

# 1 The monodomain model

The monodomain equations are given by

$$\frac{\partial \mathbf{s}}{\partial t} = \mathbf{F}(\mathbf{s}, v), \quad \mathbf{x} \in H, \quad (1)$$

$$\frac{\partial v}{\partial t} + I_{ion}(v, \mathbf{s}) = \nabla \cdot (\mathbf{M} \nabla v) + I_s, \quad \mathbf{x} \in H, \quad (2)$$

$$\mathbf{n} \cdot (\mathbf{M} \nabla v) = 0, \quad \mathbf{x} \in \delta H, \quad (3)$$

with  $v(\mathbf{x}, t)$  the transmembrane potential (in mV),  $H$  the domain,  $\delta H$  the boundary of  $H$ ,  $\mathbf{n}$  the outward pointing normal of the boundary, and with  $I_s$  the prescribed input current (in mV/ms) and  $I_{ion}$  the ionic current across the membrane (in mV/ms), both scaled by the cell membrane capacitance (in  $\mu\text{F}/(\text{mm}^2)$ ). Equation (1) is a system of ODE's that models the membrane dynamics. There exist many different cell membrane dynamics models with varying degrees of complexity that can be used to specify  $I_{ion}$ ,  $\mathbf{F}(\mathbf{s}, v)$  and the state variables  $\mathbf{s}(\mathbf{x}, t)$ , see the CellML repository [9] for an overview of different types of models.

Finally,  $\mathbf{M}$  is a conductivity tensor (in  $\text{mm}^2/\text{ms}$ ), that satisfies

$$\mathbf{M} = \frac{\lambda}{1 + \lambda} \mathbf{M}_i, \quad (4)$$

with  $\mathbf{M}_e = \lambda \mathbf{M}_i$ . Here,  $\mathbf{M}_e$  and  $\mathbf{M}_i$  are the extracellular and intracellular conductivities (in  $\text{mm}^2/\text{ms}$ ), divided by the product of the membrane capacitance (in  $\mu\text{F}/(\text{mm}^2)$ ) and the cell membrane area-to-volume ratio (in  $1/\text{mm}$ ). By assuming that there exists a  $\lambda$  such that  $\mathbf{M}_e = \lambda \mathbf{M}_i$  the monodomain equations can be derived from the more complicated bidomain equations [3].

## 2 A basic test case

For our test case, we take a square of  $5 \text{ mm} \times 5 \text{ mm}$  as our domain  $H$ . We will use the Grandi cell model to model the membrane kinetics [1]. We solve our test case with the `splittingsolver` module from the `cbcbeat` Python package [10]<sup>1</sup> with the default parameter values and default initial conditions for  $v$  and  $s$ . The `splittingsolver` solves the monodomain PDE system and its coupled cell membrane dynamics ODE system separately, using the operator splitting scheme as described in [3]. We take typical values  $\sigma_l = 0.1$

---

<sup>1</sup>This electrophysiology solver package is based on the FEniCS Project software [7] and the dolfin-adjoint software [8]

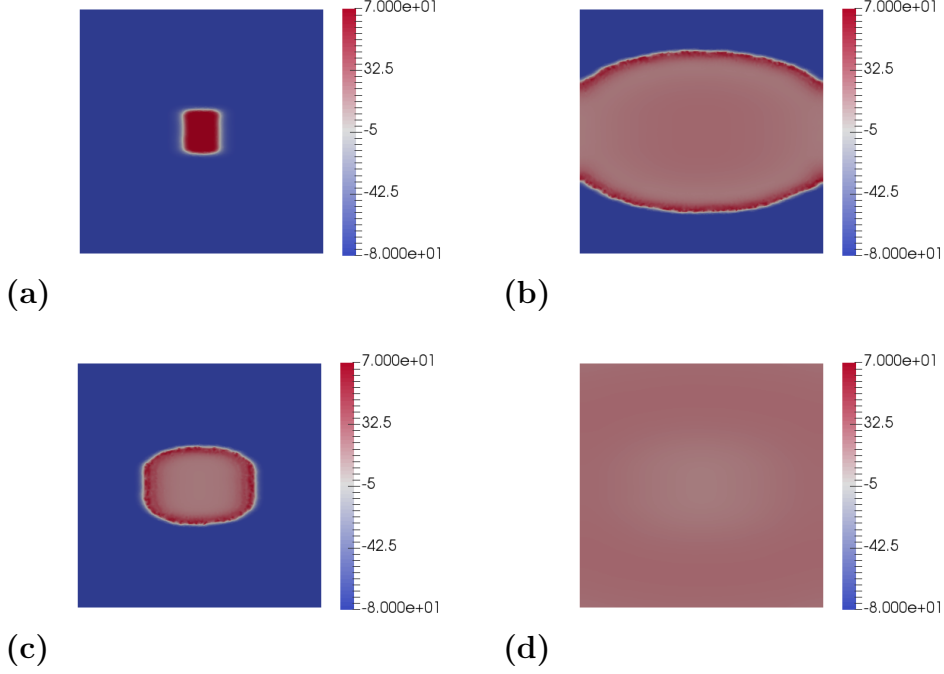


Figure 1: Heat maps of  $v$  (in mV) of our basic test case at **(a)**  $t = 5\text{ms}$ , **(b)**  $t = 15\text{ms}$ , **(c)**  $t = 40\text{ms}$  and **(d)**  $t = 80\text{ms}$ .

and  $\sigma_t = 0.02$  (in mS/mm) for the longitudinal and tangential conductivity respectively and take  $C_m = 0.2$  for the membrane capacitance (in  $\mu\text{F}/(\text{mm}^2)$ ) and  $\beta = 200$  for the cell membrane area-to-volume ratio (in  $1/\text{mm}$ ) [4], data from [5, 6]. We apply a stimulus of 10 mV/ms over  $1 \text{ mm}^2$  in the centre of the square from  $t = 0$  to  $t = 3 \text{ ms}$ . In Figure 1, we show a heat map of  $v$  at  $t = 5, 15, 40$  and  $80 \text{ ms}$ . In Figure 2, we show a heat map of  $Ca_{sl}$  at the same times. This variable is one of the 37 state variables  $\mathbf{s}$  of the Grandi cell model and measures the intracellular calcium concentration [1]. In Figures 3 and 4, we plotted  $v$  and  $Ca_{sl}$  over time, both at a point at the centre of the domain (in blue) and at a point at the left lower edge (in green). We see that the ...

### 3 The inverse problem

It is possible to obtain measurements  $u_{\text{obs}}$  of the transmembrane potential and measurements  $c_{\text{obs}}$  of the intracellular calcium concentration  $Ca_{sl}$  over the whole domain  $H$  at discrete points in time. With those measurements, we can estimate the value of the parameters in our model, using an adjoint-

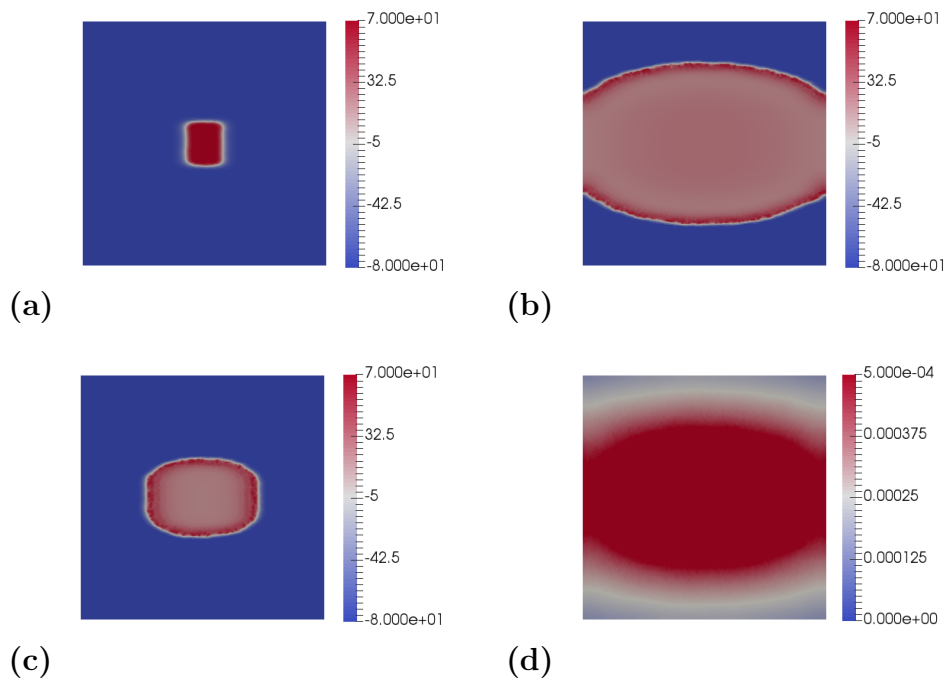
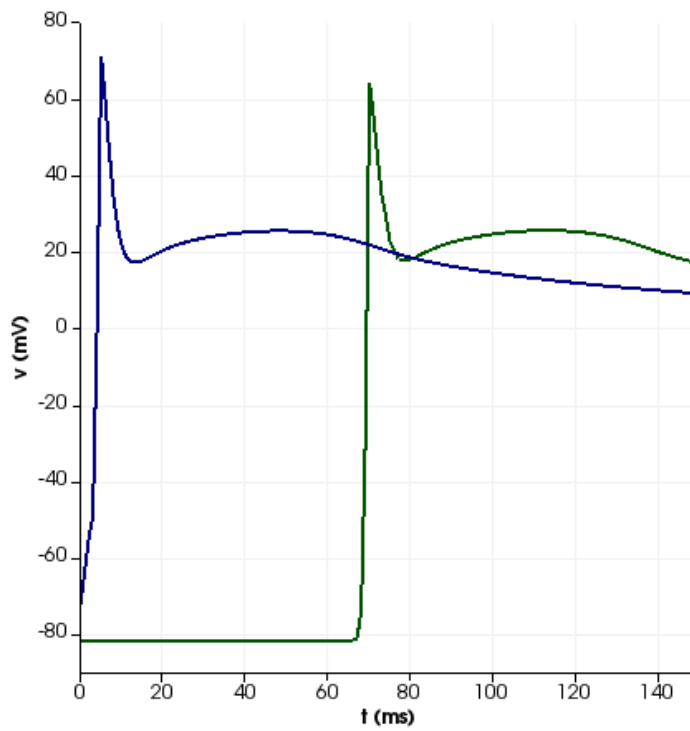
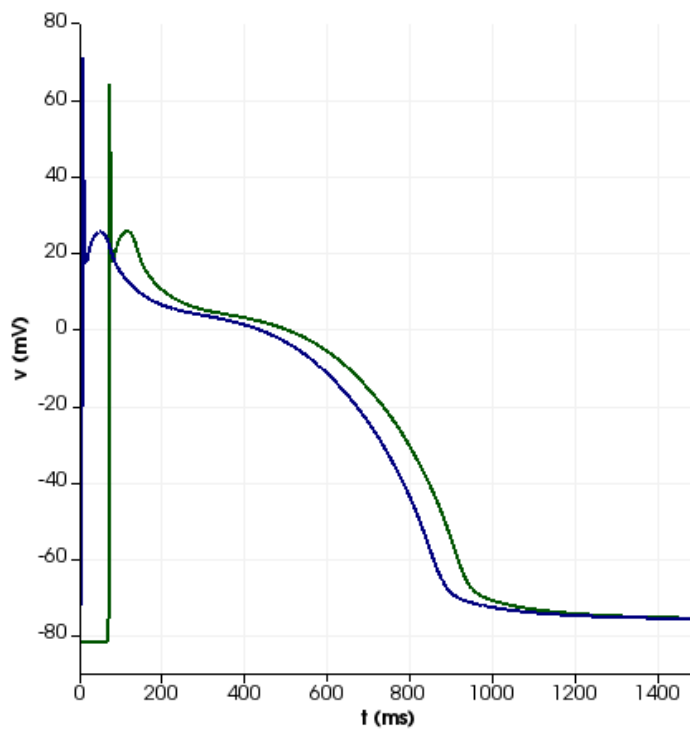


Figure 2: Heat maps of  $Ca_{sl}$  of our basic test case at (a)  $t = 5\text{ms}$ , (b)  $t = 15\text{ms}$ , (c)  $t = 40\text{ms}$  and (d)  $t = 80\text{ms}$ .

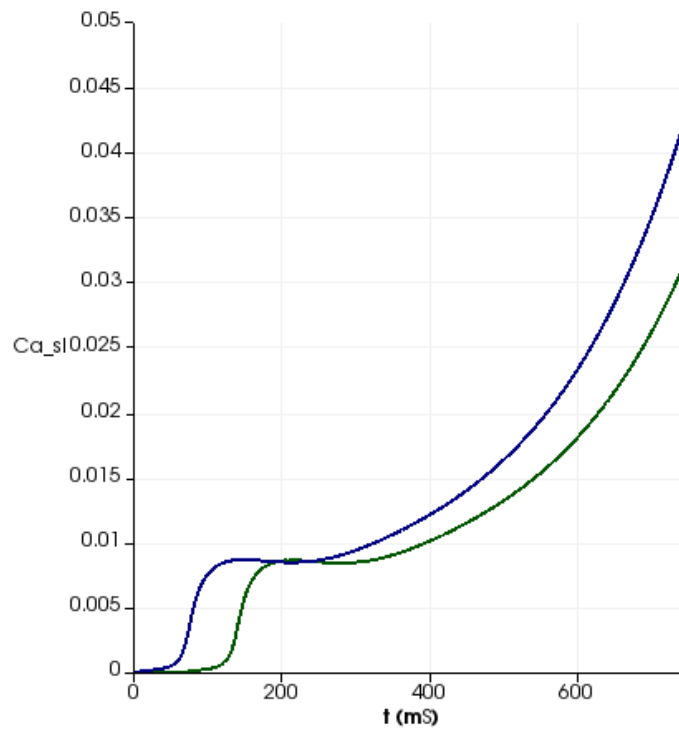


(a)

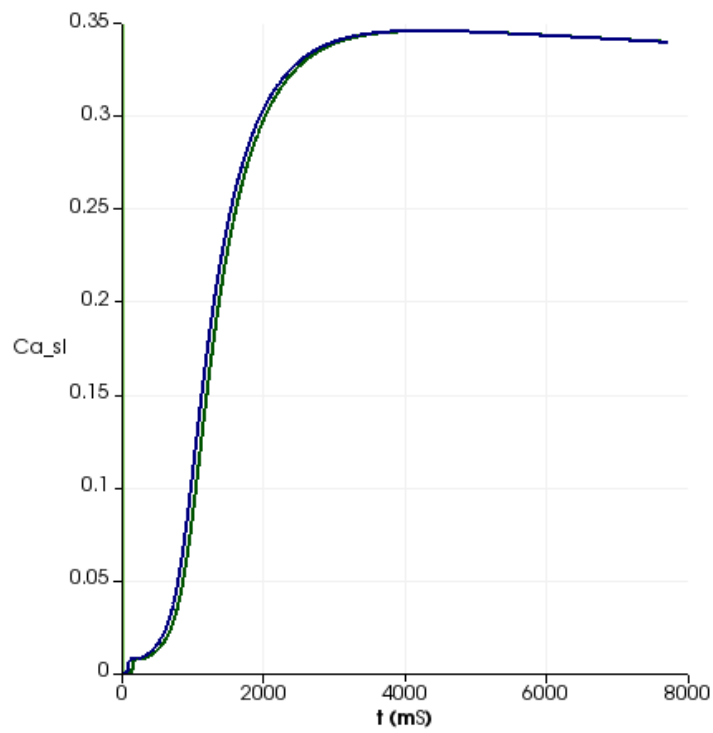


(b)

Figure 3: Plots of  $v$  against  $t$ .



(a)



(b)

Figure 4: Plots of  $Ca_{sl}$  against  $t$ .

based approach.<sup>2</sup> Here, as an example, we will try to estimate the values of the  $\sigma_l$  parameter from the Grandi cell model. We can formulate this problem as an optimisation problem: find  $\sigma_l$ , such that the functional

$$\mathcal{J}(v, c, \sigma_l) = \frac{1}{N} \sum_{i=1}^N \|v - v_{\text{obs}}(t_i)\|_{L^2(H)}^2 + \|c - c_{\text{obs}}(t_i)\|_{L^2(H)}^2, \quad (5)$$

is minimized, subject to the requirements that  $v, c$  and  $\sigma_l$  satisfy the state system of equations (1)-(3) and initial conditions  $v(\mathbf{x}, 0) = v_0(\mathbf{x})$  and  $\mathbf{s}(\mathbf{x}, 0) = \mathbf{s}_0(\mathbf{x})$ . Here,  $N$  are the number of measurements in time and  $t_i, i = 1, \dots, N$  the respective moments in time. Using `cbceat` and the `dolfin-adjoint` software on which it is based, we can automatically compute the total derivative of  $\mathcal{J}$  with respect to the optimization parameter  $\sigma_l$ . We can then use the `scipy` optimisation algorithm `minimize()` -which uses the limited memory BroydenFletcherGoldfarbShanno (BFGS) method with bound support - to find an optimal value for  $\sigma_l$ . We first generated some fake observed data for  $\sigma_l = 0.15$ , from  $t = 0.0$  to  $t = 5.0$  ms, with a timestep  $dt = 0.5$  ms. With  $\sigma_l = 0.10$  as initial guess, the `minimize()` algorithm returned  $\sigma_l = 0.150004$  after six iterations. Similarly, we can optimize for other parameters, such as  $\sigma_t$ , or  $g\_Na$ ,  $g\_CaL$ ,  $g\_K1$  or  $g\_K2$  from the Grandi cell model. With  $\sigma_t = 0.01$  as initial guess and fake data created for  $\sigma_t = 0.02$ , the `minimize()` algorithm converged to  $\sigma_t = 0.020005$  after five iterations. Optimizing for both  $\sigma_l$  and  $\sigma_t$ , with the same initial guesses and synthetic data, gave convergence in eight iterations. The value of the functional at each iteration step was consecutively 38.6, 11.7, 10.0, 3.19, 0.89, 0.023, 0.00063, 0.0000084 and 0.0000098. We also looked at synthetic spatially varying  $\sigma_l$  and  $\sigma_t$ ...

## References

- [1] Grandi, Pasqualini, Puglisi, & Bers. (2009). A Novel Computational Model of the Human Ventricular Action Potential and Ca transient. *Biophysical Journal*, 96(3), 664a-665a.
- [2] Gunzburger, M. (2003). *Perspectives in flow control and optimization* (Advances in design and control). Philadelphia, Pa.: SIAM.
- [3] Sundnes, J., Nielsen, B., Mardal, K., Cai, X., Lines, G., & Tveito, A. (2006). On the Computational Complexity of the Bidomain and the Monodomain Models of Electrophysiology. *Annals of Biomedical Engineering*, 34(7), 1088-1097.

---

<sup>2</sup>See, for example, [2] for an introductory text in adjoint-based optimization methods.

- [4] Sepulveda, Roth, & Wikswo. (1989). Current injection into a two-dimensional anisotropic bidomain. *Biophysical Journal*, 55(5), 987-999.
- [5] Plonsey, R., & Barr, R. (1982). The Four-Electrode Resistivity Technique as Applied to Cardiac Muscle. *Biomedical Engineering, IEEE Transactions on*, BME-29(7), 541-546.
- [6] Barr, & Plonsey. (1984). Propagation of excitation in idealized anisotropic two-dimensional tissue. *Biophysical Journal*, 45(6), 1191-1202.
- [7] [fenicsproject.org](http://fenicsproject.org)
- [8] [www.dolfin-adjoint.org](http://www.dolfin-adjoint.org)
- [9] [models.cellml.org/electrophysiology](http://models.cellml.org/electrophysiology)
- [10] [bitbucket.org/meg/cbcbeat](http://bitbucket.org/meg/cbcbeat)