (Reference value (90, 120)).

| | $(\tau_{\text{close}}^{\text{epi}}, \tau_{\text{close}}^{\text{RV}})$ | Value of the cost function |
|-----------------------|---|----------------------------|
| Noise of intensity 1% | (89.6, 119.7) | 0.12 |
| Noise of intensity 4% | (90.2, 119.9) | 0.8 |

Table 2. Identification of $(\tau_{\text{close}}^{\text{epi}}, \tau_{\text{close}}^{\text{RV}})$ using **M2** with \mathcal{A}_1 (Reference value (90, 120)).

is defined by $\sqrt{(c_1 - 1.8)^2 + (c_2 + 0.6)^2}$. In Table 3 columns 2 and 3, the parameters of the genetic algorithm are $N_p = 25$, $N_g = 12$ and $N_{ex} = 150$. In Table 3 columns 4 and 5, the same test is done with a larger population: $N_p = 100$, $N_g = 15$ and $N_{ex} = 1000$. For **M2**, $\mathcal{A}_3 = \{-2.5; -2; -1.5; -1; -0.5; 0; 0.5; 1; 1.5; 2; 2.5\}$ and $\mathcal{A}_4 = \{-2; -1; 0; 1; 2\}$. For **M3**, $\mathcal{A} = \{\mathcal{A}_l, 1 \leq l \leq 36\}$ where \mathcal{A}_l contains the four vertices of the element l of the cartesian grid $\{-2.5; -2; -1; 0; 1; 2; 2.5\} \times \{-2.5; -2; -1; 0; 1; 2; 2.5\}$. For this simple geometry (about 5000 nodes), the direct simulation is not very time-consuming. It is therefore possible to solve the inverse problem with the full order model rather than with the POD.

| | (c_1, c_2) | Error | (c_1, c_2) | Error |
|---|---------------|-------|---------------|-------|
| M1 with $x_0 = (0, 0)$ | (-1.6, -1.6) | 3.5 | (-1.5, -1.7) | 3.5 |
| M2 with $\mathcal{A} = \mathcal{A}_3 \times \mathcal{A}_3$ | (1.71, -0.54) | 0.11 | (1.81, -0.66) | 0.06 |
| M2 with $\mathcal{A} = \mathcal{A}_4 \times \mathcal{A}_4$ | (1.92, -0.59) | 0.12 | (2.07, -0.54) | 0.28 |
| M3 | (1.96, -0.61) | 0.16 | (1.97, -0.62) | 0.17 |
| without POD | (1.57, -0.56) | 0.23 | (1.74, -0.51) | 0.11 |

Table 3. Identification of (c_1, c_2) (Reference value (1.8, -0.6)). Columns 4 & 5 correspond to a larger population in the genetic algorithm than columns 2 & 3.

4 Discussion and conclusion

In paragraph 3.2, we saw that the simplest method **M1** allows to obtain satisfactory results, but the results depend on the arbitrary value θ_0 and the error does not seem to quickly decrease when the population goes up. Results seem more robust with **M2**. In paragraph 3.3, when the initial activation parameters vary, method **M1** is inadequate. This obviously comes from the fact that, as noticed in paragraph 3.1, POD is unable to approximate the direct problem in this case. The strategies proposed in **M2** or **M3** allow to control the strong dependence of the solution with respect to the initial stimulation. It is interesting to notice that the results with **M2** or **M3** are quite comparable to the ones obtained without the POD method. Thus, the accuracy of the POD seems to be reasonably consistent with the accuracy that can be expected from the genetic algorithm.

In conclusion, these preliminary results are promising. Nevertheless, the method still has to be tested on real data. Moreover, difficulties can be expected when increasing the number of parameters. In that case, an efficient strategy to sample the parameters space will be critical to keep using the proposed strategy of approximation by POD.

Acknowledgment: The authors wish to thank Laurent Dumas who provided them with his optimization algorithm and Charbel Farhat and David Amsallem for valuable discussions about model reduction.

References

1. D. Amsallem and C. Farhat. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA Journal-American Institute of Aeronautics and Astronautics*, 46(7):1803–1813, 2008.
2. M. Boulakia, S. Cazeau, M.A. Fernández, J.-F. Gerbeau, and N. Zemzemi. Mathematical modeling of electrocardiograms: a numerical study. *Ann. Biomed. Eng.*, 38(3):1071–1097, 2010.
3. D.E. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.
4. K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische Mathematik*, 90(1):117–148, 2001.
5. C.C. Mitchell and D.G. Schaeffer. A two-current model for the dynamics of cardiac membrane. *Bulletin Math. Bio.*, 65:767–793, 2003.
6. A. Quarteroni, R. Sacco, and F. Saleri. *Numerical mathematics*, volume 37 of *Texts in Applied Mathematics*. Springer-Verlag, Berlin, second edition, 2007.
7. M. Rathinam and L.R. Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 41(5):1893–1925, 2004.
8. F.B. Sachse. *Computational Cardiology: Modeling of Anatomy, Electrophysiology, and Mechanics*. Springer-Verlag, 2004.
9. J. Sundnes, G.T. Lines, X. Cai, B.F. Nielsen, K.-A. Mardal, and A. Tveito. *Computing the electrical activity in the heart*. Springer-Verlag, 2006.