Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

SEBASTIAN GÖTSCHEL, CHAMAKURI NAGAIAH, KARL KUNISCH,
MARTIN WEISER

Lossy Compression in Optimal Control of Cardiac Defibrillation

ZIB-Report 13-26 (May 2013)

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Takustraße 7
D-14195 Berlin-Dahlem

Telefon: 030-84185-0
Telefax: 030-84185-125

e-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Lossy Compression in Optimal Control of Cardiac Defibrillation

S. Götschel[1], Ch. Nagaiah[2], K. Kunisch[2,3], M. Weiser[1]

[1] Zuse Institute Berlin, Takustr. 7, D-14195 Berlin, Germany
[2] Radon Institute for Computational and Applied Mathematics, Altenbergerstr. 69, A-4040 Linz, Austria.
[3] Institute of Mathematics and Scientific Computing, University of Graz, Heinrichstr. 36, A-8010 Graz, Austria.

Abstract

This paper presents efficient computational techniques for solving an optimization problem in cardiac defibrillation governed by the monodomain equations. Time-dependent electrical currents injected at different spatial positions act as the control. Inexact Newton-CG methods are used, with reduced gradient computation by adjoint solves. In order to reduce the computational complexity, adaptive mesh refinement for state and adjoint equations is performed. To reduce the high storage and bandwidth demand imposed by adjoint gradient and Hessian-vector evaluations, a lossy compression technique for storing trajectory data is applied. An adaptive choice of quantization tolerance based on error estimates is developed in order to ensure convergence. The efficiency of the proposed approach is demonstrated on numerical examples.

Keywords: monodomain model, defibrillation, optimal control, Newton-CG, trajectory storage, compression

MSC2010: 35K57, 35Q92, 49M15, 65M60, 68P30, 94A29

1 Introduction

Ventricular fibrillation has been recognized as an important problem for clinical and theoretical research, due to the fact that it is a most dangerous and life threatening situation if not treated immediately to avoid sudden cardiac death. The termination of these cardiac arrhythmia is achieved by applying a strong electrical shock, a process called defibrillation. Today, many lives are saved by implantable defibrillation devices detecting the onset of fibrillation automatically. The unexpected and sudden application of a strong shock, however, is quite painful for the patient. Thus, there is a demand for designing shocks with as small an amplitude as possible while still extinguishing fibrillation. Computationally, it is therefore important to determine the time-dependent amplitude of an applied electrical field, which is able to damp gradients of transmembrane voltage in the system. This can be formulated as an optimal control problem governed by the monodomain equations describing the cardiac dynamics.

The optimal control of such models was already investigated in [25, 24, 22], where spatially distributed controls where studied. The optimization deals with the extinction of a single excitation wave as well as the termination of re-entry waves. As in practical situations distributed controls are difficult to implement, the present paper addresses a finite number of time-dependent controls which are spatially piecewise constant over different parts of the computational domain.

For the numerical realization of the optimal control problem by gradient methods, the solution of the adjoint equation, which is backward in time, is needed. Since the adjoint equation depends on the forward solution, the latter has to be stored. So-called checkpointing methods have been developed to reduce the storage demand, at the cost of multiple solves of the state equations, see e.g. [17]. In [35] a computationally inexpensive method for lossy compression of state trajectories was proposed, and compared to other approaches. In this paper, we extend the algorithm in several ways for the application to the cardiac defibrillation problem. As this is a nonlinear problem, an extended error estimation for estimating the quantization error's impact on the accuracy of the reduced gradient is derived. Here we shall use a Newton-CG method. For this purpose the action of the reduced Hessian on a given vector is computed by solving a linearized state equation as well as a second adjoint equation, incurring the need to store two additional trajectories, the adjoint and linearized state solutions. To keep the storage and memory bandwidth demands low, compression has to be applied here too. We investigate the influence of lossy compression on the convergence behavior.

Another important feature of this paper which is intimately related with compression is spatial grid adaptivity. For the direct problem space-time adaptivity of the monodomain equations was developed in e.g. [4, 13]. In the context of optimal control is was previously investigated in [24].

This paper is organized as follows. Section 2 summarizes the monodomain model used in this paper. In Section 3, we present the necessary optimality conditions for the optimization problem at hand. In Section 4 we briefly describe the basic compression method, before the error estimates and an adaptive compression technique are derived. Details of the implementation of our methods as well as numerical results are presented in Section 5.

2 Cardiac Modelling

The electrical activity of the heart is described by the well known monodomain or bidomain models [19, 28, 32]. The dynamics of the intra- and extracellular potentials are described by a set of reaction-diffusion equations and the ionic activity is expressed by ordinary differential equations. The present work is restricted to the optimal control of monodomain model equations. The monodomain model can be considered to approximate the more complex bidomain model fairly well in many situations [26, 29]. The governing equations of the monodomain model are described by the following equations.

$$\frac{\partial v}{\partial t} = \nabla \cdot \sigma \nabla v - I_{\text{ion}}(v, w) + \chi_{\Omega_c} I_e(t) \quad \text{in } Q \tag{1}$$

$$\frac{\partial w}{\partial t} = G(v, w) \quad \text{in } Q, \tag{2}$$

where $v \colon Q \to \mathbb{R}$ is the transmembrane voltage, $w \colon Q \to \mathbb{R}^n$ represents the ionic current variables, $\sigma \colon \Omega \in \mathbb{R}^2 \to \mathbb{R}^{2\times 2}$ denotes the intracellular conductivity tensor, and $Q = \Omega \times (0, T)$ denotes the space-time cylinder.

The time dependent extracellular current density stimulus $I_e(t)$ is the control variable which has a spatial support on the control domain $\Omega_c \subset \Omega$. Further χ_{Ω_c} denotes the extension by zero operator from Ω_c to Ω. The term $I_{\text{ion}}(v, w)$ describes the current density through the ionic channels of the cell membranes. The function $G(v, w)$ determines the evolution of the gating variables given by an electrophysiological cell model.

Since two decades, a variety of ionic models has been proposed to describe the cardiac electrical phenomenon in detail. In our computations, a simplified two variable model,

namely the Rogers-McCulloch variant of the Fitzhugh-Nagumo (FHN) model [30] is considered, where $I_{\text{ion}}(v,w)$ is cubic in terms of the transmembrane potential v and linear in terms of the gating variable w.

$$I_{\text{ion}}(v,w) = gv\big(1-\frac{v}{v_{th}}\big)\big(1-\frac{v}{v_p}\big)+\eta_1 vw\,, \tag{3}$$

$$G(v,w) = \eta_2\big(\frac{v}{v_p}-\eta_3 w\big)\,, \tag{4}$$

where g,η_1,η_2,η_3 are positive real coefficients, v_{th} is a threshold potential and v_p the peak potential.

In the absence of a conductive bath both intracellular and extracellular domains are electrically isolated along the tissue boundaries and homogeneous Neumann boundary conditions are appropriate to reflect this fact. Here the initial and boundary conditions are chosen as

$$\sigma\nabla v\cdot\eta \;=\; 0 \quad \text{on} \quad \partial\Omega\times[0,T] \tag{5}$$

$$v(x,0)=v_0 \qquad \text{and} \quad w(x,0)=w_0 \quad \text{on} \quad \Omega\,, \tag{6}$$

where $v_0:\Omega\to\mathbb{R}$ denotes the initial transmembrane potential, $w_0:\Omega\to\mathbb{R}^n$ is the initial gating variables at time $t=0$ and η stands for the outer normal to Ω.

We introduce some standing assumptions and the proper function space setting. It is assumed that the boundary of Ω is Lipschitz continuous and piecewise C^1 and that the conductivity tensor σ has $W^{1,\infty}$-coefficients and is uniformly elliptic. Further, let $v_0,w_0\in L^2(\Omega)$ and $I_e\in L^2(0,T)$. Then it is known (see e. g. [5, 23]), that (1) – (6) admit a weak solution pair (v,w) satisfying the a-priori estimate

$$\begin{aligned}&\|v\|^2_{C([0,T],L^2(\Omega))}+\|v\|^2_{L^2((0,T),H^1(\Omega))}+\|v\|^4_{L^4(Q)}+\|\tfrac{\partial v}{\partial t}\|^{\frac{4}{3}}_{L^{\frac{4}{3}}((0,T),(H^1)^*)}+\|w\|^2_{C([0,T],L^2(\Omega))}\\ &+\|\tfrac{\partial w}{\partial t}\|^2_{L^2((0,T),(H^1)^*)}\le C\,(1+\|v_0\|^2_{L^2(\Omega)}+\|w_0\|^2_{L^2(\Omega)}+\|I_e\|^2_{L^2((0,T),H^1(\Omega)^*)})\end{aligned} \tag{7}$$

where the constant C is independent of v_0,w_0 and I_e. If, moreover, $w_0\in L^4(\Omega)$ and $I_e\in L^\infty(0,T)$, then the weak solution is unique.

3 Optimization

We are considering the following optimal control problem:

$$\begin{cases}\min J(v,I_e)\,,\\ \\ \text{s.t. } e(v,w,I_e)=0 \quad \text{in} \quad Q\,,\end{cases} \tag{8}$$

where as before $v\colon Q\to\mathbb{R}$ is the transmembrane voltage, $w\colon Q\to\mathbb{R}$ is the gating variable, and the extracellular current I_e acts as a control variable in the optimal control problem. The coupled system of PDE and ODE constraints together with initial and boundary conditions is represented by $e(v,w,I_e)=0$. The existence of a solution to the system (1) – (6), the optimal control problem and its optimality system are discussed in this section.

As described in the previous section, the formal constraint $e(v,w,I_e)=0$ in (8) admits a solution $(v(I_e),w(I_e))$ as a function of I_e. We need to turn to the definition of the cost functional next. Here we aim at the resting potential of the transmembrane voltage in the observation domain Ω_{obs} and thus to track to zero voltage, with the control acting on a control domain $\Omega_c\subset\Omega$. We therefore use the following cost functional where the control objective consists in dampening out possible excitation waves of the transmembrane

voltage in the observation domain Ω_{obs} by applying extracellular current in the control domain Ω_c:

$$J(v, I_e) \;=\; \frac{1}{2}\int_0^T \int_{\Omega_{\text{obs}}} |v|^2 \,\mathrm{d}x\,\mathrm{d}t + \frac{\alpha}{2}\int_0^T |I_e|^2 \,\mathrm{d}t. \tag{9}$$

Here α is the weight of the cost of the control, which is used to penalize the influence of the control. With the help of the a-priori estimate (7) existence of a solution to (8) subject to (1) – (6), and $|I_e(t)| \le \gamma$ for a.e. $t \in (0,T)$ and any $\gamma > 0$ follows by standard arguments. The numerical realization is based on the optimality system which is given next.

Optimality system

The formal derivation of the optimality system is based on the Lagrangian functional given by

$$\mathcal{L} : X_1 \times X_2 \times L^\infty(0,T) \times L^4((0,T), H^1(\Omega)) \times L^2((0,T), L^2(\Omega)) \times L^2(\Omega) \times L^2(\Omega) \to \mathbb{R}$$

$$\begin{aligned}\mathcal{L}(v, w, I_e, p, q, \tilde{p}, \tilde{q}) = J(v, I_e) &+ \int_0^T \int_\Omega \left(-\frac{\partial w}{\partial t} + G(v, w)\right) q \;\mathrm{d}\Omega\,\mathrm{d}t \\ &+ \int_0^T \int_\Omega \left(\nabla \cdot \sigma \nabla v - \frac{\partial v}{\partial t} - I_{\text{ion}}(v, w) + \chi_{\Omega_c} I_e\right) p \;\mathrm{d}\Omega\,\mathrm{d}t + \int_\Omega (v(0) - v_0)\tilde{p} \;\mathrm{d}\Omega \\ &+ \int_\Omega (w(0) - w_0)\tilde{q} \;\mathrm{d}\Omega,\end{aligned}$$

where

$$X_1 = L^2((0,T), H^1(\Omega)) \cap W^{1,\frac{4}{3}}((0,T), (H^1)^*) \cap C([0,1], L^2(\Omega))$$

and

$$X_2 = L^2((0,T), L^2(\Omega)) \cap W^{1,2}((0,T), L^2(\Omega)) \cap C([0,1], L^2(\Omega)).$$

The first order optimality system is obtained by taking variations to the Lagrangian functional and invoking the appropriate Green identities. The complete optimality system is given as follows.

State equations

$$\frac{\partial v}{\partial t} \;=\; \nabla \cdot \sigma \nabla v - I_{\text{ion}}(v, w) + I_e(x, t) \quad \text{in } Q \tag{10}$$

$$\frac{\partial w}{\partial t} \;=\; G(v, w) \quad \text{in } Q. \tag{11}$$

Adjoint equations

$$\frac{\partial p}{\partial t} \;=\; -\nabla \cdot \sigma \nabla p + [I_{\text{ion}}]_v p + G_v q - v|_{\Omega_{\text{obs}}} \quad \text{in } Q\,, \tag{12}$$

$$\frac{\partial q}{\partial t} \;=\; -[I_{\text{ion}}]_w p - G_w(v, w) q \quad \text{in } Q. \tag{13}$$

Optimality condition

$$\alpha I_e + P_U\Big(\int_{\Omega_c} p\,dx\Big) = 0\,, \quad \text{a.e. in } (0,T) \tag{14}$$

The subscripts v and w denote the partial derivatives with respect to these variables, and P_U denotes the projection onto the closed interval $U = [-\gamma, \gamma]$. Further we obtain the

$$\text{terminal conditions: } p(T) = 0, \quad q(T) = 0 \quad \text{in } \Omega, \tag{15}$$

$$\text{boundary conditions: } \sigma \nabla p \cdot \eta = 0 \quad \text{on } \partial\Omega \times [0,T], \tag{16}$$

and $p(0) = -\tilde{p}$ and $q(0) = -\tilde{q}$. These computations were made rigorous in [23] and we have:

Proposition 3.1 (**First order necessary optimality conditions).** Let $(\overline{v}, \overline{w}, \overline{I}_e)$ denote a local minimizer of (8) under the constraint $|I_e| \leq \gamma$ a.e. in $(0,T)$. Then there exist adjoint states $p \in L^2(0,T;H^2(\Omega)) \cap W^{1,2}(0,T;L^2(\Omega))$, and $q \in C^1([0,T],L^2(\Omega)) \cap C^0([0,T],L^2(\Omega))$ satisfying the optimality system (1) – (6) together with (10)-(14). Moreover the gradient of the reduced mapping $j(I_e) : I_e \mapsto J(v(I_e), I_e)$ is given by $\alpha I_e + P_U(\int_{\Omega_c} p(I_e))\,dx)$.

In the numerical realization we use a Newton method to minimize the reduced functional j. Since the construction of the complete Hessian matrix is infeasible for this type of large scale problems we need to address the evaluation of the Hessian $j''(I_e^n)$ of j at I_e^n at iteration level n of the Newton algorithm in direction δI_e. For this purpose we recall from e.g. [18, 20] that $j''(I_e)$ can be represented as

$$j''(I_e) = T(I_e)^* \, \mathcal{L}(v, w, I_e, p, q)\, T(I_e), \tag{17}$$

where we dropped the last two variables $\tilde{p}, \tilde{q}$ from the Lagrangian since they do not contribute to the Hessian, and

$$T(I_e) = \begin{pmatrix} e_y(y(I_e), I_e)^{-1} e_{I_e}(y(I_e), I_e) \\ \mathrm{Id} \end{pmatrix}.$$

Here $y(I_e) = (v(I_e), w(I_e))$ and Id denotes the identity operator. Considering the structure of the Hessian given in (17) we see that due to the appearance of $T(I_e)$ and its adjoint we need to consider an additional linearized primal and a second pair of dual equations.

Linearized primal equations

$$\frac{\partial \delta v}{\partial t} = \nabla \cdot (\sigma \nabla \delta v) - ([I_{\text{ion}}]_v \, \delta v + [I_{\text{ion}}]_w \, \delta w) + \chi_{\Omega_c} \delta I_e \quad \text{in } Q, \tag{18}$$

$$\frac{\partial \delta w}{\partial t} = \frac{\eta_2}{v_p} \delta v - \eta_2 \eta_3 \delta w \quad \text{in } Q, \tag{19}$$

with homogeneous initial and Neumann boundary conditions, and

Adjoint-for-Hessian equations

$$\frac{\partial \delta p}{\partial t} = -\nabla \cdot \sigma \nabla \delta p + [I_{\text{ion}}]_v \delta p + \frac{\eta_2}{v_p} \delta q + z_1 \quad \text{in } Q, \tag{20}$$

$$\frac{\partial \delta q}{\partial t} = -[I_{\text{ion}}]_w \delta p + \eta_2 \eta_3 \delta q + z_2 \quad \text{in } Q, \tag{21}$$

with homogeneous initial and Neumann boundary conditions. Here z_1, and z_2 arise from the computation of the second derivative of the Lagrangian and are given by $z_1 = \delta v|_{\Omega_{\text{obs}}} - [I_{\text{ion}}]_{vv} p \delta v - \eta_1 \, \delta w \, p$ and $z_2 = -\eta_1 \, \delta v \, p$.

We refer e.g. to [20, 24] for details. Below we summarize the computations that are needed for the evaluation of the Hessian $j''(I_e^n)$ in direction δI_e^k:

1. solve the linearized primal equations Eqs. (18) and (19) for $\delta v, \delta w$ using δI_e^k
2. solve the adjoint equation Eqs. (20) and (21) using the computed (z_1, z_2) as right hand side
3. finally evaluate the action of j'' on δI_e^k i.e $-\int_{\Omega_c} \delta p + \alpha \delta I_e$

Remark 3.2. Let us heuristically discuss a second order sufficient optimality condition at a local minimizer $\overline{I}_e$ of (8) with associated states $(\overline{v}, \overline{w}) = (v(\overline{I}_e), w(\overline{I}_e))$. The second derivative of j at $\overline{I}_e$ in directions $(\delta I_e, \delta I_e)$ can be expressed as

$$j''(\overline{I}_e)(\delta I_e, \delta I_e) = \alpha \int_0^T |\delta I_e(t)|^2 \, dt + \int_Q |\delta v(\delta I_e)|^2 \, dx \, dt + \int_Q v(\overline{I}_e) \delta^2 v(\delta I_e, \delta I_e)| \, dx \, dt,$$

where $\delta v(\delta I_e)$ denotes the first component of the first variation of (v, w) at $\overline{I}_e$ in direction δI_e, which satisfies (18) (19), and $\delta^2 v(\delta I_e, \delta I_e)$ stands for the first component of the second variation of (v, w) at $\overline{I}_e$ in direction $(\delta I_e, \delta I_e)$. Putting aside the point that the second order condition needs only to hold on the kernel of the linearization of e, we note that, in view of the fact that $\delta^2 v(\delta I_e, \delta I_e)$ is of the order of $|\delta I_e|^2$, positive definiteness of $j''(\overline{I}_e)$ depends on the norm of $|v(\overline{I}_e)|$. A small norm of $v(\overline{I}_e)$ increases the chance of a second order sufficient optimality condition to hold.

4 Compressed Trajectory Storage

The numerical realization of the optimality system is demanding, both with respect to computation and storage. In order to reduce the demand in storage size and bandwidth, we apply lossy compression to the computed FE trajectories. In this section, we briefly describe the main ingredients of the lossy compression method, and derive error estimates for the reduced gradient as well as the Hessian-vector-multiplication.

The lossy compression algorithm consists of two main parts: quantization and prediction. We consider the spatial discretization by a nested family $\mathcal{T}_0 \subset \cdots \subset \mathcal{T}_l$ of triangulations, constructed from an initial triangulation $\mathcal{T}_0$ of a polygonal domain $\Omega \subset \mathbb{R}^d$. The set of nodes on level j is denoted by $\mathcal{N}_j$.

Quantization. For a given $\delta > 0$, we define the *quantization* $Q_\delta : \mathbb{R} \to \mathbb{Z}$ as

$$Q_\delta(y) := \left\lfloor \frac{y + \delta}{2\delta} \right\rfloor,$$

the *reconstruction* $Q_\delta^\dagger : \mathbb{Z} \to \mathbb{R}$ is given by

$$Q_\delta^\dagger(i) := 2\delta i.$$

This yields for the *quantization error* $|y - Q_\delta^\dagger(Q_\delta(y))| \leq \delta$.

Prediction. Values y_k of coarse level nodes are quantized directly to $i_k = Q_\delta(y_k)$, yielding a reconstructed value $\hat{y}_k := Q_\delta^\dagger(i_k)$. For new nodes $x_k \in \mathcal{N}_j \backslash \mathcal{N}_{j-1}$ on level $j > 0$, we make use of the grid hierarchy and quantize and store only the deviation of y_k from a prediction $P_k(\hat{y}_l : l \in \mathcal{N}_{j-1})$ obtained from reconstructed values $\hat{y}_l$ of lower level nodes. For simplicity, here we use linear interpolation between coarser grid nodes adjacent to x_k, which is the usual multigrid prolongation operator. The prediction increases the frequency of small numbers in the coefficients to be stored, such that subsequent entropy coding reduces the amount of data to be stored quite effectively. More details on the algorithmic procedure can be found in [35].

Below we denote the reconstructed functions with $\hat{\cdot}$, and the errors induced by lossy compression by $\varepsilon_\cdot$, e.g. $\hat{v} = v + \varepsilon_v$.

4.1 Inexact reduced gradient

For a scalar, linear state equation, the influence of the lossy compressed state trajectories on the reduced gradient was analyzed in [35]. In particular, a computational upper bound on the error could be obtained using the maximum principle. The strategy for adaptive selection of quantization tolerances sketched there for a simple steepest descent algorithm is extended to semilinear systems and the Newton-CG method in [14].

Let $y = (v, w)$ and $y(I_e)$ be the solution of the state equations (10), (11) for a given control I_e. Then the reduced gradient is given by

$$j'(I_e) = J_{I_e}(y(I_e), I_e) + e_{I_e}(y(I_e), I_e)^\star \underbrace{e_y(y(I_e), I_e)^{-\star}\left(-J_y(y(I_e), I_e)\right)}_{=:\lambda}.$$

As lossy compression is used, instead of the exact state only the reconstructed, inexact state $\hat{y} = (v + \varepsilon_v, w + \varepsilon_w)$ is available, leading to an inexact reduced gradient

$$\hat{j}'(I_e) = J_{I_e}(\hat{y}(I_e), I_e) + e_{I_e}(\hat{y}(I_e), I_e)^\star \underbrace{e_y(\hat{y}(I_e), I_e)^{-\star}\left(-J_y(\hat{y}(I_e), I_e)\right)}_{=:\tilde{\lambda}}.$$

For the error in the reduced gradient we have

$$\varepsilon_{j'} = \hat{j}'(I_e) - j'(I_e) = e_{I_e}(\hat{y}, I_e)^\star\big(\tilde{\lambda} - \lambda\big) = e_{I_e}(\hat{y}, I_e)^\star \varepsilon_\lambda, \tag{22}$$

using that J_{I_e}, e_{I_e} are independent of y. Thus the error in the reduced gradient is determined by the error in the solution of the adjoint equation.

From [14] we adapt the following theorem for the adaptive choice of the state quantization tolerance to our problem:

Theorem 4.1. *Denote by $\varepsilon_\lambda^{\max}$ an upper bound for the error in the adjoint for a state quantization error $\varepsilon_y^{\max} \equiv \mathbf{1}$. Let*

$$\xi := \left\| e_{I_e}(\hat{y}, I_e)^\star \varepsilon_\lambda^{\max} \right\|, \tag{23}$$

and $\theta \leq \left\| j'(I_e^{\text{next}}) \right\|$ be an estimate for the norm of the reduced gradient of the next Newton iteration. If the quantization tolerance δ^{next} fulfills

$$\delta^{\text{next}} \leq \frac{\theta \ \text{TOL}_{\text{grad}}}{\xi}, \tag{24}$$

then $\left\| \varepsilon_{j',\,\text{next}} \right\| \leq \text{TOL}_{\text{grad}} \left\| j'(I_e^{\text{next}}) \right\|$.

Remark 4.2. Assuming linear convergence of the optimization algorithm,

$$\tilde{\theta} = \frac{\left\| \hat{j}'((I_e)_i) \right\|^2}{\left\| \hat{j}'((I_e)_{i-1}) \right\|}$$

can be chosen as a computationally available approximation for θ, for $i > 1$.

To use Theorem 4.1, we need to derive a computationally available worst case error estimate $\varepsilon_\lambda^{\max}$. An important tool for this purpose is a comparison technique for the reaction-diffusion systems (see e.g. [6]). In the following, for vectors $y, z \in \mathbb{R}^m$, $y \geq z$ is defined as $y_i \geq z_i \ \forall i = 1, \ldots, m$. Other relations etc. are also defined component-wise.

Definition 4.3. A function $g : \mathbb{R}^m \to \mathbb{R}^m$ is *quasi-monotone non-decreasing*, if each component $g_i(y)$ is non-decreasing in y_j for each $i \neq j$.

Let us consider, for a moment, the abstract semilinear problem

$$\begin{aligned} \frac{\partial y_1}{\partial t} - \nabla \cdot \sigma \nabla y_1 &= g_1(y) \quad \text{in } Q \\ \frac{\partial y_2}{\partial t} &= g_2(y) \quad \text{in } Q, \end{aligned} \tag{P}$$

together with homogeneous initial- and Neumann-boundary conditions.

Definition 4.4. A function $\underline{y}$ is a *sub-solution* to (P), if in the differential equations, initial- and boundary conditions "$\leq$" holds instead of "$=$". $\overline{y}$ is a *super-solution*, if "$\geq$" holds instead of "$=$".

We introduce an auxiliary problem ($\overline{\text{P}}$) by replacing the reaction-function $g(y)$ in (P) by a super-reaction function $\overline{g}$, satisfying $\overline{g}(y) \geq g(y)\ \forall y \in \mathbb{R}^m$. The following two results are given in [6]:

Theorem 4.5. *Let $\overline{g}$ be a quasi-monotone non-decreasing, uniformly Lipschitz continuous super-reaction function . Let $\overline{y}$ be a super-solution of the auxiliary problem ($\overline{P}$) and y a sub-solution of* (P)*. Then $y \leq \overline{y}$ in $\overline{\Omega} \times [0,T]$.*

The next Lemma shows how to construct a super-reaction function.

Lemma 4.6. *Let $\overline{g}$ be defined by*

$$\overline{g}_i(y) = \sup_{\{z | \underline{y} \leq z \leq y,\ z_i = y_i\}} g_i(z)$$

for some sub-solution $\underline{y}$. Then $\overline{g}(y) \geq g(y)\ \forall y$, $\overline{g}$ is uniformly Lipschitz continuous, provided g is, and is quasi-monotone non-decreasing.

Returning to the defibrillation setting, we denote in the following by $\hat{I}_{\text{ion}}$ the evaluation of the function $I_{\text{ion}}(v,w)$ for inexact states $\hat{v}, \hat{w}$.

Lemma 4.7. *Denote by $\varepsilon_v, \varepsilon_w$ then the quantization errors for the solution of the state equations* (10), (11)*. The errors $\varepsilon_p = \hat{p} - p, \varepsilon_q = \hat{q} - q$ in the solutions p, q of the adjoint equations* (12), (13) *fulfill*

$$\frac{\partial \varepsilon_p}{\partial t} + \nabla \cdot \sigma \nabla \varepsilon_p - \left([\hat{I}_{\text{ion}}]_v \hat{p} - [I_{\text{ion}}]_v p \right) - G_v \varepsilon_q \;=\; -\varepsilon_v|_{\Omega_{obs}} \quad \textit{in } Q\,, \tag{25}$$

$$\frac{\partial \varepsilon_q}{\partial t} + \left([\hat{I}_{\text{ion}}]_w \hat{p} - [I_{\text{ion}}]_w p \right) + G_w \varepsilon_q \;=\; 0 \quad \textit{in } Q\,, \tag{26}$$

with homogeneous terminal and boundary conditions.

Proof. Linearity of the equations (12), (13). □

Using Taylor expansion, we get

$$\begin{aligned} [I_{\text{ion}}]_v p &\approx \left([\hat{I}_{\text{ion}}]_v - [\hat{I}_{\text{ion}}]_{vv} \varepsilon_v - [\hat{I}_{\text{ion}}]_{vw} \varepsilon_w \right) p \\ [I_{\text{ion}}]_w p &\approx \Big([\hat{I}_{\text{ion}}]_w - [\hat{I}_{\text{ion}}]_{wv} \varepsilon_v - \underbrace{[\hat{I}_{\text{ion}}]_{ww}}_{=0} \varepsilon_w \Big) p. \end{aligned}$$

Combined with the time transform $\tau = T - t$ (and minor abuse of notation) to get a system similar to (P), the error equations are

$$\frac{\partial \varepsilon_p}{\partial t} - \nabla \cdot \sigma \nabla \varepsilon_p = -[\hat{I}_{\text{ion}}]_v \varepsilon_p - G_v \varepsilon_q + \varepsilon_v|_{\Omega_{\text{obs}}} - [\hat{I}_{\text{ion}}]_{vv} \varepsilon_v p - [\hat{I}_{\text{ion}}]_{vw} \varepsilon_w p \quad \text{in } Q, \tag{27}$$

$$\frac{\partial \varepsilon_q}{\partial t} = [\hat{I}_{\text{ion}}]_w \varepsilon_p + G_w \varepsilon_q + [\hat{I}_{\text{ion}}]_{wv} \varepsilon_v p \quad \text{in } Q. \tag{28}$$

Using the abbreviations

$$\begin{aligned}
a(x,t) &= -[\hat{I}_{\text{ion}}]_v(x,t)\\
b(x,t) &= -\eta_2/v_p = \text{const} < 0\\
c(x,t) &= \chi_{\Omega_{\text{obs}}}(x)\varepsilon_v(x,t) - [\hat{I}_{\text{ion}}]_{vv}(x,t)\varepsilon_v(x,t)p(x,t) - \eta_1\varepsilon_w(x,t)p(x,t)\\
d(x,t) &= \eta_1\hat{v}(x,t)\\
e(x,t) &= -\eta_2\eta_3 = \text{const} < 0\\
f(x,t) &= \eta_1\varepsilon_v(x,t)p(x,t),
\end{aligned}$$

we can formulate the reaction function $g(y)$ as

$$g_1(y_1,y_2) = ay_1 + by_2 + c, \quad g_2(y_1,y_2) = dy_1 + ey_2 + f. \tag{29}$$

In our case the transmembrane potential $\hat{v}$ and thus $d(x,t)$ may be negative. Moreover, we have

$$\frac{\partial g_1}{\partial y_2} = b < 0\,.$$

Hence g is not quasi-monotone non-decreasing. With the aid of Lemma 4.6 and a sub-solution $\underline{y}$, we construct a super-reaction function:

$$\begin{aligned}
\overline{g}_1(y) &= \sup_{\{z|\underline{y}\le z\le y,\ z_1=y_1\}} g_1(z) = ay_1 + c + \sup_{\{z_2|\underline{y}_2\le z_2\le y_2\}} bz_2 \overset{b<0}{=} ay_1 + c + b\underline{y}_2\\
\overline{g}_2(y) &= \sup_{\{z|\underline{y}\le z\le y,\ z_2=y_2\}} g_2(z) = ey_2 + f + \sup_{\{z_1|\underline{y}_1\le z_1\le y_1\}} dz_1.
\end{aligned} \tag{30}$$

With regard to error estimation, we modify $\overline{g}$ given by (30) further, replacing c and f by upper bounds

$$\begin{aligned}
\overline{c}(x,t) &= \chi_{\Omega_{\text{obs}}}(x)\varepsilon_v^{\max} + |[\hat{I}_{\text{ion}}]_{vv}p(x,t)|\varepsilon_v^{\max} + |\eta_1 p(x,t)|\varepsilon_w^{\max}\\
\overline{f}(x,t) &= |\eta_1 p(x,t)|\varepsilon_v^{\max},
\end{aligned} \tag{31}$$

where $\varepsilon_\cdot^{\max} \equiv \mathbf{1}$ are upper bounds for the quantization error of the state solutions.

Moreover, we need to derive a sub-solution $\underline{y}$ to problem (P) with right-hand-side (29). For simplicity, $\underline{y}$ should be constant in time and space.

Lemma 4.8. *Let* $\varepsilon_v^{\max}, \varepsilon_w^{\max} > 0$ *and*

$$\begin{aligned}
\underline{c}(x,t) &= -\left(\varepsilon_v^{\max} + \left\|[\hat{I}_{\text{ion}}]_{vv}p\right\|_{L^\infty(\Omega\times(0,T))}\varepsilon_v^{\max} + \eta_1\left\|p\right\|_{L^\infty(\Omega\times(0,T))}\varepsilon_w^{\max}\right)\\
\underline{f}(x,t) &= -\eta_1\left\|p\right\|_{L^\infty(\Omega\times(0,T))}\varepsilon_v^{\max}.
\end{aligned}$$

Then the constant function $\underline{y} = \left(0, \min\{-\underline{c}/b,\ -\underline{f}/e\}\right)^{\mathsf{T}}$ *is a sub-solution to* (P) *with right-hand-side* (29).

Proof. As $\underline{y}$ is constant in x, and t, the derivatives vanish. It remains to show that $0 \le a\underline{y}_1 + b\underline{y}_2 + c$ and $0 \le d\underline{y}_1 + e\underline{y}_2 + f$. As $b, e < 0$, by definition $\underline{c}, \underline{f} \le 0$, and thus $\underline{y}_2 \le 0$. We have:

$$\begin{aligned}
a\underline{y}_1 + b\underline{y}_2 + c &= b\min\{-\underline{c}/b,\ -\underline{f}/e\} + c \ge b\min\{-\underline{c}/b,\ -\underline{f}/e\} + \underline{c}\\
&\begin{cases} = -\underline{c} + \underline{c} = 0, & \text{if } -\underline{c}/b \le -\underline{f}/e\\ = -b(\underline{f}/e) + \underline{c} \ge -b(\underline{c}/b) + \underline{c} \ge 0, & \text{if } -\underline{c}/b > -\underline{f}/e, \end{cases}
\end{aligned}$$

and

$$d\underline{y}_1 + e\underline{y}_2 + f = e\min\{-\underline{c}/b,\ -\underline{f}/e\} + f \geq e\min\{-\underline{c}/b,\ -\underline{f}/e\} + \underline{f}$$
$$\begin{cases} = -e(\underline{c}/b) + \underline{f} \geq -e(\underline{f}/e) + \underline{f} \geq 0, & \text{if } -\underline{c}/b \leq -\underline{f}/e \\ = -\underline{f} + \underline{f} = 0, & \text{if } -\underline{c}/b > -\underline{f}/e. \end{cases}$$

□

With this sub-solution at hand, we note for the super-reaction function $\overline{g}$, that

$$\sup_{\{z_1|0\leq z_1\leq y_1\}} dz_1 \leq \sup_{\{z_1|0\leq z_1\leq y_1\}} |d|z_1 = |d|y_1.$$

Thus we set

$$\begin{aligned} \overline{g}_1(y) &= ay_1 + b\underline{y}_2 + \overline{c} \\ \overline{g}_2(y) &= |d|y_1 + ey_2 + \overline{f}. \end{aligned} \tag{32}$$

Denote by $\varepsilon_p^{\max}, \varepsilon_q^{\max}$ the solution of the adjoint error equations (27),(28) with modified right-hand-side (32). Then, using Theorem 4.5, we get that

$$\varepsilon_p \leq \varepsilon_p^{\max}, \quad \varepsilon_q \leq \varepsilon_q^{\max}.$$

Remark 4.9. In order to choose a suitable quantization tolerance δ for encoding the state trajectory (governing $\varepsilon_v, \varepsilon_w$), we have to solve the error equation *before* solving the state equations (10), (11). In that case, $\hat{v}, \hat{w}$ needed for evaluating the coefficient functions are not known. Moreover, the exact solution p of the adjoint equation, without quantization errors, is not available. Thus we have to use estimates/heuristics for the unknown quantities to solve the error equation and get an approximate error bound. A simple choice is to substitute $\hat{p} \approx p$, to use a prescribed quantization tolerance δ_1 in the first iteration, and then, for iteration $i > 1$, to employ the values of $\hat{v}, \hat{p}$ from iteration $i-1$. With this choice, the error estimation has to be done only once per iteration before entering the CG, with the result being used for the matrix-vector-products during the CG (see Section 4.3), as well as for the gradient computation in the next optimization iteration.

4.2 Inexact Newton method

For the inexact Newton method,

$$j''(u_k)\delta u_k \approx -j'(u_k),\ u_{k+1} = u_k + \delta u_k,$$

we require for convergence that

$$\left\|j''(u_k)\delta u_k + j'(u_k)\right\| \leq \rho_k \left\|j'(u_k)\right\|,\ 0 < \rho_k \leq \overline{\rho} < 1,$$

with $\rho_k \to 0$ for super-linear convergence [9]. In addition to the truncation error r_k of the iterative solution of the Newton equation, the compression contributes to the residual by providing an inexact reduced gradient, $\hat{j}' = j' + \varepsilon_{j'}$.

Omitting the Newton iteration index k we solve

$$j''(u)\delta u = -j'(u) + \varepsilon_{j'} + r$$

and obtain the condition

$$\left\|\varepsilon_{j'} + r\right\| \leq \left\|\varepsilon_{j'}\right\| + \|r\| \leq \rho \left\|j'\right\|,$$

which is fulfilled if for some parameter $\zeta \in (0,1)$,

$$\left\|\varepsilon_{j'}\right\| \leq \zeta\rho \left\|j'\right\| \text{ and } \|r\| < (1-\zeta)\rho \left\|j'\right\|$$

hold. Thus we can fix the required gradient accuracy before entering the CG method, where we only have to control the truncation error.

4.3 CG with inexact Hessian-vector products

As the reduced Hessian $j''(u_k)$ is unavailable, the Newton system $j''(u_k)\delta u_k = -\hat{j}'(u_k)$ will be solved approximately by a conjugate gradient method. For the application of the reduced Hessian to a given vector, the solution of two linear PDEs is necessary, one linearized primal and one adjoint equation. To solve the dual equations (20), (21), the previously computed solution of the linearized state equations (18), (19), $(\delta v, \delta w)$, is required and thus needs to be stored using the lossy compression technique. The quantization error leads to an inexact matrix-vector product in the CG iterations, see step 3 of Alg. 1. Consequently, approximate residuals $\hat{r}_i$ are computed in the CG algorithm instead of the true residuals r_i. In order not to hinder convergence of Newton's method, we have to ensure that the stopping criterion in line 2 of Alg. 1, formulated in terms of $\hat{r}_i$, implies the desired bound

$$\|r_i\| < \mathrm{TOL}_{\mathrm{CG}} = (1-\zeta)\rho\|j'k\| \tag{33}$$

on the true residual r_i as well, i.e. we have to control $\|\varepsilon_{\mathrm{mv},j}\|$ for $j < i$ in order to bound the error $\|\hat{r}_i - r_i\|$.

Remark 4.10. In this paper, we only consider the influence of the inexact matrix-vector products during the solution of $j''(I_e)\delta I = -\hat{j}'(I_e)$. For simplicity of presentation, we neglect the impact of the quantization error in v, w and p on the matrix-vector-products. A detailed analysis of this influence can be found in the subsequent paper [14].

In the following, we denote by r_i the true residual in iteration i of the CG method.

Algorithm 1 CG for solving $j''(I_e)\delta I = -\hat{j}'(I_e)$ with inexact matrix-vector products

1: set $i = 0, \delta I_0 = 0, \hat{r}_0 = \hat{j}'(I_e), s_0 = -r_0$
2: **while** $\|\hat{r}_i\| > \mathrm{TOL}$ **do**
3: $\quad q_i = j''(I_e)s_i + \varepsilon_{\mathrm{mv},i}$
4: $\quad \alpha_i = (\hat{r}_i, \hat{r}_i)/(q_i, s_i)$
5: $\quad \delta I_{i+1} = \delta I_i + \alpha_i s_i$
6: $\quad \hat{r}_{i+1} = \hat{r}_i + \alpha_i q_i$
7: $\quad \beta_i = (\hat{r}_{i+1}, \hat{r}_{i+1})/(\hat{r}_i, \hat{r}_i)$
8: $\quad s_{i+1} = -\hat{r}_{i+1} + \beta_i s_i$
9: $\quad i = i + 1$
10: **end while**

Adaptive choice of quantization tolerance

The impact of inexact matrix-vector products on Krylov subspace methods has been analyzed in [31]. Adapted to our problem setting, the theory presented there leads to the following Lemma:

Lemma 4.11. *If, for a certain value l_m, in all CG iterations $i < m$,*

$$\|\varepsilon_{\mathrm{mv},i}\| \leq l_m \frac{\varepsilon}{\|\hat{r}_i\|} \tag{34}$$

holds, then $\|\hat{r}_m - r_m\| \leq \varepsilon$.

Choosing $\varepsilon = \varsigma \mathrm{TOL}_{\mathrm{CG}}$ in Lemma 4.11 for some $\varsigma \in (0,1)$ and $\mathrm{TOL} = (1-\varsigma)\mathrm{TOL}_{\mathrm{CG}}$ in Algorithm 1 guarantees the desired accuracy (33). In order to satisfy condition (34), a suitable, computable bound δ for the admissible quantization error has to be derived. To do so, we apply the same approach as in Section 4.1 above now to the linearized equations.

The first step is to evaluate the influence of the quantization tolerance on the error $\varepsilon_{\mathrm{mv},i}$. The inexact storage of the linearized-state trajectory leads to an inexactness in the computed linearized-dual equation.

Lemma 4.12. *Denote by $\varepsilon_{\delta v}, \varepsilon_{\delta w}$ the quantization errors for the solutions $\delta v, \delta w$ to the linearized state equations* (18), (19). *The errors $\varepsilon_{\delta p} = \hat{\delta p} - \delta p, \varepsilon_{\delta q} = \hat{\delta q} - \delta q$ in the linearized adjoint equations* (20),(21) *fulfill*

$$\frac{\partial \varepsilon_{\delta p}}{\partial t} + \nabla \cdot \sigma \nabla \varepsilon_{\delta p} - [\hat{I}_{ion}]_v \varepsilon_{\delta p} - \frac{\eta_2}{v_p} \varepsilon_{\delta q} \;=\; \varepsilon_{z_1} \quad \text{in } Q\,, \tag{35}$$

$$\frac{\partial \varepsilon_{\delta q}}{\partial t} + [\hat{I}_{ion}]_w \varepsilon_{\delta p} - \eta_2 \eta_3 \varepsilon_{\delta q} \;=\; \varepsilon_{z_2} \quad \text{in } Q\,, \tag{36}$$

with homogeneous terminal and boundary conditions, and

$$\begin{aligned} \varepsilon_{z_1} &= \varepsilon_{\delta v}|_{\Omega_{obs}} - \eta_1 \hat{p} \varepsilon_{\delta w} - [\hat{I}_{ion}]_{vv} \varepsilon_{\delta v} \hat{p} \\ \varepsilon_{z_2} &= -\eta_1 \hat{p} \varepsilon_{\delta v}. \end{aligned}$$

Proof. The proof uses linearity of the equations (20),(21). □

Due to the similarity of the equations, the same super-reaction function (32) as before can be used for computing upper bounds

$$\varepsilon_{\delta p} \le \varepsilon_{\delta p}^{\max}, \quad \varepsilon_{\delta q} \le \varepsilon_{\delta q}^{\max}. \tag{37}$$

The matrix-vector multiplication error induced by a quantization error not exceeding δ can now be estimated as

$$\|\varepsilon_{\mathrm{mv}}\| \le \delta \left\| e_{I_e}(\hat{v}, \hat{w}, I_e)^\star \begin{pmatrix} \varepsilon_{\delta p}^{\max} \\ \varepsilon_{\delta q}^{\max} \end{pmatrix} \right\| =: \delta\, \xi.$$

From this we obtain the quantization tolerance δ_i in iteration i of CG:

$$\delta_i \le \frac{l_m}{\xi} \frac{\varsigma\ \mathrm{TOL}_{\mathrm{CG}}}{\|\hat{r}_i\|}. \tag{38}$$

Remark 4.13. The convergence speed of the CG method with inexact matrix-vector-products is slower than with exact products, as orthogonality of the residuals is lost, see, e.g., [16]. As a remedy, re-orthogonalization of the residuals by using modified Gram-Schmidt orthogonalization can be implemented to reduce the deterioration of convergence speed.

Restart strategy. The choice (38) of the quantization tolerance suffers from the fact that neither l_m nor ξ is known exactly. The value of l_m given in [31] is, unfortunately, computationally unavailable. In order to avoid computational overhead, the error bounds $\varepsilon_{\delta p}^{\max}$ and $\varepsilon_{\delta q}^{\max}$ are best computed on a coarse mesh, which leads to an inexact value of ξ. Consequently, $\|r\| \le \mathrm{TOL}_{\mathrm{CG}}$ can not be guaranteed in practice. This may cause the norm of the true residual to stay far above the required tolerance, while $\|\hat{r}\|$ decreases further, see Fig. 1 (left) for an example.

In practice, the inaccuracy of ξ turns out to be of little consequence, as on one hand the error bounds tend to be rather smooth and well represented on coarse meshes, and on the the other hand, the computed error bounds are not particularly sharp.

The other factor l_m is of more importance. A heuristic value $l_m = \lambda_{\min}/m_{\max}$ has been proposed, where $\lambda_{\min}$ denotes the smallest eigenvalue of the reduced Hessian matrix j''. As a computational estimate thereof, the (inexact) Rayleigh quotient can be used,

$$\lambda_{\min} \lesssim \min_i \frac{(j''(I_e)(s_i) + \varepsilon_{\mathrm{mv},i}, s_i)}{(s_i, s_i)} = \min_i \frac{(q_i, s_i)}{(s_i, s_i)},$$

where the minimum is taken over all CG iterations. Note that due to the inexactness of the matrix-vector-product, underestimation of $\lambda_{\min}$ is possible, leading to a smaller-than-necessary quantization tolerance.

During the CG iterations we keep track of the approximate upper bound for $\lambda_{\min}$, and update it when necessary. To be precise, we compute an initial estimate before the first CG iteration, requiring one additional matrix-vector-product, and two scalar products. The matrix-vector-product can be computed on a coarse, fixed mesh with a user-defined quantization tolerance, or without compression, to keep the computational overhead as low as possible. In iteration i of the CG, we evaluate $(q_i, s_i)/(s_i, s_i)$, which uses only quantities already computed. When we get a significantly smaller estimate $\lambda^i_{\min} \ll \lambda^{i-1}_{\min}$, we update l_m and thus the quantization tolerance, and restart the CG algorithm by re-computing the current residual using the new quantization tolerance for the matrix-vector-product. While this causes a computational overhead of one additional matrix-vector product per restart, it is—in some situations—necessary to achieve convergence, see Fig. 1 (right).

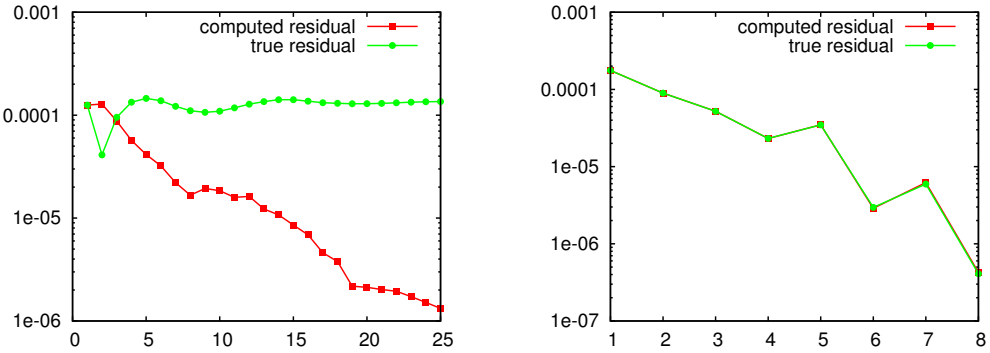

Figure 1: Behavior of computed $\hat{r}_i$ and true residuals during CG (fixed uniform grids). Left: without restart, right: with restarts.

The complete algorithm. In Alg. 2 we summarize the optimization procedure with adaptive quantization. Here, for simplicity, we use the same quantization tolerance δ_i for both the state and the adjoint solutions, despite the fact that the error estimate derived is only for the error caused by quantization of the state values. Details on an adaptive choice for the quantization tolerance for the adjoint solution will be provided in a separate paper.

Algorithm 2 Optimization with adaptive quantization tolerance

1: fix initial δ_1 (provided by user)
2: **for** $i = 1, \ldots$ **do**
3: solve the state equations (10), (11), encode v, w using δ_i
4: solve the adjoint equations (12), (13), decode v, w and encode p using δ_i
5: check optimality conditions, if optimal: end
6: solve the error estimator equations (27), (28) with right-hand-side (32) on a coarse, fixed mesh
7: use the CG method to compute the Newton update using the quantization tolerance provided by Eq. (38) for encoding and decoding the solutions $\delta v, \delta w$ (v, w, p are decoded using δ_i)
8: update the control I_e using the Armijo rule for the step size
9: compute δ_{i+1} by Eq. (24)
10: **end for**

5 Numerical Examples

In this section we give some details of the numerical realization and present results for two different test cases, the dampening of a single excitation wavefront, and the termination of re-entry waves. Both examples serve illustrative purposes and are therefore limited to a rather simple 2D unit square setup.

5.1 Implementation

Here we describe discretizations and optimization algorithms used. Two different implementations are available, based on the PDELab [3] and Kaskade 7 [15] toolbox, respectively. Both build upon the general PDE solver package Dune [2], a publicly available software package written in C++. The two implementations allow a validation of the codes and give an idea about the impact of different discretizations on the results. In essence, the results differ only in the range of the discretization errors to be expected.

Spatial discretization. Spatial discretization is done by piecewise linear finite elements on triangular meshes provided by the ALUGrid library [1] via the Dune interface. We provide results both on uniform Cartesian grids as well as adaptively refined meshes. Refinement is implemented in ALUGrid as bisection. The decision for local refinement or coarsening is based on a posteriori error estimators. We use the Zienkiewicz-Zhu (ZZ) estimator [36] after each time step in the PDELab implementation (see [24, Section 4.3] for details) and the hierarchical DLY estimator [10] in each time step of the Kaskade 7 implementation, cf. [12]. A fixed tolerance tol_x is prescribed independently of the progress of the optimization, together with limits for the maximum refinement level in the PDELab implementation and limits for the number of nodes in the Kaskade 7 version. Therefore, the accuracy attainable in the optimization is limited by the accuracy of the computed reduced gradients.

Grid refinement is performed independently for primal and dual trajectories and adapted to the respective solutions. As the primal solution enters the right hand side of the adjoint equation, the primal solution has to be interpolated on the dual grid, which is done by hierarchical search in the mesh hierarchy.

With adaptive mesh refinement, not only the state values, but also the corresponding non-uniform grids have to be stored. An efficient algorithm to encode the mesh connectivity can be found in [34], but has not been implemented here. We resort to the built-in functions of Dune/ALUGrid to write and read the adaptively refined grids directly.

Time discretization. For time stepping, linearly implicit Runge-Kutta methods of second order are used, the ROS2 Rosenbrock method in the PDELab implementation (see [24, Section 4.1 and 4.2] for details) and an extrapolated Euler method [11] in Kaskade 7. Fixed time step sizes were chosen for ease of implementation. The resulting linear systems arising after space and time discretization are solved by the BiCGStab method [33] with an ILU preconditioner.

Optimization. For the minimization of the objective, a Newton method with iterative solution of the Newton system by CG is applied. Reduced gradients and Hessian-vector products are computed by solving linearized primal and dual trajectories as outlined in Section 3. Once an approximate Newton direction is obtained, a stepsize is selected by backtracking line search (see, e.g., [27]). The optimization is terminated as soon as either $\|\nabla J_k\|_\infty \leq 10^{-3}(1+|J_k|)$ is satisfied, or the difference between the objective values of two successive optimization iterations is less than 10^{-3}.

$\sigma_{il}[\Omega^{-1}cm^{-1}]$	$\sigma_{it}[\Omega^{-1}cm^{-1}]$	$g[mS/cm^2]$	$v_{th}[mV]$	$v_p[mV]$	$\eta_1[mS/cm^2]$	η_2	η_3
$3 \cdot 10^{-3}$	$3.1525 \cdot 10^{-4}$	1.5	13	100	4.4	0.012	1

Table 1: Electrophysiological parameters used in the examples.

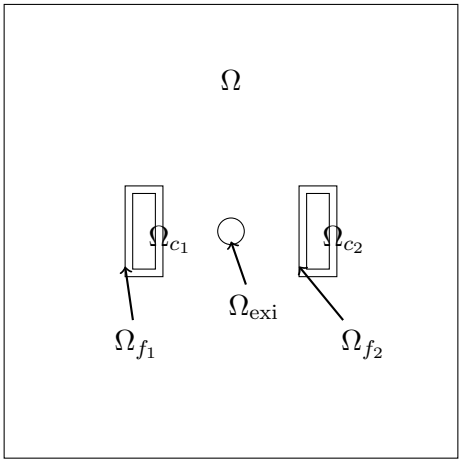

Figure 2: Computational domain for the dampening of single excitation wave.

For long time horizons as encountered in the second example, the extinction of a re-entrant spiral wave, the amount of data to be stored is quite large. One remedy is a receding horizon strategy [7, 8, 21, 22], which obtains suboptimal results by minimizing the objective on sequential, shorter time intervals individually. In this context, lossy compression as an additional means of reducing storage demand allows to use larger time horizons and thus leads to better results, such as a faster extinction of spiral waves.

5.2 Dampening of a single excitation wavefront

The first example taken from [24] deals with the extinction of the single excitation wave front. The computational domain is the unit square $\Omega = [0,1]^2$. The excitation domain Ω_{exi} is a circle with midpoint $(0.5, 0.5)$ and radius 0.04. The control domains are $\Omega_c = \Omega_{c_1} \cup \Omega_{c_2} = [0.32, 0.4] \times [0.45, 0.55] \cup [0.6, 0.68] \times [0.45, 0.55]$. Furthermore, $\Omega_{f_1} = [0.28, 0.42] \times [0.43, 0.57]$ and $\Omega_{f_2} = [0.58, 0.72] \times [0.43, 0.57]$ are neighborhoods of $\Omega_{c_1}, \Omega_{c_2}$. The observation domain is $\Omega_{\text{obs}} = \Omega \backslash (\Omega_{f_1} \cup \Omega_{f_2})$. See Fig. 2 for a sketch.

The simulated time is $T = 4$ ms. Initial conditions for the primal problem are

$$v(0) = \begin{cases} 101.0 & \text{in} \quad \Omega_{\text{exi}} \\ 0 & \text{otherwise} \end{cases}$$
$$w(0) = 0 \quad \text{in } \Omega. \tag{39}$$

The coarse grid is a 32×32 criss-cross grid. In adaptive computations, the maximum refinement level is limited to 7 in the PDELAB implementation and the maximum number of nodes is set to 24,000 in the KASKADE 7 implementation. For the regularization parameter, $\alpha = 3 \cdot 10^{-6}$ is chosen.

In the following we investigate the effect of different algorithmic ingredients. First we compare the results for the case of spatially constant controls with the results of [24], where the control is varying in time and space. Second, we show the effects of adaptive mesh refinement and lossy compressed storage.

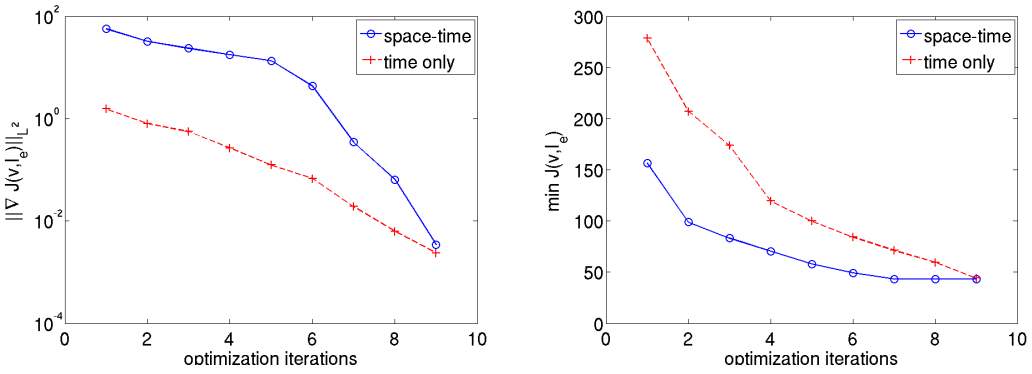

Figure 3: The norm of the gradient (y-axis shown in log scale for better comparison) and minimizational value of the cost functional are shown on left and right respectively.

Space-time vs time-only control

Both space-time and time only control are computed by minimizing the objective

$$\tilde{J}(v, I_e) \quad = \quad \frac{1}{2}\int_0^T \left(\int_{\Omega_{\text{obs}}} |v|^2 \, \mathrm{d}\Omega_{\text{obs}} + \beta \int_{\Omega_c} |I_e|^2 \, \mathrm{d}\Omega_c \right) dt. \tag{40}$$

In order to allow a direct comparison with [24], the regularization parameter $\beta = 3.75 \cdot 10^{-4}$ is used, which corresponds to $\beta \cdot \int_{\Omega_c} \mathrm{d}\Omega_c = \alpha$ in (9).

From the numerical results we observe that, in the case of space-time control, the control is active only in those parts of the control domain Ω_c where the wave enters first. When enlarging Ω_c, the control remains to be active only near that part of the boundary where the wavefront first arrives. The Newton algorithm exhibits superlinear convergence, but gets slower as Ω_c is enlarged. The L^2-norm of the gradient is shown at left hand side of Figure 3 and the minimal value of the cost functional is depicted at right. The minimal cost value is very similar for both cases.

Fixed mesh vs adaptivity

From here on, we consider spatially constant controls only. We compare results obtained on uniform static grids 128×128, 128×128, 256×256, 512×512 and 1024×1024 with those obtained on adaptively refined grids. The coarse grid is 32×32, and the maximum level of mesh refinement is set to 7. The optimal controls obtained on static and adaptive grids are shown in Figure 4. The total injected charge computed with PDELAB is -5.24 $(As)/cm^3$, -6.29 $(As)/cm^3$, -6.92 $(As)/cm^3$, -7.08 $(As)/cm^3$, and -7.27 $(As)/cm^3$ for static grids and -7.56 $(As)/cm^3$ for the adaptively refined grid. We observed that in contrast to all other grids, the excitation wave front was not completely extinguished on a 64×64 fixed grid. Clearly, we can see that adaptive mesh refinement grid shows good agreement with finer uniform meshes. Moreover, we observed mesh independence as the optimization algorithm takes 9 optimization iterations in any case.

The evolution of the excitation wave front and the adaptively refined grids is shown in Figure 5 at different time instances up to 6 ms, i.e. 2 ms beyond the control horizon. In the controlled case, the excitation wave is successfully terminated, such that at 6 ms only the coarse grid is retained. This is also reflected in the number of elements shown in Figure 6.

Compressed vs uncompressed

The compression factor, measuring the performance of the lossy storage algorithm, is defined as the ratio between uncompressed and compressed storage sizes.

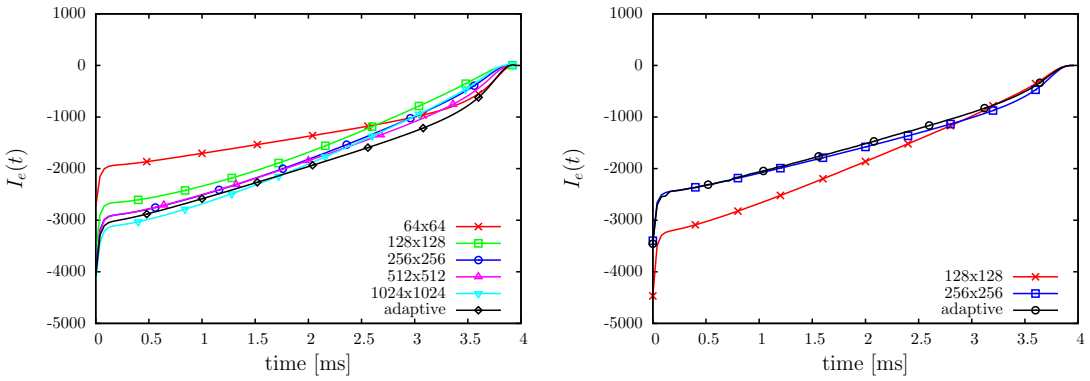

Figure 4: The control value I_e is shown for the different static grids and automatic mesh refinement strategy over the simulation time. Left: PDELab implementation. Right: Kaskade 7 implementation.

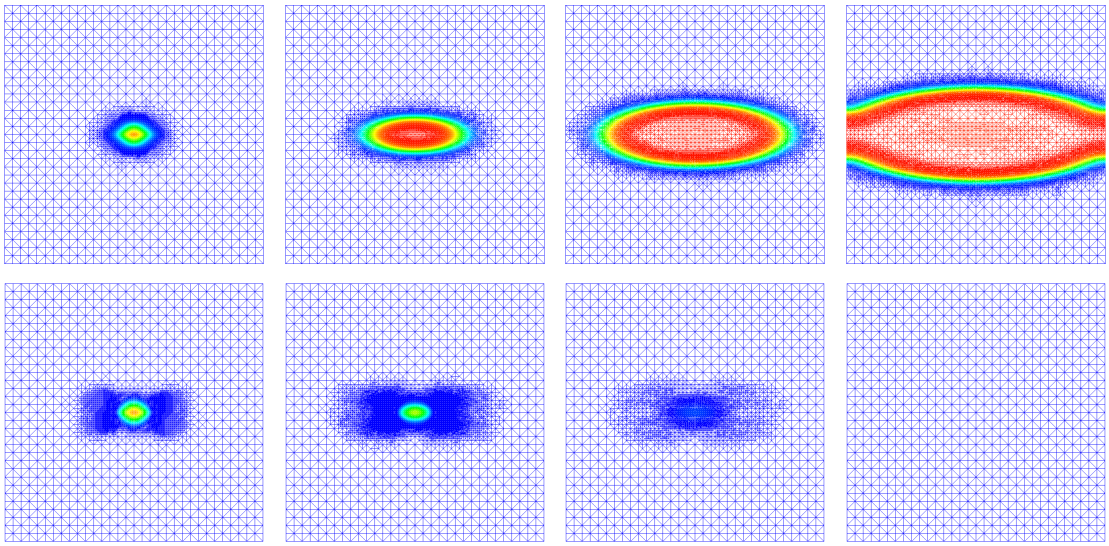

Figure 5: Transmembrane voltage at times 0.16, 2.0, 4.0 and 6 ms. Top row: uncontrolled. Bottom row: controlled.

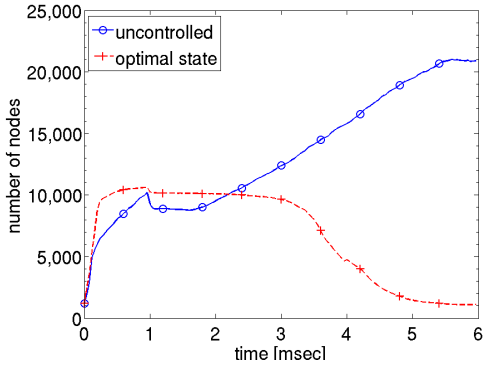

Figure 6: The number of nodes over the simulation time for the dynamic evolution of uncontrolled solution wave propagation and optimal state solution.

δ	compression factor			relative error in I_e	
	state	adjoint	avg. CG	L^2	L^∞
adaptive	13.0	11.4	10.8	$7.85 \cdot 10^{-5}$	$3.14 \cdot 10^{-4}$
10^{-4}	9.3	7.8	11.2	$9.32 \cdot 10^{-6}$	$2.95 \cdot 10^{-5}$
10^{-3}	13.0	11.4	9.9	$9.87 \cdot 10^{-5}$	$1.70 \cdot 10^{-4}$
10^{-2}	20.2	17.1	8.6	0.0054	0.0078
10^{-1}	34.1	28.7	8.3	0.0355	0.0377

Table 2: Compression factors and relative error in the computed control for different quantization tolerances.

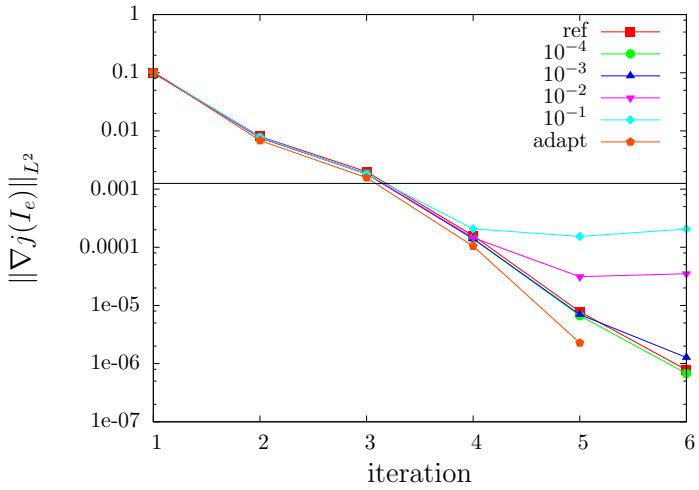

Figure 7: Gradient norm for different quantization tolerances on a fixed grid (32768 elements, 16641 vertices). The black horizontal line shows the approximated discretization error.

Fixed mesh. Test results for using compression on a fixed grid with 32768 elements and a time discretization with 100 time steps are shown in Figure 7 and Table 2. The same quantization tolerance δ was used for state and adjoint trajectories. The tolerance for the linearized state trajectory needed in the inner CG iterations was chosen adaptively according to Section 4.3. For the adaptive choice of quantization tolerance for the state solution, the averaged compression rate is reported in Table 2. The quantization tolerance and compression factors for each iteration are shown in Figure 8.

We estimate the discretization error in the reduced gradient by using a solution on a finer mesh as a reference. The relative error in the computed controls are determined by using the control computed without compression as a reference.

For smaller δ, the residual in the CG iterations becomes smaller more quickly, yielding a larger adaptively chosen quantization tolerance for the matrix-vector products, and thus better compression factors.

For state and adjoint solves, we measured the computation times with and without compression. In Table 3, we give details on the composition of the run-times for state and adjoint equations. The computations were carried out on an Intel Core2 Quad CPU with 2.83GHz, without using parallelization. The writing and reading of the grid is done by using the built-in functions of ALUGrid and is stated just for completeness; for fixed meshes, storing the grid is not necessary. We note that reading the grids is more expensive than writing, as the grid data structures have to be re-created. Setup for lossy compression

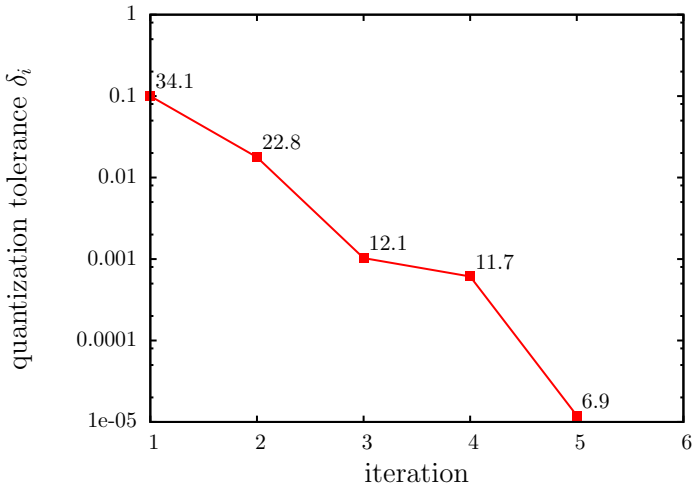

Figure 8: Adaptively chosen quantization tolerances and factors for encoding the state solutions v, w. Note that while the quantization tolerance is reduced according to the progress of the optimization, the mesh refinement tolerance remains fixed, which leads to decreasing compression rates.

	solve	grid write/read	setup	encode	decode
state	150.1	0.056	0.055	4.99	–
adjoint	735.03	0.286	0.166	2.75	2.62

Table 3: Detailed timings (in seconds) for a single state and adjoint solve (averaged over 5 iterations). Writing/reading of the grid can be ignored in the case of fixed meshes.

consists of generating data structures for prolongation, which have to be re-computed whenever the grid changes at a time step. For fixed meshes, these computations are done just once at the beginning of each time integration, and can be kept throughout all time steps. Setup takes more time for the adjoint, as here both encoding (adjoint grid) and decoding (state grid) are used on two different grids. When compressing the solution of the adjoint equations, only p has to be stored, resulting in lower CPU times than for the state solutions, where v and w have to be encoded.

Adaptive mesh. In Figure 9 we show the optimization progress on adaptively refined grids with and without compression. Compression factors are reported in Table 4.

The adaptive grids were restricted to consist of at most $24,000$ vertices. Due to local refinement there are less degrees of freedom on higher grid levels. As there is less

δ	compression factor			relative error in I_e	
	state	adjoint	avg. CG	L^2	L^∞
adaptive	8.0	5.4	5.5	0.0088	0.0156
10^{-3}	4.6	5.3	3.7	0.0067	0.0121
10^{-2}	11.1	11.4	3.5	0.0087	0.0195
10^{-1}	23.3	22.5	3.9	0.0577	0.0624

Table 4: Compression factors and relative error in the computed control for different quantization tolerances.

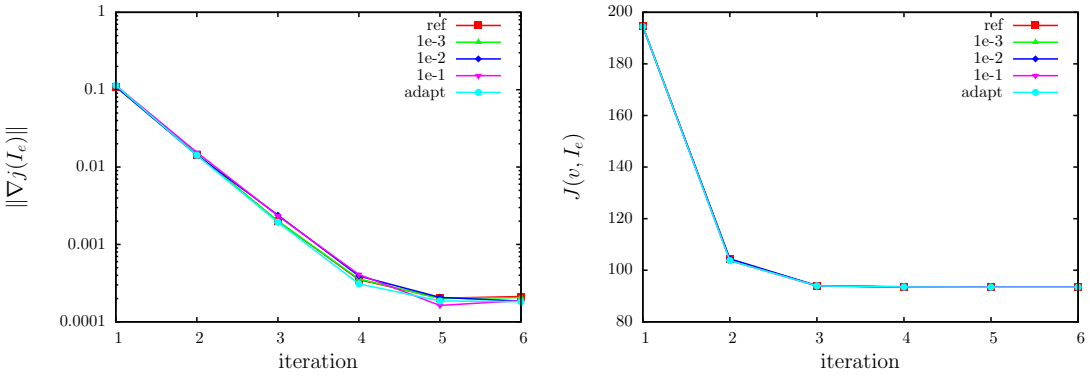

Figure 9: Gradient norm (left) and objective function (right) on adaptive meshes for fixed and adaptive quantization.

	solve	grid write/read	setup	encode	decode
state	3303.8	30.7	73.3	30.3	–
adjoint	4885.0	232.6	176.6	27.9	23.2

Table 5: Detailed timings (in seconds) for a single state and adjoint solve (averaged over 5 iterations; $\delta = 10^{-2}$). *solve* consists of time for adaptivity, solution transfer and actual solution of the linear equation systems.

redundancy in the values, the compression is not as effective as in the uniform case, leading to smaller compression factors than before.

Detailed timings for state and adjoint solves can be found in Table 5. The most expensive part for the lossy storage algorithm here is the setup of the prolongation matrices, which has to be done in every time step during the state solve, and twice per time step in the adjoint solve (for state and adjoint grids). This overhead could be reduced by reusing the prolongation matrices available in the FE-codes from the grid transfer, or used in multigrid preconditioners. As the same lossy compression code was used in the two different implementations, this was not implemented due to incompatible data structures in PDELab and Kaskade 7.

5.3 Termination of re-entry waves

The second example demonstrates the computational techniques for the termination of re-entrant waves by using a receding horizon technique. Optimal control for the termination of re-entrant waves were first presented in [22], where a spatially variable control and uniform grids have been used. Here we show numerical results using spatially constant controls and adaptive mesh refinement in a Newton-Krylov optimization algorithm. In [22], 45 receding horizon intervals of 4 ms each, i.e. 180 ms in total, were required to terminate the re-entry wave. Motivated by the reduced storage requirement due to lossy compression we allow longer time horizons of 16 ms now. With longer time horizon, the optimizer can compute better controls, such that now 9 intervals, i.e. 144 ms in total, are sufficient.

The computational domain shown in Figure 10 is $[0, 2]^2 cm^2$, the control and observation domains are $\Omega_{c_1} = [0.4, 0.5] \times [0.0, 0.65]$, $\Omega_{c_2} = [1.3, 1.4] \times [0.0, 1.0]$, $\Omega_{f_1} = [0.35, 0.55] \times [0.0, 0.70]$, and $\Omega_{f_2} = [1.25, 1.45] \times [0.0, 1.05]$, respectively. The regularization parameter is chosen as $\alpha = 8 \cdot 10^{-6}$. For this test case the maximum level of spatial grid refinement is set to 5. The spatial grid adaptivity tolerance is set to $\text{tol}_x = 10^{-3}$ and the fixed time

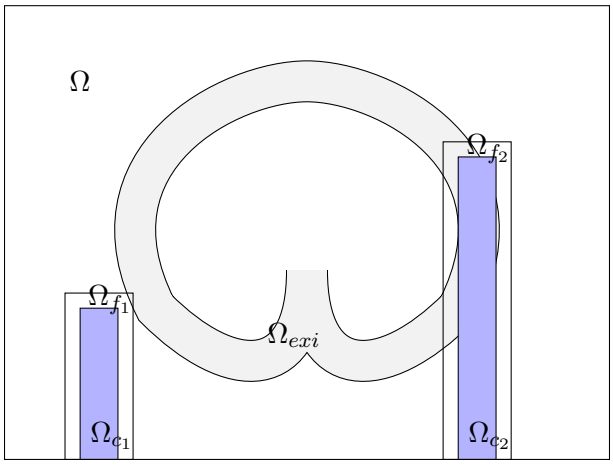

Figure 10: Computational domain for the termination of re-entry wave.

	solve	adapt	update	grid		compression		
				write	read	encode	decode	factors
primal	650	795	-	58	-	142.1	-	5.2
dual	498	683	60	40	350	116.9	120	5.7
lin primal	304	500	50	38	345	113.4	131	4.8
lin dual	285	490	135	-	1150	-	319	-

Table 6: Average computation times (in seconds) for one iteration of primal, dual, linearized primal and dual-for-Hessian equations.

step size is reduced to 0.01 ms, i.e. there are 1600 time steps in each receding time horizon of the optimization.

To induce the re-entry we followed the standard $S1 - S2$ stimulation protocol as explained in [22]. Initially the external stimulus $S1$ of $I_{\text{tr}} = 100\ \mu A/cm^2$ at time $t = 0\ ms$ is applied for a duration of 5 ms along the bottom edge of the tissue sheet to induce a planar wavefront traveling towards the top edge of the sheet. At time $t = 183\ ms$, when the critical recovery isoline arrives at the center of the sheet, a second $S2$ stimulus of $I_{\text{tr}} = 100\ \mu A/cm^2$ is applied in a small region of 0.3 cm radius at the center of the domain for a duration of 5 ms. This generates two phase singularities at the intersections of the critical recovery isoline with the boundary of the $S2$ stimulus region, leading to a so-called Figure of Eight re-entrant pattern. The solution at $t = 435\ ms$ was then chosen as the initial state.

In each time horizon, the optimization algorithm takes an average of 7 Newton iterations to satisfy the given convergence tolerance and an average of 10.5 inner CG iterations were taken at for Newton iteration. Averaged computation times for one solve of primal, dual, linearized primal and linearized dual problems are reported in Table 6. The computation time for the spatial grid adaptivity, including the error estimator, refining the grid and interpolation of the solution to the new grid, is shown in the second column. We can observe that adaptivity takes more CPU time than solving the governing equations. As primal and dual equations are solved on individually adapted grids, the primal solution has to be interpolated on the dual mesh. To accomplish this, we use a hierarchical grid search. Such grid transfers are also needed during the linearized primal solve. The third column reports these solution transfer times.

The computation times for writing and reading the grids are shown in the 4th and 5th

columns, respectively. Again, reading the grid is more expensive than writing.

The coarse grid contains 2048 triangular elements and 1089 nodes. During the initial outer iteration of a complete primal solve for the first time horizon of optimization, the maximum number of elements and nodes at a particular time frame reached 41985 and 21136 respectively. Likewise, in the dual solve the maximum number of elements and nodes are 45568 and 22979 respectively at the first outer iteration.

The computational time for the trajectory compression, including setup, as well as compression factors are shown in the last three columns of Table 6. During the primal solve one only needs to encode the solution of the state trajectory. This takes 9.8% CPU time of the complete primal solve, which is the sum of state equation solve and the expense of spatial grid adaptivity. During the dual solve one needs to decode the primal solution as well as to encode the solution of dual equations. Here we observed that the encoding of the dual solution takes less computational time compared to encoding of the primal solution. The decoding of the primal solution takes 10% CPU time of the dual solve. In the dual-for-Hessian solve one has to restore the solution of the primal, dual and linearized primal solution. It can be seen that decoding takes about 40% of overall CPU time over the solution. Note that one has to load the three different spatial grids for the reconstruction of the solution. We can observe that the reconstruction of the grids takes 150% of CPU time of the dual solution and it is the most expensive part. The overall computational time for 9 time horizons of optimization takes about 23 days in a sequential run. The quantization tolerances were chosen fixed to be $\delta = 10^{-3}$ for the primal and dual solution, and $\delta = 10^{-7}$ during the CG method. This choice leads to the reported compression factors of about 5–6, which are smaller than in the previous test case. Besides the fixed tolerance, this can be attributed to the adaptive mesh refinement, leading to fewer nodes on higher grid levels, and thus to a worse prediction than before.

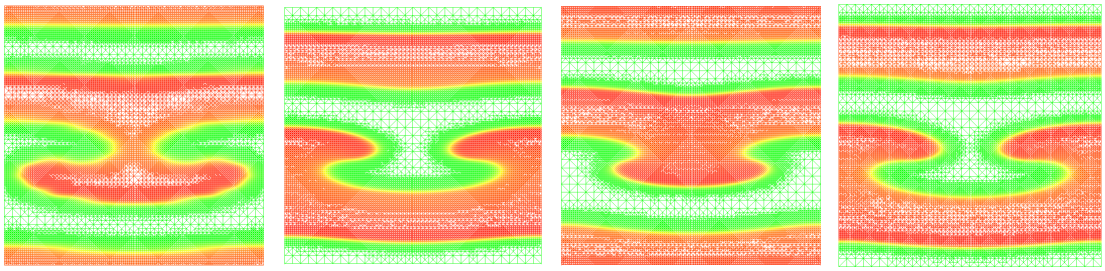

Figure 11: The uncontrolled solution of transmembrane voltage at times 0.4, 32.04, 64.04 and 110.04ms.

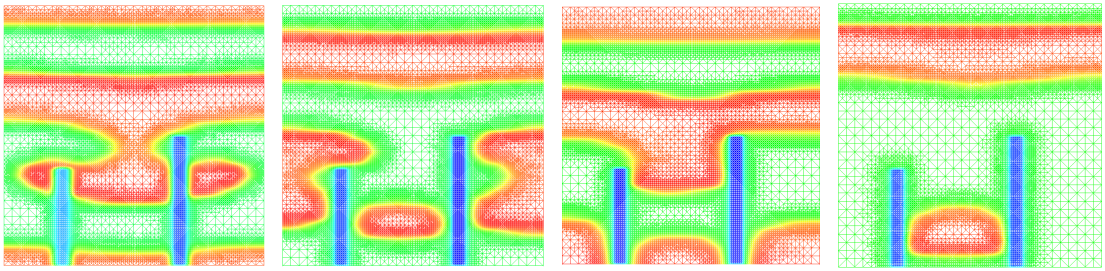

Figure 12: The optimally controlled solution of transmembrane voltage at times 0.4, 32.04, 64.04 and 110.04ms.

The 2D surface mesh plots of the transmembrane voltage are depicted at different times for the uncontrolled solution and optimally controlled solution. Clearly, we can observe that the uncontrolled dynamical solution evolves with time and maintains the re-entrant phenomena, while for the optimally controlled solution at 0.04ms the left and

right excitation wave fronts are separated. At time $32.04ms$ they are traveling towards the center of the domain. Due to some delay in the excitation wave fronts, the re-entrant wave will fit exactly in between the two control domains at time $64.04ms$. Then it reaches the boundary and leaves the domain completely. We observed that the excitation wave front of the transmembrane voltage goes to resting state after $144ms$ of simulation time, i.e. after 9 time horizons. In the previous work [22], this takes about $180ms$ to reach the resting state where each time horizon period is $4ms$. The longer time horizon computations are made possible with the help of the compression technique.

6 Conclusion

It was demonstrated that the lossy compression strategy developed in [35] for optimal control problems subject to the heat equation can be successfully extended to reaction diffusion equations arising in electro-physiology. The trajectories of these nonlinear equations exhibit an especially rich dynamical systems behavior. From the optimal control point of view the main objective consists in the dampening of arrhythmias.

By means of two typical scenarios, one related to excitation waves and another to a reentry wave phenomenon, the efficiency of the proposed compression methodology could be demonstrated. In the context of the Newton method that was used to solve the open loop optimal control problems, the trajectories of four dynamical systems need to be considered. The primal and adjoint one, just as for gradient type methods, and two additional ones, a second linearized forward and a second adjoint equation, backwards in time. It is remarkable that even for the control of the reentry wave, with highly nonlinear dynamics and adaptive mesh refinement, good compression factors could be achieved.

From the point of view of optimal control we also investigated the combination of the compression techniques with receding horizon control methods. Here compression methods allow the use of longer time horizons which give better performance compared to solutions which need to rely on many shorter time horizons.

Concluding we believe that compression techniques provide an effective method to cope with computing and storage demands in large scale optimal control problems.

Acknowledgment

The authors gratefully acknowledge support by the Austrian Science Foundation (FWF) under SFB 032, "Mathematical Optimization and Applications in Biomedical Sciences", the Austrian Academy of Sciences (ÖAW) and by the DFG Research Center MATHEON, project F9.

References

[1] ALUGrid home page. `ALUGrid:http://www.mathematik.uni-freiburg.de/IAM/Research/alugrid/`.

[2] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, R. Kornhuber, M. Ohlberger, and O. Sander. A generic grid interface for parallel and adaptive scientific computing. Part II: implementation and tests in DUNE. *Computing*, 82(2):121–138, July 2008.

[3] P. Bastian, F. Heimann, and S. Marnach. Generic implementation of finite element methods in the distributed and unified numerics environment (DUNE). *Kybernetika*, 46(2):294–315, 2010.

[4] Y. Belhamadia, A. Fortin, and Y. Bourgault. Towards accurate numerical method for monodomain models using a realistic heart geometry. *Mathematical Biosciences*, 220(2):89 – 101, 2009.

[5] Y. Bourgault, Y. Coudiére, and C. Pierre. Existence and uniqueness of the solution for the bidomain model used in cardiac electrophysiology. *Nonlinear Analysis: Real World Applications*, 10(1):458–482, 2009.

[6] N. F. Britton. *Reaction-Diffusion Equations and Their Application to Biology*. Academic Press, 1986.

[7] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica J. IFAC*, 34(10):1205–1217, 1998.

[8] H. Choi, M. Hinze, and K. Kunisch. Instantaneous control of backward-facing step flows. *Appl. Numer. Math.*, 31:133–158, October 1999.

[9] P. Deuflhard. *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*. Springer, 2nd edition, 2006.

[10] P. Deuflhard, P. Leinen, and H. Yserentant. Concepts of an adaptive hierarchical finite element code. *IMPACT Comp. Sci. Eng.*, 1(1):3–35, 1989.

[11] P. Deuflhard and U. Nowak. Extrapolation integrators for quasilinear implicit ODEs. In P. Deuflhard and B. Engquist, editors, *Large Scale Scientific Computing*, volume 7 of *Progress in Scientific Computing*, pages 37–50. Birkhäuser, 1987.

[12] P. Deuflhard and M. Weiser. *Adaptive numerical solution of PDEs*. de Gruyter, 2012.

[13] P. C. Franzone, P. Deuflhard, B. Erdmann, J. Lang, and L. F. Pavarino. Adaptivity in space and time for reaction-diffusion systems in electrocardiology. *SIAM Journal on Numerical Analysis*, 28(3):942–962, 2006.

[14] S. Götschel and M. Weiser. Lossy compression for PDE-constrained optimization: Adaptive error control. ZIB Report 13-27, 2013.

[15] S. Götschel, M. Weiser, and A. Schiela. Solving optimal control problems with the kaskade 7 finite element toolbox. In A. Dedner, B. Flemisch, and R. Klöfkorn, editors, *Advances in DUNE*, pages 101–112. Springer, 2012.

[16] S. Gratton, P. Toint, and J. Tshimanga. Inexact range-space Krylov solvers for linear systems arising from inverse problems. Technical Report 09/20, FUNDP – University of Namur, Belgium, 2009.

[17] A. Griewank and A. Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM, Philadelphia, 2008.

[18] M. Heinkenschloss and M. Herty. A spatial domain decomposition method for parabolic optimal control problems. *J. Comput. Appl. Math.*, 201:88–111, April 2007.

[19] C. S. Henriquez. Simulating the electrical behavior of cardiac tissue using the bidomain model. *Crit. Rev. Biomed. Eng.*, 21:1 77, 1993.

[20] M. Hinze and K. Kunisch. Second order methods for optimal control of time-dependent fluid flow. *SIAM J. Control Optim.*, 40(3):925–946, 2001.

[21] K. Ito and K. Kunisch. Asymptotic properties of receding horizon optimal control problems. *SIAM J. Control Optim.*, 40:1585–1610, May 2001.

[22] K. Kunisch, C. Nagaiah, and M. Wagner. A parallel Newton-Krylov method for optimal control of the monodomain model in cardiac electrophysiology. *to appear in: Computing and Visualization in Science*, 2012.

[23] K. Kunisch and M. Wagner. Optimal control of the bidomain system (I): The monodomain approximation with the Rogers–McCulloch model. *Nonlinear Analysis: Real World Applications*, 13(4):1525 – 1550, 2012.

[24] C. Nagaiah and K. Kunisch. Higher order optimization and adaptive numerical solution for optimal control of monodomain equations in cardiac electrophysiology. *Appl. Numer. Math.*, 61:53–65, January 2011.

[25] C. Nagaiah, K. Kunisch, and G. Plank. Numerical solution for optimal control of the reaction-diffusion equations in cardiac electrophysiology. *Computational Optimization and Applications*, 49:149–178, 2011. 10.1007/s10589-009-9280-3.

[26] B. F. Nielsen, T. S. Ruud, G. T. Lines, and A. Tveito. Optimal monodomain approximations of the bidomain equations. *Applied Mathematics and Computation*, 184(2):276–290, 2007.

[27] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Verlag, New York, second edition, 2006.

[28] R. Plonsey. Bioelectric sources arising in excitable fibers (ALZA lecture). *Ann Biomed Eng*, 16(6):519–46, 1988.

[29] M. Potse, B. Dube, J. Richer, A. Vinet, and R. Gulrajani. A comparison of monodomain and bidomain reaction-diffusion models for action potential propagation in the human heart. *IEEE Transactions on Biomedical Engineering*, 53(12):2425–2435, Dec. 2006.

[30] J. M. Rogers and A. D. McCulloch. A collocation-Galerkin finite element model of cardiac action potential propagation. *IEEE Trans. Biomed. Eng.*, 41:743–757, 1994.

[31] V. Simoncini and D. B. Szyld. Theory of inexact Krylov subspace methods and applications to scientific computing. *SIAM J. Scientific Computing*, 25(2):454–477, 2003.

[32] L. Tung. *A bi-domain model for describing ischemic myocardial DC potentials.* PhD thesis, MIT, Cambridge, MA, 1978.

[33] H. A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13:631–644, 1994.

[34] C. von Tycowicz, F. Kälberer, and K. Polthier. Context-based coding of adaptive multiresolution meshes. *Computer Graphics Forum*, 30(8):2231–2245, 2011.

[35] M. Weiser and S. Götschel. State trajectory compression for optimal control with parabolic pdes. *SIAM J. Scientific Computing*, 34(1), 2012.

[36] O. C. Zienkiewicz and J. Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *Int. J. Num. Meth. Eng*, 24:337–357, 1987.